

AIX เวอร์ชัน 7.2

ระบบปฏิบัติการและการจัดการกับ
อุปกรณ์

IBM

AIX เวอร์ชัน 7.2

ระบบปฏิบัติการและการจัดการกับ
อุปกรณ์

IBM

หมายเหตุ
ก่อนใช้ข้อมูลนี้ รวมถึงผลิตภัณฑ์ที่สนับสนุน โปรดอ่าน ข้อมูลใน “คำประกาศ” ในหน้า 719

This edition applies to AIX Version 7.2 and to all subsequent releases and modifications until otherwise indicated in new editions.

© ลิขสิทธิ์ของ IBM Corporation 2015, 2016.

© Copyright IBM Corporation 2015, 2016.

สารบัญ

เกี่ยวกับเอกสารนี้	v
การไฮไลต์	v
การคำนึงถึงขนาดตัวพิมพ์ใน AIX	v
ISO 9000	v
การจัดการระบบปฏิบัติการและอุปกรณ์	1
มีอะไรใหม่ในระบบปฏิบัติการและการจัดการอุปกรณ์	1
การจัดการระบบปฏิบัติการ	2
อินเตอร์เฟซการจัดการระบบที่มีอยู่	2
ข้อมูลผลิตภัณฑ์สำคัญของซอฟต์แวร์	3
อัปเดตระบบปฏิบัติการ	3
สตาร์ทอัพระบบ	4
สำเนาสำรองระบบ	23
การปิดระบบ	55
สภาวะแวดล้อมของระบบ	57
AIX Runtime Expert	70
คำสั่งและโปรเซส	140
การจัดการ hang ระบบ	163
การจัดการโปรเซส	166
การทำแอดเคาต์ระบบ	174
System Resource Controller	203
ไฟล์ระบบปฏิบัติการ	208
เซลล์ระบบปฏิบัติการ	229
การรักษาความปลอดภัยของระบบปฏิบัติการ	321
สภาวะแวดล้อมผู้ใช้	336
การอ้างอิงระบบ BSD	353
การเปลี่ยนทิศทางอินพุตและเอาต์พุต	375
การกู้คืนเคอร์เนล AIX	382
การจัดการอุปกรณ์	384
Logical Volume Manager	384
หน่วยเก็บโลจิคัลวอลุ่ม	421
พื้นที่การเลื่อนหน้าและหน่วยความจำเสมือน	453
ระบบไฟล์	464
ผู้จัดการเวิร์กโหลด	525
การติดตั้งอุปกรณ์ IDE	574

โหนดอุปกรณ์	578
โค้ดที่ตั้งอุปกรณ์	580
การตั้งค่า iSCSI	584
การจัดการฮอตปลั๊ก PCI	586
Multiple Path I/O	591
การกำหนดคอนฟิกอุปกรณ์ปลายทาง	616
เทปไดรฟ์	618
การสนับสนุนอุปกรณ์ USB	631
การแคชข้อมูลหน่วยเก็บข้อมูล	633
ชื่อล็อกอิน, IDs ระบบ, และรหัสผ่าน	641
Common Desktop Environment	649
การพิมพ์และงานพิมพ์	655
Live Partition Mobility with Host Ethernet Adapters	669
การย้ายที่อะแด็ปเตอร์สำหรับ DLPAR	671
อุปกรณ์ Loopback	672
AIX Event Infrastructure สำหรับ AIX และ AIX clusters-	
AHAFS	673
บทนำเกี่ยวกับ AIX Event Infrastructure	673
คอมโพเนนต์ AIX Event Infrastructure	673
การตั้งค่า AIX Event Infrastructure	676
มุมมองระดับสูงของวิธีการทำงานของ AIX Event Infrastructure	676
การใช้ AIX Event Infrastructure	678
การมอนิเตอร์เหตุการณ์	678
โปรติวเซอร์เหตุการณ์ที่กำหนดไว้ล่วงหน้า	690
เหตุการณ์คลัสเตอร์	707
โปรติวเซอร์เหตุการณ์ที่กำหนดไว้ล่วงหน้า producers สำหรับอินสแตนซ์ Cluster Aware AIX	708
คำประกาศ	719
ข้อควรพิจารณาเกี่ยวกับนโยบายความเป็นส่วนตัว	721
เครื่องหมายการค้า	721
ดัชนี	723

เกี่ยวกับเอกสารนี้

เอกสารนี้มีขึ้นสำหรับผู้ใช้และผู้ดูแลระบบเพื่อให้ข้อมูลที่ครบถ้วนซึ่งอาจส่งผลต่อการเลือกตัวเลือกต่างๆ ของคุณเมื่อดำเนินการปฏิบัติงานอย่างเช่น การสำรองและกู้คืนระบบ การจัดการที่จัดเก็บแบบฟิลิคัลและโลจิคัล การปรับขนาดพื้นที่เพจที่เหมาะสมและอื่นๆ คู่มือนี้ แสดงข้อมูลทั้งหมดเกี่ยวกับวิธีการทำภารกิจต่างๆ เช่น การจัดการ โลจิคัลวอลุ่ม หน่วยเก็บ และรีซอร์ส ผู้ใช้ระบบสามารถศึกษาวิธีการ ทำภารกิจต่างๆ เช่น การรันคำสั่ง การจัดการโปรเซส การจัดการ ไฟล์และไดเรกทอรี และการพิมพ์พื้นฐาน

ประโยชน์ของหัวข้ออื่นๆ ต่อผู้ใช้และผู้ดูแลระบบได้แก่ การสร้างและการปรับขนาดพื้นที่การสลับเพจ การจัดการหน่วยความจำเสมือน การสำรอง ข้อมูล และการเรียกคืนระบบ การจัดการฮาร์ดแวร์ และ pseudodevices, การใช้ System Resource Controller (SRC), การรักษาความปลอดภัยไฟล์, การใช้ ลีโอบันทิกที่เก็บข้อมูล, การกำหนดไฟล์สภาวะแวดล้อมเอง และการเขียนเซลล์สคริปต์ เอกสารนี้ยังมีอยู่บนซีดีเอกสารคู่มือที่มากับ ระบบปฏิบัติการ

การไฮไลต์

มีการใช้ระเบียนการไฮไลต์ต่อไปนี้ในเอกสารนี้:

Bold	ระบุคำสั่ง รุทินย่อย คีย์เวิร์ด ไฟล์ โครงสร้าง ไดเรกทอรี และรายการอื่นๆ ที่มีชื่อที่ได้ถูกกำหนดไว้โดยระบบ และยังมีระบุอีก
<i>Italics</i>	อบเจกต์แบบกราฟิก เช่น ปุ่ม เลเบล และไอคอนที่ผู้ใช้เลือก
Monospace	ระบุพารามิเตอร์ ที่มีชื่อหรือค่าจริงที่จะถูกจัดหาโดยผู้ใช้
	ระบุตัวอย่างของค่าข้อมูลที่ระบุเฉพาะ ตัวอย่างของข้อความที่ดูคล้ายกับที่คุณอาจมองเห็นจากที่แสดง ระบุตัวอย่างของส่วนของโค้ดโปรแกรมที่ดูคล้ายกับที่คุณอาจเขียนในฐานะเป็นโปรแกรมเมอร์ ระบุข้อความจากระบบ หรือข้อมูลที่ควรระวังพิมพ์

การคำนึงถึงขนาดตัวพิมพ์ใน AIX

ทุกสิ่งที่อยู่ในระบบปฏิบัติการ AIX® จะคำนึงถึงขนาดตัวพิมพ์ ซึ่งหมายความว่า ขนาดตัวพิมพ์จะแบ่งแยกกันระหว่างตัวอักษรตัวพิมพ์ใหญ่ และตัวพิมพ์เล็ก ตัวอย่างเช่น คุณสามารถใช้คำสั่ง ls เพื่อแสดงรายการไฟล์ ถ้าคุณพิมพ์ LS ระบบจะตอบกลับคำสั่งนั้นว่า is not found นอกจากนั้น FILEA, FiLea และ filea คือชื่อไฟล์สามชื่อที่แตกต่างกัน แม้ว่า ชื่อเหล่านั้นจะอยู่ในไดเรกทอรีเดียวกัน เพื่อหลีกเลี่ยงสาเหตุของการดำเนินการที่ไม่ต้องการดำเนินการ ตรวจสอบให้แน่ใจว่า คุณใช้กรณีที่ต้องการ

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

การจัดการระบบปฏิบัติการและอุปกรณ์

ผู้ดูแลระบบและผู้ใช้สามารถศึกษาวิธีการทำภารกิจต่างๆ เช่น การรันคำสั่ง การจัดการโปรเซส การจัดการไฟล์ และไดเรกทอรี การทำสำเนาสำรองและการเรียกคืนระบบ การจัดการหน่วยเก็บ ฟิสิคัลและโลจิคัล และการพิมพ์พื้นฐาน

หัวข้ออื่นที่เป็นประโยชน์สำหรับผู้ดูแลระบบได้แก่ การสร้างและการปรับขนาดพื้นที่การเลื่อนหน้า การจัดการหน่วย ความจำเสมือน การทำสำเนาสำรองและการเรียกคืนระบบ การจัดการฮาร์ดแวร์และอุปกรณ์เทียม การใช้ System Resource Controller (SRC), การรักษาความปลอดภัยไฟล์ การใช้สื่อบันทึกหน่วยเก็บ การกำหนดไฟล์สภาวะแวดล้อม และการเขียนเซลล์สคริปต์ หัวข้อนี้ยังมีอยู่ในแผ่นซีดีเอกสารคู่มือที่จัดส่งมาพร้อมกับ ระบบปฏิบัติการด้วย

มีอะไรใหม่ในระบบปฏิบัติการและการจัดการอุปกรณ์

อ่านข้อมูลใหม่หรือข้อมูลที่ถูกเปลี่ยนแปลงอย่างมากสำหรับชุดเรื่องระบบปฏิบัติการและการจัดการอุปกรณ์

วิธีการดู มีอะไรใหม่หรือมีอะไรที่เปลี่ยนแปลง

ในไฟล์ PDF นี้ คุณอาจเห็นแถบการแก้ไข (I) ในขอบด้านซ้ายเพื่อระบุข้อมูลใหม่และที่เปลี่ยนแปลง

October 2016

ข้อมูลต่อไปนี้เป็นสรุปของอัปเดตที่ทำให้กับชุดของหัวข้อนี้:

- เพิ่มข้อมูลเกี่ยวกับ “การสนับสนุนการเรียกคืนพื้นที่สำหรับ Logical Volume Storage” ในหน้า 425 ในหัวข้อ Logical Volume Storage

พฤษภาคม 2016

ข้อมูลต่อไปนี้เป็นสรุปของอัปเดตที่ทำให้กับชุดของหัวข้อนี้:

- เพิ่มข้อจำกัดเกี่ยวกับการแคชฝั่งเซิร์ฟเวอร์ของข้อมูลหน่วยเก็บข้อมูลในหัวข้อ “ข้อจำกัดสำหรับการแคชข้อมูลหน่วยเก็บข้อมูล” ในหน้า 634

กุมภาพันธ์ 2016

ข้อมูลต่อไปนี้เป็นสรุปของอัปเดตที่ทำให้กับชุดของหัวข้อนี้:

- เพิ่มข้อมูลเกี่ยวกับการแคชฝั่งเซิร์ฟเวอร์ของข้อมูลหน่วยเก็บข้อมูลในชุดหัวข้อ “การแคชข้อมูลหน่วยเก็บข้อมูล” ในหน้า 633

October 2014

ข้อมูลต่อไปนี้เป็นสรุปของอัปเดตที่ทำให้กับชุดของหัวข้อนี้:

- อัปเดตรายละเอียดปลีกย่อยต่างๆ เกี่ยวกับกระบวนการบูตในหัวข้อ “เฟสเฟิร์มแวร์” ในหน้า 19, “การบูตระบบ” ในหน้า 20, “ระบบไฟล์ RAM” ในหน้า 21 และ “โปรเซสบูตการบำรุงรักษา” ในหน้า 20

การจัดการระบบปฏิบัติการ

คุณสามารถใช้คำสั่งเพื่อจัดการกับสตาร์ทอัพระบบและสำเนาสำรอง การปิด ระบบ เซลล์ระบบและสภาวะแวดล้อม รีซอร์สระบบ และส่วนที่แตกต่างอื่น ของ AIX

การจัดการระบบปฏิบัติการเป็นภารกิจของแต่ละบุคคล ซึ่งโดยปกติในเนื้อหา UNIX เรียกว่าเป็นผู้ดูแลระบบ โชคดีที่นักมีกิจกรรมของผู้ดูแลระบบเพียงสองสามกิจกรรมเท่านั้นที่มีลักษณะตรงไปตรงมาเพียงพอที่สมควรเรียกว่าการจัดการ คู่มือนี้และคู่มือที่เกี่ยวข้องมีการจัดทำขึ้นเพื่อช่วย ผู้ดูแลระบบซึ่งมีหน้าที่รับผิดชอบหลายด้าน

ระบบปฏิบัติการนี้แนะนำเสนอเวอร์ชันเฉพาะของระบบเองของการสนับสนุนการจัดการ ระบบ เพื่อให้ง่ายต่อการใช้งานและปรับปรุงความปลอดภัยและคุณภาพ

อินเตอร์เฟซการจัดการระบบที่มีอยู่

นอกเหนือจากการดูแลรักษาระบบบรรทัดคำสั่งตามปกติแล้ว ระบบปฏิบัติการนี้ยังจัดให้มีอินเตอร์เฟซ SMIT

ต่อไปนี้เป็นอินเตอร์เฟซ SMIT:

- System Management Interface Tool (SMIT) ซึ่งเป็นอินเตอร์เฟซผู้ใช้บน เมนูที่สร้างคำสั่งจากอ็อปชันซึ่งคุณเลือกและดำเนินการคำสั่งนั้น

ด้วย SMIT คุณสามารถ:

- ติดตั้ง อัปเดต และบำรุงรักษาซอฟต์แวร์
- กำหนดคอนฟิกอุปกรณ์
- กำหนดคอนฟิกหน่วยเก็บดิสก์เข้าในกลุ่มมวลและโลจิคัลวอลุ่ม
- จัดทำและขยายระบบไฟล์และพื้นที่การเลื่อนหน้า
- การจัดการกับผู้ใช้และกลุ่ม
- กำหนดคอนฟิกเครือข่ายและแอปพลิเคชันการสื่อสาร
- พิมพ์
- ทำการกำหนดปัญหา
- จัดตารางเวลางาน
- จัดการกับรีซอร์สระบบและเวิร์กโหลด
- จัดการกับสภาวะแวดล้อมของระบบ
- จัดการกับข้อมูลระบบคลัสเตอร์
- อินเตอร์เฟซผู้ใช้แบบกราฟิกที่มุ่งเน้น อ็อบเจกต์สนับสนุนภารกิจจัดการระบบเช่นเดียวกับ SMIT แต่ทำให้ภารกิจจัดการระบบง่ายขึ้นโดย:
 - การลดข้อผิดพลาดผู้ใช้ผ่านการตรวจสอบข้อผิดพลาดและการออกแบบโต้ตอบ
 - การนำเสนอโปรแกรมที่ละขั้นตอนสำหรับภารกิจใหม่หรือที่ซับซ้อน
 - การนำเสนออ็อปชันขั้นสูงสำหรับผู้ดูแลระบบที่มีประสบการณ์ยาวนาน
 - ช่วยให้เห็นภาพของข้อมูลที่ซับซ้อนหรือความสัมพันธ์ระหว่างอ็อบเจกต์ระบบ ได้ง่ายขึ้น
 - การมอนิเตอร์กิจกรรมระบบและการแจ้งเตือนผู้ดูแลระบบเมื่อเกิดเหตุการณ์ที่กำหนดไว้ล่วงหน้าขึ้น

- การนำเสนอวิธีใช้แบบอิงคอนเท็กซ์ ภาพรวม เคล็ดล็บ และลิงก์ไปยังเอกสาร คู่มือทางออนไลน์

ข้อมูลผลิตภัณฑ์สำคัญของซอฟต์แวร์

ข้อมูลเกี่ยวกับผลิตภัณฑ์ซอฟต์แวร์และอ็อปชันที่ติดตั้งได้มีการรักษาไว้ในฐานข้อมูล Software Vital Product Data (SWVPD)

SWVPD ประกอบด้วยชุดของคำสั่งและคลาสอ็อบเจกต์ Object Data Manager (ODM) สำหรับการดูแลรักษาข้อมูลผลิตภัณฑ์ซอฟต์แวร์ คำสั่ง SWVPD ถูกจัดเตรียมสำหรับผู้ใช้เพื่อเคียวรี (Ispp) และตรวจสอบ (Ippchk) ผลิตภัณฑ์ซอฟต์แวร์ที่ติดตั้ง คลาสอ็อบเจกต์ ODM กำหนดขอบเขตและรูปแบบของข้อมูลผลิตภัณฑ์ซอฟต์แวร์ที่ดูแล

คำสั่ง `install` ใช้ ODM เพื่อรักษาข้อมูลดังต่อไปนี้ในฐานข้อมูล SWVPD:

- ชื่อของผลิตภัณฑ์ซอฟต์แวร์ที่ติดตั้ง
- เวอร์ชันของผลิตภัณฑ์ซอฟต์แวร์
- ระดับรีลีสของผลิตภัณฑ์ซอฟต์แวร์ ซึ่งระบุการเปลี่ยนแปลงใน อินเทอร์เน็ตเอกสารเขียนโปรแกรมภายนอกของผลิตภัณฑ์ซอฟต์แวร์
- ระดับโมดิฟิเคชันของผลิตภัณฑ์ซอฟต์แวร์ ซึ่งระบุการเปลี่ยนแปลงที่ไม่กระทบต่อ อินเทอร์เน็ตเอกสารเขียนโปรแกรมภายนอกของผลิตภัณฑ์ซอฟต์แวร์
- ระดับโปรแกรมแก้ไขของผลิตภัณฑ์ซอฟต์แวร์ ซึ่งระบุการอัปเดตเล็กน้อยที่สร้างขึ้นในระดับโมดิฟิเคชันปกติในภายหลัง
- ฟิลด์หมายเลขประจำโปรแกรมแก้ไข
- ชื่อ, checksums, และขนาดของไฟล์ที่ประกอบขึ้นเป็นผลิตภัณฑ์ซอฟต์แวร์ หรืออ็อปชัน
- สถานะการติดตั้งของผลิตภัณฑ์ซอฟต์แวร์: ใช้อยู่, ใช้งานแล้ว, committing, committed, ปฏิเสธ, หรือเสียหาย

อัปเดตระบบปฏิบัติการ

แพ็คเกจระบบปฏิบัติการแบ่งออกเป็นชุดไฟล์ต่างๆ โดยที่แต่ละชุด มีกลุ่มของไฟล์ที่จัดส่งให้แก่ลูกค้าและมีความเกี่ยวข้องกัน ทางตรรกะ ชุดไฟล์แต่ละชุดสามารถติดตั้งและอัปเดตแยกต่างหากกัน

การแก้ไขในชุดไฟล์มีการติดตามโดยใช้ระดับเวอร์ชัน รีลีส การบำรุงรักษา และระดับการแก้ไข (VRMF) ตามระเบียบที่ใช้กันมาในแต่ละครั้งที่ใช้อัปเดตชุดไฟล์ AIX จะมีการปรับระดับการแก้ไข ในแต่ละครั้งที่ใช้แพ็คเกจการบำรุงรักษา AIX หรือระดับเทคโนโลยี จะมีการปรับระดับโมดิฟิเคชัน และรีเซ็ตระดับการแก้ไขเป็นศูนย์ การติดตั้งเริ่มต้นของเวอร์ชัน AIX เช่น AIX 6.1 ถูกเรียกว่าการติดตั้งฐาน ระบบปฏิบัติการนำเสนออัปเดตในคุณลักษณะและ การทำงาน ซึ่งอาจมีการรวมแพ็คเกจเป็นแพ็คเกจการบำรุงรักษา ระดับ เทคโนโลยี โปรแกรมแก้ไขชั่วคราว (PTF) หรือเซอร์วิสแพ็คเกจ (กลุ่มของ PTF)

แพ็คเกจการบำรุงรักษาและระดับเทคโนโลยี

แพ็คเกจการบำรุงรักษาและระดับเทคโนโลยีนำเสนอฟังก์ชันใหม่ที่มีไว้เพื่ออัปเดตรีลีส ส่วนการบำรุงรักษาของ VRMF มีการอัปเดต ในแพ็คเกจการบำรุงรักษา ตัวอย่างเช่น แพ็คเกจการบำรุงรักษาแรก สำหรับ AIX 6.1 คือ 6.1.1.0; แพ็คเกจที่สองคือ 6.1.2.0 และอื่นๆ เมื่อต้องการแสดงรายการแพ็คเกจการบำรุงรักษา ให้ใช้คำสั่ง `oslevel -r` เมื่อต้องการกำหนดแพ็คเกจการบำรุงรักษา หรือระดับเทคโนโลยีที่ติดตั้งบนระบบเฉพาะ ให้พิมพ์:

```
oslevel
```

เมื่อต้องการ กำหนดชุดไฟล์ที่จำเป็นต้องอัปเดตสำหรับระบบให้ถึง แพ็คเกจการบำรุงรักษาหรือระดับเทคโนโลยีเฉพาะ (ในตัวอย่างนี้ 6.1.1.0) ให้ใช้คำสั่งต่อไปนี้:

```
oslevel -l 6.1.1.0
```

เมื่อต้องการดูว่า แพ็กเกจการบำรุงรักษาหรือระดับเทคโนโลยีใด ถูกติดตั้งไว้ (ในตัวอย่างนี้ 6100-02) ให้ใช้คำสั่งต่อไปนี้:

```
oslevel -r 6100-02
```

เมื่อต้องการดูว่า ชุดไฟล์ใดที่จำเป็นต้องอัปเดตสำหรับระบบให้ถึงแพ็กเกจการบำรุงรักษาหรือระดับเทคโนโลยี 6100-02 ให้ใช้คำสั่งต่อไปนี้:

```
oslevel -r1 6100-02
```

เมื่อต้องการกำหนดแพ็กเกจการบำรุงรักษาหรือระดับเทคโนโลยีของชุดไฟล์เฉพาะ (ในตัวอย่างนี้คือ bos.mp) ให้ใช้คำสั่งต่อไปนี้:

```
ls1pp -L bos.mp
```

PTFs ระหว่างรีลีส คุณอาจได้รับ PTFs เพื่อแก้ไขหรือป้องกันปัญหา เฉพาะ การติดตั้งบางรายการอาจต้องการ PTFs ที่มีอยู่บางส่วน ทั้งหมด หรืออาจไม่ต้องการ

แพ็กเกจการบำรุงรักษาที่แนะนำ

แพ็กเกจการบำรุงรักษาที่แนะนำคือชุดของ PTFs ระหว่างระดับเทคโนโลยี ที่ผ่านการทดสอบโดยรวมร่วมกันแล้ว และแนะนำสำหรับ การบำรุงรักษาเชิงป้องกัน

โปรแกรมแก้ไขชั่วคราว

โปรแกรมแก้ไขชั่วคราวคล้ายกับ PTF แต่โดยปกติแล้ว มีการนำเสนอเมื่อ PTF ไม่พร้อมใช้งาน โปรแกรมแก้ไขชั่วคราวยังมีการรีลีสเมื่อ PTF จะอัปเดต ระบบเป็นระดับการบำรุงรักษาถัดไปและผู้ใช้อาจต้องการให้ระบบของตนคงอยู่ที่ระดับปัจจุบัน

เมื่อต้องการกำหนดเวอร์ชันและระดับรีลีส, แพ็กเกจการบำรุงรักษา, ระดับเทคโนโลยี และระดับเซอร์วิสแพ็ก เช่นเดียวกับชุดไฟล์ ที่จำเป็นต้องอัปเดตให้ถึงระดับเฉพาะ โปรดดูที่คำสั่ง `oslevel` และ `ls1pp` in *Commands Reference*.

สตาร์ทอัพระบบ

เมื่อระบบปฏิบัติการพื้นฐานเริ่มต้น ระบบจะเริ่มต้นชุดของภารกิจ ที่ซับซ้อน ภายใต้สภาพปกติ ภารกิจเหล่านี้มีการทำให้โดยอัตโนมัติ

มีสถานการณ์บางอย่างซึ่งคุณต้องการสั่งให้ระบบรีบูต ตัวอย่างเช่น เพื่อให้ระบบรับรู้ซอฟต์แวร์ที่ติดตั้งใหม่ เพื่อรีเซ็ต อุปกรณ์รอบข้าง เพื่อทำการกิจการบำรุงรักษาอื่น เช่น การตรวจสอบ ระบบไฟล์ หรือเพื่อกู้คืนจาก hang หรือความเสียหายของระบบ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ โพรซีเจอร์เหล่านี้ ให้ดูที่:

งานที่เกี่ยวข้อง:

“การสร้างอิมเมจเปิดเครื่องใหม่ที่เสียหายขั้นใหม่” ในหน้า 38

โพรซีเจอร์ต่อไปนี้อธิบายวิธีการระบุอิมเมจเปิดเครื่องใหม่ที่เสียหาย และสร้างขึ้นใหม่

การจัดการสตาร์ทอัพระบบ

มีหลายสถานการณ์จำลองซึ่งคุณอาจพบเมื่อคุณ ต้องการบูตหรือรีบูตระบบของคุณ เมื่อต้องการปิดหรือรีบูตระบบของคุณ คุณสามารถใช้คำสั่ง `shutdown` หรือ `reboot` ใดๆอย่างหนึ่ง คุณควร ใช้คำสั่ง `shutdown` เมื่อผู้ใช้หลายๆ คนล็อกออน เข้าสู่ระบบ

การรีบูตระบบที่กำลังรัน:

เนื่องจากโปรเซสอาจกำลังรันสิ่งที่ควรยุติที่ละน้อยเกินกว่าที่ **reboot** ทำได้ ดังนั้น **shutdown** จึงเป็น เมธอดที่ดีกว่าสำหรับระบบทั้งหมด

มีสองเมธอดสำหรับการปิดและรีบูตระบบของคุณคือ **shutdown** และ **reboot** ใช้เมธอด **shutdown** เสมอเมื่อผู้ใช้หลายๆ คน ล็อกออนเข้าสู่ระบบ

ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
การรีบูตระบบหลายผู้ใช้	smit shutdown	shutdown -r
การรีบูตระบบผู้ใช้เดียว	smit shutdown	shutdown -r หรือ reboot

การรีบูตระบบที่ไม่ตอบสนองแบบรีโมต:

สิ่งอำนวยความสะดวกกรีโมตรีบูตช่วยให้สามารถรีบูตระบบผ่านทาง พอร์ตระบบ native (แบบรวม)

พอร์ตระบบแบบรวมของ POWER5 คล้ายกับพอร์ตอนุกรม ยกเว้นว่าพอร์ตระบบพร้อมใช้งาน สำหรับฟังก์ชันที่สนับสนุน โดยเฉพาะเท่านั้น

ระบบถูกรีบูตเมื่อได้รับ **reboot_string** ที่ พอร์ต สิ่งอำนวยความสะดวกนี้มีประโยชน์เมื่อระบบไม่ตอบสนองแต่ สามารถเซอร์วิสการขัดจังหวะพอร์ตระบบได้ รีโมตรีบูตสามารถเปิดใช้งานได้ บนพอร์ตระบบ native หนึ่งพอร์ตเท่านั้นในแต่ละครั้ง ผู้ใช้ต้องระบุข้อมูลรักษาความปลอดภัยภายนอก ของตนเองสำหรับพอร์ต สิ่งอำนวยความสะดวกนี้วันที่คลาสการขัดจังหวะอุปกรณ์สูงสุด และความล้มเหลวของ UART (Universal Asynchronous Receive/Transmit) ในการล้างบัฟเฟอร์การส่งผ่านอย่างรวดเร็วอาจส่งผลให้อุปกรณ์อื่น สูญเสียข้อมูลได้ถ้าบัฟเฟอร์โอเวอร์โฟลว์ในระหว่างเวลานี้ ขอแนะนำให้ผู้ใช้งานอำนวยความสะดวกนี้เพื่อรีบูตเครื่องที่อาจจะ hung และไม่สามารถล็อกอินแบบรีโมตได้เท่านั้น ระบบไฟล์จะ *ไม่* ถูกซิงโครไนซ์ และมีโอกาสสูญเสียข้อมูลบางส่วนซึ่งยังไม่ได้ฟลัช ขอแนะนำ อย่างยิ่งว่าเมื่อมีการเปิดใช้งานรีโมตรีบูต ไม่ควรใช้พอร์ตสำหรับวัตถุประสงค์อื่นใด โดยเฉพาะอย่างยิ่ง การโอนย้ายไฟล์ เพื่อป้องกันรีบูต โดยไม่ได้ตั้งใจ

แอ็ตทริบิวต์พอร์ตระบบ native สองรายการควบคุมการดำเนินงานของรีโมตรีบูต

reboot_enable

บ่งชี้ว่าพอร์ตนี้มีการเปิดใช้งานเพื่อรีบูตเครื่องเมื่อได้รับ รีโมต **reboot_string** หรือไม่ และถ้าใช่ จะใช้ดั้มพีร์ระบบก่อน การรีบูตหรือไม่

- no - Indicates remote reboot is disabled
- reboot - Indicates remote reboot is enabled
- dump - Indicates remote reboot is enabled, and prior to rebooting a system dump will be taken on the primary dump device

reboot_string

ระบุรีโมต **reboot_string** ที่พอร์ตอนุกรมจะสแกนหา เมื่อมีการเปิดใช้งานคุณลักษณะรีโมตรีบูต เมื่อมีการเปิดใช้งานคุณลักษณะรีโมตรีบูต และได้รับ **reboot_string** บนพอร์ต อีกซระ > จะถูก ส่งผ่านและระบบพร้อมที่จะรีบูต ถ้าได้รับอีกซระ 1 ระบบ จะถูกรีบูต อีกซระอื่นใดที่ไม่ใช่ 1 จะยกเลิก โปรเซสรีบูต **reboot_string** มีความยาวสูงสุด 16 อักซระ และต้องไม่มีช่องว่าง, เครื่องหมายจุดคู่, เครื่องหมายเท่ากับ, null, บรรทัดใหม่, หรืออักซระ Ctrl-\

รีโมตรีบูตสามารถเปิดใช้งานได้โดยใช้ SMIT หรือบรรทัดคำสั่ง สำหรับ SMIT อาจใช้พาธ สภาวะแวดล้อมของระบบ -> จัดการสิ่งอำนวยความสะดวกรีโมตรีบูต สำหรับ TTY ที่กำหนดคอนฟิก หรือเมื่อกำหนดคอนฟิก TTY ใหม่ รีโมตรีบูต อาจเปิดใช้งานได้จากเมนู เพิ่ม TTY หรือ เปลี่ยน/แสดง ลักษณะของ TTY เมนูเหล่านี้สามารถเข้าถึงได้ผ่านทาง พาธ อุปกรณ์ -> TTY

จากบรรทัดคำสั่ง ใช้คำสั่ง `mkdev` หรือ `chdev` เพื่อเปิดใช้งานรีโมตรีบูต ตัวอย่างเช่น คำสั่งต่อไปนี้เปิดใช้งาน รีโมตรีบูต (ด้วยอ็อพชันดัมพ์) และตั้งค่ารีบูตสตริงเป็น `ReBoOtMe` บน `tty1`

```
chdev -l tty1 -a remreboot=dump -a reboot_string=ReBoOtMe
```

ตัวอย่างนี้เปิดใช้งานรีโมตรีบูตบน `tty0` ด้วย `reboot_string` ปัจจุบันใน ฐานข้อมูลเท่านั้น (จะมีผลบังคับใช้ในรีบูตครั้งถัดไป)

```
chdev -P -l tty0 -a remreboot=reboot
```

หากกำลังใช้ `tty` เป็นพอร์ตปกติ คุณจะต้องใช้คำสั่ง `pdisable` ก่อนเปิดใช้งานรีโมตรีบูต คุณอาจใช้ `penable` เพื่อเปิดใช้งานพอร์ตอีกครั้งหลังจากนั้น

ข้อมูลที่เกี่ยวข้อง:

ความแตกต่างของฟังก์ชันระหว่างพอร์ตระบบและพอร์ตอนุกรม

การบูตจากฮาร์ดดิสก์สำหรับการบำรุงรักษา:

คุณสามารถบูตเครื่องในโหมดการบำรุงรักษาจากฮาร์ดดิสก์ได้

ข้อกำหนดเบื้องต้น

สื่อบันทึกที่บูตและถอดออกได้ (เทปหรือ ซีดีรอม) ต้องไม่อยู่ในไดรฟ์ นอกจากนี้ โปรดอ้างอิงเอกสารคู่มือฮาร์ดแวร์เพื่อดูคำแนะนำเฉพาะในการเปิดใช้งานบูตโหมดการบำรุงรักษาบนรุ่น เฉพาะของคุณ

โพรซีเจอร์

เมื่อต้องการบูตเครื่องในโหมดการบำรุงรักษาจากฮาร์ดดิสก์:

1. เมื่อต้องการรีบูตให้ปิดเครื่อง แล้วเปิดใหม่ หรือกดปุ่มรีเซ็ต
2. กดลำดับคีย์เพื่อรีบูตในโหมดการบำรุงรักษาตามที่ระบุไว้ในเอกสารคู่มือฮาร์ดแวร์ของคุณ
3. เครื่องจะบูตถึงจุดที่มีการกำหนดคอนฟิก อุปกรณ์คอนโซล หากมีดัมพ์ระบบที่จำเป็นต้องดึงข้อมูล เมนูดัมพ์ระบบจะแสดงขึ้นบนคอนโซล

หมายเหตุ:

- a. ถ้าคอนโซลล่มเหลวในการกำหนดคอนฟิกเมื่อมีดัมพ์ที่จะดึงข้อมูล ระบบจะไม่ตอบสนอง ต้องบูตระบบจากสื่อที่ถอดออกได้เพื่อ ดึงข้อมูลดัมพ์
 - b. ระบบทำการดัมพ์ไปยังอุปกรณ์ดัมพ์ที่ระบุไว้โดยอัตโนมัติ เมื่อกดปุ่มรีเซ็ต เมื่อต้องการเปลี่ยนการกำหนดอุปกรณ์ดัมพ์หลัก หรือรองในระบบที่รันอยู่ โปรดดูที่คำสั่ง `sysdumpdev`
4. หากไม่มีดัมพ์ระบบ หรือถ้าถูกคัดลอกแล้ว คำแนะนำการดำเนินงาน วิเคราะห์จะแสดงขึ้น กด Enter เพื่อทำเมนู การเลือกฟังก์ชัน ต่อไป
 5. จากเมนู การเลือกฟังก์ชัน คุณสามารถ เลือกโหมดการวิเคราะห์หรือผู้ใช้เดี่ยว:

Single-User Mode: เมื่อต้องการทำการบำรุงรักษาในสถานะแวดล้อมผู้ใช้เดี่ยว ให้เลือก อ็อปชันนี้ (อ็อปชัน 5) ระบบทำการบูตต่อไปและเข้าสู่โหมด ผู้ใช้เดี่ยว การบำรุงรักษาที่ระบบจะต้องอยู่ในโหมดสแตนด์อะโลนสามารถทำได้ใน โหมดนี้ และสามารถรีเซ็ตคำสั่ง **bosboot** ได้ ถ้าต้องการ

ข้อมูลที่เกี่ยวข้อง:

การเริ่มต้นดัมพ์ระบบ

การบูตระบบที่เสียหาย:

ในบางอินสแตนซ์ คุณอาจต้องบูตระบบที่หยุด (เสียหาย) โดยไม่ได้ปิดอย่างถูกต้อง

ข้อกำหนดเบื้องต้นสำหรับโพรซีเดอร์นี้คือ:

- ระบบของคุณเสียหายและไม่ได้ปิดอย่างถูกต้องเนื่องจากสภาพไม่ปกติ
- ระบบของคุณถูกปิด

โพรซีเดอร์นี้ครอบคลุมข้อมูลพื้นฐานเกี่ยวกับวิธีการบูต ถ้าระบบของคุณไม่สามารถกู้คืนจากความเสียหายได้ ปฏิบัติตามขั้นตอนต่อไปนี้:

1. ตรวจสอบให้แน่ใจว่าฮาร์ดแวร์และอุปกรณ์รอบข้างทั้งหมดมีการเชื่อมต่ออย่างถูกต้อง
2. เปิดอุปกรณ์รอบข้างทั้งหมด
3. เฝ้านำจอเพื่อดูข้อมูลเกี่ยวกับการวิเคราะห์ฮาร์ดแวร์อัตโนมัติ
 - a. หากการทดสอบวิเคราะห์ฮาร์ดแวร์ใดๆ ไม่สำเร็จ ให้อ้างอิง เอกสารคู่มือฮาร์ดแวร์
 - b. หากการทดสอบวิเคราะห์ฮาร์ดแวร์ทั้งหมดสำเร็จ ให้เปิดยูนิท ระบบ

การรีเซ็ตรหัสผ่านรากที่ไม่รู้จัก:

โพรซีเดอร์ต่อไปนี้อธิบายวิธีการกู้คืนการเข้าถึงสิทธิ์ราก เมื่อไม่มีหรือไม่ทราบรหัสผ่านรากของระบบ

โพรซีเดอร์ต่อไปนี้ต้องการดาวนโหลดระบบบางเวลา ถ้าเป็นไปได้ ให้จัดตารางเวลา ดาวนโหลดของคุณเป็นเวลาที่ไม่ส่งผลกระทบต่อเวิร์กโพลของคณน้อยที่สุด เพื่อป้องกัน การสูญหายของข้อมูลหรือการทำงาน

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

1. ใส่สื่อบันทึกผลิตภัณฑ์ที่เป็นเวอร์ชันและระดับเดียวกันกับ การติดตั้งปัจจุบันเข้าในไดรฟ์ที่เหมาะสม
2. เปิดเครื่อง
3. เมื่อนำจอของไอคอนปรากฏขึ้น หรือเมื่อคุณได้ยินเสียงบี๊บสองครั้ง ให้กดปุ่ม F1 ซ้ำๆ จนกว่าเมนู เซอร์วิสการจัดการระบบ ปรากฏขึ้น
4. เลือก **มัลติบูต**
5. เลือก **ติดตั้งจาก**
6. เลือกอุปกรณ์ที่มีสื่อบันทึกผลิตภัณฑ์ แล้วเลือก **ติดตั้ง**
7. เลือกไอคอนเวอร์ชัน AIX
8. กำหนดระบบปัจจุบันของคุณเป็นคอนโซลระบบโดยกดปุ่ม F1 แล้วกด Enter
9. เลือกหมายเลขของภาษาที่ต้องการ และกด Enter

10. เลือก เริ่มต้นโหมดการบำรุงรักษาสำหรับการกู้คืนระบบ โดย พิมพ์ 3 และกด Enter
11. เลือก เข้าสู่กลุ่มวอลุ่มราก ข้อความแสดงขึ้นเพื่ออธิบายว่าคุณจะไม่สามารถกลับไปยังเมนู การติดตั้ง โดยไม่ได้รีบูตถ้าคุณเปลี่ยนกลุ่มวอลุ่มราก ณ จุดนี้
12. พิมพ์ 0 และกด Enter
13. พิมพ์หมายเลขของกลุ่มวอลุ่มที่เหมาะสมจากรายการ และ กด Enter
14. เลือก เข้าสู่กลุ่มวอลุ่มนี้และเริ่มต้นเซลล์โดย พิมพ์ 1 และกด Enter
15. ที่พร้อมท์ # (เครื่องหมายตัวเลข) ให้พิมพ์ คำสั่ง passwd ที่พร้อมท์บรรทัดคำสั่งเพื่อรีเซ็ต รหัสผ่านราก ตัวอย่างเช่น:

```
# passwd
Changing password for "root"
root's New password:
Enter the new password again:
```
16. เมื่อต้องการเขียนทุกสิ่งจากบัพเฟอร์ลิ่งในฮาร์ดดิสก์และรีบูต ระบบ ให้พิมพ์ดังต่อไปนี้:

```
sync;sync;sync;reboot
```

เมื่อนำจอล็อกอินปรากฏขึ้น ขณะนี้ รหัสผ่านซึ่งคุณตั้งค่าในขั้นตอนที่ 15 ควรช่วยให้เข้าถึง สิทธิรากได้

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง passwd d

คำสั่ง reboot

การบูตระบบโดยใช้กราฟิก planar:

หากเครื่องมีการติดตั้งพร้อมกับระบบย่อยกราฟิก planar เพียงอย่างเดียว และในเวลาต่อมา มีการเพิ่มอะแดปเตอร์กราฟิกเพิ่มเติมลงในระบบ สถานการณ์ ต่อไปนี้จะเกิดขึ้น:

1. มีการเพิ่มอะแดปเตอร์กราฟิกใหม่ลงในระบบ และติดตั้งซอฟต์แวร์ไดรเวอร์ อุปกรณ์ที่เชื่อมโยง
2. ระบบถูกรีบูต และอย่างใดอย่างหนึ่งต่อไปนี้เกิดขึ้น:
 - a. หากมีการกำหนดคอนโซลระบบเป็น /dev/lft0 (lscns แสดง ข้อมูลนี้) จะมีการขอให้ผู้ใช้เลือกจอแสดงผลที่จะใช้เป็นคอนโซลระบบ ในเวลารีบูต ถ้าผู้ใช้เลือกอะแดปเตอร์กราฟิก (อุปกรณ์ที่ไม่ใช่ TTY) อะแดปเตอร์กราฟิกนั้นจะกลายเป็นจอแสดงผลฟอลต์ใหม่ด้วย ถ้าผู้ใช้เลือกอุปกรณ์ TTY แทนอุปกรณ์ LFT ล็อกอินระบบจะไม่ปรากฏขึ้น รีบูตอีกครั้ง และหน้าจอ ล็อกอิน TTY จะแสดงขึ้น มีการสมมติว่าถ้าผู้ใช้เพิ่มอะแดปเตอร์กราฟิกเพิ่มเติม ลงในระบบ และคอนโซลระบบเป็นอุปกรณ์ LFT ผู้ใช้จะไม่เลือกอุปกรณ์ TTY เป็นคอนโซลระบบ
 - b. หากคอนโซลระบบมีการกำหนดเป็น TTY ผลคือในเวลารีบูต อะแดปเตอร์จอแสดงผลที่เพิ่มเข้ามาใหม่จะกลายเป็นจอแสดงผลฟอลต์

หมายเหตุ: เนื่องจาก TTY เป็นคอนโซลระบบ TTY จึงยังคงเป็นคอนโซลระบบต่อไป

3. หากคอนโซลระบบเป็น /dev/lft0 ผลคือหลังจากรีบูต DPMS จะถูกปิดใช้งานเพื่อแสดงข้อความการเลือกคอนโซลระบบบนหน้าจอ ในช่วงเวลาที่ไม่จำกัด เมื่อต้องการเปิดใช้งาน DPMS อีกครั้ง ให้รีบูตระบบ อีกครั้งหนึ่ง

การใช้การดำเนินการสคริปต์ระดับ:

สคริปต์ระดับการรันช่วยให้ผู้ใช้สามารถเริ่มต้นและหยุดแอ็พพลิเคชันที่เลือก ในขณะที่เปลี่ยนระดับการรัน

วางสคริปต์ระดับการรันในไดเรกทอรีย่อยของ /etc/rc.d ที่ใช้เฉพาะสำหรับระดับการรัน:

- /etc/rc.d/rc2.d
- /etc/rc.d/rc3.d
- /etc/rc.d/rc4.d
- /etc/rc.d/rc5.d
- /etc/rc.d/rc6.d
- /etc/rc.d/rc7.d
- /etc/rc.d/rc8.d
- /etc/rc.d/rc9.d

/etc/rc.d/rc จะรันสคริปต์ที่พบใน ไดเรกทอรีที่ระบุเมื่อระดับการรันเปลี่ยนไป - โดยรันสคริปต์หยุดแอ็พพลิเคชัน ก่อนตามด้วยการรันสคริปต์เริ่มต้นแอ็พพลิเคชัน

หมายเหตุ: สคริปต์ที่ขึ้นต้นด้วย K เป็นสคริปต์หยุด ในขณะที่สคริปต์ ที่ขึ้นต้นด้วย S เป็นสคริปต์เริ่มต้น

การแก้ไขไฟล์ /etc/inittab:

มีอยู่สี่คำสั่งสำหรับการแก้ไขเร็กคอร์ดในไฟล์ etc/inittab

Adding records – mkitab command

เมื่อต้องการเพิ่ม เร็กคอร์ดลงในไฟล์ /etc/inittab ให้พิมพ์ดังต่อไปนี้ที่พร้อมต์คำสั่ง:

```
mkkitab Identifier:Run Level:Action:Command
```

ตัวอย่าง เช่น เมื่อต้องการเพิ่มเร็กคอร์ดสำหรับ tty2 ให้พิมพ์ดังต่อไปนี้ที่พร้อมต์คำสั่ง:

```
mkkitab tty002:2:respawn:/usr/sbin/getty /dev/tty2
```

ในตัวอย่างข้างบน:

ไอเท็ม

```
tty002
2
respawn
/usr/sbin/getty /dev/tty2
```

คำอธิบาย

ระบุอ็อบเจกต์ที่มีระดับการรันซึ่งคุณกำลังกำหนด
ระบุระดับการรันซึ่งโปรเซสนี้รัน
ระบุการดำเนินการที่คำสั่ง `init` ควรจะใช้สำหรับโปรเซสนี้
ระบุคำสั่งเชลล์ที่จะดำเนินการ

Changing records – chitab command

เมื่อต้องการ เปลี่ยนเร็กคอร์ดในไฟล์ /etc/inittab ให้พิมพ์ดังต่อไปนี้ที่พร้อมต์คำสั่ง:

```
chitab Identifier:Run Level:Action:Command
```

ตัวอย่าง เช่น เมื่อต้องการเปลี่ยนเร็กคอร์ดสำหรับ tty2 เพื่อให้โปรเซสนี้รันที่ระดับการรัน 2 และ 3 ให้พิมพ์:

```
chitab tty002:23:respawn:/usr/sbin/getty /dev/tty2
```

ใน ตัวอย่างข้างบน:

ไอเอ็ม
tty002
23
respawn
/usr/sbin/getty /dev/tty2

คำอธิบาย
ระบุออบเจกต์ที่มีระดับการรันซึ่งคุณกำลังกำหนด
ระบุระดับการรันซึ่งโปรเซสนี้รัน
ระบุการดำเนินการที่คำสั่ง `init` ควรจะใช้สำหรับโปรเซสนี้
ระบุคำสั่งเซลล์ที่จะดำเนินการ

Listing records – `lsitab` command

เมื่อต้องการแสดงรายการเร็กคอร์ดทั้งหมดในไฟล์ `/etc/inittab` ให้พิมพ์ดังต่อไปนี้พร้อมคำสั่ง:

```
lsitab -a
```

เมื่อต้องการแสดงรายการเร็กคอร์ดเฉพาะในไฟล์ `/etc/inittab` ให้พิมพ์:

```
lsitab Identifier
```

ตัวอย่าง เช่น เมื่อต้องการแสดงรายการเร็กคอร์ดสำหรับ `tty2` ให้พิมพ์: `lsitab tty2`

Removing records – `rmitab` command

เมื่อต้องการลบเร็กคอร์ดออกจากไฟล์ `/etc/inittab` ให้พิมพ์ดังต่อไปนี้พร้อมคำสั่ง:

```
rmitab Identifier
```

ตัวอย่าง เช่น เมื่อต้องการลบเร็กคอร์ดสำหรับ `tty2` ให้พิมพ์: `rmitab tty2`

หลักการที่เกี่ยวข้อง:

“ระดับการรันของระบบ” ในหน้า 16

ระดับการรันของระบบระบุสถานะระบบและกำหนดโปรเซสที่จะเริ่มต้น

การเรียกใช้ระบบที่ไม่ได้ใช้งานอยู่อีกครั้ง:

ระบบของคุณอาจไม่ได้ใช้งานอยู่เนื่องจากปัญหาฮาร์ดแวร์ ปัญหาซอฟต์แวร์ หรือทั้งสองปัญหา

โปรซีเดอร์นี้แนะนำคุณผ่านขั้นตอนต่างๆ ในการแก้ไขปัญหาและรีสตาร์ทระบบ หากระบบของคุณยังคงไม่ได้ใช้งานอยู่หลังจากทำโปรซีเดอร์นี้เสร็จแล้ว ให้อ้างอิงข้อมูลการระบุปัญหาในเอกสารคู่มือฮาร์ดแวร์ของคุณ

ใช้โปรซีเดอร์ต่อไปนี้เพื่อเรียกใช้ระบบที่ไม่ได้ใช้งานอยู่อีกครั้ง:

การตรวจสอบฮาร์ดแวร์:

มีหลายโปรซีเดอร์ซึ่งคุณสามารถใช้เพื่อตรวจสอบฮาร์ดแวร์ของคุณ

ตรวจสอบฮาร์ดแวร์ของคุณโดย:

การตรวจสอบไฟฟ้า:

หากไฟ เปิดเครื่อง บนระบบของคุณติดอยู่ให้ไป ยัง การตรวจสอบจอแสดงผลพาวเนลผู้ใช้งาน ข้างล่าง

หากไฟ เปิดเครื่อง บนระบบของคุณไม่ติด ให้ตรวจสอบว่าเครื่องเปิดอยู่ และเสียบปลั๊กระบบแล้ว

การตรวจสอบจอแสดงผลพาเนลผู้ใช้งาน:

หากระบบของคุณมีจอแสดงผลพาเนลผู้ใช้งานให้ตรวจสอบหาข้อความใดๆ

หากจอแสดงผลพาเนลผู้ใช้งานบนระบบของคุณว่างเปล่าให้ไปยัง การเรียกใช้จอแสดงผลหรือเทอร์มินัลของคุณ ด้านล่าง

หากจอแสดงผลพาเนลผู้ใช้งานบนระบบของคุณไม่ว่างเปล่าให้ไปยังคู่มือ เซอร์วิสสำหรับยูนิตของคุณเพื่อค้นหาข้อมูลเกี่ยวกับตำแหน่งในจอแสดงผลพาเนลผู้ใช้งาน

การเรียกใช้จอแสดงผลหรือเทอร์มินัลของคุณ:

ตรวจสอบชิ้นส่วนต่างๆ ของจอแสดงผลหรือเทอร์มินัลของคุณ ดังนี้:

- ตรวจสอบให้แน่ใจว่าสายเคเบิลจอแสดงผลต่อพ่วงเข้ากับจอแสดงผลและยูนิตระบบแน่นดีแล้ว
- ตรวจสอบให้แน่ใจว่าต่อพ่วงสายเคเบิลคีย์บอร์ดแน่นดีแล้ว
- ตรวจสอบให้แน่ใจว่าต่อพ่วงสายเคเบิลเมาส์แน่นดีแล้ว
- ตรวจสอบให้แน่ใจว่าจอแสดงผลเปิดอยู่และไฟ เปิดเครื่อง ติดอยู่
- ปรับตัวควบคุมความสว่างบนจอแสดงผล
- ตรวจสอบให้แน่ใจว่าค่าติดตั้งการสื่อสารของเทอร์มินัลถูกต้อง

หากระบบของคุณใช้งานอยู่ในขณะนี้ การตรวจสอบฮาร์ดแวร์ของคุณ ได้แก้ไขปัญหาแล้ว

งานที่เกี่ยวข้อง:

“การรีเซ็ตระบบ” ในหน้า 13

นอกเหนือจากการตรวจสอบฮาร์ดแวร์และการตรวจสอบโปรเซสแล้ว คุณสามารถรีเซ็ตระบบเพื่อเรียกใช้ระบบที่ไม่ทำงานอีกครั้ง

“การตรวจสอบโปรเซส”

โปรเซสที่หยุดหรือสะดุดอาจทำให้ระบบของคุณไม่ได้ใช้งานอยู่

การตรวจสอบโปรเซส:

โปรเซสที่หยุดหรือสะดุดอาจทำให้ระบบของคุณไม่ได้ใช้งานอยู่

ตรวจสอบโปรเซสระบบของคุณโดย:

1. การรีเซ็ตการเลื่อนหน้าจอบรรทัด
2. การใช้ลำดับคีย์ Ctrl+D
3. การใช้ลำดับคีย์ Ctrl+C
4. การลือกอินจากรีโมตเทอร์มินัลหรือโฮสต์
5. การสิ้นสุดโปรเซสที่สะดุดแบบรีโมต

การรีเซ็ตการเลื่อนหน้าจอบรรทัด:

รีเซ็ต การเลื่อนหน้าจอบรรทัดที่ถูกหยุดด้วยคีย์ Ctrl-S ตามลำดับโดยทำดังต่อไปนี้:

1. เรียกใช้งานหน้าต่างหรือเซลล์ที่มีโปรเซสปัญหา

- กดลำดับคีย์ Ctrl-Q เพื่อรีเซ็ตการเลื่อนหน้าจอ ลำดับคีย์ Ctrl-S หยุดการเลื่อนหน้าจอบรรทัด และลำดับคีย์ Ctrl-Q รีเซ็ตการเลื่อนหน้าจอบรรทัด

ถ้าการตรวจสอบการเลื่อนหน้าจอไม่ได้แก้ไขปัญหากับระบบที่ไม่ได้ใช้งาน ให้ไปยังส่วนถัดไป การใช้ลำดับคีย์ Ctrl-D

การใช้ลำดับคีย์ Ctrl-D:

- เรียกใช้งานหน้าต่างหรือเซลล์ที่มีโปรเซสปัญหา
- กดลำดับคีย์ Ctrl-D ลำดับคีย์ Ctrl-D ส่งสัญญาณการสิ้นสุดไฟล์ (EOF) ไปยังโปรเซส ลำดับคีย์ Ctrl-D อาจปิดหน้าต่างของเซลล์และล็อกคุณเอาต์

ถ้าลำดับคีย์ Ctrl-D ไม่ได้แก้ไขปัญหากับระบบที่ไม่ได้ใช้งาน ให้ไปยังส่วนถัดไป การใช้ลำดับคีย์ Ctrl-C

การใช้ลำดับคีย์ Ctrl-C:

สิ้นสุดโปรเซสที่หยุดโดยทำต่อไปนี้:

- เรียกใช้งานหน้าต่างหรือเซลล์ที่มีโปรเซสปัญหา
- กดลำดับคีย์ Ctrl-C ลำดับคีย์ Ctrl-C หยุดการค้นหาปัจจุบัน หรือตัวกรอง

ถ้าลำดับคีย์ Ctrl-C ไม่ได้แก้ไขปัญหากับระบบที่ไม่ได้ใช้งาน ให้ไปยังส่วนถัดไป การล็อกอินจากรีโมต เทอร์มินัลหรือโฮสต์:

การล็อกอินจากรีโมตเทอร์มินัลหรือโฮสต์:

ล็อกอินแบบรีโมตในวิธีอย่างใดอย่างหนึ่งจากสองวิธี:

- ล็อกอินเข้าสู่ระบบจากเทอร์มินัลอื่นถ้ามีเทอร์มินัลต่อพ่วงกับระบบของคุณ มากกว่าหนึ่งเทอร์มินัล
- ล็อกอินจากโฮสต์อื่นบนเครือข่าย (ถ้าระบบของคุณเชื่อมต่อกับเครือข่าย) โดยการพิมพ์คำสั่ง `tn` ดังนี้:

```
tn YourSystemName
```

ระบบจะขอชื่อล็อกอินและรหัสผ่านปกติของคุณ เมื่อคุณใช้คำสั่ง `tn`

หาก你不能ล็อกอินเข้าสู่ระบบ จากรีโมตเทอร์มินัลหรือโฮสต์ ให้ไปยังส่วนถัดไป การสิ้นสุดโปรเซสที่สะดุดแบบรีโมต

หาก你不能ล็อกอินเข้าสู่ระบบจากรีโมตเทอร์มินัลหรือโฮสต์ คุณต้องรีเซ็ตระบบ

คุณยังสามารถเริ่มต้นดีพระบบเพื่อกำหนดว่า เพราะเหตุใดระบบของคุณจึงไม่ได้ใช้งานอยู่

การสิ้นสุดโปรเซสที่สะดุดแบบรีโมต:

สิ้นสุดโปรเซสที่สะดุดจากรีโมตเทอร์มินัลโดยทำต่อไปนี้:

- แสดงรายการโปรเซสที่ใช้งานอยู่โดยการพิมพ์คำสั่ง `ps` ต่อไปนี้:

```
ps -ef
```

แฟล็ก `-e` และ `-f` ระบุโปรเซสที่ใช้งานอยู่และที่ไม่ได้ใช้งานอยู่ทั้งหมด

2. ระบุ ID โพรเซสของโพรเซสที่สะดุด

หากต้องการวิธีใช้ในการระบุโพรเซสให้ใช้คำสั่ง `grep` พร้อมด้วยสตริงการค้นหา ตัวอย่างเช่น เมื่อต้องการสิ้นสุดโพรเซส `xlock` ให้พิมพ์ดังต่อไปนี้ เพื่อค้นหา ID โพรเซส:

```
ps -ef | grep xlock
```

คำสั่ง `grep` ช่วยให้คุณสามารถค้นหาเอาต์พุตจากคำสั่ง `ps` เพื่อระบุ ID โพรเซสของโพรเซสเฉพาะ

3. สิ้นสุดโพรเซสโดยพิมพ์คำสั่ง `kill` ต่อไปนี้:

หมายเหตุ: คุณต้องมีสิทธิ์ผู้ใช้งานเพื่อใช้คำสั่ง `kill` บน โพรเซสซึ่งคุณไม่ได้เริ่มต้น

```
kill -9 ProcessID
```

หาก คุณไม่สามารถระบุโพรเซสปัญหาได้ โพรเซสที่เรียกใช้ล่าสุด อาจเป็นสาเหตุของระบบที่ไม่ได้ใช้งานอยู่ สิ้นสุดโพรเซสล่าสุดถ้า คุณคิดว่าโพรเซสนั้นเป็นปัญหา

ถ้าการตรวจสอบโพรเซสของคุณไม่ได้แก้ไขปัญหากับ ระบบที่ไม่ได้ใช้งานอยู่ คุณต้องรีสตาร์ทระบบ

หลักการที่เกี่ยวข้อง:

“การตรวจสอบฮาร์ดแวร์” ในหน้า 10

มีหลายโพรซีเจอร์ซึ่งคุณสามารถใช้เพื่อตรวจสอบฮาร์ดแวร์ของคุณ

งานที่เกี่ยวข้อง:

“การรีสตาร์ทระบบ”

นอกเหนือจากการตรวจสอบฮาร์ดแวร์และการตรวจสอบโพรเซสแล้ว คุณสามารถรีสตาร์ทระบบเพื่อเรียกใช้ระบบที่ไม่ทำงานอีกครั้ง

ข้อมูลที่เกี่ยวข้อง:

สิ่งอำนวยความสะดวกตามพีระบทย

การรีสตาร์ทระบบ:

นอกเหนือจากการตรวจสอบฮาร์ดแวร์และการตรวจสอบโพรเซสแล้ว คุณสามารถรีสตาร์ทระบบเพื่อเรียกใช้ระบบที่ไม่ทำงานอีกครั้ง

หากโพรซีเจอร์สำหรับ “การตรวจสอบฮาร์ดแวร์” ในหน้า 10 และ “การตรวจสอบโพรเซส” ในหน้า 11 ล้มเหลวในการแก้ไขปัญหาที่ทำให้ระบบของคุณไม่ทำงาน คุณต้องรีสตาร์ทระบบ

หมายเหตุ: ก่อน รีสตาร์ทระบบ ให้ทำดัมพ์ระบบให้สมบูรณ์

1. ตรวจสอบสถานะของอุปกรณ์บูต

ระบบของคุณบูต ด้วยสื่อที่ถอดออกได้ อุปกรณ์ภายนอก อุปกรณ์ small computer system interface (SCSI), อุปกรณ์ integrated device electronics (IDE), หรือ local area network (LAN) เลือกเมธอดที่จะใช้กับระบบของคุณ และ ใช้คำแนะนำต่อไปนี้เพื่อตรวจสอบอุปกรณ์บูต:

- สำหรับสื่อที่ถอดออกได้ เช่น เทป ตรวจสอบให้แน่ใจว่าใส่สื่อ อย่างถูกต้อง
- สำหรับอุปกรณ์ IDE ตรวจสอบว่าค่าติดตั้ง ID ของอุปกรณ์ IDE ไม่ซ้ำกันสำหรับ แต่ละอะแดปเตอร์ หากมีอุปกรณ์ที่ต่อพ่วงกับอะแดปเตอร์เพียงรายการเดียว ต้องตั้งค่า ID ของ อุปกรณ์ IDE เป็นอุปกรณ์หลัก

- สำหรับอุปกรณ์ที่ต่อพ่วงภายนอก เช่น เทปไดรฟ์ ตรวจสอบให้แน่ใจว่า:
 - อุปกรณ์เปิดอยู่
 - สายไฟของอุปกรณ์มีการต่อเข้ากับอุปกรณ์และยูนิตรบบอย่างถูกต้อง
 - ไฟแสดงความพร้อมติดอยู่ (ถ้าอุปกรณ์มีไฟแสดงนี้)
 - สำหรับอุปกรณ์ SCSI ภายนอก ตรวจสอบว่าค่าติดตั้ง SCSI แอดเดรสไม่ซ้ำกัน
 - สำหรับ LAN ตรวจสอบว่าเครือข่ายอัปและใช้งานได้
- ถ้าอุปกรณ์บูตทำงานถูกต้องให้ทำขั้นตอน ถัดไป
2. โหลดระบบปฏิบัติการของคุณโดยทำต่อไปนี้:
- a. ปิดระบบ
 - b. รอหนึ่งนาที
 - c. เปิดระบบ
 - d. รอให้ระบบบูต

หากระบบปฏิบัติการล้มเหลวในการโหลด ให้บูตฮาร์ดดิสก์จาก โหมดการบำรุงรักษาหรือการวินิจฉัยฮาร์ดแวร์

ถ้าคุณยังคงไม่สามารถรีสตาร์ทระบบให้ใช้ SRN เพื่อรายงานปัญหาเกี่ยวกับระบบที่ไม่ทำงานให้ ผู้แทนเซอร์วิสรับทราบ
 หลักการที่เกี่ยวข้อง:

“การตรวจสอบฮาร์ดแวร์” ในหน้า 10

มีหลายโปรแกรมซึ่งคุณสามารถใช้เพื่อตรวจสอบฮาร์ดแวร์ของคุณ
 งานที่เกี่ยวข้อง:

“การตรวจสอบโปรเซส” ในหน้า 11

โปรเซสที่หยุดหรือสะดุดอาจทำให้ระบบของคุณไม่ได้ใช้งานอยู่

ข้อมูลที่เกี่ยวข้อง:

สิ่งอำนวยความสะดวกัมพ์ระบบ y

การสร้างอิมเมจเปิดเครื่องใหม่

เพื่อติดตั้งระบบปฏิบัติการพื้นฐาน หรือเพื่อเข้าถึงระบบที่ไม่บูตจากฮาร์ดไดรฟ์ของระบบ คุณต้องการอิมเมจเปิดเครื่องใหม่ โปรแกรมนี้ อธิบายวิธีการสร้างอิมเมจเปิดเครื่องใหม่ อิมเมจเปิดเครื่องใหม่แตกต่างกันไปสำหรับ อุปกรณ์แต่ละชนิด

เมื่อติดตั้งระบบเป็นครั้งแรก คำสั่ง **bosboot** จะสร้างอิมเมจเปิดเครื่องใหม่จากรูปภาพระบบไฟล์ดิสก์ RAM (random access memory) และเคอร์เนลระบบปฏิบัติการ จากนั้น อิมเมจเปิดเครื่องใหม่ถูกโอนย้ายไปยังสื่อบันทึก เฉพาะ เช่น ฮาร์ดดิสก์ เมื่อรีบูตเครื่อง อิมเมจเปิดเครื่องใหม่จะ ถูกโหลดจากสื่อบันทึกเข้าในหน่วยความจำ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง **bosboot** ให้ดูที่ **bosboot**

ระบบไฟล์ ดิสก์ RAM ที่เชื่อมโยงมีรูทีนคอนฟิกูเรชันอุปกรณ์สำหรับ อุปกรณ์ต่อไปนี้:

- ดิสก์
- เทป
- ซีดีรอม
- อุปกรณ์ Token-Ring เครือข่าย อีเทอร์เน็ต หรือ FDDI

- คุณต้องมีสิทธิ์ผู้ใช้รากเพื่อใช้คำสั่ง `bosboot`
- ระบบไฟล์ `/tmp` ต้องมีพื้นที่ว่างอย่างน้อย 20 MB
- ฟิสิคัลดิสก์ต้องมีบูตโลจิคัลวอลุ่ม เมื่อต้องการกำหนดว่าจะระบุ อุปกรณ์ดิสก์ใด ให้พิมพ์ดังต่อไปนี้ที่พร้อมต์คำสั่ง:

```
lsvg -l rootvg
```

คำสั่ง `lsvg -l` แสดงรายการโลจิคัลวอลุ่มบนกลุ่มวอลุ่มราก (`rootvg`) จากรายการนี้ คุณสามารถค้นหาชื่อของบูตโลจิคัลวอลุ่มได้

จากนั้น พิมพ์ดังต่อไปนี้ ที่พร้อมต์คำสั่ง:

```
lsvg -M rootvg
```

คำสั่ง `lsvg -M` แสดงรายการฟิสิคัลดิสก์ที่มีโลจิคัลวอลุ่มต่างๆ

การสร้างอิมเมจเปิดเครื่องใหม่บนบูตโลจิคัลวอลุ่ม:

หากกำลังติดตั้งระบบปฏิบัติการพื้นฐาน (การติดตั้งใหม่หรืออัปเดต อย่างใดอย่างหนึ่ง) คำสั่ง `bosboot` จะถูกเรียกเพื่อวางอิมเมจเปิดเครื่องใหม่บนบูตโลจิคัลวอลุ่ม บูตโลจิคัลวอลุ่มเป็นพื้นที่ต่อเนื่องทางกายภาพ บนดิสก์ซึ่งสร้างขึ้นผ่านทาง Logical Volume Manager (LVM) ในระหว่างการติดตั้ง

หากต้องการรายการข้อกำหนดเบื้องต้นสำหรับโปรซีเดอรันี้ ให้ดูที่ “การสร้างอิมเมจเปิดเครื่องใหม่” ในหน้า 14

คำสั่ง `bosboot` ทำ ดังต่อไปนี้:

1. ตรวจสอบระบบไฟล์เพื่อดูว่ามีพื้นที่ว่างเพียงพอที่จะสร้าง อิมเมจเปิดเครื่องใหม่หรือไม่
2. สร้างระบบไฟล์ RAM โดยใช้คำสั่ง `mkfs` และไฟล์ต้นแบบ
3. เรียกคำสั่ง `mkboot` ซึ่งจะผสมแอสเซมบลีและระบบไฟล์ RAM เข้าในอิมเมจเปิดเครื่องใหม่
4. เขียนอิมเมจเปิดเครื่องใหม่ลงในบูตโลจิคัลวอลุ่ม

เมื่อต้องการสร้างอิมเมจเปิดเครื่องใหม่บนดิสก์บูตโลจิคัลวอลุ่ม บนดิสก์ที่ระบุ ให้พิมพ์ดังต่อไปนี้ที่พร้อมต์คำสั่ง:

```
bosboot -a
```

หรือ:

```
bosboot -ad /dev/ipldevice
```

หมายเหตุ: อย่า รีบูตเครื่องถ้าคำสั่ง `bosboot` ล้มเหลวในขณะที่สร้างอิมเมจเปิดเครื่องใหม่ แก้ไขปัญหาและรันคำสั่ง `bosboot` เพื่อให้เสร็จสมบูรณ์

คุณต้องรีบูต ระบบเพื่อให้อิมเมจเปิดเครื่องใหม่รูปใหม่พร้อมใช้งานได้

การสร้างอิมเมจเปิดเครื่องใหม่สำหรับอุปกรณ์เครือข่าย:

คุณสามารถสร้างอิมเมจเปิดเครื่องใหม่สำหรับบูตอีเทอร์เน็ตหรือบูต Token-Ring

หากต้องการรายการข้อกำหนดเบื้องต้นสำหรับโปรซีเดอรันี้ ให้ดูที่ “การสร้างอิมเมจเปิดเครื่องใหม่” ในหน้า 14

เมื่อต้องการสร้างอิมเมจเปิดเครื่องใหม่สำหรับบูตอีเทอร์เน็ต ให้พิมพ์ ดังต่อไปนี้ที่พร้อมต์คำสั่ง:

bosboot -ad /dev/ent

สำหรับบูต Token-Ring:

bosboot -ad /dev/tok

ระดับการรันของระบบ

ระดับการรันของระบบระบุสถานะระบบและกำหนด โพรเซสที่จะเริ่มต้น

ตัวอย่างเช่น เมื่อระดับการรันของระบบเป็น 3 โพรเซสทั้งหมดที่กำหนดให้ดำเนินการที่ระดับการรันนั้นจะเริ่มต้นขึ้น ใกล้เคียงกับตอนท้ายของระยะบูตระบบของ บูตโพรเซส ระดับการรันจะถูกอ่านจากรายการ initdefault ของไฟล์ /etc/inittab ระบบดำเนินการที่ระดับการรันนั้นจนกว่าระบบได้รับสัญญาณให้เปลี่ยนแปลง ระดับการรันของระบบสามารถเปลี่ยนได้โดยใช้คำสั่ง `init` ไฟล์ /etc/inittab มีเรกคอร์ดของแต่ละโพรเซสที่กำหนดระดับการรันสำหรับโพรเซสนั้น เมื่อระบบบูต คำสั่ง `init` จะอ่านไฟล์ /etc/inittab เพื่อ กำหนดโพรเซสที่จะเริ่มต้น

ข้อมูลต่อไปนี้เป็นระดับการรันที่กำหนดในปัจจุบัน:

ไอเท็ม	คำอธิบาย
0-9	เมื่อคำสั่ง <code>init</code> เปลี่ยนเป็นระดับการรัน 0-9 คำสั่งจะทำลายโพรเซสทั้งหมดที่ระดับการรันปัจจุบัน จากนั้นรีสตาร์ท โพรเซสใดๆ ที่เชื่อมโยงกับระดับการรันใหม่
0-1	สงวนไว้สำหรับการใช้งานในอนาคตของระบบปฏิบัติการ
2	ระดับการรันดีฟอลต์
3-9	สามารถกำหนดตามการกำหนดค่าตามความชอบของผู้ใช้
a, b, c	เมื่อคำสั่ง <code>init</code> รองขอการเปลี่ยนแปลงเป็นระดับการรัน a, b, หรือ c คำสั่งจะไม่ทำลายโพรเซสที่ ระดับการรันปัจจุบัน เพียงแต่เริ่มต้นโพรเซสใดๆ ที่กำหนดให้กับระดับการรัน ใหม่
Q, q	บอกให้คำสั่ง <code>init</code> ตรวจสอบไฟล์ /etc/inittab อีกครั้ง

งานที่เกี่ยวข้อง:

“การแก้ไขไฟล์ /etc/inittab” ในหน้า 9

มีอยู่สี่คำสั่งสำหรับการแก้ไขเรกคอร์ดในไฟล์ /etc/inittab

การระบุระดับการรันระบบ:

ก่อนทำการบำรุงรักษาบนระบบปฏิบัติการหรือเปลี่ยน ระดับการรันของระบบ คุณอาจต้องตรวจสอบระดับการรันต่างๆ

โพรซีเดอรั้นอธิบายวิธีการระบุระดับการรันซึ่งระบบ กำลังดำเนินการ และวิธีการแสดงประวัติของระดับการรันก่อนหน้านี้ คำสั่ง `init` กำหนดระดับการรันของระบบ

การระบุระดับการรันปัจจุบัน

ที่ บรรทัดคำสั่ง ให้พิมพ์ `cat /etc/.init.state` ระบบ แสดงหนึ่งตำแหน่ง ซึ่งเป็นระดับการรันปัจจุบัน ให้ดูที่คำสั่ง `init` หรือไฟล์ /etc/inittab สำหรับข้อมูลเพิ่มเติมเกี่ยวกับระดับการรัน

การแสดงประวัติของระดับการรันก่อนหน้านี้:

คุณสามารถแสดงประวัติของระดับการรันก่อนหน้านี้โดยใช้คำสั่ง `fwtmp`

หมายเหตุ: ต้องมีการติดตั้งโค้ด `bosext2.acct.obj` บนระบบของคุณเพื่อใช้คำสั่งนี้

1. ล็อกอินเป็นผู้ใช้ราก

16 AIX เวอร์ชัน 7.2: ระบบปฏิบัติการและการจัดการกับอุปกรณ์

2. พิมพ์ดังต่อไปนี้ที่พร้อมคำสั่ง:

```
/usr/lib/acct/fwtmp </var/adm/wtmp |grep run-level
```

ระบบแสดงข้อมูลคล้ายกับตัวอย่าง ต่อไปนี้:

```
run-level 2 0 1 0062 0123 697081013 Sun Feb 2 19:36:53 CST 1992
run-level 2 0 1 0062 0123 697092441 Sun Feb 2 22:47:21 CST 1992
run-level 4 0 1 0062 0123 698180044 Sat Feb 15 12:54:04 CST 1992
run-level 2 0 1 0062 0123 698959131 Sun Feb 16 10:52:11 CST 1992
run-level 5 0 1 0062 0123 698967773 Mon Feb 24 15:42:53 CST 1992
```

การกำหนดคอนฟิกระดับการรันบนระบบหลายผู้ใช้:

คุณสามารถเปลี่ยนระดับการรันบนระบบหลายผู้ใช้ได้

1. ตรวจสอบไฟล์ `/etc/inittab` เพื่อยืนยันว่าระดับการรันซึ่งคุณกำลังเปลี่ยนสนับสนุนโปรเซสที่คุณกำลังรัน กระบวนการ `getty` มีความสำคัญเป็นพิเศษ เนื่องจากกระบวนการนี้ควบคุม การเข้าถึงบรรทัดเทอร์มินัลสำหรับคอนโซลระบบ และล็อกอินอื่น ตรวจสอบให้แน่ใจว่ากระบวนการ `getty` มีการเปิดใช้งานที่ทุกระดับการรัน
2. ใช้คำสั่ง `wall` เพื่อแจ้งให้ผู้ใช้ทุกรายทราบว่า คุณตั้งใจจะเปลี่ยนระดับการรันและร้องขอให้ผู้ใช้ล็อกออฟสำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `wall` ใหญ่ที่ `wall`
3. ใช้พาด่วน `smit telinit` เพื่อเข้าถึงเมนู ตั้งค่า ระดับการรันระบบ
4. พิมพ์ระดับการรันใหม่ในฟิลด์ `System RUN LEVEL`
5. กด Enter เพื่อนำค่าที่ตั้งทั้งหมดในโปรซีเดอร์นี้ไปใช้ ระบบตอบกลับโดยการบอกคุณว่าโปรเซสใดกำลังยุติ หรือกำลังเริ่มต้นอันเป็นผลมาจากการเปลี่ยนแปลงในระดับการรัน และโดยการแสดงข้อความ:

```
INIT: New run level: n
```

โดยที่ *n* คือ หมายเลขระดับการรันใหม่

การกำหนดคอนฟิกระดับการรันบนระบบผู้ใช้เดียว:

คุณสามารถเปลี่ยนระดับการรันบนระบบผู้ใช้เดียวได้

1. ตรวจสอบไฟล์ `/etc/inittab` เพื่อยืนยันว่าระดับการรันซึ่งคุณกำลังเปลี่ยนสนับสนุนโปรเซส ซึ่งคุณกำลังรัน กระบวนการ `getty` มีความสำคัญเป็นพิเศษ เนื่องจากกระบวนการนี้ควบคุม การเข้าถึงบรรทัดเทอร์มินัลสำหรับคอนโซลระบบ และล็อกอินอื่น ตรวจสอบให้แน่ใจว่ากระบวนการ `getty` มีการเปิดใช้งานที่ทุกระดับการรัน สำหรับ ข้อมูลเพิ่มเติมเกี่ยวกับไฟล์ `inittab` ให้อูที่ `inittab`
2. ใช้พาด่วน `smit telinit` เพื่อเข้าถึงเมนู ตั้งค่าระดับการรันระบบ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `telinit` ให้อูที่ `telinit`
3. พิมพ์ระดับการรันระบบใหม่ในฟิลด์ `System RUN LEVEL`
4. กด Enter เพื่อนำค่าที่ตั้งทั้งหมดในโปรซีเดอร์นี้ไปใช้ ระบบตอบกลับโดยการบอกคุณว่าโปรเซสใดกำลังยุติ หรือกำลังเริ่มต้นอันเป็นผลมาจากการเปลี่ยนแปลงในระดับการรัน และโดยการแสดงข้อความ:

```
INIT: New run level: n
```

โดยที่ *n* คือ หมายเลขระดับการรันใหม่

โปรเซสบูต

มีบุตรระบบสามชนิดและรีซอร์สสองรายการที่ต้องการเพื่อบุตรระบบปฏิบัติการ

ในระหว่างโปรเซสบูต ระบบจะทดสอบฮาร์ดแวร์ โหลดและรัน ระบบปฏิบัติการ และกำหนดคอนฟิกอุปกรณ์ เพื่อบุตรระบบปฏิบัติการ ต้องการรีซอร์สต่อไปนี้:

- อิมเมจเปิดเครื่องใหม่ที่สามารถโหลดได้หลังจากเปิดหรือรีเซ็ต เครื่อง
- สิทธิเข้าถึงระบบไฟล์ราก (/) และ /usr

บุตรระบบมีอยู่สามชนิดดังนี้:

ไอเท็ม	คำอธิบาย
บุตรฮาร์ดดิสก์	เครื่องที่เริ่มต้นสำหรับการดำเนินงานปกติ
บุตรเครือข่ายที่ไม่มีดิสก์	เวิร์กสเตชันที่ไม่มีดิสก์หรือไม่มีข้อมูลมีการเริ่มต้นแบบรีโมตบนเครือข่าย เครื่องที่เริ่มต้นสำหรับการดำเนินงานปกติ เซิร์ฟเวอร์ไฟล์รีโมตตั้งแต่หนึ่งเครื่องขึ้นไป จัดเตรียมไฟล์และโปรแกรมซึ่งเวิร์กสเตชันที่ไม่มีดิสก์หรือไม่มีข้อมูลต้องใช้ ในการบูต
บุตรการบำรุงรักษา	เครื่องที่เริ่มต้นจากฮาร์ดดิสก์ เครือข่าย เทป หรือซีดีรอมใน โหมดการบำรุงรักษา ผู้ดูแลระบบสามารถทำภารกิจต่างๆ เช่น การติดตั้ง ซอฟต์แวร์ใหม่หรือที่อัปเดต และการรันการตรวจสอบเชิงวิเคราะห์

ในระหว่างบุตรฮาร์ดดิสก์ อิมเมจเปิดเครื่องใหม่มีอยู่บนโลคัลดิสก์ที่สร้างขึ้น เมื่อติดตั้งระบบปฏิบัติการ ในระหว่างโปรเซสบูต ระบบจะ กำหนดคอนฟิกอุปกรณ์ทั้งหมดที่พบในเครื่องและเริ่มต้นซอฟต์แวร์พื้นฐานอื่น ที่ต้องใช้เพื่อให้ระบบทำงาน (เช่น Logical Volume Manager) ใน ตอนท้ายของโปรเซสนี้ ระบบไฟล์จะมีการติดตั้งและพร้อมใช้งาน

ข้อกำหนดทั่วไปเดียวกันใช้กับโคลเอ็นต์เครือข่ายที่ไม่มีดิสก์ นอกจากนี้ยัง ต้องการอิมเมจเปิดเครื่องใหม่และสิทธิเข้าถึงแผนผังไฟล์ระบบปฏิบัติการด้วย โคลเอ็นต์ เครือข่ายที่ไม่มีดิสก์ไม่มีระบบไฟล์โลคัลและเรียกใช้ข้อมูลทั้งหมดของตน ด้วยวิธีการเข้าถึงแบบรีโมต

หลักการที่เกี่ยวข้อง:

“การประมวลผลบุตรระบบ”

ผู้ใช้งานใหญ่ทำฮาร์ดดิสก์บูตเมื่อเริ่มต้นระบบสำหรับการดำเนินงานทั่วไป ระบบค้นหาข้อมูลที่จำเป็นทั้งหมดสำหรับโปรเซสบูต บนดิสก์ไดรฟ์

“โปรเซสบูตการบำรุงรักษา” ในหน้า 20

ในบางโอกาส อาจจำเป็นต้องบูตเพื่อทำภารกิจพิเศษ เช่น การติดตั้งซอฟต์แวร์ใหม่หรือที่อัปเดต การทำการตรวจสอบวินิจฉัยหรือสำหรับการบำรุงรักษา ในกรณีนี้ ระบบจะเริ่มต้นจาก สื่อบันทึกที่สามารถบูตได้ เช่น ซีดีรอม ดีวีดี เทปไดรฟ์ เครือข่ายหรือดิสก์ไดรฟ์

“ระบบไฟล์ RAM” ในหน้า 21

ระบบไฟล์ RAM ซึ่งเป็นส่วนหนึ่งของอิมเมจเปิดเครื่องใหม่ อยู่ในหน่วย ความจำทั้งหมด และมีโปรแกรมทั้งหมดที่ช่วยให้บูตโปรเซสทำต่อไปได้ ไฟล์ในระบบไฟล์ RAM ขึ้นอยู่กับชนิดของบูต

การประมวลผลบุตรระบบ:

ผู้ใช้งานใหญ่ทำฮาร์ดดิสก์บูตเมื่อเริ่มต้นระบบสำหรับการดำเนินงานทั่วไป ระบบค้นหาข้อมูลที่จำเป็นทั้งหมดสำหรับโปรเซสบูต บนดิสก์ไดรฟ์

เมื่อระบบเริ่มต้นขึ้นโดยการเปิดสวิตช์เปิด/ปิด (cold boot) หรือรีสตาร์ทโดยใช้คำสั่ง **reboot** หรือ **shutdown** (warm boot) เหตุการณ์ต่างๆ ต้องเกิดขึ้นก่อนระบบพร้อมใช้งาน เหตุการณ์เหล่านี้สามารถแบ่งออกเป็นระยะต่อไป:

หลักการที่เกี่ยวข้อง:

“โปรเซสบูต” ในหน้า 18

มีบูตระบบสามชนิดและรีซอร์สสองรายการที่ต้องการเพื่อบูตระบบปฏิบัติการ

เฟสเฟิร์มแวร์:

เฟิร์มแวร์จัดเตรียมระบบเพื่อโหลดและรันระบบปฏิบัติการ

ระยะการเริ่มต้นของเคอร์เนลเกี่ยวข้องกับขั้นตอนต่อไปนี้:

1. เฟิร์มแวร์ดำเนินการทดสอบขั้นพื้นฐานบนรีซอร์สระบบที่จำเป็นสำหรับการเริ่มต้นระบบปฏิบัติการ
2. เฟิร์มแวร์ตรวจสอบรายการบูตของผู้ใช้ รายการของอุปกรณ์บูตที่พร้อมใช้งาน รายการบูตนี้สามารถเปลี่ยนแปลงให้เหมาะสมตามความต้องการของคุณ โดยใช้คำสั่ง **bootlist** หากรายการบูตของผู้ใช้ใน non-volatile random access memory (NVRAM) ไม่ถูกต้อง หรือถ้าไม่พบอุปกรณ์บูตที่ถูกต้อง จะมีการตรวจสอบรายการบูตดีฟอลต์ในกรณีใดก็ตาม จะใช้อุปกรณ์บูตที่ถูกต้องลำดับแรกๆ ที่พบในรายการบูตสำหรับสตาร์ทอัพระบบ ถ้ามีรายการบูตของผู้ใช้ที่ถูกต้องอยู่ใน NVRAM อุปกรณ์ในรายการ จะถูกตรวจสอบตามลำดับ ถ้าไม่มีรายการบูตของผู้ใช้ อะแดปเตอร์และอุปกรณ์ทั้งหมดบนบัสจะถูกตรวจสอบ ในกรณีใดก็ตาม อุปกรณ์จะถูกตรวจสอบแบบวนซ้ำอย่างต่อเนื่อง จนกว่าพบอุปกรณ์บูตที่ถูกต้องสำหรับสตาร์ทอัพระบบ

หมายเหตุ: ระบบ เก็บรักษารายการบูตดีฟอลต์ที่จัดเก็บไว้ใน NVRAM สำหรับบูตโหมด ปกติ รายการบูตโหมดเซอรัวิสที่แยกต่างหากถูกเก็บอยู่ใน NVRAM และคุณต้องการอ้างอิงคำสั่งฮาร์ดแวร์ที่ระบุเฉพาะสำหรับโมเดลของคุณ เพื่อศึกษาถึงวิธีการเข้าถึงรายการบูตในโหมดเซอรัวิส

3. เมื่อพบอุปกรณ์บูตที่ถูกต้อง เร็กคอร์ดแรกหรือ program sector number (PSN) จะถูกตรวจสอบ หากเป็นเร็กคอร์ดบูตที่ถูกต้อง เร็กคอร์ดจะถูกอ่านเข้าในหน่วยความจำ และเพิ่มลงในบล็อกควบคุม IPL ในหน่วยความจำ สิ่งที่อยู่ในข้อมูลเร็กคอร์ดบูตคือที่ตั้งเริ่มต้นของอิมเมจเปิดเครื่องใหม่บน อุปกรณ์บูต ความยาวของอิมเมจเปิดเครื่องใหม่ และคำแนะนำเกี่ยวกับตำแหน่ง การโหลดอิมเมจเปิดเครื่องใหม่ในหน่วยความจำ
4. อิมเมจเปิดเครื่องใหม่ถูกอ่านตามลำดับจากอุปกรณ์บูตเข้าในหน่วยความจำ โดยเริ่มต้นที่ที่ตั้งซึ่งระบุใน NVRAM อิมเมจเปิดเครื่องใหม่ของดิสก์ประกอบด้วยเคอร์เนล ระบบไฟล์ RAM และข้อมูลอุปกรณ์พื้นฐานแบบกำหนดเอง
5. การควบคุมถูกส่งผ่านไปยังเคอร์เนล ซึ่งเริ่มต้นการเริ่มต้นระบบ
6. เคอร์เนลรัน **init** ซึ่งจะรันระยะที่ 1 ของสคริปต์ **rc.boot**

เมื่อระยะการเริ่มต้นเคอร์เนลเสร็จสมบูรณ์แล้ว คอนฟิกูเรชันอุปกรณ์พื้นฐาน จะเริ่มขึ้น

ระยะคอนฟิกูเรชันอุปกรณ์พื้นฐาน:

โปรเซส **init** เริ่มต้นสคริปต์ **rc.boot** ระยะ 1 ของสคริปต์ **rc.boot** ทำคอนฟิกูเรชัน อุปกรณ์พื้นฐาน

ระยะ 1 ของสคริปต์ **rc.boot** มีขั้นตอน ต่อไปนี้:

1. บูตสคริปต์เรียกโปรแกรม **restbase** เพื่อสร้าง ฐานข้อมูล Object Data Manager (ODM) แบบกำหนดเองในระบบไฟล์ RAM จากข้อมูลแบบกำหนดเองที่บีบอัด

2. บุตสคริปต์เริ่มต้นตัวจัดการคอนฟิกูเรชัน ซึ่งเข้าถึงระยะ 1 ของกฎคอนฟิกูเรชัน ODM เพื่อกำหนดคอนฟิกูปรแกรมพื้นฐาน
3. ตัวจัดการคอนฟิกูเรชันเริ่มต้น sys, bus, disk, SCSI, และ Logical Volume Manager (LVM) และเมธอดคอนฟิกูเรชันกลุ่ม วอลุ่ม rootvg
4. เมธอดคอนฟิกูเรชันโหนดไดเรกทอรีอุปกรณ์ สร้างไฟล์พิเศษ และอัปเดตข้อมูลแบบกำหนดเองในฐานข้อมูล ODM

การบูตระบบ:

โพรซีเดอร์นี้ทำให้เฟสการบูตระบบ เสร็จสิ้น

1. โพรเซส `init` เริ่มต้นการรันระยะ 2 ของสคริปต์ `rc.boot` ระยะ 2 ของ `rc.boot` มีขั้นตอนต่อไปนี้:
 - a. เรียกโปรแกรม `ipl_varyon` เพื่อ vary on กลุ่ม วอลุ่ม rootvg
 - b. ติดตั้งระบบไฟล์ฮาร์ดดิสก์บนจุดต่อปกติ
 - c. รันโปรแกรม `swapon` เพื่อเริ่มต้นการเลื่อนหน้า
 - d. คัดลอกข้อมูลที่กำหนดเองจากฐานข้อมูล ODM ในระบบไฟล์ RAM ไปยังฐานข้อมูล ODM ในระบบไฟล์ฮาร์ดดิสก์
 - e. ออกจากสคริปต์ `rc.boot`
2. หลังจากเฟส 2 ของสคริปต์ `rc.boot` เสร็จสิ้น กระบวนการบูตจะสลับจากระบบไฟล์ RAM ไปเป็น ระบบไฟล์ที่เก็บบนฮาร์ดดิสก์
3. จากนั้น โพรเซส `init` รันโพรเซสที่กำหนดโดยเร็กคอร์ดในไฟล์ `/etc/inittab` คำแนะนำอย่างใดอย่างหนึ่ง ในไฟล์ `/etc/inittab` จะรันระยะ 3 ของสคริปต์ `rc.boot` ซึ่งมีขั้นตอนต่อไปนี้:
 - a. ติดตั้งระบบไฟล์ฮาร์ดดิสก์ `/tmp`
 - b. เริ่มต้นผู้จัดการคอนฟิกูเรชันระยะ 2 เพื่อกำหนดคอนฟิกูปรแกรมที่เหลือทั้งหมด
 - c. ใช้คำสั่ง `savebase` เพื่อบันทึกข้อมูลที่กำหนดเอง ไว้ในบูตโลจิคัลวอลุ่ม
 - d. ออกจากสคริปต์ `rc.boot`

ที่ตอนท้ายของโพรเซสนี้ ระบบออฟและพร้อมใช้งาน

โพรเซสบูตการบำรุงรักษา:

ในบางโอกาส อาจจำเป็นต้องบูตเพื่อทำการกิจพิเศษ เช่น การติดตั้งซอฟต์แวร์ใหม่หรือที่อัปเดต การทำการตรวจสอบวินิจฉัย หรือสำหรับการบำรุงรักษา ในกรณีนี้ ระบบจะเริ่มต้นจาก สื่อบันทึกที่สามารถบูตได้ เช่น ซีดีรอม ดีวีดี เทปไดร์ฟ เครื่องช่วยหรือดิสก์ไดร์ฟ

ลำดับบูตการบำรุงรักษาของเหตุการณ์คล้ายกับลำดับของ บูตปกติ

1. เพิ่มแอร์ด้าเนินการทดสอบขั้นพื้นฐานบนรีซอร์สระบบ ที่จำเป็นสำหรับการเริ่มต้นระบบปฏิบัติการ
2. เพิ่มแอร์ดตรวจสอบรายการบูตของผู้ใช้ คุณสามารถใช้คำสั่ง `bootlist` เพื่อเปลี่ยนรายการบูตของผู้ใช้ให้เหมาะสมกับความต้องการของคุณ หากรายการบูตของผู้ใช้ใน non-volatile random access memory (NVRAM) ไม่ถูกต้อง หรือหาไม่พบอุปกรณ์บูตที่ถูกต้อง รายการบูตดีฟอลต์จะถูกตรวจสอบ ในกรณีนี้ อุปกรณ์บูตที่ถูกต้องอันดับแรกที่พบในรายการบูตถูกใช้เพื่อเริ่มทำงานระบบ

หมายเหตุ: สำหรับบูตแบบปกติ ระบบปฏิบัติการยังเก็บรายการบูตดีฟอลต์และรายการบูตของผู้ใช้ไว้ ซึ่งจะเก็บอยู่ใน NVRAM การแยกรายการบูตดีฟอลต์กับรายการบูตของผู้ใช้จะถูกเก็บไว้สำหรับเริ่มต้นทำงานกับระบบในโหมดการซ่อม บำรุง

- เมื่อพบอุปกรณ์บูตที่ถูกต้อง เร็กคอร์ดแรกหรือ program sector number (PSN) จะถูกตรวจสอบ หากเป็นเร็กคอร์ดบูตที่ถูกต้อง เร็กคอร์ดจะถูกอ่านเข้าในหน่วยความจำ และเพิ่มลงในบล็อกควบคุม initial program load (IPL) ในหน่วยความจำ สิ่งทั้งหมดอยู่ใน ข้อมูลเร็กคอร์ดบูตคือที่ตั้งเริ่มต้นของอิมเมจเปิดเครื่องใหม่บน อุปกรณ์บูต ความยาวของอิมเมจเปิดเครื่องใหม่ และออฟเซตของจุดการป้อน เพื่อเริ่มต้นการรันเมื่ออิมเมจเปิดเครื่องใหม่อยู่ในหน่วยความจำ
- อิมเมจเปิดเครื่องใหม่ถูกอ่านตามลำดับจากอุปกรณ์บูตเข้าในหน่วยความจำ โดยเริ่มต้นที่ที่ตั้งซึ่งระบุใน NVRAM
- การควบคุมถูกส่งผ่านไปยังเคอร์เนล ซึ่งเริ่มต้นการรันโปรแกรมใน ระบบไฟล์ RAM
- เนื้อหาของข้อมูล ODM กำหนดอุปกรณ์ที่มีอยู่ และคำสั่ง `cfgmgr` กำหนด คอนฟิกอุปกรณ์ทั้งหมดที่พบแบบไดนามิก รวมถึงดิสก์ทั้งหมดซึ่งมี ระบบไฟล์ราก
- ถ้าใช้ซีดีรอม ดีวีดี เทป หรือเครือข่ายเพื่อบูตระบบ กลุ่มวอลุ่ม `rootvg` (หรือ `rootvg`) ถูก vary on เนื่องจาก `rootvg` อาจไม่มีอยู่ (ดังนั้นจึงเป็นกรณีเมื่อติดตั้งระบบปฏิบัติการ บนระบบใหม่) คอนฟิกูเรชัน เครือข่ายสามารถเกิดขึ้น ณ เวลานี้ไม่มี การเลื่อนหน้าเกิดขึ้นเมื่อทำการบูต การบำรุงรักษา

ที่ตอนท้ายของโปรเซสนี้ ระบบพร้อมสำหรับการติดตั้ง การบำรุงรักษา หรือการวินิจฉัย

หมายเหตุ: ถ้าเริ่มต้นระบบจากฮาร์ดดิสก์แล้ว `rootvg` จะถูก vary on ระบบไฟล์รากของฮาร์ดดิสก์และระบบไฟล์ /usr ของฮาร์ดดิสก์ถูกเมาท์ในระบบไฟล์ RAM เมนูจะแสดงขึ้นซึ่งอนุญาตให้คุณเข้าสู่โหมดการวินิจฉัย หรือโหมดผู้ใช้เดี่ยว ถ้าคุณเลือกโหมดผู้ใช้เดี่ยว คุณสามารถดำเนินการกับกระบวนการบูตต่อไปได้ และเข้าสู่โหมดผู้ใช้เดี่ยว โดยที่ระดับการรัน `init` ถูกตั้งค่าเป็นตัวอักษร S ระบบพร้อมสำหรับการซ่อมบำรุง ซอฟต์แวร์อัปเดต หรือสำหรับคำสั่ง `bosboot`

หลักการที่เกี่ยวข้อง:

“โปรเซสบูต” ในหน้า 18

มีบูตระบบสามชนิดและรีซอร์สสองรายการที่ต้องการ เพื่อบูตระบบปฏิบัติการ

ระบบไฟล์ RAM:

ระบบไฟล์ RAM ซึ่งเป็นส่วนหนึ่งของอิมเมจเปิดเครื่องใหม่ อยู่ในหน่วย ความจำทั้งหมด และมีโปรแกรมทั้งหมดที่ช่วยให้บูตโปรเซสทำต่อไปได้ ไฟล์ในระบบไฟล์ RAM ขึ้นอยู่กับชนิดของบูต

ระบบไฟล์ RAM บูตการบำรุงรักษาอาจไม่มีโลจิคัลวอลุ่มรูทีน เนื่องจาก `rootvg` อาจไม่ต้องถูก varied on อย่างไรก็ตาม ในระหว่างบูตฮาร์ดดิสก์ `rootvg` ควรจะถูก varied on และเรียกใช้การเลื่อนหน้า โดยเร็วที่สุด แม้วามีความแตกต่างในสถานการณ์ จำลองบูตสองสถานการณ์เหล่านี้ แต่โครงสร้างของระบบไฟล์ RAM ไม่ได้แตกต่างกันมากนัก

คำสั่ง `init` ซึ่งวางอยู่บนระบบไฟล์ RAM คือโปรแกรมการตีความคำสั่งบูตระดับต้น ซึ่งออกแบบมาสำหรับใช้ในระหว่างกระบวนการบูต โปรแกรมการตีความคำสั่งบูต ควบคุมกระบวนการบูตโดยเรียกสคริปต์ `rc.boot` สคริปต์ `rc.boot` กำหนดอุปกรณ์ที่เครื่อง ต้องเริ่มทำงาน อุปกรณ์บูตกำหนดอุปกรณ์ ที่ต้องกำหนดคอนฟิกบนระบบไฟล์ RAM ถ้าเครื่อง เริ่มต้นทำงานผ่านเครือข่าย อุปกรณ์เครือข่ายต้องถูกกำหนดคอนฟิกไว้ ดังนั้น ระบบไคลเอ็นต์ไฟล์สามารถเมาท์แบบรีโมตในกรณีของเทป ซีดีรอม หรือบูต DVD คอนโซลถูกกำหนดคอนฟิกไว้เพื่อแสดง เมนูการติดตั้ง base operating system (BOS) หลังจากทีสคริปต์ `rc.boot` ระบุอุปกรณ์บูตแล้ว รูทีนคอนฟิกูเรชันที่เหมาะสม จะถูกเรียกจากระบบไฟล์ RAM สคริปต์ `rc.boot` ถูกเรียก

สองครั้งโดยโปรแกรมการตีความคำสั่งบูตเพื่อให้อัปกับเฟส คอนฟิกูเรชันสองเฟสในระหว่างกระบวนการบูต การเรียก rc.boot ครั้งที่สามเกิดขึ้น ในระหว่างบูตดิสก์หรือเครือข่ายเมื่อเรียกคำสั่ง init จริง ไฟล์ inittab มี rc.boot stanza ที่ทำคอนฟิกูเรชันสุดท้ายของเครื่องให้เสร็จสิ้น

ระบบไฟล์ RAM สำหรับอุปกรณ์บูตแต่ละรายการยังไม่ซ้ำกันด้วย เนื่องจาก อุปกรณ์ที่จะกำหนดคอนฟิกมีอยู่หลายชนิด มีการเชื่อมโยงไฟล์ต้นแบบ กับอุปกรณ์บูตแต่ละชนิด ไฟล์ต้นแบบเป็นเทมเพลตของไฟล์ซึ่งประกอบขึ้น เป็นระบบไฟล์ RAM คำสั่ง bosboot ใช้คำสั่ง mkfs เพื่อสร้างระบบไฟล์ RAM โดยใช้ไฟล์ต้นแบบหลายชนิด โปรดดูที่คำสั่ง bosboot สำหรับรายละเอียดเพิ่มเติม

หลักการที่เกี่ยวข้อง:

“โปรเซสบูต” ในหน้า 18

มีบูตระบบสามชนิดและรีซอร์สสองรายการที่ต้องการ เพื่อบูตระบบปฏิบัติการ

การแก้ไขปัญหาสตาร์ทอ็อประบบ

ใช้วิธีการแก้ไขปัญหาเหล่านี้เพื่อจัดการกับ ปัญหาพื้นฐานที่อาจเกิดขึ้นเมื่อระบบของคุณเริ่มทำงาน หากข้อมูลการแก้ไขปัญหาไม่ได้ช่วยแก้ไข ปัญหา โปรดติดต่อผู้แทนเซอร์วิสของคุณ

ระบบที่จะไม่บูต:

หากระบบจะไม่บูตจากฮาร์ดดิสก์ คุณอาจยังคงสามารถเข้าถึงระบบ เพื่อตรวจสอบและแก้ไขปัญหาได้

หากคุณมีระบบที่จะไม่บูตจากฮาร์ดดิสก์ ให้อัปที่โพรซีเดอรวีธีการเข้าถึงระบบที่จะไม่บูตใน การแก้ไขปัญหาการติดตั้งของคุณ ใน *Installation and migration*

โพรซีเดอรวีธีการนี้ช่วยให้คุณได้รับพร้อมระบบเพื่อให้ คุณสามารถพยายามกู้คืนข้อมูลจากระบบ หรือทำการดำเนินการแก้ไข ที่เปิดทางให้ระบบบูตจากฮาร์ดดิสก์ได้

หมายเหตุ:

- โพรซีเดอรวีธีการนี้เหมาะสมกับผู้จัดการระบบที่มีประสบการณ์ซึ่งมีความรู้เกี่ยวกับวิธีการบูตหรือกู้คืนข้อมูลจากระบบที่ไม่สามารถบูต จากฮาร์ดดิสก์ได้ ผู้ใช้ส่วนใหญ่ไม่ควรพยายามใช้โพรซีเดอรวีธีการนี้ แต่ควร ติดต่อผู้แทนเซอร์วิส
- โพรซีเดอรวีธีการนี้ไม่เหมาะสมกับผู้จัดการระบบที่เพิ่งติดตั้งใหม่เสร็จสิ้น เนื่องจากในกรณีนี้ ระบบไม่มีข้อมูลที่จำเป็น ต้อง กู้คืน หาก你不能บูตจากฮาร์ดดิสก์หลังจาก ทำการติดตั้งใหม่เสร็จสมบูรณ์แล้ว โปรดติดต่อผู้แทนเซอร์วิสของคุณ

สิ่งอ้างอิงที่เกี่ยวข้อง:

“การวิเคราะห์ปัญหาการบูต”

มีปัจจัยหลายอย่าง ที่อาจส่งผลให้ระบบไม่สามารถ บูตได้

การวิเคราะห์ปัญหาการบูต:

มีปัจจัยหลายอย่าง ที่อาจส่งผลให้ระบบไม่สามารถ บูตได้

ปัจจัยดังกล่าวบางอย่างคือ:

- ปัญหาฮาร์ดแวร์
- เทปหรือซีดีรอมบูตบกพร่อง
- เซิร์ฟเวอร์บูตเครือข่ายที่กำหนดคอนฟิกไม่ถูกต้อง

- ระบบไฟล์เสียหาย
- ข้อผิดพลาดในสคริปต์ เช่น /sbin/rc.boot

ถ้าโปรเซสหยุดทำงานด้วยโค้ดการอ้างอิง 2702 และแสดงข้อความ "INSUFFICIENT ENTITLED MEMORY" ให้ใช้ HMC เพื่อเพิ่มจำนวนของหน่วยความจำที่ใช้ได้สำหรับพาร์ติชันนั้น

หลักการที่เกี่ยวข้อง:

“ระบบที่จะไม่บูต” ในหน้า 22

หากระบบจะไม่บูตจากฮาร์ดดิสก์ คุณอาจยังคงสามารถเข้าถึงระบบ เพื่อตรวจสอบและแก้ไขปัญหาได้

สำเนาสำรองระบบ

หลังจากใช้ระบบของคุณแล้ว ข้อควรพิจารณาถัดไปของคุณควรจะเป็น การทำสำเนาสำรองระบบไฟล์ไดเรกทอรี และไฟล์ หากคุณทำสำเนาสำรองระบบไฟล์ คุณสามารถเรียกคืนไฟล์หรือระบบไฟล์ในกรณีที่ฮาร์ดดิสก์ เสียหายได้ มีเมธอดที่ต่างกักันสำหรับการทำสำเนาสำรองข้อมูล

การทำสำเนาสำรองระบบไฟล์ไดเรกทอรี และไฟล์ถือเป็นการลงทุนทั้ง เวลาและความพยายามที่สำคัญ ในเวลาเดียวกัน ไฟล์คอมพิวเตอร์ทั้งหมดมีโอกาสอย่างมาก ที่จะถูกเปลี่ยนแปลงหรือลบออก โดยตั้งใจหรือโดยอุบัติเหตุ

ข้อควรสนใจ: เมื่อฮาร์ดดิสก์เสียหาย ข้อมูลที่มีอยู่บนดิสก์นั้นจะถูกทำลาย วิธีเดียวในการกู้คืนข้อมูลที่เสียหายคือ การดึงข้อมูลมาจากสำเนาสำรองของคุณ

หากคุณใช้ความระมัดระวังและปฏิบัติตามวิธีการทำสำเนาสำรองระบบไฟล์ของคุณ คุณควรจะสามารถเรียกคืนเวอร์ชันล่าสุดของไฟล์หรือระบบไฟล์ได้ไม่ยากนัก เสมอ

มีหลายเมธอดสำหรับการทำสำเนาสำรองข้อมูล เมธอดหนึ่งที่ใช้กันบ่อยที่สุด เรียกว่า *สำเนาสำรองตามชื่อ ที่เก็บถาวรชื่อไฟล์* หรือ *สำเนาสำรอง สมบูรณ์เสมอ* นี่เป็นสำเนาของระบบไฟล์ ไดเรกทอรี หรือไฟล์ซึ่งเก็บไว้สำหรับการโอนย้ายไฟล์หรือในกรณีที่ข้อมูลต้นฉบับถูกเปลี่ยนแปลง หรือทำลายโดยไม่ได้ตั้งใจ เมธอดการทำสำเนาสำรองนี้ทำเมื่อมีการระบุแฟล็ก *i* และใช้เพื่อจัดทำสำเนาสำรองของแต่ละไฟล์และไดเรกทอรี นี่เป็น เมธอดที่ใช้กันทั่วไปโดยผู้ใช้แต่ละรายเพื่อทำสำเนาสำรองแอดเคาต์ของตน

อีกเมธอดหนึ่งที่ใช้บ่อยเรียกว่า *สำเนาสำรองตาม i-node ที่เก็บถาวร ระบบไฟล์* หรือ *สำเนาสำรองเก็บถาวร* เมธอดการทำสำเนาสำรองนี้ทำ เมื่อ *ไม่ได้* ระบุแฟล็ก *i* เมธอดนี้ใช้สำหรับการอ้างอิงในอนาคต ใช้เป็นข้อมูลประวัติ หรือสำหรับการกู้คืนถ้าข้อมูลต้นฉบับเสียหายหรือหายไป เมธอดนี้ใช้เพื่อจัดทำสำเนาสำรองของทั้งระบบไฟล์ และเป็นเมธอดที่ใช้กันทั่วไปโดยผู้ดูแลระบบเพื่อทำสำเนาสำรองกลุ่มไฟล์ขนาดใหญ่ เช่น แอดเคาต์ผู้ใช้ทั้งหมดใน /home สำเนาสำรองระบบไฟล์ ช่วยให้สามารถทำสำเนาสำรองเพิ่มเติมได้ง่าย สำเนาสำรองเพิ่มเติม จะทำสำเนาสำรองไฟล์ทั้งหมดที่ถูกแก้ไขตั้งแต่สำเนาสำรองก่อนหน้านี้ที่ระบุ

คำสั่ง **compress** และ **pack** ช่วยให้คุณสามารถ บีบอัดไฟล์ในหน่วยเก็บ และคำสั่ง **uncompress** และ **unpack** คลายการบีบอัดไฟล์หลังจากถูก เรียกคืนแล้ว กระบวนการบีบอัดและคลายบีบอัดไฟล์ใช้เวลานานพอสมควร แต่ หลังจากบีบอัดแล้ว ข้อมูลใช้พื้นที่ว่างบนสื่อสำเนาสำรองน้อยลง สำหรับข้อมูลเพิ่มเติม เกี่ยวกับคำสั่งเหล่านี้ ให้อูที่ **compress, pack, uncompress, และ unpack**

หลายคำสั่งสร้างสำเนาสำรองและที่เก็บถาวร ด้วยเหตุนี้ ข้อมูลที่ทำสำเนาสำรองแล้ว จึงต้องมีการติดเลเบลแสดงคำสั่งที่ใช้เพื่อเริ่มต้น สำเนาสำรอง และวิธีการทำสำเนาสำรอง (โดยชื่อหรือโดยระบบไฟล์)

ไอเท็ม	คำอธิบาย
สำเนาสำรอง	สำเนาสำรองไฟล์โดยชื่อหรือโดยระบบไฟล์สำหรับข้อมูลเพิ่มเติม ให้อูที่ backup
mksysb	สร้างรูปภาพที่ติดตั้งไดของ rootvg สำหรับข้อมูลเพิ่มเติม ให้อูที่ mksysb
cpio	คัดลอกไฟล์เข้าในและออกจากหน่วยเก็บถาวร สำหรับข้อมูลเพิ่มเติม ให้อูที่ cpio
dd	แปลงและคัดลอกไฟล์ใช้กันทั่วไปเพื่อแปลงและคัดลอกข้อมูลไปยัง และออกจากระบบที่กำลังรันระบบปฏิบัติการอื่น ตัวอย่างเช่น เมินเฟรม dd ไม่ได้ จัดกลุ่มหลายไฟล์เข้าเป็นหนึ่งหน่วยเก็บถาวร แต่ใช้เพื่อจัดวางและ ย้ายข้อมูล สำหรับข้อมูลเพิ่มเติม ให้อูที่ dd
tar	สร้างหรือจัดวางที่เก็บถาวรรูปแบบ tar สำหรับข้อมูลเพิ่มเติม ให้อูที่ tar
rdump	ทำสำเนาสำรองไฟล์โดยระบบไฟล์บนอุปกรณ์ของเครื่องรีโมต สำหรับ ข้อมูลเพิ่มเติม ให้อูที่ rdump
pax	(POSIX-ยูทิลิตี้ที่เก็บถาวรแบบสอดคล้องกัน) อ่านและเขียนที่เก็บถาวร tar และ cpio สำหรับข้อมูลเพิ่มเติม ให้อูที่ pax

หลักการที่เกี่ยวข้อง:

“สำเนาสำรองสำหรับผู้จัดการระบบ BSD 4.3” ในหน้า 360

ผู้จัดการระบบ BSD 4.3 สามารถทำสำเนาสำรองข้อมูลได้

งานที่เกี่ยวข้อง:

“การทำสำเนาสำรองไฟล์ผู้ใช้หรือระบบไฟล์” ในหน้า 29

สามารถใช้สองโปรแกรมเพื่อทำสำเนาสำรองไฟล์และระบบไฟล์: พาดวน SMIT **smit backfile** หรือ **smit backfilesys**, และคำสั่ง **backup**

แนวคิดสำเนาสำรอง

ก่อนคุณเริ่มต้นการทำสำเนาสำรองข้อมูล คุณต้องเข้าใจ ชนิดของข้อมูล นโยบาย และสื่อบันทึกซึ่งคุณสามารถใช้ได้

นโยบายสำเนาสำรอง:

นโยบายสำเนาสำรองเดียวไม่สามารถตอบสนองต่อความต้องการของผู้ใช้ทั้งหมดได้ ตัวอย่างเช่น นโยบายที่ทำงานได้ดี สำหรับระบบที่มีผู้ใช้หนึ่งรายอาจไม่พอเพียง สำหรับระบบที่ให้บริการผู้ใช้หนึ่งร้อยราย ในลักษณะคล้ายกัน นโยบายที่จัดทำขึ้นสำหรับ ระบบซึ่งไฟล์จำนวนมากถูกเปลี่ยนแปลงรายวันอาจจะไม่มีประสิทธิภาพ สำหรับระบบซึ่งข้อมูลเปลี่ยนแปลงไม่บ่อยนัก

ไม่ว่ากลยุทธ์สำเนาสำรองที่เหมาะสมสำหรับไซต์ของคุณจะเป็นอะไรก็ตาม สิ่งที่สำคัญมากคือต้องมีสำเนาสำรองและมีการทำสำเนาสำรองนั้นบ่อยครั้งและสม่ำเสมอ การกู้คืนข้อมูลที่สูญหายไป เป็นเรื่องยากถ้ายังไม่มีการใช้กลยุทธ์สำเนาสำรองที่ดี

คุณเท่านั้นที่สามารถกำหนดกลยุทธ์สำเนาสำรองที่ดีที่สุดสำหรับระบบของคุณ แต่แนวทางทั่วไปต่อไปนี้อาจช่วยคุณได้:

- ตรวจสอบให้แน่ใจว่าคุณสามารถกู้คืนจากการสูญหายที่สำคัญได้

ระบบของคุณสามารถ รันต่อไปได้หรือไม่หลังจากดิสก์ที่ระบุตัวหนึ่งล้มเหลว? คุณสามารถกู้คืนระบบได้หรือไม่ ถ้าดิสก์ที่ระบุทั้งหมดอาจล้มเหลวได้? คุณสามารถกู้คืนระบบได้หรือไม่ถ้า ดิสก์เกิดสำเนาสำรองสูญหายหรือเทปถูกไฟไหม้หรือถูกขโมยไป? หากข้อมูลสูญหาย การสร้างข้อมูลนั้นขึ้นใหม่เป็นเรื่องยากเพียงใด? ให้คิดและเลือกคำตอบระหว่าง เป็นไปได้เกิดขึ้นได้ยาก การสูญเสียนโยบาย สำเนาสำรองที่ช่วยให้คุณกู้คืน ระบบเนื่องจากสาเหตุใดๆ ข้างต้น

- ตรวจสอบสำเนาสำรองของคุณเป็นครั้งคราว

สื่อบันทึกสำเนาสำรองและฮาร์ดแวร์ อาจเชื่อถือไม่ได้ โลกบริษัขนาดใหญ่ของเทปหรือดิสก์เกิดสำเนาสำรองเป็นสิ่งที่ไม่ประโยชน์ ถ้าไม่สามารถอ่านข้อมูลกลับไปบนดิสก์ที่ระบุได้ เพื่อให้แน่ใจว่าสำเนาสำรองของคุณ สามารถใช้ได้ ให้แสดงสารบัญจากเทปสำเนาสำรองเป็นครั้งคราว (โดยใช้ **restore -T** หรือ **tar -t** สำหรับเทป เก็บถาวร) หากคุณใช้ดิสก์เกิดสำเนาสำรองสำเนาสำรองและมีมากกว่าหนึ่งดิสก์เกิด ไดรฟ์ให้อ่านดิสก์เกิดจากไดรฟ์อื่นซึ่งไม่ใช่ไดรฟ์ที่สร้างดิสก์เกิดนั้น คุณอาจต้องการ

ความปลอดภัยโดยการทำซ้ำสำเนาสำรองแต่ละระดับ 0 ด้วยสื่อ บันทึกชุดที่สอง หากคุณใช้อุปกรณ์เทป streaming สำหรับ สำเนาสำรอง คุณสามารถใช้ คำสั่ง `tapechk` เพื่อทำการตรวจสอบความสม่ำเสมอส่วนที่เหลือ บนเทป สำหรับข้อมูลเพิ่มเติม เกี่ยวกับคำสั่งเหล่านี้ ให้ดูที่ `restore -T, tar -t, และ tapechk`

- **เก็บสำเนาสำรองเก่า**

จัดทำรอบสม่ำเสมอสำหรับการใช้สื่อบันทึกสำเนาสำรอง ของคุณซ้ำ อย่างไรก็ตาม อย่าใช้ซ้ำสื่อบันทึกสำเนาสำรองทั้งหมด ในบางครั้ง อาจผ่านไป หลายเดือนก่อนที่คุณหรือผู้ใช้รายอื่นของระบบจะสังเกตเห็นว่าไฟล์ที่สำคัญ เสียหายหรือหายไป ดังนั้นการเก็บรักษาสำเนาสำรองเก่าจึงเป็นประโยชน์สำหรับกรณีดังกล่าว ตัวอย่างเช่น คุณอาจมีรอบของเทปหรือดิสเก็ต สำเนาสำรองสามารถต่อไปนี้:

- หนึ่งครั้งต่อสัปดาห์ รีไซเคิลดิสเก็ตรายวันทั้งหมดยกเว้นดิสเก็ตวันศุกร์
- หนึ่งครั้งต่อเดือน รีไซเคิลดิสเก็ตวันศุกร์ทั้งหมดยกเว้นดิสเก็ตจากวันศุกร์ สุดท้ายของเดือน ซึ่งทำให้มีสำเนาสำรองวันศุกร์สี่ชุดล่าสุดอยู่เสมอ
- หนึ่งครั้งต่อไตรมาส รีไซเคิลดิสเก็ตรายเดือนทั้งหมดยกเว้นสำหรับเดือนล่าสุด ควรเก็บดิสเก็ตรายเดือนล่าสุดแยกต่างหากจากแต่ละไตรมาส ถ้าเป็นไปได้ ควรเก็บในอาคารที่แตกต่างกัน

- **ตรวจสอบระบบไฟล์ก่อนทำสำเนาสำรอง**

สำเนาสำรองที่ทำจาก ระบบไฟล์ที่เสียหายอาจไร้ประโยชน์ ก่อนทำสำเนาสำรองของคุณ ควรมีนโยบาย ตรวจสอบคุณภาพของระบบไฟล์ด้วยคำสั่ง `fsck` สำหรับข้อมูลเพิ่มเติม ให้ดูที่ `fsck`

- **ตรวจสอบให้แน่ใจว่าไฟล์ไม่ได้ถูกใช้อยู่ในระหว่างสำเนาสำรอง**

อย่าใช้ระบบ เมื่อคุณทำสำเนาสำรองของคุณ หากระบบถูกใช้อยู่ ไฟล์สามารถเปลี่ยนแปลง ในขณะที่กำลังถูกทำสำเนาสำรอง และสำเนาสำรองจะไม่ถูกต้อง

- **ทำสำเนาสำรองระบบของคุณก่อนทำการเปลี่ยนแปลงที่สำคัญบนระบบ**

ควรมีนโยบาย ทำสำเนาสำรองทั้งระบบเสมอก่อนทำการทดสอบฮาร์ดแวร์ หรืองานซ่อมแซม หรือก่อนคุณติดตั้งอุปกรณ์ใหม่ โปรแกรม หรือ คุณลักษณะระบบอื่นใด

- **ปัจจัยอื่น**

เมื่อวางแผนและนำกลยุทธ์สำเนาสำรองไปใช้ ให้พิจารณาปัจจัยต่อไปนี้:

- ข้อมูลเปลี่ยนแปลงบ่อยเพียงใด? ข้อมูลระบบปฏิบัติการไม่ได้เปลี่ยนแปลงบ่อยนัก ดังนั้นคุณไม่จำเป็นต้องทำสำเนาสำรองบ่อย ในทางกลับกัน ข้อมูลผู้ใช้ มีการเปลี่ยนแปลงครั้งเป็นปกติ ดังนั้นคุณควรทำสำเนาสำรองข้อมูลผู้ใช้บ่อยๆ
- มีผู้ใช้อยู่บนระบบจำนวนเท่าใด? จำนวนผู้ใช้มีผลกระทบต่อจำนวน ของสื่อบันทึกหน่วยเก็บและความถี่ที่จำเป็นสำหรับสำเนาสำรอง
- การสร้างข้อมูลขึ้นใหม่เป็นเรื่องยากเพียงใด? เป็นสิ่งสำคัญที่ต้องพิจารณา ว่าข้อมูลบางอย่างไม่สามารถสร้างขึ้นใหม่ได้ถ้าไม่มีสำเนาสำรอง

การมีกลยุทธ์สำเนาสำรองเพื่อสำรองข้อมูลของคุณจึงเป็นเรื่องที่สำคัญมาก การประเมินความต้องการของไซต์จะช่วยให้คุณกำหนดนโยบายสำเนาสำรอง ที่ดีที่สุดสำหรับคุณ ควรทำสำเนาสำรองข้อมูลผู้ใช้บ่อยครั้งและสม่ำเสมอ การกู้คืนข้อมูลที่สูญหายไปเป็นเรื่องยากมาก ถ้ายังไม่มีการใช้กลยุทธ์สำเนาสำรองที่ดี

หมายเหตุ: สำหรับสำเนาสำรองของไฟฟ์ที่ระบุ (ไฟฟ์พิเศษ FIFO) ไฟฟ์สามารถ เปิดหรือปิด อย่างไรก็ตาม การเรียกคืนจะล้มเหลวเมื่อทำสำเนาสำรอง บนไฟฟ์ที่ระบุและเปิด เมื่อเรียกคืนไฟฟ์พิเศษ FIFO i-node เป็นเพียง ข้อมูลเดียวที่ต้องการในการสร้างไฟล์ขึ้นใหม่ เนื่องจาก i-node มีข้อมูลลักษณะ ครบทั้งหมด เนื้อหาของไฟฟ์ที่ระบุไม่เกี่ยวข้องกับการเรียกคืน ดังนั้นขนาดไฟล์ในระหว่างสำเนาสำรองจึงเป็นศูนย์ (FIFOs ทั้งหมดปิด) ก่อน ทำสำเนาสำรอง

ข้อควรสนใจ: โพรซีเจอร์การทำสำเนาสำรองระบบและการเรียกคืนกำหนดว่า ระบบจะถูกเรียกคืนบนแพลตฟอร์มชนิดเดียวกันกับที่ทำ สำเนาสำรอง โดยเฉพาะ CPU และ I/O planar boards ต้องเป็นชนิดเดียวกัน

สื่อบันทึกสำเนาสำรอง:

สื่อบันทึกสำเนาสำรองมีอยู่หลายชนิด ชนิดของสื่อบันทึก สำเนาสำรองที่มีอยู่สำหรับคอนฟิกูเรชันระบบเฉพาะของคุณขึ้นอยู่กับ ทั้งซอฟต์แวร์และฮาร์ดแวร์ของคุณ

สื่อบันทึกสำเนาสำรองมีอยู่หลายชนิด ชนิดของสื่อบันทึก สำเนาสำรองที่มีอยู่สำหรับคอนฟิกูเรชันระบบเฉพาะของคุณขึ้นอยู่กับ ซอฟต์แวร์และฮาร์ดแวร์ของคุณ ชนิดที่ใช้กันมากที่สุดคือเทป (เทป 8-มม. และเทป 9-แตรีก) ดิสเก็ต (ดิสเก็ต 5.25-นิ้วและดิสเก็ต 3.5-นิ้ว) ที่เก็บถาวรแบบรีโมต และโลคัลฮาร์ดดิสก์อื่น ยกเว้นว่าคุณระบุอุปกรณ์ที่แตกต่างอื่นโดยใช้ คำสั่ง **backup -f** คำสั่ง **backup** จะเขียนเอาต์พุตไปยัง `/dev/rfd0` ซึ่งเป็นดิสเก็ตไดรฟ์โดยอัตโนมัติ

ข้อควรสนใจ: การรันคำสั่ง **backup** ส่งผลให้ วัสดุทั้งหมดที่จัดเก็บไว้ก่อนหน้านี้บนสื่อสำเนาสำรองที่เลือกสูญหายไป

ดิสเก็ต

ดิสเก็ตเป็นสื่อสำเนาสำรองมาตรฐาน ยกเว้นว่าคุณระบุอุปกรณ์ที่แตกต่างอื่นโดยใช้ คำสั่ง **backup -f** คำสั่ง **backup** จะเขียนเอาต์พุตไปยังอุปกรณ์ `/dev/rfd0` ซึ่งเป็นดิสเก็ตไดรฟ์โดยอัตโนมัติ เมื่อต้องการทำสำเนาสำรองข้อมูลไว้บน อุปกรณ์เทปดีฟอลต์ให้พิมพ์ `/dev/rmt0` และ กด Enter

โปรดใช้ความระมัดระวังเมื่อคุณจัดการกับดิสเก็ต เนื่องจากข้อมูลแต่ละชิ้นส่วน ใช้พื้นที่บนดิสเก็ตเพียงเล็กน้อย ดังนั้น รอยขีดข่วนขนาดเล็ก ผุ่น เศษอาหาร หรือแถบหรืออาจทำให้ข้อมูลไม่สามารถใช้งานได้ โปรดจดจำ สิ่งต่อไปนี้:

- อย่าสัมผัสกับพื้นผิวการบันทึก
- เก็บดิสเก็ตให้ห่างจากแม่เหล็กและแหล่งกำเนิดสนามแม่เหล็ก เช่น โทรศัพท เครื่องมือจัดบันทึก และเครื่องคิดเลขไฟฟ้า
- เก็บดิสเก็ตให้ห่างจากอากาศร้อนจัดและเย็นจัด ช่วงอุณหภูมิที่แนะนำ คือ 10 องศาเซลเซียสถึง 60 องศาเซลเซียส (50 องศาฟาเรนไฮต์ถึง 140 องศาฟาเรนไฮต์)
- การดูแลอย่างถูกต้องช่วยป้องกันการสูญหายของข้อมูล
- จัดทำสำเนาสำรองของดิสเก็ตอย่างสม่ำเสมอ

ข้อควรสนใจ: ดิสเก็ตไดรฟ์และดิสเก็ตต้องเป็น ชนิดที่ถูกต้องเพื่อให้จัดเก็บข้อมูลได้สำเร็จ ถ้าคุณใช้ดิสเก็ตที่ไม่ถูกต้องใน ดิสเก็ตไดรฟ์ 3.5-นิ้ว ข้อมูลบนดิสเก็ตอาจถูกทำลายได้

ดิสเก็ต ไดรฟ์ใช้ดิสเก็ตขนาด 3.5-นิ้วต่อไปนี้:

- ความจุ 1 MB (จัดเก็บข้อมูลประมาณ 720 KB)
- ความจุ 2 MB (จัดเก็บข้อมูลประมาณ 1.44 MB)

เทป

เนื่องจากความจุและความทนทานสูง จึงมักมีการเลือกใช้เทป สำหรับการจัดเก็บไฟล์ขนาดใหญ่หรือไฟล์จำนวนมาก เช่น สำเนาเก็บถาวรของระบบไฟล์ นอกจากนี้ ยังใช้เทปสำหรับการโอนย้ายไฟล์จำนวนมากจากระบบหนึ่งไปยังอีก ระบบหนึ่งด้วย อย่างไรก็ตาม ไม่นิยมใช้เทปสำหรับการจัดเก็บไฟล์ที่เข้าถึงบ่อยครั้งเนื่องจาก สื่อบันทึกอื่นสามารถเข้าถึงได้เร็วกว่า

ไฟล์เทปมีการสร้างขึ้นโดยใช้คำสั่ง เช่น `backup`, `cpio`, และ `tar` ซึ่งเปิดเทปไดรฟ์ เขียนลงในเทปไดรฟ์ และปิดเทปไดรฟ์ตามลำดับ

กลยุทธ์สำเนาสำรอง:

มีสองเมธอดสำหรับการทำสำเนาสำรองข้อมูลจำนวนมาก

- สำเนาสำรองระบบที่สมบูรณ์
- สำเนาสำรองเพิ่มเติม

เพื่อความเข้าใจกับสำเนาสำรองสองชนิดเหล่านี้และชนิดที่เหมาะสมสำหรับ โซต์หรือระบบ สิ่งสำคัญคือการทำ ความเข้าใจกับโครงสร้างระบบไฟล์ และการจัดวางข้อมูล หลังจากคุณตัดสินใจเลือกกลยุทธ์สำหรับการจัดวางข้อมูลแล้ว คุณสามารถจัดทำกลยุทธ์สำเนาสำรองสำหรับข้อมูลนั้นได้

งานที่เกี่ยวข้อง:

“การนำสำเนาสำรองที่จัดตารางเวลาไปใช้” ในหน้า 48

โพธิ์เดอรันอธิบายวิธีการจัดทำและใช้สคริปต์เพื่อทำ สำเนาสำรองแบบเต็มรายสัปดาห์และสำเนาสำรองส่วนเพิ่มรายวันของไฟล์ผู้ใช้

ข้อมูลระบบและข้อมูลผู้ใช้:

ข้อมูลมีการกำหนดเป็นโปรแกรมหรือข้อความ และสำหรับการอธิบายในที่นี้ มีการแบ่งย่อยเป็นสองคลาสดังนี้:

- ข้อมูลระบบ ซึ่งประกอบขึ้นเป็นระบบปฏิบัติการและส่วนขยาย ข้อมูลนี้มีการเก็บไว้ในระบบไฟล์ระบบเสมอ ซึ่งมีชื่อว่า `/` (ราก), `/usr`, `/tmp`, `/var`, และอื่นๆ
- ข้อมูลผู้ใช้โดยปกติเป็นข้อมูลโลคัลที่ผู้ใช้แต่ละรายต้องการเพื่อทำภารกิจเฉพาะ ของตนให้เสร็จสมบูรณ์ ข้อมูลนี้มีการเก็บไว้ในระบบไฟล์ `/home` หรือในระบบไฟล์ที่สร้างขึ้นสำหรับข้อมูลผู้ใช้โดยเฉพาะ

โปรแกรมผู้ใช้และข้อความจะไม่ถูกวางไว้ในระบบไฟล์ที่ออกแบบมาสำหรับ ข้อมูลระบบ ตัวอย่างเช่น ผู้จัดการระบบอาจสร้างระบบไฟล์ใหม่ และติดตั้งบน `/local` ซอยกเว้นคือ `/tmp` ซึ่งใช้สำหรับหน่วยเก็บชั่วคราวของข้อมูลระบบและผู้ใช้

สำเนาสำรอง:

โดยทั่วไป การสำรองข้อมูลของผู้ใช้และระบบจะถูกเก็บไว้ใน กรณีที่ข้อมูลถูกลบโดยไม่ได้ตั้งใจหรือมีดิสก์ล้มเหลว การจัดการสำเนาสำรองสามารถทำได้ง่ายขึ้นเมื่อเก็บข้อมูลผู้ใช้แยกต่างหากจากข้อมูลระบบ

ข้อมูลต่อไปนี้เป็นเหตุผลสำหรับการเก็บข้อมูลระบบแยกต่างหากจากข้อมูลผู้ใช้:

- ข้อมูลผู้ใช้มีแนวโน้มที่จะเปลี่ยนบ่อยกว่าข้อมูลระบบปฏิบัติการมาก รูปภาพสำเนาสำรองจะมีขนาดเล็กลงมากถ้าไม่ได้ทำสำเนาสำรองข้อมูลระบบไว้ใน รูปภาพเดียวกันกับข้อมูลผู้ใช้ จำนวนผู้ใช้มีผลต่อสื่อเก็บข้อมูล และความถี่ที่จำเป็นสำหรับการสำรองข้อมูล
- การเรียกคืนข้อมูลผู้ใช้สามารถทำได้เร็วกว่าและง่ายกว่าเมื่อข้อมูลผู้ใช้ถูกจัดเก็บไว้แยกต่างหาก การเรียกคืนระบบปฏิบัติการ การควบคุมไปกับข้อมูลผู้ใช้ต้องใช้เวลาและความพยายาม มากขึ้น เหตุผลที่วิธีการที่ใช้ กู้คืนข้อมูลของระบบปฏิบัติการเกี่ยวข้องกับ การเริ่มต้น ระบบจากสื่อบันทึกที่ถอดเปลี่ยนได้ (เทปหรือซีดี) และติดตั้งสำเนาของระบบ

เมื่อต้องการทำสำเนาสำรองข้อมูลระบบ ให้ถอนติดตั้งระบบไฟล์ผู้ใช้ทั้งหมด รวมถึง `/home` ด้วย คำสั่ง `umount` หากระบบไฟล์เหล่านี้ ถูกใช้งานอยู่คุณไม่สามารถถอนติดตั้งได้ จัดตารางเวลาสำเนาสำรองในเวลาที่มีการใช้งานน้อย เพื่อให้สามารถถอนติดตั้ง

ตั้งได้ ถ้าระบบไฟล์ข้อมูลผู้ใช้ยังคงติดตั้งอยู่ ข้อมูล จะถูกทำสำเนาสำรองควไปกับข้อมูลระบบปฏิบัติการ ใช้คำสั่ง `mount` เพื่อให้แน่ใจว่ามีารติดตั้งเฉพาะระบบไฟล์ระบบปฏิบัติการเท่านั้น

ระบบไฟล์ที่เมาท์ได้แก่ `/`, `/usr`, `/var`, และ `/tmp` และผลลัพธ์ของคำสั่ง `mount` จะเหมือนกับเอาต์พุตต่อไปนี้:

```
node mounted mounted over vfs      date      options
/dev/hd4      /          jfs      Jun 11 10:36  rw,log=/dev/hd8
/dev/hd2      /usr      jfs      Jun 11 10:36  rw,log=/dev/hd8
/dev/hd9var   /var      jfs      Jun 11 10:36  rw,log=/dev/hd8
/dev/hd       /tmp      jfs      Jun 11 10:36  rw,log=/dev/hd8
```

หลังจากที่คุณแน่ใจว่าระบบไฟล์ของผู้ใช้ทั้งหมดถูกยกเลิกการเมาท์แล้ว คุณก็พร้อมที่จะสำรองข้อมูลของระบบปฏิบัติการ

เมื่อคุณเสร็จสิ้นการสำรองข้อมูลของระบบปฏิบัติการแล้ว ให้เมาท์ระบบไฟล์ ของผู้ใช้โดยใช้คำสั่ง `smit mount` ถัดไป คุณ สามารถทำสำเนาสำรองไฟล์ ระบบไฟล์ หรือกลุ่มวอลุ่มอื่น ตามความต้องการของคุณ

หลักการที่เกี่ยวข้อง:

“รูปภาพระบบและสำเนาสำรองกลุ่มวอลุ่มที่ผู้ใช้กำหนด” ในหน้า 44

Rootvg มีการจัดเก็บไว้บนฮาร์ดดิสก์ หรือกลุ่มของดิสก์ และประกอบด้วย ไฟล์สตาร์ทอัพ, BOS, ข้อมูลคอนฟิกูเรชัน และผลิตภัณฑ์ซอฟต์แวร์ที่เป็นทาง เลือกใดๆ โดยปกติ *กลุ่มวอลุ่มที่ผู้ใช้กำหนด* (เรียกอีกอย่างว่า *กลุ่มวอลุ่ม nonrootvg*) ประกอบด้วย ไฟล์ข้อมูลและแอปพลิเคชันซอฟต์แวร์

การทำซ้ำระบบ (การลอกแบบ):

การลอกแบบจะบันทึกข้อมูลการกำหนดคอนฟิกพร้อมข้อมูลผู้ใช้ หรือข้อมูลระบบ ตัวอย่างเช่น คุณอาจต้องการทำซ้ำระบบ หรือกลุ่มวอลุ่ม ซึ่งบางครั้งเรียกว่าการลอกแบบ

จากนั้น คุณสามารถติดตั้งรูปภาพนี้บนระบบอื่นและสามารถใช้ได้เช่นเดียวกับ ระบบแรก คำสั่ง `mksysb` ใช้เพื่อลอกแบบกลุ่มวอลุ่ม `rootvg` ซึ่งมีระบบปฏิบัติการ ในขณะที่คำสั่ง `savevg` ใช้ เพื่อลอกแบบกลุ่มวอลุ่ม

ข้อมูลสรุปคำสั่งสำหรับไฟล์สำเนาสำรองและสื่อบันทึกหน่วยเก็บ:

มีคำสั่งสำหรับการทำสำเนาสำรองไฟล์และการจัดเก็บข้อมูล

ไอเท็ม	คำอธิบาย
<code>backup</code>	ทำสำเนาสำรองไฟล์และระบบไฟล์
<code>compress</code>	บีบอัดและขยายข้อมูล
<code>cpio</code>	คัดลอกไฟล์ไปยังและออกจากหน่วยเก็บถาวรและไดเรกทอรี
<code>fdformat</code>	จัดรูปแบบดิสเก็ต
<code>flcopy</code>	คัดลอกไปยังและออกจากดิสเก็ต
<code>format</code>	จัดรูปแบบดิสเก็ต
<code>fsck</code>	ตรวจสอบความสอดคล้องกันของระบบไฟล์และซ่อมแซม ระบบไฟล์แบบโต้ตอบ
<code>pack</code>	บีบอัดไฟล์
<code>restore</code>	คัดลอกระบบไฟล์หรือไฟล์ที่ทำสำเนาสำรองไว้ก่อนหน้านั้น ซึ่งถูก สร้างขึ้นโดยคำสั่ง <code>backup</code> จากอุปกรณ์โลดัล
<code>tapechk</code>	ตรวจสอบความสอดคล้องกันของอุปกรณ์เทปการสตรีม
<code>tar</code>	จัดการหน่วยเก็บถาวร
<code>tcopy</code>	คัดลอกเทปแม่เหล็ก
<code>uncompress</code>	บีบอัดและขยายข้อมูล

การจัดการสำเนาสำรองระบบ

มีหลายวิธีในการทำสำเนาสำรองระบบของคุณและเรียกคืน สำเนาสำรองระบบ

การทำสำเนาสำรองไฟล์ผู้ใช้หรือระบบไฟล์:

สามารถใช้สองโปรแกรมเพื่อทำสำเนาสำรองไฟล์และระบบไฟล์: พาด่วน SMIT `smit backfile` หรือ `smit backfilesys`, และคำสั่ง `backup`

- หากคุณกำลังทำสำเนาสำรองโดยระบบไฟล์ `i-node` ที่อาจใช้งานอยู่ให้ถอนติดตั้ง ระบบไฟล์นั้นก่อนเพื่อป้องกันความไม่สอดคล้องกัน
ข้อควรสนใจ: หากคุณพยายามสำรองระบบไฟล์ที่ถูกติดตั้งอยู่จะมีข้อความเตือนแสดง คำสั่ง `backup` ยังทำต่อไป แต่อาจเกิดความไม่สอดคล้องกันขึ้นในระบบไฟล์ คำเตือนนี้ไม่ใช่กับระบบไฟล์ราก (`/`)
- ตรวจสอบให้แน่ใจว่าอุปกรณ์สำเนาสำรองถูกทำความสะอาดเมื่อเร็วๆ นี้เพื่อ ป้องกันข้อผิดพลาด

ในการทำสำเนาสำรองไฟล์ผู้ใช้และระบบไฟล์ คุณสามารถใช้พาด่วน SMIT `smit backfile` หรือ `smit backfilesys`

คุณสามารถใช้อินเตอร์เฟส SMIT เพื่อทำสำเนาสำรอง ระบบไฟล์เดี่ยวและระบบไฟล์ขนาดเล็กตามชื่อ เช่น `/home` บนระบบโลคัลของคุณ โปรดทราบว่า SMIT ไม่สามารถจัดทำที่เก็บถาวรในรูปแบบอื่นนอกเหนือ จากที่ระบุโดยคำสั่ง `backup` นอกจากนี้ บาง แฟล็กของคำสั่ง `backup` ไม่มีอยู่ผ่านทาง SMIT SMIT อาจไม่ตอบสนองถ้าต้องการเทปหรือดิสก์หลายแผ่นในระหว่างสำเนาสำรอง สำหรับข้อมูลเพิ่มเติม ใหญ่ที่คำอธิบายคำสั่ง `backup` ใน *Commands Reference, Volume 1*

ใช้คำสั่ง `backup` เมื่อคุณต้องการทำสำเนาสำรองระบบไฟล์ขนาดใหญ่และหลายระบบ คุณสามารถระบุ ตัวเลขระดับเพื่อควบคุมจำนวนข้อมูลที่จะทำสำเนาสำรอง (สมบูรณ์, 0; เพิ่มเติม, 1-9) การใช้คำสั่ง `backup` เป็นวิธีเดียวที่คุณสามารถ ระบุตัวเลขระดับบนสำเนาสำรองได้

คำสั่ง `backup` สร้างสำเนาในรูปแบบสำเนาสำรองอย่างใดอย่างหนึ่งจากสองรูปแบบต่อไปนี้:

- ไฟล์เฉพาะที่สำรองข้อมูลตามชื่อโดยใช้ แฟล็ก `-i`
- ทั้งระบบไฟล์ที่ทำสำเนาสำรองตาม `i-node` โดยใช้พารามิเตอร์ `-Level` และ `FileSystem` ระบบไฟล์ถูกยกเลิกแฟร็กเมนต์เมื่อถูกเรียกคืนจากสำเนาสำรอง

ข้อควรสนใจ: การทำสำเนาสำรองตาม `i-node` ทำงานไม่ถูกต้องสำหรับ ไฟล์ที่มี ID ผู้ใช้ (UID) หรือ ID กลุ่ม (GID) มากกว่า 65535 ไฟล์เหล่านี้ มีการทำสำเนาสำรองด้วย UID หรือ GID ที่ตัดสั้นลง ดังนั้นจะมีแอ็ททริบิวต์ UID หรือ GID ที่ไม่ถูกต้องเมื่อเรียกคืน สำหรับกรณีเหล่านี้ คุณต้องทำสำเนาสำรอง ตามชื่อ

ภารกิจการทำสำเนาสำรองไฟล์ผู้ใช้ หรือระบบไฟล์

ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
ทำสำเนาสำรองไฟล์ผู้ใช้	smit backfile	<ol style="list-style-type: none"> 1. ล็อกอินเข้าสู่แอคเคาต์ผู้ใช้ของคุณ 2. สำเนาสำรอง: <code>find . -print backup -ivf /dev/rmt0</code>
ทำสำเนาสำรองระบบไฟล์ผู้ใช้	smit backfilesys	<ol style="list-style-type: none"> 1. ถอนติดตั้งระบบไฟล์ซึ่งคุณวางแผนจะทำสำเนาสำรอง ตัวอย่างเช่น: ถอนติดตั้ง ทั้งหมด หรือ ถอนติดตั้ง <code>/home /filesys1</code> 2. ตรวจสอบระบบไฟล์ ตัวอย่างเช่น: <code>fsck /home /filesys1</code> 3. ทำสำเนาสำรองตาม i-node ตัวอย่างเช่น: <code>backup -5 -uf /dev/rmt0 /home/libr</code> 4. เรียกคืนไฟล์โดยใช้คำสั่งต่อไปนี้: <code>restore -t</code>

หมายเหตุ: หากคำสั่งนี้สร้างข้อความแสดงข้อผิดพลาด คุณต้องทำซ้ำ ทั้งสำเนาสำรอง

หลักการที่เกี่ยวข้อง:

“สำเนาสำรองระบบ” ในหน้า 23

หลังจากใช้ระบบของคุณแล้ว ข้อควรพิจารณาถัดไปของคุณควรจะเป็น การทำสำเนาสำรองระบบไฟล์ ไดรฟ์ทอริ และไฟล์ หาก你做สำเนาสำรองระบบไฟล์ คุณสามารถเรียกคืนไฟล์หรือระบบไฟล์ในกรณีที่ฮาร์ดดิสก์ เสียหายได้ มีเมธอดที่แตกต่างกันสำหรับการทำสำเนาสำรองข้อมูล

การเรียกคืนไฟล์ที่ทำสำเนาสำรอง:

หลังจากข้อมูลมีการทำสำเนาสำรองอย่างถูกต้อง มีหลายเมธอดในการ เรียกคืนข้อมูล ทั้งนี้ขึ้นอยู่กับชนิดของคำสั่ง backup ที่คุณใช้

คุณจำเป็นต้องทราบวิธีการสร้างสำเนาสำรองหรือที่เก็บถาวรของคุณ เพื่อให้ สามารถเรียกคืนได้อย่างถูกต้อง แต่ละโปรซีเจอร์ สำเนาสำรองให้ข้อมูลเกี่ยวกับการเรียกคืนข้อมูล ตัวอย่างเช่น ถ้าคุณใช้คำสั่ง **backup** คุณสามารถระบุ สำเนาสำรองโดยใช้ระบบไฟล์หรือโดยใช้ชื่อ อย่างใดอย่างหนึ่ง สำเนาสำรองนั้นต้องถูกเรียกคืนในวิธี ที่ทำ โดยใช้ระบบไฟล์หรือโดยใช้ชื่อ สำหรับข้อมูลเกี่ยวกับคำสั่ง **backup** ให้ดูที่ **backup**

หลายคำสั่งเรียกคืนข้อมูลที่ทำสำเนาสำรองไว้ เช่น:

ไอเท็ม	คำอธิบาย
restore	คัดลอกไฟล์ที่สร้างขึ้นโดยคำสั่ง backup สำหรับ ข้อมูลเพิ่มเติมเกี่ยวกับการใช้คำสั่งนี้ ให้ดูที่ส่วนข้างล่าง
rrestore	คัดลอกระบบไฟล์ที่ทำสำเนาสำรองบนเครื่องรีโมตไปยังเครื่องโลคัล สำหรับข้อมูลเพิ่มเติม ให้ดูที่ rrestore
cpio	คัดลอกไฟล์เข้าในและออกจากหน่วยเก็บถาวร สำหรับข้อมูลเพิ่มเติม ให้ดูที่ cpio
tar	สร้างหรือจัดวางที่เก็บถาวร tar สำหรับข้อมูลเพิ่มเติม ให้ดูที่ tar
pax	(POSIX-ยูทิลิตี้ที่เก็บถาวรแบบสอดคล้องกัน) อ่านและเขียนที่เก็บถาวร tar และ cpio สำหรับข้อมูลเพิ่มเติม ให้ดูที่ pax

ส่วนต่อไปนี้อธิบายคำสั่ง **restore** และ **smit**

หมายเหตุ:

- ไฟล์ต้องถูกเรียกคืนโดยใช้เมธอดเดียวกับที่ทำสำเนาสำรอง ตัวอย่างเช่น ถ้าระบบไฟล์มีการทำสำเนาสำรองโดยใช้ชื่อระบบไฟล์นั้นต้อง มีการเรียกคืนโดยใช้ชื่อ
- เมื่อต้องการมากกว่าหนึ่งดิสเก็ต คำสั่ง **restore** จะอ่าน ดิสเก็ตที่ติดตั้ง พร้อมทำให้คุณป้อนดิสเก็ตใหม่ และรอการตอบกลับของคุณ หลังจากใส่ดิสเก็ตใหม่แล้ว ให้กดปุ่ม Enter เพื่อทำการเรียกคืนไฟล์ ต่อไป

การเรียกคืนไฟล์โดยใช้คำสั่ง restore

ใช้คำสั่ง **restore** เพื่ออ่านไฟล์ที่เขียนโดย คำสั่ง **backup** และเรียกคืนไฟล์นั้นบนระบบโลคัลของคุณ

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการแสดงรายการชื่อไฟล์ที่ทำสำเนาสำรองไว้ก่อนหน้านี้ ให้พิมพ์ดังต่อไปนี้:

```
restore -T
```

ข้อมูลถูกอ่านจากอุปกรณ์สำเนาสำรองดีฟอลต์ `/dev/rfd0` หากทำสำเนาสำรองที่ละไฟล์ เฉพาะชื่อไฟล์เท่านั้นจะแสดงขึ้น หากทำสำเนาสำรองทั้งระบบไฟล์ หมายเลขไอ-โหนดจะแสดงขึ้น ด้วย

- เมื่อต้องการเรียกคืนไฟล์ไปยังระบบไฟล์หลัก ให้พิมพ์ดังต่อไปนี้:

```
restore -x -v
```

แฟล็ก `-x` แยกไฟล์ทั้งหมดออกจากสื่อบันทึกสำเนาสำรอง และเรียกคืนไฟล์ดังกล่าวไปยัง ตำแหน่งที่เหมาะสมในระบบไฟล์ แฟล็ก `-v` แสดงรายงานความคืบหน้าเมื่อ แต่ละไฟล์ถูกเรียกคืน หากกำลังเรียกคืนสำเนาสำรองของระบบไฟล์ ไฟล์ จะถูกตั้งชื่อด้วยหมายเลขไอ-โหนด มิฉะนั้น จะแสดงเฉพาะชื่อเท่านั้น

- เมื่อต้องการคัดลอกไฟล์ `/home/mike/manual/chap1` ให้พิมพ์ ดังต่อไปนี้:

```
restore -xv /home/mike/manual/chap1
```

คำสั่งนี้ แยกไฟล์ `/home/mike/manual/chap1` ออกจาก สื่อสำเนาสำรองและเรียกคืนไฟล์ `/home/mike/manual/chap1` ต้องเป็นชื่อที่คำสั่ง **restore -T** สามารถแสดงได้

- เมื่อต้องการคัดลอกไฟล์ทั้งหมดในไดเรกทอรีที่ชื่อ `manual` ให้พิมพ์ดังต่อไปนี้:

```
restore -xdv manual
```

คำสั่งนี้ เรียกคืนไดเรกทอรี `manual` และไฟล์ในไดเรกทอรี นั้น หากไม่มีไดเรกทอรี ไดเรกทอรีที่ชื่อ `manual` จะถูกสร้างขึ้นในไดเรกทอรีปัจจุบันเพื่อเก็บไฟล์ที่กำลังเรียกคืน

ให้ดูที่คำสั่ง **restore** ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

การเรียกคืนไฟล์โดยใช้คำสั่ง smit

ใช้คำสั่ง **smit** เพื่อรัน คำสั่ง **restore** ซึ่งจะอ่านไฟล์ที่เขียนโดยคำสั่ง **backup** และเรียกคืนไฟล์ดังกล่าวไปบนระบบโลคัลของคุณ

1. ที่พร้อมท์ ให้พิมพ์ดังต่อไปนี้:

```
smit restore
```

2. จัดทำรายการของคุณในฟิลด์ **Target DIRECTORY** นี้เป็นไดเรกทอรีซึ่งคุณต้องการจัดวางไฟล์ที่เรียกคืน
3. ไปต่อที่ฟิลด์ **BACKUP device** หรือ **FILE** และป้อนชื่ออุปกรณ์เอาต์พุต ดังเช่นในตัวอย่างต่อไปนี้สำหรับ อุปกรณ์เทปแม่เหล็ก ดิบ:

/dev/rmt0

ถ้าอุปกรณ์ไม่มีอยู่ข้อความคล้ายกับตัวอย่างต่อไปนี้จะแสดงขึ้น:

Cannot open /dev/rmtX, no such file or directory.

ข้อความนี้บ่งชี้ว่าระบบไม่สามารถเข้าถึงไดรเวอร์อุปกรณ์ได้เนื่องจากไม่มีไฟล์ rmtX อยู่ในไดเรกทอรี /dev เฉพาะไอเท็มที่มีสถานะเป็น available เท่านั้นที่มีอยู่ในไดเรกทอรี /dev

4. สำหรับไฟล์จำนวนบล็อกที่จะอ่านในอินพุตเดียว แนะนำให้ใช้ค่าดีฟอลต์
5. กด Enter เพื่อเรียกคืนระบบไฟล์หรือไดเรกทอรีที่ระบุ

การสร้างที่เก็บถาวรรีโมต:

ใช้โปรแกรมนี้เพื่อเก็บถาวรไฟล์ไว้ในอุปกรณ์เทปรีโมต

การรันระบบ AIX ไม่สามารถติดตั้งอุปกรณ์เทปรีโมตทราบเท่าที่เป็นแบบโลคัลในระบบ อย่างไรก็ตาม สามารถส่งข้อมูลไปยังอุปกรณ์เทปบนเครื่องรีโมตได้โดยใช้คำสั่ง rsh โปรแกรมต่อไปนี้เขียนลงในเทปเดียวเท่านั้น ที่เก็บถาวรหลายเทปต้องการแอปพลิเคชันซอฟต์แวร์พิเศษ

ในโปรแกรมต่อไปนี้มีการสมมติต่อไปนี้:

blocksize

แสดงถึงขนาดบล็อกอุปกรณ์เทปเป้าหมาย

remotehost

เป็นชื่อของระบบเป้าหมาย (ระบบที่มีเทปไดร์ฟ)

sourcehost

เป็นชื่อของระบบต้นทาง (ระบบที่กำลังถูกเก็บถาวร)

/dev/rmt0

เป็นชื่อของอุปกรณ์เทปรีโมต

pathname

แสดงถึงชื่อพาธแบบเต็มของไดเรกทอรีหรือไฟล์ที่ต้องการ

เมื่อใช้คำแนะนำต่อไปนี้ให้สมมติว่าทั้ง ผู้ใช้โลคัลและรีโมตเป็นผู้ใช้ราก

1. ตรวจสอบให้แน่ใจว่าคุณมีสิทธิ์เข้าถึงเครื่องรีโมต เครื่องต้นทาง ต้องมีสิทธิ์เข้าถึงระบบที่มีเทปไดร์ฟ (ระบบเป้าหมายสามารถเข้าถึงได้โดยใช้ผู้ใช้ที่กำหนดใดๆ บนระบบนั้น แต่ชื่อผู้ใช้ต้องมีสิทธิ์การใช้งานรากเพื่อทำขั้นตอนหลายอย่างต่อไปนี้)
2. ใช้โปรแกรมแก้ไขที่โปรดปรานเพื่อสร้างไฟล์ในไดเรกทอรี / (ราก) ของระบบเป้าหมายที่เรียกว่า .rhosts ซึ่งช่วยให้ระบบต้นทางเข้าถึงระบบเป้าหมายได้ คุณต้องเพิ่ม ชื่อโฮสต์ที่ได้รับอนุญาตและ ID ผู้ใช้ลงในไฟล์นี้ เมื่อต้องการกำหนดชื่อของ เครื่องต้นทางสำหรับไฟล์ .rhosts คุณสามารถใช้คำสั่งต่อไปนี้:

```
host SourceIPAddress
```

สำหรับวัตถุประสงค์ของตัวอย่างนี้ สมมติว่าคุณเพิ่มบรรทัดต่อไปนี้ลงในไฟล์ .rhosts:

```
sourcehost.mynet.com root
```

3. บันทึกไฟล์ แล้วเปลี่ยนสิทธิ์อนุญาตของไฟล์โดยใช้คำสั่ง ต่อไปนี้:

32 AIX เวอร์ชัน 7.2: ระบบปฏิบัติการและการจัดการกับอุปกรณ์

```
chmod 600 .rhosts
```

4. ใช้คำสั่ง `rsh` เพื่อทดสอบสิทธิเข้าถึงของคุณจาก เครื่องต้นทาง ตัวอย่างเช่น:

```
rsh remotehost
```

หากตั้งค่าทุกอย่างถูกต้อง คุณควรได้รับสิทธิเข้าถึงเซลล์ ในเครื่องรีโมต คุณไม่ควรเห็นล็อกอินพร้อมที่ร้องขอ ชื่อผู้ใช้ พิมพ์ `exit` เพื่อล็อกเอาต์ออกจากเซลล์ทดสอบนี้

5. ตัดสินใจเลือกขนาดบล็อกอุปกรณ์เทปที่เหมาะสม ข้อมูล ต่อไปนี้เป็นค่าที่แนะนำ:

ไอเท็ม	คำอธิบาย
ขนาดบล็อกสี่บันทึก 9-แทร็ก หรือ 0.25-นิ้ว:	512
ขนาดบล็อกสี่บันทึก 8-mm หรือ 4-mm:	1024

หากคุณไม่แน่ใจและต้องการตรวจสอบขนาดบล็อกปัจจุบันของ อุปกรณ์เทป ให้ใช้คำสั่ง `tctl` ตัวอย่างเช่น:

```
tctl -f /dev/rmt0 status
```

ถ้าคุณต้องการเปลี่ยนขนาดบล็อกเทป ให้ใช้คำสั่ง `chdev` ตัวอย่างเช่น:

```
chdev -l rmt0 -a block_size=1024
```

6. สร้างที่เก็บถาวรของคุณโดยใช้เมธอดอย่างใดอย่างหนึ่งต่อไปนี้:

สำเนาสำรองตามชื่อ

เมื่อต้องการสร้างที่เก็บถาวรสำเนาสำรองแบบรีโมตตามชื่อ ให้ใช้คำสั่งต่อไปนี้:

```
find pathname -print | backup -ivqf | rsh remotehost \  
"dd of=/dev/rmt0 bs=blocksize conv=sync"
```

สำเนาสำรองตามไอโฟนด

เมื่อต้องการสร้างที่เก็บถาวรสำเนาสำรองแบบรีโมตตามไอโฟนด อันดับแรก ให้ถอนติดตั้งระบบไฟล์ จากนั้นใช้คำสั่ง `backup` ตัวอย่างเช่น:

```
umount /myfs  
backup -0 -uf- /myfs | rsh remotehost \  
"dd of=/dev/rmt0 bs=blocksize conv=sync"
```

สร้างและคัดลอกที่เก็บถาวรไปยังเทปรีโมต

เมื่อต้องการสร้างและคัดลอกที่เก็บถาวรไปยังอุปกรณ์เทปรีโมต ให้ใช้คำสั่ง ต่อไปนี้:

```
find pathname -print | cpio -ovcB | rsh remotehost \  
"dd ibs=5120 obs=blocksize of=/dev/rmt0"
```

สร้างที่เก็บถาวร tar

เมื่อต้องการสร้างที่เก็บถาวร `tar` แบบรีโมต ให้ใช้คำสั่งต่อไปนี้:

```
tar -cvdf- pathname | rsh remotehost \  
"dd of=/dev/rmt0 bs=blocksize conv=sync"
```

สร้างรีโมตดัมพ์

เมื่อต้องการสร้างรีโมตดัมพ์แบบรีโมตของระบบไฟล์ `/myfs` ให้ใช้คำสั่งต่อไปนี้:

```
rdump -u -0 -f remotehost:/dev/rmt0 /myfs
```

แฟล็ก `-u` บอกให้ระบบอัปเดตเรียกคอร์ตระดับสำเนาสำรองปัจจุบันในไฟล์ `/etc/dumpdates -O` เป็นค่าติดตั้งของแฟล็ก ระดับ ระดับสำเนาสำรอง 0 ระบุว่าไฟล์ทั้งหมดในไดเรกทอรี `/myfs` จะถูกทำสำเนา สำรอง สำหรับข้อมูลเพิ่มเติม ให้ดูที่คำอธิบายคำสั่ง `rdump` ใน *Commands Reference, Volume 4*

7. เรียกคืนที่เก็บถาวรรีโมตของคุณโดยใช้เมธอดอย่างใดอย่างหนึ่งต่อไปนี้:

เรียกคืนสำเนาสำรองตามชื่อ

เมื่อต้องการเรียกคืนที่เก็บถาวรสำเนาสำรองแบบรีโมตตามชื่อ ให้ใช้คำสั่งต่อไปนี้:

```
rsh remotehost "dd if=/dev/rmt0 bs=blocksize" | restore \
-xvqdf- pathname
```

เรียกคืนสำเนาสำรองตามไอโหนด

เมื่อต้องการเรียกคืนที่เก็บถาวรสำเนาสำรองแบบรีโมตตามไอโหนด ให้ใช้คำสั่งต่อไปนี้:

```
rsh remotehost "dd if=/dev/rmt0 bs=blocksize" | restore \
-xvqf- pathname
```

เรียกคืนที่เก็บถาวร cpio รีโมต

เมื่อต้องการเรียกคืนที่เก็บถาวรรีโมตที่สร้างขึ้นด้วยคำสั่ง `cpio` ให้ใช้คำสั่งต่อไปนี้:

```
rsh remotehost "dd if=/dev/rmt0 ibs=blocksize obs=5120" | \
cpio -icvdumB
```

เรียกคืนที่เก็บถาวร tar

เมื่อต้องการเรียกคืนที่เก็บถาวร `tar` แบบรีโมต ให้ใช้คำสั่งต่อไปนี้:

```
rsh remotehost "dd if=/dev/rmt0 bs=blocksize" | tar -xvpf- pathname
```

เรียกคืนรีโมตเต็มพี

เมื่อต้องการเรียกคืนรีโมตเต็มพีของระบบไฟล์ `/myfs` ให้ใช้คำสั่งต่อไปนี้:

```
cd /myfs
rrestore -rvf remotehost:/dev/rmt0
```

การเรียกคืนไฟล์ผู้ใช้จากรูปภาพสำเนาสำรอง:

ถ้าคุณต้องการเรียกคืนรูปภาพสำเนาสำรองที่ถูกทำลายโดยอุบัติเหตุ ปัญหาที่ยากที่สุดของคุณคือการกำหนดเทปสำเนาสำรองที่มีไฟล์นี้ คำสั่ง `restore -T` สามารถใช้เพื่อแสดงรายการเนื้อหาของที่เก็บถาวร ถือเป็นแนวคิดที่ดีในการเรียกคืนไฟล์ในไดเรกทอรี `/tmp` เพื่อให้คุณไม่เขียนทับไฟล์อื่นของผู้ใช้โดยไม่ได้ตั้งใจ

ตรวจสอบให้แน่ใจว่าอุปกรณ์เชื่อมต่อและมีอยู่ เมื่อต้องการตรวจสอบการมีอยู่ให้พิมพ์:

```
lsdev -C | pg
```

หากกลยุทธ์สำเนาสำรองรวมสำเนาสำรองส่วนเพิ่ม การค้นหาเวลาที่ไฟล์ถูกแก้ไขครั้งล่าสุดจากผู้ใช้นับเป็นสิ่งที่มีความสำคัญ เนื่องจากช่วยในการกำหนดสำเนาสำรองส่วนเพิ่มที่มีไฟล์ หากไม่สามารถหาข้อมูลนี้ได้ หรือพบว่าไม่ถูกต้อง ให้เริ่มต้นการค้นหาสำเนาสำรองส่วนเพิ่มในลำดับย้อนกลับ (7, 6, 5, ...). สำหรับสำเนาสำรองระบบไฟล์ส่วนเพิ่ม แฟล็ก `-i` (โหมดโต้ตอบ) ของคำสั่ง `restore` มีประโยชน์มากทั้งในการระบุตำแหน่งและการเรียกคืนไฟล์ที่หายไป (โหมดโต้ตอบ ยังมีประโยชน์สำหรับการเรียกคืนแอคเคาต์ของผู้ใช้แต่ละรายจากสำเนาสำรอง ของระบบไฟล์ `/home` ด้วย)

โปรดดูในตารางต่อไปนี้อธิบายวิธีการนำการเรียกคืนระดับ 0 (เต็ม) ของไดเรกทอรีหรือระบบไฟล์ไปใช้

ภารกิจการเรียกคืนจากรูปภาพ สำเนาสำรอง		
ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
เรียกคืนไฟล์ผู้ใช้ที่ละไฟล์	smit restfile	ให้ดูที่คำสั่ง restore
การเรียกคืนระบบไฟล์ผู้ใช้	smit restfilesystem	<ol style="list-style-type: none"> mkfs /dev/hd1 mount /dev/hd1 /filesystem cd /filesystem restore -r
การเรียกคืนกลุ่มวอลุ่มผู้ใช้	smit restvg	ให้ดูที่คำสั่ง restvg -q

งานที่เกี่ยวข้อง:

“การแก้ไขระบบไฟล์ที่เสียหาย” ในหน้า 487

ระบบไฟล์อาจเกิดความเสียหายได้เมื่อข้อมูลไอ-โหนดหรือซูเปอร์บล็อก สำหรับโครงสร้างไดเรกทอรีของระบบไฟล์เสียหาย

การเรียกคืนสิทธิ์เข้าถึงไลบรารีระบบที่ไม่ได้ลิงก์หรือที่ลบ:

เมื่อไลบรารี `libc.a` ที่มีอยู่ไม่พร้อมใช้งาน จะส่งผลให้คำสั่ง ระบบปฏิบัติการส่วนใหญ่ไม่เป็นที่รู้จัก

สาเหตุที่เป็นไปได้มากที่สุดสำหรับปัญหานี้มีดังต่อไปนี้:

- ลิงก์ใน `/usr/lib` ไม่มีอยู่อีกต่อไป
- ไฟล์ใน `/usr/ccs/lib` ถูกลบออกแล้ว

โปรดดูต่อไปนี้อธิบายวิธีการเรียกคืนสิทธิ์เข้าถึงไลบรารี `libc.a` โปรดดูว่านี่ต้องการดาวน์โหลดใหม่ระบบ ถ้าเป็นไปได้ให้จัดตารางเวลา ดาวน์โหลดของคุณเป็นเวลาที่จะส่งผลกระทบต่อเวิร์กโพลด์ของคุณน้อยที่สุด เพื่อป้องกัน การสูญหายของข้อมูลหรือการทำงาน

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง mount

คำสั่ง unmount

คำสั่ง reboot

การเรียกคืนลิงก์สัญลักษณ์ที่ลบออก:

ใช้พร็อกซีต่อไปนี่เพื่อเรียกคืนลิงก์สัญลักษณ์จากไลบรารี `/usr/lib/libc.a` ไปยังพาส `/usr/ccs/lib/libc.a`

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

1. ด้วยสิทธิการใช้งานราก ตั้งค่าตัวแปรสถานะแวดล้อม `LIBPATH` เพื่อชี้ไปยังไดเรกทอรี `/usr/ccs/lib` โดยพิมพ์ คำสั่งต่อไปนี้:

```
# LIBPATH=/usr/ccs/lib:/usr/lib
# export LIBPATH
```

ณ จุดนี้ คุณควรจะสามารถดำเนินการคำสั่งระบบได้

- เมื่อต้องการเรียกคืนลิงก์จากไลบรารี /usr/lib/libc.a และไดเรกทอรี /lib ไปยังไดเรกทอรี /usr/lib ให้พิมพ์คำสั่งต่อไปนี้:

```
ln -s /usr/ccs/lib/libc.a /usr/lib/libc.a
ln -s /usr/lib /lib
```

ณ จุดนี้ คำสั่งควรจะรัน เหมือนแต่ก่อน หากคุณยังคงไม่มีสิทธิเข้าถึงเซลล์ให้ข้ามส่วนที่เหลือของ โพรซีเดอร์นี้และทำส่วนถัดไป “การเรียกคืนไฟล์ไลบรารีระบบที่ลบบอก”

- พิมพ์คำสั่งต่อไปนี้เพื่อยกเลิกการตั้งค่าตัวแปรสภาวะแวดล้อม LIBPATH

```
unset LIBPATH
```

การเรียกคืนไฟล์ไลบรารีระบบที่ลบบอก:

โพรซีเดอร์ในการเรียกคืนไฟล์ไลบรารีระบบที่ลบบอกนี้ต้องการ ดาวนโหลดระบบ มีการบูตระบบ จากนั้นเรียกคืนไลบรารีจาก เทป mksysb ล่าสุด

- ก่อนการรีบูต ตรวจสอบให้แน่ใจว่าฟิลด์ **PROMPT** ใน ไฟล์ bosinst.data มีการตั้งค่าเป็น yes
- ใส่เทป mksysb ล่าสุดเข้าในเทปไดรฟ์ mksysb ต้อง มี OS และแพ็คเกจการบำรุงรักษา หรือระดับเทคโนโลยีเดียวกันกับระบบที่ติดตั้ง หากคุณเรียกคืนไลบรารี libc.a จาก mksysb ที่ขัดแย้งกับระดับบนระบบที่ติดตั้ง คุณจะไม่สามารถออกใช้คำสั่งได้
- รีบูตเครื่อง
- เมื่อหน้าจอของไคคอนปรากฏขึ้น หรือเมื่อคุณได้ยินเสียงบี๊บสองครั้ง ให้กดปุ่ม F1 ซ้ำๆ จนกว่าเมนู เซอร์วิสการจัดการระบบ แสดงขึ้น
- เลือก **มัลติบูต**
- เลือก **ติดตั้งจาก**
- เลือกอุปกรณ์เทปที่มี mksysb แล้ว เลือก **ติดตั้ง** อาจใช้เวลาหลายนาที ก่อนพร้อมท์ถัดไปแสดงขึ้น
- กำหนดระบบปัจจุบันของคุณเป็นคอนโซลระบบโดยกดปุ่ม F1 แล้วกด Enter
- เลือกหมายเลขของภาษาที่ต้องการ และกด Enter
- เลือก **เริ่มต้นโหมดการบำรุงรักษาสำหรับการกู้คืนระบบ** โดย พิมพ์ 3 และกด Enter
- เลือก **เข้าถึงกลุ่มวอลุ่มราก** ข้อความแสดงขึ้น เพื่ออธิบายว่าคุณจะไม่สามารถกลับไปยังเมนู การติดตั้ง โดยไม่ได้รีบูต ถ้าคุณเปลี่ยนกลุ่มวอลุ่มราก ณ จุดนี้
- พิมพ์ 0 และกด Enter
- พิมพ์หมายเลขของกลุ่มวอลุ่มที่เหมาะสมจากรายการ และ กด Enter
- เลือก **เข้าถึงกลุ่มวอลุ่มนี้** โดย พิมพ์ 2 และกด Enter
- ติดตั้งระบบไฟล์ / (ราก) และ /usr โดยพิมพ์คำสั่งต่อไปนี้:

```
mount /dev/hd4 /mnt
mount /dev/hd2 /mnt/usr
cd /mnt
```

16. เมื่อต้องการเรียกคืนลิงก์สัญลักษณ์สำหรับไลบรารี libc.a ถ้าต้องการให้พิมพ์คำสั่งต่อไปนี้:

```
ln -s /usr/ccs/lib/libc.a /mnt/usr/lib/libc.a
```

หลังจากคำสั่งรันให้ทำอย่างใดอย่างหนึ่งต่อไปนี้:

- ถ้าคำสั่งสำเร็จให้ข้ามไปยังขั้นตอนที่ 20
- ถ้าข้อความแสดงว่ามีลิงก์อยู่แล้วให้ทำขั้นตอนที่ 17 ต่อไป

17. ตั้งค่าขนาดบล็อกของเทปไดรฟ์โดยออกใช้คำสั่งต่อไปนี้ โดยที่ X คือจำนวนของเทปไดรฟ์ที่เหมาะสม

```
tctl -f /dev/rmtX rewind
tctl -f /dev/rmtX.1 fsf 1
restbyname -xvqf /dev/rmtX.1 ./tapeblksz
cat tapeblksz
```

หากค่าจากคำสั่ง `cat tapeblksz` ไม่เท่ากับ 512 ให้พิมพ์คำสั่งต่อไปนี้ เพื่อแทนที่ Y ด้วยค่าจากคำสั่ง `cat tapeblksz`:

```
ln -sf /mnt/usr/lib/methods /etc/methods
/etc/methods/chgdevn -l rmtX -a block_size=Y
```

คุณควรได้รับข้อความว่า rmtX ถูก เปลี่ยนแล้ว

18. ตรวจสอบให้แน่ใจว่าเทปอยู่ในที่ตั้งที่ถูกต้องสำหรับการเรียกคืนไลบรารีโดยพิมพ์คำสั่งต่อไปนี้ (โดยที่ X คือจำนวนของเทปไดรฟ์ที่เหมาะสม):

```
tctl -f /dev/rmtX rewind
tctl -f /dev/rmtX.1 fsf 3
```

19. เรียกคืนไลบรารีที่หายไปโดยใช้คำสั่งอย่างใดอย่างหนึ่งต่อไปนี้ (โดยที่ X คือจำนวนของเทปไดรฟ์ที่เหมาะสม):

- เมื่อต้องการเรียกคืนไลบรารี libc.a อย่างเดียว ให้พิมพ์ คำสั่งต่อไปนี้:

```
restbyname -xvqf /dev/rmtX.1 ./usr/ccs/lib/libc.a
```

- เมื่อต้องการเรียกคืนไตรีกทอรี /usr/ccs/lib ให้พิมพ์คำสั่ง ต่อไปนี้:

```
restbyname -xvqf /dev/rmtX.1 ./usr/ccs/lib
```

- เมื่อต้องการเรียกคืนไตรีกทอรี /usr/ccs/bin ให้พิมพ์คำสั่ง ต่อไปนี้:

```
restbyname -xvqf /dev/rmtX.1 ./usr/ccs/bin
```

20. พลัซข้อมูลไปยังดิสก์โดยพิมพ์คำสั่ง ต่อไปนี้:

```
cd /mnt/usr/sbin
./sync;./sync;./sync
```

21. ถอนติดตั้งระบบไฟล์ /usr และ / (ราก) โดยพิมพ์คำสั่งต่อไปนี้:

```
cd/
umount /dev/hd2
umount /dev/hd4
```

หากคำสั่ง `umount` อย่างใดอย่างหนึ่ง ล้มเหลว ให้เปิดเครื่องใหม่และเริ่มต้นโพรซีเดอรันอีกครั้ง

22. รีบูตระบบโดยพิมพ์คำสั่งต่อไปนี้:

```
reboot
```

หลังจากรีบูตระบบแล้ว คำสั่งระบบปฏิบัติการควร พร้อมใช้งานได้

การสร้างอิมเมจเปิดเครื่องใหม่ที่เสียหายขึ้นใหม่:

โปรดทำตามขั้นตอนต่อไปนี้อธิบายวิธีการระบุอิมเมจเปิดเครื่องใหม่ที่เสียหาย และสร้างขึ้นใหม่

หากเครื่องของคุณกำลังรันอยู่ในปัจจุบันและคุณทราบว่าอิมเมจเปิดเครื่องใหม่เสียหายหรือถูกลบออก ให้สร้างอิมเมจเปิดเครื่องใหม่ขึ้นใหม่โดยรันคำสั่ง `bosboot` ด้วยสิทธิการใช้งานราก

ข้อควรสนใจ: ห้ามรีบูตระบบเมื่อคุณสงสัยว่า อิมเมจเปิดเครื่องใหม่เสียหาย

โปรดทำตามขั้นตอนต่อไปนี้ สมมติว่าระบบของคุณไม่ได้รีบูตอย่างถูกต้องเนื่องจากอิมเมจเปิดเครื่องใหม่ที่เสียหาย ถ้าเป็นไปได้ ควรป้องกันระบบของคุณไม่ให้สูญเสียข้อมูลหรือฟังก์ชันที่อาจเกิดขึ้นได้โดยการจัดตารางเวลาดาวน์ไทม์เป็นเวลาที่สุดผลกระทบต่อเวิร์กโหลดของคุณน้อยที่สุด

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

1. ใส่สื่อบันทึกผลิตภัณฑ์เข้าในไดรฟ์ที่เหมาะสม
2. เปิดเครื่องตามคำแนะนำที่จัดส่งมาพร้อมกับ ระบบของคุณ
3. จากเมนู เซอร์วิสการจัดการระบบ ให้เลือก **มัลติบูต**
4. จากหน้าจอถัดไป ให้เลือก **ติดตั้งจาก**
5. เลือกอุปกรณ์ที่มีสื่อบันทึกผลิตภัณฑ์ แล้วเลือก **ติดตั้ง**
6. เลือกไอคอนเวอร์ชัน AIX
7. ปฏิบัติตามคำแนะนำทางออนไลน์จนกว่าคุณสามารถเลือกโหมดซึ่งคุณจะใช้ สำหรับการติดตั้ง ณ จุดนั้น เลือก **เริ่มต้นโหมดการบำรุงรักษา สำหรับการกู้คืนระบบ**
8. เลือก **เข้าถึงกลุ่มมวลรวมราก**
9. ปฏิบัติตามคำแนะนำทางออนไลน์จนกว่าคุณสามารถเลือก **เข้าถึง กลุ่มมวลรวมนี้และเริ่มต้นเซลล์**
10. ใช้คำสั่ง `bosboot` เพื่อสร้างอิมเมจเปิดเครื่องใหม่ขึ้นใหม่ ตัวอย่าง:

```
bosboot -a -d /dev/hdisk0
```

หาก คำสั่งล้มเหลวและคุณได้รับข้อความต่อไปนี้:

```
0301-165 bosboot: WARNING! bosboot failed - do not attempt to boot device.
```

ลองแก้ไขปัญหาโดยใช้ข้อพจนานุกรมอย่างใดอย่างหนึ่งต่อไปนี้ จากนั้นรันคำสั่ง `bosboot` อีกครั้งจนกว่าคุณสามารถสร้างอิมเมจเปิดเครื่องใหม่ได้สำเร็จ:

- ลบดีฟอลต์บูตโลจิคัลมวลรวม (hd5) จากนั้นสร้าง hd5 ใหม่

หรือ

- รันการวิเคราะห์บนฮาร์ดดิสก์ ซ่อมแซมหรือเปลี่ยน ตามความจำเป็น

ถ้าคำสั่ง `bosboot` ยังล้มเหลวต่อไป โปรดติดต่อผู้แทนฝ่ายสนับสนุนลูกค้าของคุณ

ข้อควรสนใจ: หาก คำสั่ง `bosboot` ล้มเหลวในขณะที่สร้างอิมเมจเปิดเครื่องใหม่ อย่ารีบูตเครื่อง

11. เมื่อคำสั่ง `bosboot` เสร็จเรียบร้อยแล้ว ให้ใช้คำสั่ง `reboot` เพื่อรีบูตระบบของคุณ

หลักการที่เกี่ยวข้อง:

“สตาร์ทอัพระบบ” ในหน้า 4

เมื่อระบบปฏิบัติการพื้นฐานเริ่มต้น ระบบจะเริ่มต้นชุดของภารกิจที่ซับซ้อน ภายใต้สภาพปกติ ภารกิจเหล่านี้มีการทำให้โดยอัตโนมัติ

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `bosboot`

การจัดทำการสำรองแบบออนไลน์ของ JFS:

การจัดทำการสำรองแบบออนไลน์ของระบบไฟล์ที่เจอร์นัล (JFS) ที่ติดตั้ง หรือระบบไฟล์ที่เจอร์นัลที่เพิ่มประสิทธิภาพ (JFS2) สร้างรูปภาพสแตติกของ โลจิคัลวอลุ่มที่มีระบบไฟล์

เพื่อจัดทำการสำรองแบบออนไลน์ของ JFS ที่ติดตั้ง ต้องมีรีเรอร์ โลจิคัลวอลุ่มที่ระบบไฟล์ตั้งอยู่และโลจิคัลวอลุ่มที่บันทึกตั้งอยู่

หมายเหตุ: เนื่องจากการเขียนไฟล์เป็นแบบอะซิงโครนัส สำเนาที่แบ่งจึงอาจไม่มีข้อมูลทั้งหมดที่ถูกเขียนทันทีก่อนหน้านี้ การแบ่ง การแก้ไขใดๆ ที่เริ่มขึ้นหลังจากการแบ่งเริ่มต้นอาจไม่มีอยู่ในสำเนาสำรอง ดังนั้นขอแนะนำให้ดำเนินการกิจกรรมต่างๆ ในระบบไฟล์ให้น้อยที่สุดขณะที่การแบ่งเกิดขึ้น

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

เมื่อต้องการแบ่งสำเนาที่มีรีเรอร์ของระบบไฟล์ `/home/xyz` ไปยังจุดต่อใหม่ที่มีชื่อ `/jfsstaticcopy` ให้พิมพ์ ดังต่อไปนี้:

```
chfs -a splitcopy=/jfsstaticcopy /home/xyz
```

คุณสามารถควบคุมสำเนาที่มีรีเรอร์ซึ่งจะใช้เป็นสำเนาสำรองได้โดยใช้แอตทริบิวต์ สำเนา สำเนาที่มีรีเรอร์ลำดับที่สองคือค่าดีฟอลต์ถ้าผู้ใช้ไม่ได้ระบุสำเนา ตัวอย่างเช่น:

```
chfs -a splitcopy=/jfsstaticcopy -a copy=1 /home/xyz
```

ณ จุดนี้ สำเนาแบบอ่านอย่างเดียวของระบบไฟล์มีอยู่ใน `/jfsstaticcopy` การเปลี่ยนแปลงใดๆ ที่ทำในระบบไฟล์ต้นฉบับ หลังจากการแบ่งสำเนาจะไม่ สะท้อนให้เห็นในสำเนาสำรอง

เมื่อต้องการรวมรูปภาพการแบ่ง JFS อีกครั้ง เป็นสำเนาที่มีรีเรอร์ที่จุดต่อ `/testcopy` ให้ใช้ คำสั่งต่อไปนี้:

```
rmfs /testcopy
```

คำสั่ง `rmfs` ลบสำเนาระบบไฟล์ออกจากสถานะการแบ่งและอนุญาตให้รวมใหม่ เป็นสำเนาที่มีรีเรอร์ได้

การจัดทำและการทำสำเนาสำรอง snapshot ของ JFS2:

คุณสามารถจัดทำ snapshot ของ JFS2 ที่ติดตั้งซึ่งสร้างรูปภาพระดับ บล็อกที่สอดคล้องกันของระบบไฟล์ ณ จุดเวลาหนึ่ง

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

รูปภาพ snapshot ยังคงมั่นคงแม้ว่าระบบไฟล์ที่ใช้ ในการสร้าง snapshot ที่เรียกว่า *snappedFS* มีการเปลี่ยนแปลง อย่างต่อเนื่อง Snapshot คงเสถียรความปลอดภัยเดียวกันกับที่ *snappedFS* มี เมื่อ snapshot ถูกสร้างขึ้น

ในสถานการณ์จำลองต่อไปนี้จะสร้าง snapshot และทำสำเนาสำรอง snapshot ไว้บนสื่อบันทึกที่ถอดออกได้โดยไม่ต้องถอนติดตั้งหรือหยุดระบบไฟล์ ทั้งหมดนี้ด้วยคำสั่งเดียว: **backsnap** คุณยังสามารถใช้ snapshot สำหรับวัตถุประสงค์อื่นได้ด้วย เช่น การเข้าถึงไฟล์หรือไดเรกทอรีถ้ามีอยู่ เมื่อใช้ snapshot คุณสามารถดำเนินการขั้นตอนการสแน็ปช็อตแบบต่างๆได้โดยใช้ SMIT หรือคำสั่ง **backsnap** และ **snapshot**

เมื่อต้องการสร้าง snapshot ของระบบไฟล์ /home/abc/test และทำสำเนาสำรอง (ตามชื่อ) ไว้บนอุปกรณ์เทป /dev/rmt0 ให้ใช้คำสั่งต่อไปนี้:

```
backsnap -m /tmp/snapshot -s size=16M -i f/dev/rmt0 /home/abc/test
```

คำสั่งนี้สร้างโลจิคัลวอลุ่มขนาด 16 เมกะไบต์สำหรับ snapshot ของระบบไฟล์ JFS2 (/home/abc/test) Snapshot มีการติดตั้งบน /tmp/snapshot จากนั้นจึงจัดทำสำเนาสำรองของ snapshot ตามชื่อไว้บนอุปกรณ์เทป หลังจากสำเนาสำรองเสร็จสมบูรณ์แล้ว snapshot ยังคงติดตั้งอยู่ใช้แฟล็ก **-R** พร้อมกับคำสั่ง **backsnap** ถ้าต้องการลบ snapshot เมื่อสำเนาสำรองเสร็จสมบูรณ์แล้ว

หลักการที่เกี่ยวข้อง:

“ระบบไฟล์” ในหน้า 464

ระบบไฟล์คือโครงสร้างตามลำดับชั้น (แผนผังไฟล์) ของ ไฟล์และไดเรกทอรี

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง **backsnap**

คำสั่ง **chfs**

คำสั่ง **rmfs**

คำสั่ง **snapshot**

การจัดทำและการทำสำเนาสำรอง snapshot ภายนอกของ JFS2:

คุณสามารถจัดทำ snapshot ของ JFS2 ที่ติดตั้งซึ่งสร้างรูปภาพระดับ บล็อกที่สอดคล้องกันของระบบไฟล์ ณ จุดเวลาหนึ่ง

รูปภาพ snapshot ยังคงมั่นคงแม้ว่าระบบไฟล์ที่ใช้ในการสร้าง snapshot ที่เรียกว่า *snappedFS* มีการเปลี่ยนแปลงอย่างต่อเนื่อง Snapshot คงสิทธิความปลอดภัยเดียวกันกับที่ *snappedFS* มี เมื่อ snapshot ถูกสร้างขึ้น

ในสถานการณ์จำลองต่อไปนี้จะใช้คำสั่ง **backsnap** เพื่อสร้าง snapshot ภายนอก และสำรองข้อมูล snapshot ไปยังสื่อบันทึกที่ถอดออกได้โดยไม่ต้องถอนการติดตั้งอยู่หยุดระบบไฟล์ คุณยังสามารถใช้ snapshot สำหรับวัตถุประสงค์อื่นได้ด้วย เช่น การเข้าถึงไฟล์หรือไดเรกทอรีถ้ามีอยู่ เมื่อใช้ snapshot คุณยังสามารถทำหลากหลายโปรซีเจอร์ snapshot โดยใช้ SMIT หรือคำสั่ง **backsnap** และ **snapshot**

เมื่อต้องการสร้าง snapshot ภายนอกของระบบไฟล์ /home/abc/test และทำสำเนาสำรองตามชื่อไว้บนอุปกรณ์เทป /dev/rmt0 ให้รันคำสั่งต่อไปนี้:

```
backsnap -m /tmp/snapshot -s size=16M -if/dev/rmt0 /home/abc/test
```

คำสั่ง ก่อนหน้านี้สร้างโลจิคัลวอลุ่มขนาด 16 MB สำหรับ snapshot ของระบบไฟล์ JFS2 /home/abc/test Snapshot ติดตั้งบน ไดเรกทอรี /tmp/snapshot จากนั้น จัดทำสำเนาสำรองของ snapshot ตามชื่อไว้บนอุปกรณ์เทป หลังจาก สำเนาสำรองเสร็จสมบูรณ์แล้ว snapshot จะถูกถอนติดตั้งแต่ยังคงพร้อมใช้งานได้ ใช้ แฟล็ก **-R** พร้อมกับคำสั่ง **backsnap** ถ้าต้องการลบ snapshot เมื่อสำเนาสำรองเสร็จสมบูรณ์แล้ว

หลักการที่เกี่ยวข้อง:

“ระบบไฟล์” ในหน้า 464

ระบบไฟล์คือโครงสร้างตามลำดับชั้น (แผนผังไฟล์) ของไฟล์และไดเรกทอรี

การจัดทำและการทำสำเนาสำรอง snapshot ภายในของ JFS2:

คุณสามารถจัดทำ snapshot ของ JFS2 ที่ติดตั้งซึ่งสร้างรูปภาพระดับ บล็อกที่สอดคล้องกันของระบบไฟล์ ณ จุดเวลาหนึ่ง

รูปภาพ snapshot ยังคงมั่นคงแม้ว่าระบบไฟล์ที่ใช้ในการสร้าง snapshot ที่เรียกว่า *snappedFS* มีการเปลี่ยนแปลงอย่างต่อเนื่อง Snapshot คงสิทธิความปลอดภัยเดียวกันกับที่ *snappedFS* มี เมื่อ snapshot ถูกสร้างขึ้น

ในสถานการณ์จำลองต่อไปนี้ คุณใช้คำสั่ง **backsnap** เพื่อสร้าง snapshot ภายใน และสำรองข้อมูล snapshot ไปยังสื่อบันทึกที่ถอดออกได้โดยไม่ต้องถอนการติดตั้งอยู่หยุดระบบไฟล์ คุณยังสามารถใช้ snapshot สำหรับวัตถุประสงค์อื่นได้ด้วย เช่น การเข้าถึงไฟล์หรือไดเรกทอรีถ้ามีอยู่ เมื่อใช้ snapshot คุณยังสามารถทำหลากหลายโปรซีเจอร์ snapshot โดยใช้ SMIT หรือคำสั่ง **backsnap** และ **snapshot**

เมื่อต้องการสร้าง snapshot ภายในของระบบไฟล์ /home/abc/test และทำสำเนาสำรองตามชื่อไวนอุปกรณ์เทป /dev/rmt0 ให้รันคำสั่งต่อไปนี้:

```
backsnap -n mysnapshot -if/dev/rmt0 /home/abc/test
```

คำสั่ง ก่อนหน้านี้สร้าง snapshot ภายในที่ชื่อว่า mysnapshot ของระบบไฟล์ /home/abc/test Snapshot มีการเข้าถึงจาก ไดเรกทอรี /home/abc/test/.snapshot/mysnapshot จากนั้นจัดทำสำเนาสำรองไวนอุปกรณ์เทปใช้ แฟล็ก **-R** พร้อมกับคำสั่ง **backsnap** ถ้าคุณต้องการลบ snapshot เมื่อสำเนาสำรองเสร็จสมบูรณ์แล้ว

หลักการที่เกี่ยวข้อง:

“ระบบไฟล์” ในหน้า 464

ระบบไฟล์คือโครงสร้างตามลำดับชั้น (แผนผังไฟล์) ของไฟล์และไดเรกทอรี

การบีบอัดไฟล์ (คำสั่ง compress และ pack):

ใช้คำสั่ง **compress** และคำสั่ง **pack** เพื่อบีบอัดไฟล์สำหรับหน่วยเก็บ

ใช้คำสั่ง **uncompress** และคำสั่ง **unpack** เพื่อขยายไฟล์ที่เรียกคืน

กระบวนการบีบอัดและคลายบีบอัดไฟล์ใช้เวลาพอสมควร แต่หลังจากบีบอัดแล้ว ข้อมูลใช้พื้นที่ว่างบนสื่อสำเนาสำรองน้อยลง

เมื่อต้องการบีบอัดระบบไฟล์ให้ใช้เมธอดอย่างใดอย่างหนึ่งต่อไปนี้:

- ใช้แฟล็ก **-p** พร้อมกับคำสั่ง **backup**
- ใช้คำสั่ง **compress** หรือ **pack**

ข้อดีของการบีบอัดไฟล์ได้แก่:

- ประหยัดเงินและเวลาโดยการบีบอัดไฟล์ก่อนส่งไฟล์นั้นบนเครือข่าย
- ประหยัดหน่วยเก็บและใช้รีซอร์สระบบให้เกิดประโยชน์สูงสุด:
 - บีบอัดระบบไฟล์ก่อนทำสำเนาสำรองเพื่อสงวนพื้นที่ว่างเทป

- บีบอัดบล็อกไฟล์ที่สร้างขึ้นโดยเซลล์สคริปต์ซึ่งรันในตอนกลางคืน เป็นการง่ายที่จะให้สคริปต์บีบอัดไฟล์ก่อนออก
- บีบอัดไฟล์ที่ไม่ได้เข้าถึงอยู่ในปัจจุบัน ตัวอย่างเช่น ไฟล์ของผู้ใช้ซึ่งไม่ได้ใช้งานเป็นเวลานานสามารถถูกบีบอัดและวางไว้ในที่เก็บถาวร tar บนดิสก์หรือบนเทป และเรียกคืนในภายหลัง

หมายเหตุ:

- คำสั่ง **compress** อาจขาดพื้นที่ทำงานในระบบไฟล์ในขณะที่บีบอัด เนื่องจากคำสั่งสร้างไฟล์บีบอัด ก่อนจะลบไฟล์ที่ไม่ได้บีบอัดใดๆ ดังนั้นคำสั่งจึงต้องการพื้นที่ว่างมากกว่า ขนาดทั้งหมดของไฟล์ประมาณ 50%
- ไฟล์อาจล้มเหลวในการบีบอัดเนื่องจากไฟล์ถูกบีบอัดแล้ว หากคำสั่ง **compress** ไม่สามารถลดขนาดไฟล์ได้ คำสั่งจะล้มเหลว

โปรดดูที่คำสั่ง **compress** สำหรับรายละเอียดเกี่ยวกับ คำสั่งคืน แต่โดยทั่วไป ปัญหาที่พบเมื่อบีบอัดไฟล์ สามารถสรุปได้ดังนี้:

- คำสั่งอาจขาดพื้นที่ทำงานในระบบไฟล์ในขณะที่บีบอัด เนื่องจากคำสั่ง **compress** สร้างไฟล์บีบอัด ก่อนจะลบไฟล์ที่ไม่ได้บีบอัดใดๆ ดังนั้นคำสั่งจึงต้องการพื้นที่ว่างมากกว่า ขนาดของไฟล์ที่กำหนดประมาณ 50% ถึง 100%
- ไฟล์อาจล้มเหลวในการบีบอัดเนื่องจากไฟล์ถูกบีบอัดแล้ว หากคำสั่ง **compress** ไม่สามารถลดขนาดไฟล์ได้ คำสั่งจะล้มเหลว

*การบีบอัดไฟล์โดยใช้คำสั่ง **compress**:*

ใช้คำสั่ง **compress** เพื่อลดขนาด ของไฟล์โดยใช้การเข้ารหัส Lempel-Zev ที่ปรับได้

ไฟล์ต้นฉบับแต่ละไฟล์ที่ระบุโดยพารามิเตอร์ *File* จะถูกแทนที่ด้วยไฟล์บีบอัด โดยมี .Z ต่อท้าย ชื่อ ไฟล์บีบอัดจะยังคงมีความเป็นเจ้าของ โหมด และเวลาการเข้าถึงและแก้ไข เหมือนกับของไฟล์ต้นฉบับ หากไม่ระบุไฟล์ อินพุตมาตรฐานจะถูกบีบอัดไปยังเอาต์พุตมาตรฐาน หากการบีบอัดไม่ลดขนาดของไฟล์ จะมีข้อความถูกเขียนไปยัง ข้อผิดพลาดมาตรฐานและไฟล์ต้นฉบับไม่ถูกแทนที่

ใช้คำสั่ง **uncompress** เพื่อเรียกคืนไฟล์บีบอัดเป็นรูปแบบดั้งเดิม

ขนาดการบีบอัดนั้นขึ้นอยู่กับขนาดของอินพุต จำนวนบิตต่อรหัสที่ระบุโดยตัวแปร *Bits* และการแจกแจงของสตริงย่อย ที่มีร่วมกัน โดยปกติ ซอร์สโค้ดหรือข้อความภาษาอังกฤษจะถูกลดขนาดได้ 50 ถึง 60 เปอร์เซ็นต์ การบีบอัดของคำสั่ง **compress** โดยปกติมีความกระชับมากกว่า และใช้เวลาในการคำนวณน้อยกว่าการบีบอัด ที่กระทำโดยการเข้ารหัส Huffman แบบปรับได้ ดังที่ใช้ในคำสั่ง **pack**

ตัวอย่างเช่น เมื่อต้องการบีบอัดไฟล์ foo และเขียนเปอร์เซ็นต์การบีบอัดไปยังข้อผิดพลาดมาตรฐาน ให้พิมพ์ดังต่อไปนี้:

```
compress -v foo
```

ให้ดูที่คำสั่ง **compress** ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

*การบีบอัดไฟล์โดยใช้คำสั่ง **pack**:*

ใช้คำสั่ง **pack** เพื่อจัดเก็บไฟล์ ที่ระบุโดยพารามิเตอร์ *File* ในรูปแบบที่บีบอัดซึ่งใช้การเข้ารหัส Huffman

ไฟล์อินพุตถูกแทนที่โดยไฟล์ที่แพ็ก ซึ่งมีชื่อตามชื่อไฟล์ต้นฉบับ (*File.z*) โดยมีโหมดการเข้าถึง วันที่เข้าถึงและแก้ไข และเจ้าของเหมือนกับของไฟล์ต้นฉบับ ชื่อไฟล์อินพุตสามารถมีได้ไม่เกิน 253 ไบต์เพื่อให้มี พื้นที่ว่างสำหรับค่าเดิมหลัง .z ที่เพิ่ม หากคำสั่ง **pack** สำเร็จ ไฟล์ต้นฉบับจะถูกลบออก

ใช้คำสั่ง **unpack** เพื่อเรียกคืนไฟล์ที่แพ็คเป็นรูปแบบดั้งเดิม

หาก คำสั่ง **pack** ไม่สามารถสร้างไฟล์ที่เล็กลงได้ คำสั่งจะหยุด การประมวลผลและรายงานว่าไม่สามารถประหยัดพื้นที่ว่างได้ (โดยทั่วไป ความล้มเหลว ในการประหยัดพื้นที่ว่างเกิดขึ้นกับไฟล์ขนาดเล็กที่มีการกระจายอักขระรูปแบบเดียว) จำนวนของพื้นที่ว่างที่ประหยัดได้ขึ้นอยู่กับขนาดของไฟล์อินพุตและการกระจาย ความถี่อักขระ เนื่องจากแผนผังการถอดรหัสเป็นส่วนแรกของแต่ละไฟล์ .z คุณจึงไม่สามารถประหยัดพื้นที่ว่างในไฟล์ที่เล็กกว่า สามบล็อกรหัส โดยปกติ ไฟล์ข้อความมีขนาดลดลง 25 ถึง 40 เปอร์เซ็นต์

การออก ของคำสั่ง **pack** เป็นจำนวนไฟล์ที่ไม่สามารถแพ็คได้ การแพ็คไม่ทำภายใต้เงื่อนไขต่อไปนี้:

- ไฟล์ถูกแพ็คแล้ว
- ชื่อไฟล์อินพุตมีมากกว่า 253 ไบต์
- ไฟล์มีลิงก์
- ไฟล์เป็นไดเรกทอรี
- ไม่สามารถเปิดไฟล์ได้
- การแพ็คไม่ช่วยประหยัดบล็อกหน่วยเก็บ
- มีไฟล์ที่ชื่อ *File.z* อยู่แล้ว
- ไม่สามารถสร้างไฟล์ .z ได้
- เกิดข้อผิดพลาด I/O ขึ้นในระหว่างการประมวลผล

ตัวอย่างเช่น เมื่อต้องการบีบอัดไฟล์ chap1 และ chap2 ให้พิมพ์ดังต่อไปนี้:

```
pack chap1 chap2
```

คำสั่งนี้ บีบอัด chap1 และ chap2 และแทนที่ ด้วยไฟล์ที่ชื่อ chap1.z และ chap2.z คำสั่ง **pack** แสดงเปอร์เซ็นต์ขนาดที่ลดลงสำหรับแต่ละไฟล์

ให้ดูที่คำสั่ง **pack** ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

การขยายไฟล์ที่บีบอัด (คำสั่ง **uncompress** และ **unpack**):

ใช้คำสั่ง **uncompress** และ **unpack** เพื่อขยายไฟล์ที่บีบอัด

การขยายไฟล์โดยใช้คำสั่ง **uncompress**

ใช้คำสั่ง **uncompress** เพื่อเรียกคืนไฟล์ต้นฉบับที่ถูกบีบอัดโดยคำสั่ง **compress** แต่ละไฟล์บีบอัดที่ระบุโดยพารามิเตอร์ *File* ถูกเอาออก และแทนที่โดยสำเนาที่ขยาย ไฟล์ที่ขยายมีชื่อเหมือนกับ เวอร์ชันที่บีบอัดแต่ไม่มีนามสกุล .Z ไฟล์ที่ขยายยังคงมีความเป็นเจ้าของ โหมด และเวลาการเข้าถึงและแก้ไข เหมือนกับของไฟล์ต้นฉบับ หากไม่ระบุไฟล์อินพุตมาตรฐานจะถูกขยาย ไปยังเอาต์พุตมาตรฐาน

แม้ว่าคล้ายกับคำสั่ง **uncompress** แต่คำสั่ง **zcat** เขียนเอาต์พุตที่ขยายไปยังเอาต์พุตมาตรฐาน เสมอ

ตัวอย่างเช่น เมื่อต้องการคลายบีบอัดไฟล์ foo ให้พิมพ์ดังต่อไปนี้:

```
uncompress foo
```

ให้ดูที่คำสั่ง **uncompress** ใน *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

การขยายไฟล์โดยใช้คำสั่ง `unpack`

ใช้คำสั่ง `unpack` เพื่อขยาย ไฟล์ที่สร้างขึ้นโดยคำสั่ง `pack` สำหรับแต่ละไฟล์ที่ระบุ คำสั่ง `unpack` จะค้นหาไฟล์ที่ชื่อ `File.z` หากไฟล์นี้เป็นไฟล์ที่แพ็คเกจ คำสั่ง `unpack` จะแทนที่ไฟล์นี้ด้วย เวอร์ชันที่ขยาย คำสั่ง `unpack` เปลี่ยนชื่อ ไฟล์ใหม่โดยการลบคำเติมหลัง `.z` ออกจาก `File` ไฟล์ใหม่มีโหนดเข้าถึง วันที่เข้าถึงและแก้ไข และเจ้าของ เหมือนกับของไฟล์ต้นฉบับที่แพ็คเกจ

คำสั่ง `unpack` ดำเนินงานเฉพาะบนไฟล์ที่ลงท้ายด้วย `.z` เท่านั้น ผลคือเมื่อ คุณระบุชื่อไฟล์ที่ไม่ได้ลงท้ายด้วย `.z` คำสั่ง `unpack` จะเพิ่มคำเติมหลังและค้นหาไดเรกทอรีเพื่อหาชื่อไฟล์ที่มีคำเติมหลังนั้น

ค่าออกคือจำนวนของไฟล์ที่คำสั่ง `unpack` ไม่สามารถคลายแพ็คเกจได้ ไฟล์ไม่สามารถถูกคลายแพ็คเกจถ้ามีสถานการณ์ใดๆ ต่อไปนี้:

- ชื่อไฟล์ (เฉพาะของ `.z`) มีมากกว่า 253 ไบต์
- ไม่สามารถเปิดไฟล์ได้
- ไฟล์ไม่ใช่ไฟล์ที่แพ็คเกจ
- ไม่มีไฟล์ซึ่งมีชื่อไฟล์ที่คลายแพ็คเกจ
- ไม่สามารถสร้างไฟล์ที่คลายแพ็คเกจ

หมายเหตุ: คำสั่ง `unpack` เขียนคำเตือนไปยังข้อผิดพลาดมาตรฐานถ้าไฟล์ที่คำสั่งกำลังคลายแพ็คเกจมีลิงก์ ไฟล์ที่คลายแพ็คเกจใหม่มีหมายเลขไอ-โหนด (โหนดดัชนี) ที่แตกต่างจาก ไฟล์ที่แพ็คเกจซึ่งใช้ในการสร้าง อย่างไรก็ตาม ไฟล์อื่นใดที่ลิงก์ไปยัง หมายเลขไอ-โหนดดั้งเดิมของไฟล์ที่แพ็คเกจยังคงมีอยู่ และยังคงแพ็คเกจอยู่

ตัวอย่างเช่น เมื่อต้องการคลายแพ็คเกจไฟล์ที่แพ็คเกจ `chap1.z` และ `chap2.z` ให้พิมพ์ดังต่อไปนี้:

```
unpack chap1.z chap2
```

คำสั่งนี้ ขยายไฟล์ที่แพ็คเกจ `chap1.z` และ `chap2.z` และแทนที่ด้วยไฟล์ที่ชื่อ `chap1` และ `chap2`

หมายเหตุ: คุณสามารถระบุคำสั่ง `unpack` พร้อมด้วยชื่อไฟล์ที่มีหรือที่ไม่มีคำเติมหลัง `.z`

ให้ดูที่คำสั่ง `unpack` ใน *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

รูปภาพระบบและสำเนาสำรองกลุ่มวอลุ่มที่ผู้ใช้กำหนด

Rootvg มีการจัดเก็บไว้บนฮาร์ดดิสก์ หรือกลุ่มของดิสก์ และประกอบด้วย ไฟล์สตาร์ทอัพ, BOS, ข้อมูลคอนฟิกูเรชัน และผลิตภัณฑ์ซอฟต์แวร์ที่เป็นทางเลือกใดๆ โดยปกติ *กลุ่มวอลุ่มที่ผู้ใช้กำหนด* (เรียกอีกอย่างว่า *กลุ่มวอลุ่ม nonrootvg*) ประกอบด้วยไฟล์ข้อมูลและแอปพลิเคชันซอฟต์แวร์

คุณสามารถสำรองอิมเมจของระบบและกลุ่มวอลุ่มโดยใช้ SMIT หรือโปรซีเดอร์คำสั่ง อิมเมจสำเนาสำรองมีวัตถุประสงค์สองข้อ ข้อหนึ่งคือเพื่อเรียกคืนระบบที่เสียหายโดยใช้อิมเมจสำเนาสำรองระบบ อีกข้อหนึ่ง คือเพื่อโอนย้ายซอฟต์แวร์ที่ติดตั้งและกำหนดคอนฟิกจากระบบหนึ่งไปยังระบบอื่น

โปรซีเดอร์ SMIT ใช้คำสั่ง `mksysb` เพื่อสร้างอิมเมจสำเนาสำรองที่สามารถจัดเก็บไว้บนเทปหรือในไฟล์ก็ได้ หากคุณเลือกเทป โปรแกรมสำเนาสำรองจะเขียน *อิมเมจเปิดเครื่องใหม่* ลงในเทป ซึ่งทำให้ เทปเหมาะสมสำหรับการติดตั้ง

หมายเหตุ:

- เทปสตาร์ทอัพไม่สามารถจัดทำหรือใช้เพื่อเริ่มต้นคอมพิวเตอร์ส่วนบุคคล บน PowerPC

- หากคุณเลือกเมธอด SMIT สำหรับสำเนาสำรอง คุณต้องติดตั้งชุดไฟล์ sysbr ในซอฟต์แวร์แพ็คเกจ bos.sysmgt ก่อน
- หลักการที่เกี่ยวข้อง:

“สำเนาสำรอง” ในหน้า 27

โดยทั่วไป การสำรองข้อมูลของผู้ใช้และระบบจะถูกเก็บไว้ใน กรณีที่ข้อมูลถูกลบโดยไม่ตั้งใจหรือมีดิสก์ล้มเหลว การจัดการสำเนาสำรองสามารถทำได้ง่ายขึ้นเมื่อเก็บข้อมูลผู้ใช้แยกต่างหากจากข้อมูลระบบ

ข้อมูลที่เกี่ยวข้อง:

การติดตั้งผลิตภัณฑ์ซอฟต์แวร์ทางเลือกหรืออัปเดตเซอวิริส

การทำสำเนาสำรองรูปภาพระบบและกลุ่มวอลุ่มที่ผู้ใช้กำหนด:

คุณสามารถทำสำเนาสำรองของรูปภาพระบบและกลุ่มวอลุ่มที่ผู้ใช้กำหนด

ก่อนการทำสำเนาสำรองกลุ่มวอลุ่ม rootvg:

- ต้องติดตั้งฮาร์ดแวร์ทั้งหมดไว้แล้ว รวมถึงอุปกรณ์ภายนอก เช่น เทปและซีดีรอมไดรฟ์
- โพรซีเดอร์สำเนาสำรองนี้ต้องการชุดไฟล์ sysbr ซึ่งอยู่ในซอฟต์แวร์แพ็คเกจ BOS System Management Tools and Applications พิมพ์คำสั่งต่อไปนี้เพื่อกำหนดว่ามีการติดตั้งชุดไฟล์ sysbr บนระบบของคุณหรือไม่:

```
lsllp -l bos.sysmgt.sysbr
```

หาก ระบบของคุณมีการติดตั้งชุดไฟล์ sysbr ไว้แล้ว ให้ทำโพรซีเดอร์สำเนาสำรองต่อไป

ถ้าคำสั่ง lsllp ไม่ได้แสดงชุดไฟล์ sysbr ให้ติดตั้งชุดไฟล์ก่อนทำโพรซีเดอร์สำเนาสำรองต่อไป

```
installp -agqXd device bos.sysmgt.sysbr
```

โดยที่ device คือที่ตั้งของซอฟต์แวร์ ตัวอย่างเช่น /dev/rmt0 สำหรับ เทปไดรฟ์

ก่อนการทำสำเนาสำรองกลุ่มวอลุ่มที่ผู้ใช้กำหนด:

- ก่อนการบันทึก ต้อง varied on กลุ่มวอลุ่มและต้องติดตั้ง ระบบไฟล์
- ข้อควรสนใจ: การดำเนินการคำสั่ง savevg ส่งผลให้วัสดุทั้งหมดที่จัดเก็บไว้ก่อนหน้านี้บนสื่อเอาต์พุตที่เลือก สูญหายไป
- ตรวจสอบให้แน่ใจว่าอุปกรณ์สำเนาสำรองถูกทำความสะอาดเมื่อเร็วๆ นี้เพื่อ ป้องกันข้อผิดพลาด

โพรซีเดอร์ต่อไปนี้อธิบายวิธีการจัดทำรูปภาพที่ติดตั้งได้ ของระบบของคุณ

ภารกิจการทำสำเนาสำรอง ระบบของคุณ

ภารกิจ	พาด่วน SMIT	คำสั่งหรือไฟล์
การทำสำเนาสำรองกลุ่มวอลุ่ม rootvg	<ol style="list-style-type: none"> 1. ล็อกอินเป็นราก 2. ติดตั้งระบบไฟล์สำหรับสำเนาสำรอง <code>smitsmit mountfs</code> 3. ถอนติดตั้งโลคัลไดเรกทอรีใดๆ ที่ติดตั้งผ่านโลคัลไดเรกทอรีอื่น <code>smitsmit umountfs</code> 4. จัดหาพื้นที่ว่างดิสก์อย่างน้อย 8.8MB ในไดเรกทอรี <code>/tmp²</code> 5. ทำสำเนาสำรอง: <code>smitsmit mksysb</code> 6. ป้องกันการเขียนสื่อบันทึกสำเนาสำรอง 7. บันทึกผู้ใช้รากและรหัสผ่านผู้ใช้ที่สำรองไว้ 	<ol style="list-style-type: none"> 1. ล็อกอินเป็นราก 2. ติดตั้งระบบไฟล์สำหรับสำเนาสำรอง¹ ให้ดูที่คำสั่ง <code>mount</code> 3. ถอนติดตั้งโลคัลไดเรกทอรีใดๆ ที่ติดตั้งผ่านโลคัลไดเรกทอรีอื่น ให้ดูที่คำสั่ง <code>umount</code> 4. จัดหาพื้นที่ว่างดิสก์อย่างน้อย 8.8MB ในไดเรกทอรี <code>/tmp²</code> 5. ทำสำเนาสำรอง ให้ดูที่คำสั่ง <code>mksysb</code> 6. ป้องกันการเขียนสื่อบันทึกสำเนาสำรอง 7. บันทึกผู้ใช้รากและรหัสผ่านผู้ใช้ที่สำรองไว้
ตรวจสอบเทปสำเนาสำรอง ³	<code>smitsmit lsmksysb</code>	
การทำสำเนาสำรองกลุ่มวอลุ่มที่ผู้ใช้กำหนด ⁴	<code>smitsmit savevg</code>	<ol style="list-style-type: none"> 1. แก้ไขขนาดระบบไฟล์ก่อนทำสำเนาสำรอง ถ้าจำเป็น⁵ <code>mkvgdata VGName</code> จากนั้นแก้ไข <code>/tmp/vgdata/VGName/VGName.data</code> 2. บันทึกกลุ่มวอลุ่ม โปรดดูที่คำสั่ง <code>savevg</code>

หมายเหตุ:

1. คำสั่ง `mksysb` ไม่สำรองระบบไฟล์ที่ mount ข้ามเน็ตเวิร์ก NFS
2. คำสั่ง `mksysb` ต้องการพื้นที่ทำงานนี้ในระหว่างการสำรองข้อมูล ใช้คำสั่ง `df` ซึ่งจะรายงานในหน่วยของบล็อกต่างๆ ที่มีขนาด 512 ไบต์ต่อบล็อก เพื่อกำหนดพื้นที่ว่างในไดเรกทอรี `/tmp` ใช้คำสั่ง `chfs` เพื่อเปลี่ยนขนาดของระบบไฟล์ ถ้าจำเป็น
3. โพรซีเดิร์นนี้แสดงรายการเนื้อหาของเทปสำเนาสำรอง `mksysb` รายการ เนื้อหาจะตรวจสอบข้อมูลส่วนใหญ่บนเทปแต่ไม่ได้ตรวจสอบว่า สามารถบูตเทปสำหรับการติดตั้ง วิธีเดียวในการตรวจสอบว่า อิมเมจเปิดเครื่องใหม่บนเทป `mksysb` ทำงานอย่างถูกต้องคือการบูตจาก เทป
4. ถ้าคุณต้องการแยกไฟล์ในกลุ่มวอลุ่มที่ผู้ใช้กำหนดออกจากรูปภาพสำเนาสำรอง ให้สร้างไฟล์ที่ชื่อ `/etc/exclude.volume_group_name` โดยที่ `volume_group_name` คือชื่อของกลุ่มวอลุ่ม ซึ่งคุณต้องการทำสำเนาสำรอง จากนั้นแก้ไข `/etc/exclude.volume_group_name` และ ป้อนรูปแบบของชื่อไฟล์ซึ่งคุณไม่ต้องการรวมไว้ในรูปภาพสำเนาสำรองของคุณ รูปแบบในไฟล์นี้เป็นอินพุตไปยังแบบแผนการจับคู่รูปแบบ ของคำสั่ง `grep` เพื่อกำหนดว่าไฟล์ใด จะถูกแยกออกจากสำเนาสำรอง
5. หากคุณเลือกที่จะแก้ไขไฟล์ `VGName.data` เพื่อ เปลี่ยนขนาดของระบบไฟล์ คุณต้องไม่ระบุแฟล็ก `-i` หรือแฟล็ก `-m` ด้วยคำสั่ง `savevg` เนื่องจาก ไฟล์ `VGName.data` ถูกเขียนทับ

ข้อมูลที่เกี่ยวข้อง:

การติดตั้งผลิตภัณฑ์ซอฟต์แวร์ทางเลือกหรืออัปเดตเซอร์วิส

การติดตั้งการสำรองข้อมูลระบบ

คอนฟิกูเรชันก่อนสำเนาสำรอง:

กำหนดคอนฟิกูเรชันก่อนการสร้างรูปภาพสำเนาสำรองของระบบนั้น อย่างไรก็ตาม ถ้าคุณวางแผนจะใช้รูปภาพสำเนาสำรองสำหรับการติดตั้งระบบ เป้าหมายอื่นที่กำหนดคอนฟิกูเรชันแตกต่างกัน ให้สร้างรูปภาพ ก่อน กำหนดคอนฟิกูเรชันก่อน

ระบบ *ต้นทาง* คือระบบที่คุณสร้างสำเนาสำรอง ระบบ *เป้าหมาย* คือระบบที่กำลังติดตั้งสำเนาสำรอง

โปรแกรมการติดตั้งจะติดตั้งเฉพาะการสนับสนุนอุปกรณ์ที่จำเป็นสำหรับการตั้งค่าฮาร์ดแวร์ของเครื่องที่ติดตั้งเท่านั้นโดยอัตโนมัติ ด้วยเหตุนี้ ถ้าคุณกำลังใช้สำเนาสำรองระบบเพื่อติดตั้งเครื่องอื่น คุณอาจต้อง ติดตั้งอุปกรณ์เพิ่มเติมบนระบบต้นทางก่อนทำรูปภาพสำเนาสำรอง และใช้รูปภาพนั้นเพื่อติดตั้งระบบเป้าหมายตั้งแต่หนึ่งระบบขึ้นไป

ใช้พาทควอน SMIT, `smit devinst` เพื่อติดตั้งการสนับสนุนอุปกรณ์เพิ่มเติมบนระบบต้นทาง

- หากมีพื้นที่ว่างดิสก์เพียงพอบนระบบต้นทางและระบบเป้าหมาย ให้ติดตั้ง การสนับสนุนอุปกรณ์ทั้งหมด
- หากมีพื้นที่ว่างดิสก์ที่จำกัดบนระบบต้นทางและระบบเป้าหมาย ให้เลือกติดตั้ง การสนับสนุนอุปกรณ์บางรายการ

การสำรองข้อมูลจะโอนย้ายคอนฟิกูเรชันต่อไปนี้จากระบบต้นทาง ไปยังระบบเป้าหมาย:

- ข้อมูลพื้นที่การเพจ
- ข้อมูลโลจิกัลวอลุ่ม
- ข้อมูล `rootvg`
- การวางตำแหน่งของพาร์ติชันโลจิกัล (ถ้าคุณได้เลือกอ็อปชันแม่พิมพ์)

ข้อมูลที่เกี่ยวข้อง:

การติดตั้งซอฟต์แวร์ทางเลือกและอัปเดตเซอวิสเซอ

การปรับแต่งการติดตั้งของคุณเอง

การติดตั้งและการถอนติดตั้งระบบไฟล์:

ก่อนทำสำเนาสำรอง คุณต้องติดตั้งระบบไฟล์ทั้งหมดซึ่งคุณต้องการ ทำสำเนาสำรองและถอนติดตั้งระบบไฟล์ทั้งหมดซึ่งคุณไม่ต้องการทำสำเนาสำรอง

โปรซีเดอร์ เมธอดสำเนาสำรอง ทำสำเนาสำรองเฉพาะระบบไฟล์ที่ติดตั้งไว้ใน `rootvg` เท่านั้น ด้วยเหตุนี้ คุณต้องติดตั้ง ระบบไฟล์ทั้งหมดซึ่งคุณต้องการทำสำเนาสำรองก่อนเริ่มต้น ในลักษณะคล้ายกัน คุณต้อง ถอนติดตั้งระบบไฟล์ซึ่งคุณ ไม่ต้องการทำสำเนาสำรอง

โปรซีเดอร์สำเนาสำรองนี้จะสำรองข้อมูลไฟล์สองครั้ง ถ้าโลคัลไดเรกทอรีถูกติดตั้งข้างบนโลคัลไดเรกทอรีอื่นในระบบไฟล์เดียวกัน ตัวอย่างเช่น ถ้าคุณ ติดตั้ง `/tmp` ข้างบน `/usr/tmp` ไฟล์ ในไดเรกทอรี `/tmp` จะถูกทำสำเนาสำรองสองครั้ง การทำซ้ำนี้อาจมีจำนวนที่เกินกว่าจำนวนไฟล์ที่ระบบสามารถจัดการได้ ซึ่งอาจเป็นสาเหตุที่ทำให้เกิดความล้มเหลวในการติดตั้งอิมเมจของการสำรองข้อมูลในอนาคตได้

ข้อควรพิจารณาด้านความปลอดภัยสำหรับสำเนาสำรอง:

ถ้าคุณติดตั้งรูปภาพสำเนาสำรองบนระบบอื่น คุณอาจไม่ต้องการรหัสผ่านและ แอดเดรสเครือข่ายที่คัดลอกไปยังระบบเป้าหมายเนื่องจากเหตุผล ด้านความปลอดภัย

นอกจากนี้ การคัดลอกแอดเดรสเครือข่ายไปยังระบบเป้าหมายยังสร้างแอดเดรสซ้ำ ที่อาจทำให้การสื่อสารเครือข่ายหยุดชะงักได้

การเรียกคืนรูปภาพสำเนาสำรอง:

ขณะที่ติดตั้งรูปภาพสำเนาสำรอง ระบบจะตรวจเช็คระบบเป้าหมายว่ามีพื้นที่ดิสก์เพียงพอที่จะสร้างโลจิคัลวอลุ่มทั้งหมดที่เก็บไว้บนการสำรองข้อมูลหรือไม่ ถ้าระบบเป้าหมายมีพื้นที่เพียงพอ การสำรองข้อมูลทั้งหมดจะถูกกู้คืน มิฉะนั้น การติดตั้งจะหยุดทำงานและระบบพร้อมให้คุณเลือกฮาร์ดดิสก์ปลายทางเพิ่มเติม

ระบบไฟล์ถูกสร้างขึ้นบนระบบเป้าหมายด้วยขนาดเดียวกันกับที่อยู่บนระบบต้นทาง ยกเว้นว่าตัวแปร **SHRINK** มีการตั้งค่าเป็น ใช่ ในไฟล์ `image.data` ก่อนทำรูปภาพสำเนาสำรอง ข้อยกเว้นคือไดเรกทอรี `/tmp` ซึ่งสามารถเพิ่มขึ้นเพื่อจัดสรรพื้นที่ให้พอเพียงสำหรับคำสั่ง `bosboot` สำหรับข้อมูลเกี่ยวกับการตั้งค่าตัวแปรให้ดูที่ไฟล์ `image.data`

เมื่อระบบติดตั้งรูปภาพสำเนาสำรองเสร็จแล้ว โปรแกรมการติดตั้ง จะกำหนดคอนฟิก ODM อีกครั้งบนระบบเป้าหมาย หากระบบเป้าหมายไม่มีการตั้งค่าฮาร์ดแวร์เหมือนกันทุกประการกับระบบต้นทาง โปรแกรม อาจแก้ไขแอ็ทริบิวต์อุปกรณ์ในไฟล์ระบบเป้าหมายต่อไปนี้:

- ไฟล์ทั้งหมดใน `/etc/objrepos` ที่ขึ้นต้นด้วย `Cu`
- ไฟล์ทั้งหมดในไดเรกทอรี `/dev`

ข้อมูลที่เกี่ยวข้อง:

การติดตั้งการสำรองข้อมูลระบบ

การนำสำเนาสำรองที่จัดตารางเวลาไปใช้:

โพรซีเดอร์นี้อธิบายวิธีการจัดทำและใช้สคริปต์เพื่อทำ สำเนาสำรองแบบเต็มรายสัปดาห์และสำเนาสำรองส่วนเพิ่มรายวันของไฟล์ผู้ใช้

- จำนวนข้อมูลที่จัดตารางเวลาสำหรับสำเนาสำรองต้องไม่เกินหนึ่งเทปเมื่อใช้ สคริปต์นี้
- ตรวจสอบว่ามีกิลดเทปเข้าในอุปกรณ์สำเนาสำรองก่อนคำสั่ง `cron` รันสคริปต์
- ตรวจสอบให้แน่ใจว่าอุปกรณ์เชื่อมต่อและมีอยู่ โดยเฉพาะเมื่อใช้ สคริปต์ที่รันตอนกลางคืน ใช้คำสั่ง `lsdev -C | pg` เพื่อตรวจสอบการมีอยู่
- ตรวจสอบให้แน่ใจว่าอุปกรณ์สำเนาสำรองมีการทำความสะอาดเมื่อเร็วๆ นี้เพื่อป้องกันข้อผิดพลาด
- หากคุณกำลังทำสำเนาสำรองระบบไฟล์ที่อาจใช้งานอยู่ให้ลองติดตั้ง ระบบไฟล์นั้นก่อนเพื่อป้องกันความเสียหายของระบบไฟล์
- ตรวจสอบระบบไฟล์ก่อนทำสำเนาสำรอง ใช้โพรซีเดอร์ “การตรวจสอบระบบไฟล์” ในหน้า 481 หรือรันคำสั่ง `fsck`

สคริปต์ที่รวมอยู่ในโพรซีเดอร์นี้มีไว้ใช้เป็นโมเดลเท่านั้น และต้องได้รับการปรับแต่งอย่างรอบคอบเพื่อให้ตรงกับความต้องการของไซต์เฉพาะ

หลักการที่เกี่ยวข้อง:

“กลยุทธ์สำเนาสำรอง” ในหน้า 27

มีสองเมธอดสำหรับการทำสำเนาสำรองข้อมูลจำนวนมาก

การทำสำเนาสำรองระบบไฟล์โดยใช้คำสั่ง cron:

โพธิ์เดอร์นี่อธิบายวิธีการเขียนสคริปต์ crontab ซึ่งคุณสามารถส่งผ่านไปยังคำสั่ง cron เพื่อดำเนินการได้

สคริปต์ทำสำเนาสำรองระบบไฟล์ผู้ใช้สองระบบคือ /home/plan และ /home/run ในคืนวันจันทร์ถึงวันเสาร์ ระบบไฟล์ทั้งสองระบบมีการทำสำเนาสำรองไว้บนเทปเดียวกัน และในทุกเช้าจะมีการใส่เทปม้วนใหม่สำหรับสำเนาสำรองในคืนถัดไป สำเนาสำรองของคืนวันจันทร์ เป็นที่เก็บถาวรแบบสมบูรณ์ (ระดับ 0) สำเนาสำรองของวันอังคารถึง วันเสาร์เป็นสำเนาสำรองเพิ่มเติม

1. ขั้นตอนแรกในการจัดทำสคริปต์ crontab คือ ออกใช้คำสั่ง crontab -e คำสั่งนี้เปิดไฟล์ว่างซึ่งคุณสามารถจัดทำรายการที่จะถูกส่ง ไปยังสคริปต์ cron สำหรับการดำเนินการแต่ละคืน (โปรแกรมแก้ไข ดีฟอลต์คือ vi) โดยพิมพ์:

```
crontab -e
```

2. ตัวอย่างต่อไปนี้แสดงฟิลด์ crontab หกฟิลด์ ฟิลด์ 1 ใช้สำหรับนาฬิกา, ฟิลด์ 2 ใช้สำหรับชั่วโมงบนนาฬิกา 24-ชั่วโมง, ฟิลด์ 3 ใช้สำหรับวันของเดือน, และฟิลด์ 4 ใช้สำหรับเดือนของปี ฟิลด์ 3 และ 4 มี * (เครื่องหมายดอกจัน) เพื่อแสดงว่าสคริปต์รันทุกเดือนใน วันที่ระบุไว้ในฟิลด์ วัน/สัปดาห์ ฟิลด์ 5 ใช้สำหรับวันของ สัปดาห์และยังสามารถระบุเป็นช่วงของวัน ตัวอย่างเช่น 1-6 ได้ด้วย ฟิลด์ 6 ใช้สำหรับคำสั่งเชลล์ที่กำลังรัน

```
min hr day/mo mo/yr day/wk shell command
```

```
0 2 * * 1 backup -0 -uf /dev/rmt0.1 /home/plan
```

บรรทัดคำสั่งที่แสดงสมมติว่าคุณลากรที่ไซต์มีอยู่ เพื่อตอบกลับพร้อมต์เมื่อเหมาะสม แฟล็ก -0 (ศูนย์) สำหรับคำสั่ง backup หมายถึงระดับศูนย์ หรือสำเนาสำรองแบบสมบูรณ์ แฟล็ก -u อัปเดตเร็กคอร์ดสำเนาสำรองในไฟล์ /etc/dumpdates และแฟล็ก ระบุชื่ออุปกรณ์ อุปกรณ์เทปแม่เหล็กดิบ 0.1 ดังเช่นในตัวอย่างข้างบน

3. พิมพ์บรรทัดที่คล้ายกับบรรทัดในขั้นตอนที่ 2 สำหรับแต่ละระบบไฟล์ที่ทำสำเนา สำรองในวันเฉพาะ ตัวอย่างต่อไปนี้ แสดงสคริปต์แบบเต็มที่ทำสำเนาสำรอง หกวันบนสองระบบไฟล์:

```
0 2 * * 1 backup -0 -uf/dev/rmt0.1 /home/plan
0 3 * * 1 backup -0 -uf/dev/rmt0.1 /home/run
0 2 * * 2 backup -1 -uf/dev/rmt0.1 /home/plan
0 3 * * 2 backup -1 -uf/dev/rmt0.1 /home/run
0 2 * * 3 backup -2 -uf/dev/rmt0.1 /home/plan
0 3 * * 3 backup -2 -uf/dev/rmt0.1 /home/run
0 2 * * 4 backup -3 -uf/dev/rmt0.1 /home/plan
0 3 * * 4 backup -3 -uf/dev/rmt0.1 /home/run
0 2 * * 5 backup -4 -uf/dev/rmt0.1 /home/plan
0 3 * * 5 backup -4 -uf/dev/rmt0.1 /home/run
0 2 * * 6 backup -5 -uf/dev/rmt0.1 /home/plan
0 3 * * 6 backup -5 -uf/dev/rmt0.1 /home/run
```

4. บันทึกไฟล์ซึ่งคุณสร้างขึ้นและออกจากโปรแกรมแก้ไข ระบบปฏิบัติการ ส่งผ่านไฟล์ crontab ไปยังสคริปต์ cron

ข้อมูลที่เกี่ยวข้อง:

ไฟล์พิเศษ rmt

สำเนาสำรองของไฟล์บนระบบไฟล์ DMAPI-managed JFS2:

มีอ็อปชันในคำสั่ง tar และ backbyinode ที่อนุญาตให้คุณสำรองข้อมูลแอดทริบิวต์ส่วนขยาย (EA)

ด้วยคำสั่ง `backbyinode` บนระบบไฟล์ DMAPI สามารถทำสำเนาสำรองได้เฉพาะข้อมูลที่มีอยู่ในระบบไฟล์ในเวลาที่ใช้คำสั่งเท่านั้น คำสั่ง `backbyinode` ตรวจสอบสถานะปัจจุบันของข้อมูลเมตาเพื่อทำงาน การทำเช่นนี้อาจเป็นประโยชน์กับ DMAPI เนื่องจาก จะทำสำเนาสำรองสถานะของระบบไฟล์ที่จัดการ อย่างไรก็ตาม จะไม่ทำสำเนาสำรอง ข้อมูลออฟไลน์ใดๆ

เมื่อต้องการทำสำเนาสำรองข้อมูลทั้งหมดในระบบไฟล์ DMAPI ให้ใช้คำสั่งที่อ่าน ทั้งไฟล์ เช่น คำสั่ง `tar` คำสั่งนี้อาจส่งผลให้แอปพลิเคชันที่เปิดใช้งาน DMAPI เรียกคืนข้อมูลของทุกไฟล์ที่เข้าถึงโดยคำสั่ง `tar` ซึ่งย้ายข้อมูลไปมาระหว่างหน่วยเก็บที่สองและที่สาม ดังนั้นจึงอาจส่งผลกระทบต่อประสิทธิภาพการทำงานได้

การจัดรูปแบบดิสเก็ต (คำสั่ง `format` หรือ `fdformat`):

คุณสามารถจัดรูปแบบดิสเก็ตในดิสเก็ตไดรฟ์ที่ระบุโดยพารามิเตอร์ *Device* (อุปกรณ์ `/dev/rfd0` โดยค่าดีฟอลต์) ได้โดยใช้คำสั่ง `format` และ `fdformat`

ข้อควรสนใจ: การจัดรูปแบบดิสเก็ตทำลายข้อมูลที่มีอยู่บนดิสเก็ตนั้น

คำสั่ง `format` พิจารณานิตอุปกรณ์ ซึ่งเป็นอย่างไรต่อไปนี้:

- ดิสเก็ตความจุต่ำขนาด 5.25 นิ้ว (360 KB) ที่มี 40x2 แทร็ก แต่ละแทร็กมี 9 เซ็กเตอร์
- ดิสเก็ตความจุสูงขนาด 5.25 นิ้ว (1.2 MB) ที่มี 80x2 แทร็ก แต่ละแทร็กมี 15 เซ็กเตอร์
- ดิสเก็ตความจุต่ำขนาด 3.5 นิ้ว (720 KB) ที่มี 80x2 แทร็ก แต่ละแทร็กมี 9 เซ็กเตอร์
- ดิสเก็ตความจุสูงขนาด 3.5 นิ้ว (2.88 MB) ที่มี 80x2 แทร็ก แต่ละแทร็กมี 36 เซ็กเตอร์

ขนาดเซ็กเตอร์คือ 512 ไบต์สำหรับดิสเก็ตทุกชนิด

ใช้คำสั่ง `format` เพื่อจัดรูปแบบดิสเก็ตความหนาแน่นสูง ยกเว้นว่าพารามิเตอร์ *Device* ระบุความหนาแน่นอื่น

ใช้คำสั่ง `fdformat` เพื่อจัดรูปแบบดิสเก็ตความหนาแน่นต่ำ ยกเว้นว่ามีการระบุแฟล็ก `-h` พารามิเตอร์ *Device* ระบุอุปกรณ์ที่มีดิสเก็ตซึ่งจะ ถูกจัดรูปแบบ (เช่น อุปกรณ์ `/dev/rfd0` สำหรับไดรฟ์ 0)

ก่อนการจัดรูปแบบดิสเก็ต คำสั่ง `format` และ `fdformat` จะพร้อมท์สำหรับการตรวจสอบ ซึ่งอนุญาตให้คุณสิ้นสุดการดำเนินการได้อย่างเรียบร้อยถ้าจำเป็น

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการจัดรูปแบบดิสเก็ตในอุปกรณ์ `/dev/rfd0` ให้พิมพ์ ดังต่อไปนี้:

```
format -d /dev/rfd0
```
- เมื่อต้องการจัดรูปแบบดิสเก็ตโดยไม่ตรวจสอบแทร็กที่ไม่ดี ให้พิมพ์ดังต่อไปนี้:

```
format -f
```
- เมื่อต้องการจัดรูปแบบดิสเก็ต 360 KB ในดิสเก็ตไดรฟ์ขนาด 5.25 นิ้ว 1.2 MB ในอุปกรณ์ `/dev/rfd1` ให้พิมพ์ดังต่อไปนี้:

```
format -l -d /dev/rfd1
```
- เมื่อต้องการบังคับใช้การจัดรูปแบบความหนาแน่นสูงของดิสเก็ตเมื่อใช้คำสั่ง `fdformat` ให้พิมพ์ดังต่อไปนี้:

```
fdformat -h
```

ให้ดูที่คำสั่ง `format` ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

การเปลี่ยนบูรณภาพของระบบไฟล์ (คำสั่ง fsck):

ใช้คำสั่ง fsck เพื่อตรวจสอบและซ่อมแซมระบบไฟล์ที่ไม่สอดคล้องกันแบบโต้ตอบ

สิ่งสำคัญคือการรันคำสั่งนี้บนทุกระบบไฟล์ เป็นส่วนหนึ่งของการเริ่มต้นระบบ คุณต้องสามารถอ่านไฟล์อุปกรณ์ที่มีระบบไฟล์อยู่ (ตัวอย่างเช่น อุปกรณ์ /dev/hd0) โดยปกติ ระบบไฟล์จะสอดคล้องกัน และคำสั่ง fsck เพียงรายงานเกี่ยวกับจำนวนไฟล์บล็อกที่ใช้และบล็อกที่ว่างในระบบไฟล์ หากระบบไฟล์ไม่สอดคล้อง คำสั่ง fsck จะแสดงข้อมูลเกี่ยวกับความไม่สอดคล้องกันที่พบและพร้อมให้คุณเพื่อขอสิทธิในการซ่อมแซม คำสั่ง fsck ะมัดระวังเมื่อพยายามจะทำการซ่อมแซม และพยายามหลีกเลี่ยงการดำเนินการที่อาจส่งผลให้เกิดการสูญเสียข้อมูลที่ถูกต้อง อย่างไรก็ตาม ก็มีบางกรณี ที่คำสั่ง fsck จะแนะนำ ให้ทำลายไฟล์ที่เสียหายนั้น

ข้อควรสนใจ: รันคำสั่ง fsck บนระบบไฟล์เสมอ หลังจากระบบทำงานไม่ปกติ การดำเนินการแก้ไขอาจส่งผลให้บางข้อมูลสูญหายไป การดำเนินการดีฟอลต์สำหรับ การแก้ไขความไม่สอดคล้องกันแต่ละครั้งคือการขอให้ผู้ปฏิบัติงานพิมพ์ yes หรือ no หากคุณไม่มีสิทธิ การเขียน ในไฟล์ที่ได้รับผลกระทบ คำสั่ง fsck จะใช้ค่าดีฟอลต์เป็นการตอบกลับ no

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการตรวจสอบระบบไฟล์ดีฟอลต์ทั้งหมด ให้พิมพ์ดังต่อไปนี้:

```
fsck
```

คำสั่ง fsck รูปแบบนี้จะขอสิทธิจากคุณ ก่อนทำการเปลี่ยนแปลงใดๆ ในระบบไฟล์

- เมื่อต้องการแก้ไขปัญหาเล็กน้อยโดยอัตโนมัติด้วยระบบไฟล์ดีฟอลต์ ให้พิมพ์ ดังต่อไปนี้:

```
fsck -p
```

- เมื่อต้องการตรวจสอบระบบไฟล์ /dev/hd1 ให้พิมพ์ดังต่อไปนี้:

```
fsck /dev/hd1
```

คำสั่งนี้ตรวจสอบ ระบบไฟล์ที่ไม่ได้ติดตั้งซึ่งอยู่บนอุปกรณ์ /dev/hd1

หมายเหตุ: คำสั่ง fsck ไม่ได้ทำการแก้ไขใน ระบบไฟล์ที่ติดตั้ง

ให้ดูที่คำสั่ง fsck ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

การคัดลอกไปยังหรือจากดิสเก็ต (คำสั่ง flcopy):

ใช้คำสั่ง flcopy เพื่อคัดลอกดิสเก็ต (ที่เปิด เป็น /dev/rfd0) ไปยังไฟล์ที่ชื่อ floppy ซึ่งสร้างขึ้น ในไต่เรียกทอรีปัจจุบัน

ข้อความ Change floppy, hit return when done แสดงขึ้นเมื่อจำเป็น จากนั้นคำสั่ง flcopy คัดลอกไฟล์ floppy ไปยังดิสเก็ต

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการคัดลอก /dev/rfd1 ไปยังไฟล์ floppy ในไต่เรียกทอรีปัจจุบัน ให้พิมพ์ดังต่อไปนี้:

```
flcopy -f /dev/rfd1 -r
```

- เมื่อต้องการคัดลอก 100 แทร็กแรกของดิสเก็ต ให้พิมพ์ดังต่อไปนี้:

```
flcopy -f /dev/rfd1 -t 100
```

ให้ดูที่คำสั่ง `fcopy` ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

การคัดลอกไฟล์ไปยังเทปหรือดิสก์ (คำสั่ง `cpio -o`):

ใช้คำสั่ง `cpio -o` เพื่ออ่านชื่อพาธไฟล์จากอินพุตมาตรฐาน และคัดลอกไฟล์เหล่านี้ไปยังเอาต์พุตมาตรฐาน พร้อมกับ ชื่อพาธ และข้อมูลสถานะ

ชื่อพาธต้องไม่เกิน 128 อักขระ หลีกเลี่ยง การกำหนดชื่อพาธคำสั่ง `cpio` ที่ประกอบด้วยไฟล์ลิงก์เฉพาะจำนวนมาก เนื่องจาก อาจไม่มีหน่วยความจำเพียงพอจะเก็บข้อมูลชื่อพาธ และอาจทำให้สูญเสียข้อมูลการลิงก์

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการคัดลอกไฟล์ในไดเรกทอรีปัจจุบันที่มีชื่อสิ้นสุดด้วย `.c` บน ดิสเก็ต ให้พิมพ์ดังต่อไปนี้:

```
ls *.c | cpio -ov >/dev/rfd0
```

แฟล็ก `-v` แสดงชื่อของแต่ละไฟล์

- เมื่อต้องการคัดลอกไดเรกทอรีปัจจุบันและไดเรกทอรีย่อยทั้งหมดบนดิสเก็ต ให้พิมพ์ดังต่อไปนี้:

```
find . -print | cpio -ov >/dev/rfd0
```

ค่านี้จะบันทึกแผนผังไดเรกทอรีที่เริ่มต้นด้วยไดเรกทอรีปัจจุบัน (`.`) และรวมไดเรกทอรีย่อย และไฟล์ทั้งหมด

- เมื่อต้องการใช้สตริ่งคำสั่งที่สั้นลง ให้พิมพ์ดังต่อไปนี้:

```
find . -cpio /dev/rfd0 -print
```

รายการ `-print` จะแสดงชื่อของแต่ละไฟล์ที่ถูกคัดลอก

ให้ดูที่คำสั่ง `cpio` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

การคัดลอกไฟล์จากเทปหรือดิสก์ (คำสั่ง `cpio -i`):

ใช้คำสั่ง `cpio -i` เพื่ออ่านจากไฟล์เก็บถาวรอินพุต มาตรฐานที่สร้างขึ้นโดยคำสั่ง `cpio -o` และคัดลอก จากไฟล์ที่มีชื่อตรงกับ พารามิเตอร์ *Pattern*

ไฟล์เหล่านี้ถูกคัดลอกไปยังแผนผังไดเรกทอรีปัจจุบัน คุณสามารถแสดงรายการมากกว่าหนึ่งพารามิเตอร์ *Pattern* โดยใช้สัญลักษณ์ชื่อไฟล์ ที่อธิบายไว้ในคำสั่ง `ksh` ค่าดีฟอลต์สำหรับพารามิเตอร์ *Pattern* คือเครื่องหมายดอกจัน (*) ซึ่งจะเลือกไฟล์ทั้งหมดในไดเรกทอรีปัจจุบัน ในนิพจน์เช่น `[a-z]` เครื่องหมายยัติภังค์ (-) หมายถึง จนถึง ตามลำดับการจัดเรียงปัจจุบัน

หมายเหตุ: รูปแบบ `"*.c"` และ `"*.o"` ต้องอยู่ในเครื่องหมาย อัฒประกาศ เพื่อป้องกันไม่ให้เชลล์จัดการกับเครื่องหมายดอกจัน (*) เป็นอักขระการจับคู่รูปแบบ นี่เป็นกรณีพิเศษที่ตัวคำสั่ง `cpio` เองจะถอดรหัสอักขระการจับคู่รูปแบบ

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการแสดงรายการไฟล์ที่อุทกบันทึกไว้บนดิสเก็ตด้วยคำสั่ง `cpio` ให้พิมพ์ดังต่อไปนี้:

```
cpio -itv </dev/rfd0
```

ค่านี้แสดงตารางเนื้อหาของข้อมูลที่บันทึก ก่อนหน้านั้นบนไฟล์ `/dev/rfd0` ในรูปแบบคำสั่ง `cpio` รายการจะคล้ายกับการแสดงรายการไดเรกทอรี แบบยาวที่สร้างโดยคำสั่ง `ls -l`

- ในการแสดงเฉพาะชื่อพาธไฟล์ให้ใช้เฉพาะแฟล็ก `-it`
- เมื่อต้องการคัดลอกไฟล์ที่บันทึกไว้ก่อนหน้านี้ด้วยคำสั่ง `cpio` จากดิสเก็ต ให้พิมพ์ดังต่อไปนี้:

```
cpio -idmv </dev/rfd0
```

คำสั่งนี้คัดลอกไฟล์ที่บันทึกบนไฟล์ `/dev/rfd0` ก่อนหน้านี้โดยคำสั่ง `cpio` กลับมาไว้ที่ระบบไฟล์ (ระบบแฟล็ก `-i`) แฟล็ก `-d` อนุญาตให้คำสั่ง `cpio` สร้างไดเรกทอรีที่เหมาะสมถ้ามีการบันทึกแผนผังไดเรกทอรี แฟล็ก `-m` รักษาเวลาการแก้ไขล่าสุดเพื่อใช้เมื่อไฟล์ถูกบันทึก แฟล็ก `-v` ทำให้คำสั่ง `cpio` แสดงชื่อของแต่ละไฟล์ ขณะที่ถูกคัดลอก

- เมื่อต้องการคัดลอกไฟล์ที่เลือกจากดิสเก็ต ให้พิมพ์ดังต่อไปนี้:

```
cpio -i "*.c" "*.o" </dev/rfd0
```

คำสั่งนี้คัดลอกไฟล์ที่ลงท้ายด้วย `.c` หรือ `.o` จากดิสเก็ต

ให้ดูที่คำสั่ง `cpio` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

การคัดลอกไปยังหรือจากเทป (คำสั่ง `tcopy`):

ใช้คำสั่ง `tcopy` เพื่อคัดลอกเทปแม่เหล็ก

ตัวอย่างเช่น เมื่อต้องการคัดลอกจากเทปการสตรีมหนึ่ง ไปยังเทป 9-แทร็ก ให้พิมพ์ดังต่อไปนี้:

```
tcopy /dev/rmt0 /dev/rmt8
```

ให้ดูที่คำสั่ง `tcopy` ใน *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

การตรวจสอบคุณภาพของเทป (คำสั่ง `tapechk`):

ใช้คำสั่ง `tapechk` เพื่อทำการตรวจสอบความสอดคล้องกัน ซึ่งค้างอยู่บนอุปกรณ์เทปการสตรีมที่ต่อพ่วง

ความบกพร่องของฮาร์ดแวร์บางอย่างของเทปไดร์ฟการสตรีม สามารถตรวจพบได้โดยการอ่านเทปเท่านั้น คำสั่ง `tapechk` นำเสนอวิธีทำการอ่านเทปที่ระดับไฟล์

ตัวอย่าง เช่น เมื่อต้องการตรวจสอบสามไฟล์แรกบนอุปกรณ์เทปการสตรีม ให้พิมพ์ดังต่อไปนี้:

```
tapechk 3
```

ให้ดูที่คำสั่ง `tapechk` ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

การเก็บถาวรไฟล์ (คำสั่ง `tar`):

เมธอดสำเนาสำรองถาวรใช้สำหรับสำเนาของไฟล์ตั้งแต่หนึ่งไฟล์ขึ้นไป หรือทั้งฐานข้อมูลที่บันทึกสำหรับการอ้างอิงในอนาคต ใช้เป็นข้อมูลประวัติ หรือสำหรับการกู้คืนถ้าข้อมูลต้นฉบับเสียหายหรือหายไป

โดยปกติ หน่วยเก็บถาวรใช้เมื่อข้อมูลเฉพาะนั้นถูกลบออกจากระบบ

ใช้คำสั่ง `tar` เพื่อเขียนไฟล์หรือดึงข้อมูลไฟล์จากหน่วยเก็บถาวร คำสั่ง `tar` ค้นหาหน่วยเก็บถาวรบนอุปกรณ์ดีฟอลต์ (โดยปกติคือเทป) ยกเว้นว่าคุณระบุอุปกรณ์อื่น

เมื่อเขียนลงในหน่วยเก็บถาวร คำสั่ง `tar` จะใช้ไฟล์ชั่วคราว (ไฟล์ `/tmp/tar*`) และรักษาตารางของไฟล์ที่มีหลายลิงก์ไว้ในหน่วยความจำ คุณจะได้รับความแสดงข้อผิดพลาด ถ้าคำสั่ง `tar` ไม่สามารถสร้างไฟล์ชั่วคราว หรือ ถ้ามีหน่วยความจำไม่เพียงพอที่จะจัดเก็บตารางลิงก์

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการเขียนไฟล์ `file1` และ `file2` ลงใน หน่วยเก็บถาวรใหม่บนดีฟอลต์เทปไดรฟ์ ให้พิมพ์ดังต่อไปนี้:

```
tar -c file1 file2
```
- เมื่อต้องการแยกไฟล์ทั้งหมดในไดเรกทอรี `/tmp` ออกมาจาก ไฟล์หน่วยเก็บถาวรบนอุปกรณ์เทป `/dev/rmt2` และใช้เวลาของการแยกเป็นเวลากการแก้ไข ให้พิมพ์ดังต่อไปนี้:

```
tar -xm -f/dev/rmt2 /tmp
```
- เมื่อต้องการแสดงชื่อของไฟล์ในไฟล์หน่วยเก็บถาวร `out.tar` จากไดเรกทอรีปัจจุบัน ให้พิมพ์ดังต่อไปนี้:

```
tar -vtf out.tar
```

ให้ดูที่คำสั่ง `tar` ใน *Commands Reference, Volume 5* สำหรับ ข้อมูลเพิ่มเติมและไวยากรณ์ทั้งหมด

สำเนาสำรองไฟล์

ใช้คำสั่ง `backup` หรือ `smit` อย่างไม่อย่างหนึ่ง เพื่อสร้างสำเนาของไฟล์บนสื่อบันทึกสำเนาสำรอง เช่น เทปแม่เหล็กหรือ ดิสเก็ต

ข้อควรสนใจ: หากคุณพยายามทำสำเนาสำรองระบบไฟล์ที่ติดตั้ง ข้อความ จะแสดงขึ้น คำสั่ง `backup` ยังทำต่อไป แต่อาจเกิด ความไม่สอดคล้องกันขึ้นในระบบไฟล์ สถานการณ์นี้ไม่ใช่กับระบบไฟล์ราก (`/`)

สำเนาที่คุณสร้างด้วยคำสั่ง `backup` หรือคำสั่ง `smit` อยู่ในรูปแบบสำเนาสำรองอย่างไม่อย่างหนึ่งต่อไปนี้:

- ไฟล์เฉพาะที่ทำสำเนาสำรองตามชื่อโดยใช้แฟล็ก `-i`
- ทั้งระบบไฟล์ที่ทำสำเนาสำรองตามหมายเลขไอ-โหนดโดยใช้พารามิเตอร์ `-Level` และ `FileSystem`

หมายเหตุ:

- ข้อมูลอาจเสียหายได้เสมอถ้าไฟล์ถูกแก้ไข ในระหว่างทำสำเนาสำรองระบบ ด้วยเหตุนี้ ต้องแน่ใจว่าระบบมีกิจกรรมน้อยที่สุดในระหว่างโปรแกรมสำเนาสำรองข้อมูล
- หากทำสำเนาสำรองไว้บนเทป 8-มม. ที่มีการตั้งค่าขนาดบล็อกอุปกรณ์เป็น 0 (ศูนย์) จะไม่สามารถเรียกคืนข้อมูลจากเทปโดยตรงได้ หากคุณทำสำเนาสำรอง ด้วยค่าติดตั้ง 0 คุณสามารถเรียกคืนข้อมูลจากสำเนาสำรองนั้นได้โดยใช้โปรแกรมพิเศษที่อธิบายภายใต้คำสั่ง `restore`

ข้อควรสนใจ: ต้องแน่ใจว่าแฟล็กที่คุณระบุตรงกับสื่อบันทึกสำเนาสำรอง

การทำสำเนาสำรองไฟล์โดยใช้คำสั่ง `backup`:

ใช้คำสั่ง `backup` เพื่อสร้างสำเนาของไฟล์ บนสื่อบันทึกสำเนาสำรอง

ตัวอย่างเช่น เมื่อต้องการทำสำเนาสำรองไฟล์ที่เลือกไว้ในไดเรกทอรี `$HOME` ตาม ชื่อ ให้พิมพ์ดังต่อไปนี้:

```
find $HOME -print | backup -i -v
```

แฟล็ก `-i` พร้อมต์ระบบให้อ่านชื่อไฟล์ที่จะทำสำเนาสำรองจากอินพุตมาตรฐาน คำสั่ง `find` จะสร้างรายการของไฟล์ในไดเรกทอรีของผู้ใช้ รายการนี้ถูกไฟฟ้กับคำสั่ง `backup` เป็นอินพุตมาตรฐาน แฟล็ก `-v` แสดงรายงานความคืบหน้าเมื่อ คัดลอกแต่ละไฟล์ ไฟล์มีการทำสำเนาสำรองบนอุปกรณ์สำเนาสำรองดีฟอลต์สำหรับระบบ โลคัล

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการทำสำเนาสำรองระบบไฟล้ราก ให้พิมพ์ดังต่อไปนี้:

```
backup -0 -u /
```

ระดับ 0 และ / บอกให้ระบบทำสำเนาสำรองระบบไฟล้ / (ราก) ระบบไฟล้ถูกทำสำเนาสำรองไว้บนไฟล้ /dev/rfd0 แฟล็ก `-u` บอกให้ระบบอัปเดตเรกคอร์ดระดับสำเนาสำรองปัจจุบันในไฟล้ /etc/dumpdates

- เมื่อต้องการทำสำเนาสำรองไฟล้ทั้งหมดในระบบไฟล้ / (ราก) ที่ถูกแก้ไขตั้งแต่ทำสำเนาสำรองระดับ 0 ครั้งล่าสุด ให้พิมพ์ดังต่อไปนี้:

```
backup -1 -u /
```

ให้ดูที่คำสั่ง `backup` ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

การทำสำเนาสำรองไฟล้โดยใช้คำสั่ง `smit`:

ใช้คำสั่ง `smit` เพื่อรันคำสั่ง `backup` ซึ่งจะสร้างสำเนาของไฟล้ไว้บนสื่อบันทึกสำเนาสำรอง

- ที่พร้อมต์ให้พิมพ์ดังต่อไปนี้:

```
smit backup
```

- พิมพ์ชื่อพารของไดเรกทอรีซึ่งโดยปกติแล้ว มีการติดตั้งระบบไฟล้ ลงในฟิลด์ **DIRECTORY full pathname:**

```
/home/bill
```

- ในฟิลด์อุปกรณ์ **BACKUP** หรือฟิลด์ **FILE** ให้ป้อนชื่ออุปกรณ์เอาต์พุต ดังเช่นในตัวอย่างต่อไปนี้สำหรับอุปกรณ์เทปแม่เหล็ก ดิบ:

```
/dev/rmt0
```

- ใช้ปุ่ม Tab เพื่อสลับฟิลด์ รายงานแต่ละระยะของสำเนาสำรอง ที่เป็นทางเลือกถ้าคุณต้องการพิมพ์ข้อความแสดงข้อผิดพลาดไปยังหน้าจอ

- ในสภาวะแวดล้อมการจัดการระบบ ให้ใช้ค่าดีฟอลต์ของฟิลด์ จำนวน บล็อกสูงสุดที่จะเขียนบนสื่อสำเนาสำรอง เนื่องจากฟิลด์นี้ไม่ได้ใช้กับสำเนาสำรองเทป

- กด Enter เพื่อทำสำเนาสำรองไดเรกทอรีที่ระบุหรือระบบไฟล้

- รันคำสั่ง `restore -t` หากคำสั่งนี้สร้างข้อความแสดงข้อผิดพลาด คุณต้องทำซ้ำ ทั้งสำเนาสำรอง

การปิดระบบ

คำสั่ง `shutdown` เป็นวิธีที่เร็วและสะดวกที่สุด ในการหยุดระบบปฏิบัติการ

คุณอาจต้องการปิดระบบของคุณ:

- หลังจากติดตั้งซอฟต์แวร์ใหม่หรือเปลี่ยนคอนฟิกูเรชันสำหรับซอฟต์แวร์ ที่มีอยู่
- เมื่อมีปัญหาฮาร์ดแวร์
- เมื่อระบบ hung โดยหลีกเลี่ยงไม่ได้

- เมื่อประสิทธิภาพของระบบต้อยลง
- เมื่อระบบไฟล์อาจเสียหายได้

เมื่อคุณกำหนดแฟล็กที่เหมาะสม คำสั่งนี้แจ้งให้ผู้ใช้ทราบว่า ระบบกำลังจะดาวน์โหลดโปรแกรมทั้งหมดที่มีอยู่ ถอนติดตั้งระบบไฟล์ และหยุดระบบ โปรดดูที่ **shutdown** สำหรับข้อมูลเพิ่มเติม

ตรวจทานข้อมูลรายละเอียดต่อไปนี้จะเกี่ยวกับสถานการณ์การปิดเฉพาะ:

การปิดระบบโดยไม่ต้องรีบูต

การปิดระบบโดยไม่ต้องรีบูตมีอยู่สองวิธี

คุณสามารถใช้สองเมธอดเพื่อปิดระบบโดยไม่ต้องรีบูต: พาด่วน **SMIT** หรือคำสั่ง **shutdown**

ข้อกำหนดเบื้องต้น

คุณ ต้องมีสิทธิ์ผู้ใช้รากเพื่อปิดระบบ

เมื่อต้องการปิด ระบบโดยใช้ **SMIT**:

1. ล็อกอินเป็นราก
2. ที่พร้อมต์คำสั่ง ให้พิมพ์:

```
smit shutdown
```

เมื่อต้องการปิดระบบโดยใช้คำสั่ง **shutdown**:

1. ล็อกอินเป็นราก
2. ที่พร้อมต์คำสั่ง ให้พิมพ์:

```
shutdown
```

การปิดระบบในโหมดผู้ใช้เดียว

ในบางกรณี คุณอาจต้องการปิด ระบบและป้อนโหมดผู้ใช้เดียวเพื่อทำการบำรุงรักษาและการวินิจฉัย ซอฟต์แวร์

1. พิมพ์ `cd /` เพื่อเปลี่ยนเป็นไดเรกทอรีราก คุณต้องอยู่ในไดเรกทอรีรากเพื่อปิดระบบในโหมดผู้ใช้เดียว เพื่อให้มั่นใจว่าระบบไฟล์ถูกถอนติดตั้งโดยสมบูรณ์
2. พิมพ์ `shutdown -m` ระบบ ปิดในโหมดผู้ใช้เดียว

พร้อมต์ระบบแสดงขึ้นและคุณสามารถทำกิจกรรมการบำรุงรักษาได้

การปิดระบบในกรณีฉุกเฉิน

ใช้คำสั่ง **shutdown** เพื่อหยุดระบบอย่างรวดเร็ว โดยไม่แจ้งให้ผู้อื่นรับทราบ

คุณสามารถใช้คำสั่ง **shutdown** เพื่อปิดระบบ ภายใต้สถานการณ์ฉุกเฉิน

พิมพ์ `shutdown -F` แฟล็ก `-F` สั่งให้คำสั่ง **shutdown** บายพาสการส่งข้อความไปยังผู้อื่น และปิดระบบโดยเร็วที่สุด เท่าที่เป็นไปได้

สภาวะแวดล้อมของระบบ

โดยหลักแล้ว สภาวะแวดล้อมของระบบคือชุดของตัวแปรที่กำหนด หรือควบคุมลักษณะบางอย่างของการดำเนินการโปรเซส

สภาวะแวดล้อมของระบบมีการตั้งค่าหรือรีเซ็ตในทุกครั้งที่เซลล์เริ่มต้นขึ้น จากมุมมองด้านการจัดการระบบ สิ่งสำคัญคือการทำให้อันใจว่ามีการตั้งค่าผู้ใช้ด้วยค่าที่ถูกต้อง เมื่อล็อกอิน ตัวแปรเหล่านี้ส่วนใหญ่มีการตั้งค่าในระหว่างการเริ่มต้นระบบ มีการอ่านนิยามของตัวแปรจากไฟล์ `/etc/profile` หรือตั้งค่าโดยดีฟอลต์

โปรไฟล์

เซลล์ใช้ไฟล์โปรไฟล์สองชนิดเมื่อคุณล็อกอินเข้าสู่ระบบปฏิบัติการ

เซลล์ประเมินคำสั่งที่มีอยู่ในไฟล์จากนั้นรัน คำสั่งเพื่อตั้งค่าสภาวะแวดล้อมของระบบ ไฟล์มีฟังก์ชันคล้ายกัน ยกเว้นว่าไฟล์ `/etc/profile` ควบคุมตัวแปรโปรไฟล์ สำหรับผู้ใช้ทั้งหมดบนระบบ ในขณะที่ไฟล์ `.profile` อนุญาตให้คุณกำหนดสภาวะแวดล้อมของคุณเอง

มีการแสดงข้อมูลโปรไฟล์และสภาวะแวดล้อมของระบบ ต่อไปนี้:

- ไฟล์ `/etc/profile`
- ไฟล์ `.profile`
- เชื้ออัมพ์ตัวแปรสภาวะแวดล้อมของระบบ
- การเปลี่ยนข้อความของวัน
- “เซอวีส์การจัดการข้อมูลเวลา” ในหน้า 58.

ไฟล์ `/etc/profile`

ไฟล์แรกซึ่งระบบปฏิบัติการใช้ในเวลาล็อกอินคือไฟล์ `/etc/profile` ไฟล์นี้ควบคุม ตัวแปรดีฟอลต์ทั่วทั้งระบบ เช่น:

- ตัวแปรเอ็กซ์พอร์ต
- มาสก์การสร้างไฟล์ (`umask`)
- ชนิดเทอร์มินัล
- ข้อความเมลเพื่อบ่งชี้เวลาที่ได้รับเมลใหม่

ผู้ดูแลระบบกำหนดคอนฟิกไฟล์ `profile` สำหรับผู้ใช้ทั้งหมดบนระบบ เฉพาะผู้ดูแลระบบเท่านั้นสามารถเปลี่ยนไฟล์นี้ได้

ไฟล์ `.profile`

ไฟล์ลำดับที่สองซึ่งระบบปฏิบัติการใช้ในเวลาล็อกอินคือไฟล์ `.profile` ไฟล์ `.profile` มีอยู่ในไดเรกทอรีโฮม (`$HOME`) ของคุณและอนุญาตให้คุณกำหนดสภาวะแวดล้อมการทำงานเองได้ ไฟล์ `.profile` ยังยกเลิกคำสั่งและตัวแปรที่ตั้งค่าไว้ในไฟล์ `/etc/profile` ด้วย เนื่องจากไฟล์ `.profile` ถูกซ่อนไว้ให้ใช้คำสั่ง `ls -a` เพื่อแสดงไฟล์ใช้ไฟล์ `.profile` เพื่อ ควบคุมค่าดีฟอลต์ต่อไปนี้:

- เซลล์ที่จะเปิด
- ลักษณะปรากฏของพร้อมต์

- ตัวแปรสถานะแวดล้อม (ตัวอย่างเช่น ตัวแปรพาธการค้นหา)
- เสี่ยงภัยบอร์ด

ตัวอย่างต่อไปนี้แสดงไฟล์ .profile ปกติ:

```
PATH=/usr/bin:/etc:/home/bin1:/usr/lpp/tps4.0/user:/home/gsc/bin::
epath=/home/gsc/e3:
export PATH epath
csh
```

ตัวอย่างนี้กำหนดสองพาธ (PATH และ epath) เอ็กซ์พอร์ตพาธนั้น และเปิด C เซลล์ (csh)

คุณยังสามารถใช้ไฟล์ .profile (หรือถ้าไม่มีอยู่ไฟล์ .profile) เพื่อกำหนดตัวแปรเซลล์การลือกอื่นได้ด้วย คุณยังสามารถกำหนดสถานะแวดล้อมเซลล์อื่นเองได้ด้วยตัวอย่างเช่น ใช้ไฟล์ .chsrc และไฟล์ .kshrc เพื่อกำหนด C เซลล์และ Korn เซลล์เอง ตามลำดับ เมื่อเซลล์แต่ละชนิด เริ่มต้นขึ้น

เซอร์วิสการจัดการข้อมูลเวลา

ฟังก์ชันเวลาเข้าถึงและจัดรูปแบบวันที่และเวลาของระบบปัจจุบัน อีกครั้ง

คุณไม่จำเป็นต้องระบุแฟล็กพิเศษใดๆ ให้กับคอมไพเลอร์เพื่อใช้ฟังก์ชัน เวลา อย่างไรก็ตาม คุณต้องรวมไฟล์ส่วนหัวสำหรับฟังก์ชันเหล่านี้ไว้ในโปรแกรม เมื่อต้องการรวมไฟล์ส่วนหัว ให้ใช้คำสั่งต่อไปนี้:

```
#include <time.h>
```

เซอร์วิสเวลามีดังต่อไปนี้:

ไอเท็ม

adjtime

ctime, localtime, gmtime, mktime, difftime, asctime, tzset

getinterval, incinterval, absinterval, resinc, resabs, alarm, ualarm,

getitimer, setitimer

gettimer, settimer, restimer, stime, time

gettimerid

gettimeofday, settimeofday, ftime

nsleep, usleep, sleep

realtimerid

คำอธิบาย

แก้ไขเวลาเพื่อให้สามารถซิงโครไนซ์นาฬิการะบบได้

แปลงวันที่และเวลาเป็นการแสดงแทนด้วยสตริง

จัดการเวลาหมดอายุของตัวจับเวลา

เรียกใช้หรือตั้งค่าปัจจุบันสำหรับตัวจับเวลาที่ระบบที่ระบุ

จัดสรรตัวจับเวลาสำหรับแต่ละโปรเซส

เรียกใช้และตั้งค่าวันที่และเวลา

พักโปรเซสปัจจุบันจากการรัน

รีเซ็ตตัวจับเวลาที่จัดสรรก่อนหน้านี้

ชุดไฟล์และฮาร์ดแวร์ที่ต้องการสำหรับโหมด 64-บิต

เคอร์เนลรันในโหมด 64-บิต เพื่อให้สามารถเข้าถึงข้อมูลจำนวนมากได้อย่างรวดเร็วและจัดการข้อมูลชนิด 64-บิตได้อย่างมีประสิทธิภาพ

ชุดไฟล์รันไทม์ 64-บิตของระบบปฏิบัติการพื้นฐานคือ bos.64bit การติดตั้ง bos.64bit ยังติดตั้งไฟล์ /etc/methods/cfg64 ด้วย ไฟล์ /etc/methods/cfg64 เป็นคำสั่งที่เปิดใช้งานสภาพแวดล้อมรันไทม์ 64 บิต คำสั่งนี้ถูกเรียกใช้โดยสคริปต์ rc.boot ในระหว่างเฟส 3 ของกระบวนการบูท

เริ่มต้นด้วย AIX 6.1 เคอร์เนล 32-บิตได้ถูกคัดค้าน การติดตั้งระบบปฏิบัติการพื้นฐาน AIX 6.1 จะเปิดใช้งานโหมด 64-บิต

หมายเหตุ: ฮาร์ดแวร์ต้องมีความสามารถ 64-บิต เพื่อรัน AIX 6.1 รุ่น RS/6000® ต่อไปนี้ใช้ตัวประมวลผล 604e และไม่มี ความสามารถ 64-บิต:

- 7025 F50 Series

- 7026 H50 Series
- 9076 H50 Series
- 7043 150 Series
- 7046 B50 Series

เมื่อต้องการตรวจสอบความสามารถของตัวประมวลผลของคุณ ให้รันคำสั่งต่อไปนี้:

```
/usr/sbin/prtconf -c
```

คำสั่ง **prtconf** ส่งคืน 32 หรือ 64 อย่างใดอย่างหนึ่ง ขึ้นอยู่กับ ความสามารถของตัวประมวลผล ถ้าระบบของคุณไม่มีคำสั่ง **prtconf** คุณสามารถใช้คำสั่ง **bootinfo** พร้อมด้วยแฟล็ก **-y**

ฮาร์ดแวร์ที่ต้องการสำหรับโหมด 64-บิต

คุณต้องมีฮาร์ดแวร์ 64-บิตเพื่อรันแอปพลิเคชัน 64-บิต

เมื่อต้องการกำหนดว่าระบบของคุณมีสถาปัตยกรรมฮาร์ดแวร์ 32-บิตหรือ 64-บิต:

1. ล็อกอินเป็นผู้ใช้ราก
2. ที่บรรทัดคำสั่ง ให้ป้อน `bootinfo -y`

คำสั่งนี้จัดทำเอาต์พุตเป็น 32 หรือ 64 ขึ้นอยู่กับว่า สถาปัตยกรรมฮาร์ดแวร์เป็น 32-บิตหรือ 64-บิต นอกจากนี้ ถ้า คุณป้อน `lsattr -El proc0` ที่เวอร์ชันใดๆ ของ AIX ชนิดของตัวประมวลผลสำหรับเซิร์ฟเวอร์ ของคุณจะแสดงขึ้น

การเปรียบเทียบประสิทธิภาพระหว่าง 32 บิตและ 64 บิต

ในกรณีส่วนใหญ่ การรันแอปพลิเคชัน 32-บิตบนฮาร์ดแวร์ 64-บิตไม่เป็นปัญหา เนื่องจากฮาร์ดแวร์ 64-บิตสามารถรันได้ ทั้งซอฟต์แวร์ 64-บิตและ 32-บิต อย่างไรก็ตาม ฮาร์ดแวร์ 32-บิตไม่สามารถรันซอฟต์แวร์ 64-บิต

เมื่อต้องการทราบว่าปัญหาประสิทธิภาพในแอปพลิเคชันที่กำลังรันบน ระบบหรือไม่ให้อ้างอิงคู่มือผู้ใช้ของแอปพลิเคชัน เพื่อดูสถานะแวดล้อมการรัน ที่แนะนำ

Dynamic Processor Deallocation

AIX สามารถตรวจพบและหยุดการใช้ตัวประมวลผลที่บกพร่องโดยอัตโนมัติ

เริ่มต้นด้วยเครื่องชนิด 7044 รุ่น 270 ฮาร์ดแวร์ของระบบทั้งหมด ที่มีตัวประมวลผลตั้งแต่สองตัวขึ้นไปสามารถตรวจพบข้อผิดพลาดที่สามารถแก้ไขได้ ซึ่งถูกรวบรวมโดยเฟิร์มแวร์ ข้อผิดพลาดเหล่านี้ไม่ใช่ข้อผิดพลาดที่รุนแรง แต่ข้อผิดพลาดนี้ยังเกิดขึ้นได้เสมอ ซึ่งสามารถละเว้นได้อย่างปลอดภัย อย่างไรก็ตาม เมื่อรูปแบบของ ความล้มเหลวมีแนวโน้มที่จะเกิดขึ้นบนตัวประมวลผลเฉพาะ รูปแบบนี้อาจ บ่งชี้ว่าคอมโพเนนต์นี้อาจเกิดความล้มเหลวรุนแรงในอนาคต อันไกล การคาดการณ์นี้มีการทำโดยเฟิร์มแวร์ตามข้อมูลอัตราความล้มเหลว และการวิเคราะห์ threshold

บนระบบเหล่านี้ AIX ทำการสอดส่องดูแล ฮาร์ดแวร์อย่างต่อเนื่องและสำรวจเฟิร์มแวร์อย่างสม่ำเสมอเพื่อตรวจหา ข้อผิดพลาดฮาร์ดแวร์ เมื่อจำนวนของข้อผิดพลาดตัวประมวลผลตรงกับ threshold และเฟิร์มแวร์รับรู้ว่ามีแนวโน้มจะเป็นที่คอมโพเนนต์ของระบบนี้จะล้มเหลว เฟิร์มแวร์จะส่งคืนรายงานข้อผิดพลาด ในทุกกรณี ข้อผิดพลาดจะถูก บันทึกไว้ในบันทึกข้อผิดพลาดระบบ นอกจากนี้ สำหรับระบบแบบมัลติโพรเซสเซอร์ขึ้นอยู่กับชนิดของความล้มเหลว AIX พยายามหยุดการใช้ตัวประมวลผลที่ไม่แน่ใจ และจัดสรรคืน คุณลักษณะนี้ เรียกว่า *Dynamic Processor Deallocation*

ณ จุดนี้ เฟิร์มแวร์จะแฟล็กตัวประมวลผลสำหรับการจัดสรรคืนที่ยังคงอยู่สำหรับการรีบูตในลำดับต่อมา จนกว่าบุคลากรฝ่ายซ่อมบำรุงจะเปลี่ยนตัวประมวลผล

ผลกระทบของการยกเลิกจัดสรรตัวประมวลผลต่อแอสพิลเคชัน:

การยกเลิกจัดสรรตัวประมวลผลเป็นสิ่งที่เห็นได้ชัดเจนสำหรับแอสพิลเคชัน ส่วนใหญ่ รวมถึงไดรเวอร์และส่วนขยายเคอร์เนล อย่างไรก็ตาม คุณสามารถใช้ อินเทอร์เน็ตที่เผยแพร่เพื่อกำหนดว่า แอสพิลเคชันหรือส่วนขยายเคอร์เนล กำลังรันอยู่บนเครื่องแบบหลายตัวประมวลผลหรือไม่ ค้นหาจำนวนของตัวประมวลผลที่มีอยู่ และเชื่อมโยงเรดกับตัวประมวลผลที่ระบุเฉพาะ

อินเทอร์เน็ต `bindprocessor` สำหรับการเชื่อมโยงการประมวลผลหรือเรดกับตัวประมวลผล จะใช้การเชื่อมโยงจำนวนของ CPU การเชื่อมโยงจำนวนของ CPU จะอยู่ในช่วง $[0..N-1]$ โดยที่ N คือจำนวนทั้งหมดของ CPU หากต้องการหลีกเลี่ยงการพักแอสพิลเคชัน หรือส่วนขยายเคอร์เนลที่ไม่มี "ช่องโหว่" ในการกำหนดหมายเลข CPU AIX จะทำให้ปรากฏขึ้นสำหรับแอสพิลเคชันเสมอ หาก CPU คือตัว "ล่าสุด" (หมายเลขสูงสุด) ที่เชื่อมโยง CPU ซึ่งต้องการยกเลิกจัดสรร สำหรับกรณีนี้ SMP แบบ 8 ทิศทาง หมายเลขการเชื่อมโยง CPU คือ $[0..7]$ ถ้าจัดสรรคืนหนึ่งตัวประมวลผลแล้ว จำนวนทั้งหมดที่ CPU พร้อมใช้งาน กลายเป็น 7 และจะมีหมายเลข $[0..6]$ จากภายนอก จะดูเหมือนว่า CPU 7 ไม่ปรากฏขึ้น โดยไม่คำนึงว่า ตัวประมวลผลฟิสิคัล ไตล์มเหลว

หมายเหตุ: สำหรับส่วนที่เหลือของคำอธิบายนี้ คำว่า *CPU* จะถูกใช้สำหรับเอนทิตีแบบโลจิคัล และคำศัพท์ *ตัวประมวลผล* จะใช้สำหรับเอนทิตีแบบฟิสิคัล

แอสพิลเคชันหรือส่วนขยายเคอร์เนลที่เป็นโปรเซสหรือเรดที่เชื่อมโยงสามารถขาดได้ หาก AIX ยกเลิกเรดที่เชื่อมโยงไว้ หรือย้ายเรดเหล่านั้นไปยัง CPU อื่น เมื่อหนึ่งในตัวประมวลผลจำเป็นต้องถูกยกเลิกจัดสรร Dynamic Processor Deallocation นำเสนอ อินเทอร์เน็ตเขียนโปรแกรมเพื่อให้แอสพิลเคชันและส่วนขยายเคอร์เนลดังกล่าว สามารถได้รับแจ้งว่า การยกเลิกจัดสรรตัวประมวลผลกำลังจะเกิดขึ้น เมื่อแอสพิลเคชันและ ส่วนขยายเคอร์เนลเหล่านี้ได้รับการแจ้งเตือน แอสพิลเคชันและ ส่วนขยายจะรับผิดชอบต่อการย้าย เรดข้อจำกัดและรีซอร์สที่เชื่อมโยงไว้ (เช่น ตัวจับเวลาการร้องขอบล็อก) ออกจาก CPU ID ที่โยงไว้ล่าสุด และปรับเรดและรีซอร์สเหล่านั้นให้เป็นคอนฟิกูเรชัน CPU ใหม่

หลังการแจ้งเตือน หากเรดบางส่วนยังคงโยงกับ CPU ID 'ล่าสุด' ที่โยงไว้ การยกเลิกจัดสรรจะถูกยกเลิก การยกเลิกจัดสรรที่ถูกยกเลิกมีการบันทึกไว้ในบันทึก ข้อผิดพลาด และ AIX ใช้ ตัวประมวลผลที่จัดสรรต่อไป ในที่สุด เมื่อตัวประมวลผลล้มเหลว อาจส่งผลให้เกิดความล้มเหลวของระบบทั้งหมด ดังนั้น จึงมีความสำคัญที่แอสพิลเคชันหรือส่วนขยายเคอร์เนล ต้องได้รับการแจ้งเตือนเกี่ยวกับการยกเลิกจัดสรรตัวประมวลผลที่ใกล้เข้ามา และดำเนินการกับการแจ้งเตือนนี้

แม้ในกรณีที่ยากจะเกิดขึ้นซึ่งไม่สามารถดำเนินการยกเลิกจัดสรรได้ Dynamic Processor Deallocation ยังคงให้คำเตือนล่วงหน้าแก่ผู้ดูแลระบบ ด้วยการบันทึกข้อผิดพลาดในบันทึกข้อผิดพลาด บันทึกข้อผิดพลาดนี้จะกำหนดโอกาสให้กับผู้ดูแลระบบ เพื่อจัดกำหนดตารางเวลาสำหรับการดำเนินการดูแลรักษาระบบ เพื่อแทนที่คอมโพเนนต์ที่มีข้อบกพร่อง ก่อนที่ความล้มเหลวระบบโกลบอลจะเกิดขึ้น

โปรเซสการยกเลิกจัดสรรตัวประมวลผล:

AIX สามารถ หยุดตัวประมวลผลที่ล้มเหลวโดยการยกเลิกจัดสรรตัวประมวลผลนั้น

การไหลของเหตุการณ์สำหรับการจัดสรรคืนตัวประมวลผลมีดังต่อไปนี้:

1. เฟิร์มแวร์ตรวจพบ threshold ข้อผิดพลาดที่สามารถกู้คืนได้ ซึ่งจะเข้าถึงด้วยหนึ่งในตัวประมวลผล

2. รายงานข้อผิดพลาดเพิ่มเติมจะมีการบันทึกไว้ในบันทึกข้อผิดพลาดของระบบ และเมื่อ AIX กำลังดำเนินการบนเครื่องที่สนับสนุนการยกเลิก จัดสรรตัวประมวลผล AIX จะเริ่มต้น โพรเซสการยกเลิกจัดสรรตัวประมวลผล
3. AIX แจ้งเตือนการประมวลผลที่ไม่ใช่เคอร์เนลและเธรดที่โยงไว้กับ CPU ที่โยงไว้ล่าสุด
4. AIX รอานสูงสุดลิบนาที่ เพื่อให้เธรดที่โยงไว้ทั้งหมดย้ายออกจาก CPU ที่โยงไว้ล่าสุด หากเธรด ยังคงเชื่อมโยงอยู่ AIX จะยกเลิก การยกเลิกจัดสรร
5. หากโพรเซสหรือเธรดทั้งหมดถูกยกเลิกการโยงจากตัวประมวลผลที่จัดสรร จะมีการ เรียกใช้ High Availability Event Handlers (HAEHs) ที่ลงทะเบียนไว้ก่อนหน้านี้ HAEH อาจส่งคืนข้อผิดพลาดที่ยกเลิกการยกเลิกจัดสรร
6. ในท้ายที่สุด โพรเซสการยกเลิกจัดสรรจะหยุดตัวประมวลผลที่ล้มเหลว ยกเว้นว่า โพรเซสถูกยกเลิก

หากเกิดความล้มเหลวที่จุดใดๆ ของการยกเลิกจัดสรร ความล้มเหลวและสาเหตุของความล้มเหลวนั้นจะมีการบันทึกไว้ ผู้ดูแลระบบสามารถมองบันทึกข้อผิดพลาด ดำเนินการแก้ไข (เมื่อสามารถทำได้) และรีสตาร์ทการจัดสรรคืน ตัวอย่างเช่น ถ้าการยกเลิกจัดสรรถูกยกเลิก เนื่องจากแอปพลิเคชันไม่ได้ยกเลิกการโยงเธรดข้อจำกัด ผู้ดูแลระบบสามารถหยุดแอปพลิเคชัน รีสตาร์ทการยกเลิกจัดสรร จากนั้นรีสตาร์ทแอปพลิเคชัน

การเปิดใช้งาน Dynamic Processor Deallocation:

ถ้าเครื่องของคุณสนับสนุน Dynamic Processor Deallocation คุณสามารถใช้ SMIT หรือคำสั่งระบบเพื่อ เปิด หรือ ปิด คุณลักษณะได้

Dynamic Processor Deallocation ที่ถูกเปิดใช้งาน ตามดีฟอลต์ระหว่างการติดตั้ง ช่วยให้เครื่องมีฮาร์ดแวร์ และเฟิร์มแวร์ที่ถูกต่อรองรับ

โพรซีเจอร์พาดผ่าน SMIT

1. ด้วยสิทธิการใช้งานราก พิมพ์ `smit system` ที่พร้อมระบบ แล้วกด Enter
2. ในหน้าต่าง สภาวะแวดล้อมของระบบ ให้เลือก เปลี่ยน / แสดงลักษณะของระบบปฏิบัติการ
3. ใช้ไดอะล็อก SMIT เพื่อทำการกิจให้เสร็จสมบูรณ์

เมื่อต้องการข้อมูลเพิ่มเติมเกี่ยวกับการทำการกิจให้สมบูรณ์ คุณสามารถ เลือกปุ่มวิธีใช้ F1 ในไดอะล็อก SMIT

โพรซีเจอร์คำสั่ง

ด้วยสิทธิการใช้งานราก คุณสามารถใช้ คำสั่งต่อไปนี้เพื่อทำงานกับ Dynamic Processor Deallocation:

- ใช้คำสั่ง `chdev` เพื่อเปลี่ยนลักษณะ ของอุปกรณ์ที่ระบุ สำหรับข้อมูลเกี่ยวกับการใช้คำสั่งนี้ ให้ดูที่ `chdev` ใน *Commands Reference, Volume 1*
- หากการยกเลิกการจัดสรรตัวประมวลผลล้มเหลวด้วยเหตุผลใดก็ตาม คุณสามารถใช้คำสั่ง `ha_star` เพื่อรีสตาร์ทหลังจากความล้มเหลวนั้นได้รับการแก้ไขแล้ว สำหรับข้อมูลเกี่ยวกับการใช้คำสั่งนี้ ให้ดูที่ `ha_star` ใน *Commands Reference, Volume 2*
- ใช้คำสั่ง `errpt` เพื่อสร้างรายงานของข้อผิดพลาด ที่บันทึกไว้ สำหรับข้อมูลเกี่ยวกับการใช้คำสั่งนี้ ให้ดูที่ `errpt` ใน *Commands Reference, Volume 2*

เมธอดการเปิดและการปิดการยกเลิกจัดสรรตัวประมวลผล:

Dynamic Processor Deallocation สามารถ เปิดใช้งานหรือปิดใช้งานโดยการเปลี่ยนค่าแอตทริบิวต์ `cpuguard` ของอ็อบเจกต์ `ODM sys0`

ค่าที่เป็นไปได้สำหรับแอตทริบิวต์คือ เปิดใช้งาน และ ปิดใช้งาน

ดีฟอลต์คือ `enabled` (แอตทริบิวต์ `cpuguard` มี ค่าเป็น `enable`) ผู้ดูแลระบบที่ต้องการปิดใช้งานคุณลักษณะนี้ ต้องใช้เมนูระบบ, เมนู `SMIT System Environments` หรือคำสั่ง `chdev` (ในเวอร์ชัน AIX ก่อนหน้านี้ ค่าดีฟอลต์คือปิดใช้งาน)

หมายเหตุ: หากปิด (ปิดใช้งาน) การยกเลิกจัดสรรตัวประมวลผล ข้อผิดพลาดเหล่านี้ยังคง มีการบันทึกไว้ บันทึกข้อผิดพลาดจะมีข้อผิดพลาด เช่น `CPU_FAILURE_PREDICTED` ที่บ่งชี้ว่า AIX ได้รับแจ้งถึง ปัญหาเกี่ยวกับ CPU

การรีสตาร์ทการยกเลิกจัดสรรตัวประมวลผลที่ยกเลิก:

บางครั้ง การยกเลิกจัดสรรตัวประมวลผลล้มเหลวเนื่องจากแอ็พพลิเคชัน ไม่ได้ย้ายเธรดข้อจำกัดออกจากโลจิคัล CPU ล่าสุด

หลังจากแก้ไขปัญหาเหล่านี้แล้ว โดยการยกเลิกผูก (เมื่อสามารถทำได้ อย่างปลอดภัย) หรือโดยการหยุดแอ็พพลิเคชัน ผู้ดูแลระบบสามารถรีสตาร์ทโปรเซสการยกเลิกจัดสรรตัวประมวลผลได้โดยใช้คำสั่ง `ha_star`

ไวยากรณ์ สำหรับคำสั่งนี้คือ:

```
ha_star -C
```

โดยที่ `-C` คือ เหตุการณ์ความล้มเหลวที่คาดการณ์ได้ของ CPU

ข้อควรพิจารณาเกี่ยวกับสถานะตัวประมวลผล:

มีหลายสิ่งซึ่งคุณควรพิจารณาเกี่ยวกับสถานะตัวประมวลผล

ตัวประมวลผลฟิสิกส์มีการแสดงแทนในฐานข้อมูล ODM โดยอ็อบเจกต์ที่ชื่อ `procn` โดยที่ `n` คือ ตัวเลขฐานสิบที่แสดงถึงหมายเลขตัวประมวลผลฟิสิกส์ เช่นเดียวกับอุปกรณ์อื่น ที่แสดงแทนในฐานข้อมูล ODM อ็อบเจกต์ตัวประมวลผลมีสถานะ เช่น กำหนดไว้/มีอยู่ และแอตทริบิวต์

สถานะของอ็อบเจกต์ `proc` เป็น มีอยู่ เสมอตราบเท่าที่มีตัวประมวลผลที่ สอดคล้องกัน โดยไม่คำนึงว่าสามารถใช้งานได้หรือไม่ แอตทริบิวต์ สถานะ ของอ็อบเจกต์ `proc` บ่งชี้ว่าตัวประมวลผลถูกใช้อยู่หรือไม่ และถ้าไม่ แสดง เหตุผล แอตทริบิวต์นี้สามารถมีสามค่าดังนี้:

ไอเท็ม	คำอธิบาย
เปิดใช้งาน	ตัวประมวลผลถูกใช้งาน
ปิดใช้งาน	ตัวประมวลผลถูกยกเลิกจัดสรรแบบไดนามิก
บกพร่อง	ตัวประมวลผลถูกประกาศว่าบกพร่อง โดยเฟิร์มแวร์ในเวลาสแตร์ทอัพ

หากยกเลิกจัดสรรตัวประมวลผลที่บกพร่องได้สำเร็จ สถานะของตัวประมวลผลนั้นจะเปลี่ยนจาก `เปิดใช้งาน` เป็น `ปิดใช้งาน` โดยเป็นอิสระจาก AIX ตัวประมวลผลนี้ยังถูกแฟล็กในเฟิร์มแวร์เป็นบกพร่องด้วย หลังจากรีบูต ตัวประมวลผลที่ยกเลิกจัด

สรรจะไม่มีอยู่และจะมีการตั้งค่าสถานะเป็น บกพร่อง อย่างไรก็ตาม อ็อบเจ็กต์ ODM proc ยังคงมีการทำเครื่องหมายเป็น มีอยู่ คุณต้องเอา CPU ที่บกพร่องออกจากบอร์ดระบบทางกายภาพ หรือเอาบอร์ด CPU (ถ้าเป็นไปได้) สำหรับอ็อบเจ็กต์ proc ออก เพื่อเปลี่ยนเป็น กำหนดไว้

ในตัวอย่างต่อไปนี้ ตัวประมวลผล proc4 กำลังทำงานอย่างถูกต้องและ ใช้อ้อยู่โดยระบบปฏิบัติการ ดังแสดงในเอาต์พุตต่อไปนี้:

```
# lsattr -EH -l proc4
attribute value  description user_settable

state enable Processor state False
type PowerPC_RS64-III Processor type False
#
```

เมื่อมีการคาดการณ์ว่าตัวประมวลผล proc4 จะล้มเหลว ตัวประมวลผลจะถูก ยกเลิกจัดสรรโดยระบบปฏิบัติการ ดังแสดงในตัวอย่างต่อไปนี้:

```
# lsattr -EH -l proc4
attribute value  description user_settable

state disable Processor state False
type PowerPC_RS64-III Processor type False
#
```

ในการรีสตาร์ทระบบครั้งถัดไป เฟิร์มแวร์จะรายงานตัวประมวลผล proc4 เป็นบกพร่อง ดังแสดงในตัวอย่างต่อไปนี้:

```
# lsattr -EH -l proc4
attribute value  description user_settable

state faulty Processor state False
type PowerPC_RS64-III Processor type False
#
```

แต่สถานะของตัวประมวลผล proc4 ยังคงเป็น มีอยู่ ดังแสดงใน ตัวอย่างต่อไปนี้:

```
# lsdev -CH -l proc4
name status  location description

proc4 Available 00-04 Processor
#
```

รายการบันทึกข้อผิดพลาดการยกเลิกจัดสรร:

ข้อความบันทึกข้อผิดพลาดที่เชื่อมโยงกับการยกเลิกจัดสรร CPU มีอยู่สามข้อความ

ตัวอย่างมีดังต่อไปนี้

รูปแบบย่อ errpt - ข้อมูลสรุป

ข้อมูลต่อไปนี้เป็นตัวอย่างเป็นตัวอย่างของรายการที่แสดงขึ้นโดยคำสั่ง errpt (โดยไม่มีอ็อพชัน):

```
# errprt
IDENTIFIER      TIMESTAMP      T    C    RESOURCE_NAME  DESCRIPTION
804E987A        1008161399    I    O    proc4           CPU DEALLOCATED
8470267F        1008161299    T    S    proc4           CPU DEALLOCATION ABORTED
1B963892        1008160299    P    H    proc4           CPU FAILURE PREDICTED
#
```

- หากเปิดใช้งานการยกเลิกจัดสรรตัวประมวลผล ข้อความ CPU FAILURE PREDICTED จะมีข้อความ CPU DEALLOCATED หรือ ข้อความ CPU DEALLOCATION ABORTED ตามหลังเสมอ
- หากไม่เปิดใช้งานการยกเลิกจัดสรรตัวประมวลผล จะมีการบันทึกเฉพาะข้อความ CPU FAILURE PREDICTED เท่านั้น การเปิดใช้งานการยกเลิกจัดสรรตัวประมวลผล ในเวลาใดๆ หลังจากมีการบันทึกข้อความ CPU FAILURE PREDICTED แล้ว ตั้งแต่หนึ่งข้อความขึ้นไป จะเริ่มต้นโปรเซสการยกเลิกจัดสรรและส่งผลให้เกิด รายการบันทึกข้อผิดพลาดที่สำเร็จหรือล้มเหลว ดังอธิบายข้างบน สำหรับแต่ละความล้มเหลวของตัวประมวลผลที่รายงาน

รูปแบบยาว errprt - คำอธิบายโดยละเอียด

ข้อมูลต่อไปนี้เป็นรูปแบบของเอาต์พุตที่ได้รับเมื่อใช้ errprt -a:

- CPU_FAIL_PREDICTED

คำอธิบายข้อผิดพลาด: ความล้มเหลวของ ตัวประมวลผลที่คาดการณ์

ข้อผิดพลาดนี้บ่งชี้ว่าฮาร์ดแวร์ตรวจพบว่า ตัวประมวลผลมีโอกาสที่จะล้มเหลวสูงในอนาคตอันใกล้นี้ มีการบันทึกเสมอไม่ว่าการยกเลิกจัดสรรตัวประมวลผลจะเปิดใช้งานหรือไม่

ข้อมูล รายละเอียด: หมายเลขตัวประมวลผลฟิสิกส์, ที่ตั้ง

ตัวอย่าง รายการบันทึกข้อผิดพลาด - รูปแบบยาว

```
LABEL: CPU_FAIL_PREDICTED
IDENTIFIER: 1655419A
```

```
Date/Time: Thu Sep 30 13:42:11
Sequence Number: 53
Machine Id: 00002F0E4C00
Node Id: auntbea
Class: H
Type: PEND
Resource Name: proc25
Resource Class: processor
Resource Type: proc_rspc
Location: 00-25
```

```
Description
CPU FAILURE PREDICTED
```

```
Probable Causes
CPU FAILURE
```

```
Failure Causes
CPU FAILURE
```

```
Recommended Actions
ENSURE CPU GARD MODE IS ENABLED
RUN SYSTEM DIAGNOSTICS.
```

```
Detail Data
```

```
PROBLEM DATA
0144 1000 0000 003A 8E00 9100 1842 1100 1999 0930 4019
0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 4942 4D00 5531
2E31 2D50 312D 4332 0000
0002 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000
... ..
```

- CPU_DEALLOC_SUCCESS

คำอธิบายข้อผิดพลาด: ตัวประมวลผล ถูกยกเลิกจัดสรรเสร็จเรียบร้อยแล้วหลังจากการตรวจพบความล้มเหลวของตัวประมวลผลที่คาดการณ์ ข้อความนี้มีการบันทึกเมื่อเปิดใช้งานการยกเลิกจัดสรรตัวประมวลผล และเมื่อยกเลิกจัดสรร CPU เสร็จเรียบร้อยแล้ว

ข้อมูล รายละเอียด: หมายเลขโลจิคัล CPU ของตัวประมวลผลที่ยกเลิกจัดสรร

ตัวอย่าง: รายการบันทึกข้อผิดพลาด - รูปแบบยาว

```
LABEL: CPU_DEALLOC_SUCCESS
IDENTIFIER: 804E987A
```

```
Date/Time: Thu Sep 30 13:44:13
Sequence Number: 63
Machine Id: 00002F0E4C00
Node Id: auntbea
Class: 0
Type: INFO
Resource Name: proc24
```

```
Description
CPU DEALLOCATED
```

```
Recommended Actions
MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE
```

```
Detail Data
LOGICAL DEALLOCATED CPU NUMBER
```

```
0
```

ในตัวอย่างนี้ **proc24** ถูกยกเลิกจัดสรรเรียบร้อยแล้ว และเป็นโลจิคัล CPU 0 เมื่อเกิดความล้มเหลวขึ้น

- CPU_DEALLOC_FAIL

คำอธิบายข้อผิดพลาด: การยกเลิกจัดสรรตัวประมวลผลเนื่องจากความล้มเหลวของตัวประมวลผลที่คาดการณ์ ไม่สำเร็จ ข้อความนี้มีการบันทึกเมื่อเปิดใช้งานการยกเลิกจัดสรร CPU และเมื่อยกเลิกจัดสรร CPU เสร็จเรียบร้อยแล้ว

ข้อมูล รายละเอียด: โค้ดเหตุผล หมายเลขโลจิคัล CPU และข้อมูลเพิ่มเติม ขึ้นอยู่กับชนิดของความล้มเหลว
โค้ดเหตุผล เป็นค่าตัวเลขฐานสิบหก โค้ดเหตุผลที่เป็นไปได้ คือ:

ไอเท็ม

2

คำอธิบาย

โปรเซส/เธรดตั้งแต่หนึ่งรายการขึ้นไปยังคงผูกอยู่กับโลจิคัล CPU ล่าสุด ในกรณีนี้ ข้อมูลรายละเอียดแสดง PIDs ของโปรเซสที่ทำ

3

ผิด
ไดรเวอร์หรือส่วนขยายเคอร์เนลที่ลงทะเบียนส่งคืนข้อผิดพลาดเมื่อได้รับแจ้ง ในกรณีนี้ ฟีดข้อมูลรายละเอียดมีชื่อของ ไดรเวอร์

4

หรือส่วนขยายเคอร์เนลที่ทำผิด (ที่เข้ารหัส ASCII)
การยกเลิกจัดสรรตัวประมวลผลส่งผลให้เครื่องมี CPUs ที่พร้อมใช้งาน น้อยกว่าสองเครื่อง ระบบปฏิบัติการนี้ไม่ยกเลิกจัดสรร

200 (0xC8)

มากกว่า $N-2$ ตัวประมวลผลบนเครื่อง N -way เพื่อหลีกเลี่ยงการสร้างความเสี่ยงให้แก่อุปพลิเคชันหรือส่วนขยายเคอร์เนลที่ใช้
จำนวนทั้งหมดของตัวประมวลผลที่มีอยู่เพื่อพิจารณาว่า กำลังรันอยู่บนระบบ Uni Processor (UP) ซึ่งสามารถละเว้นการใช้ล็อก
หลาย ตัวประมวลผลได้อย่างปลอดภัย หรือ Symmetric Multi Processor (SMP) หรือไม่
การยกเลิกจัดสรรตัวประมวลผลถูกปิดใช้งาน (แอตทริบิวต์ ODM cpuguard มีค่าเป็น disable) โดยปกติ คุณไม่เห็นข้อผิดพลาดนี้ ยกเว้นว่าคุณเริ่มต้น ha_star ด้วยตนเอง

ตัวอย่าง: รายการบันทึกข้อผิดพลาด - รูปแบบยาว

ตัวอย่างที่ 1:

```
LABEL: CPU_DEALLOC_ABORTED
IDENTIFIER: 8470267F
Date/Time: Thu Sep 30 13:41:10
Sequence Number: 50
Machine Id: 00002F0E4C00
Node Id: auntbea
Class: S
Type: TEMP
Resource Name: proc26
```

Description

CPU DEALLOCATION ABORTED

Probable Causes

SOFTWARE PROGRAM

Failure Causes

SOFTWARE PROGRAM

Recommended Actions

MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE
SEE USER DOCUMENTATION FOR CPU GARD

Detail Data

```
DEALLOCATION ABORTED CAUSE
0000 0003
DEALLOCATION ABORTED DATA
6676 6861 6568 3200
```

ในตัวอย่างนี้ การยกเลิกจัดสรรสำหรับ proc26 ล้มเหลว โค้ดเหตุผล 3 หมายความว่า ส่วนขยายเคอร์เนลส่งคืน
ข้อผิดพลาดกลับไปยังรูทีนการแจ้งเตือน เคอร์เนล DEALLOCATION ABORTED DATA ข้างบนแสดง fvhaeh2 ซึ่งเป็น
ชื่อของส่วนขยายที่ใช้เมื่อลงทะเบียนกับเคอร์เนล

ตัวอย่างที่ 2:

```
LABEL: CPU_DEALLOC_ABORTED
IDENTIFIER: 8470267F
Date/Time: Thu Sep 30 14:00:22
Sequence Number: 71
Machine Id: 00002F0E4C00
```

Node Id: auntbea
Class: S
Type: TEMP
Resource Name: proc19

Description
CPU DEALLOCATION ABORTED

Probable Causes
SOFTWARE PROGRAM

Failure Causes
SOFTWARE PROGRAM

Recommended Actions
MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE;
SEE USER DOCUMENTATION FOR CPU GARD

Detail Data
DEALLOCATION ABORTED CAUSE
0000 0002
DEALLOCATION ABORTED DATA
0000 0000 0000 4F4A

ในตัวอย่างนี้ การยกเลิกจัดสรร สำหรับ proc19 ล้มเหลว โค้ดเหตุผล 2 บ่งชี้ว่า เเรตถูกผูกไว้กับตัวประมวลผลโลจิคัลล่าสุดและไม่ได้แยกออก หลังจากได้รับสัญญาณ SIGCPUFAIL DEALLOCATION ABORTED DATA แสดงว่าเเรตเหล่านี้เป็นของโปรเซส 0x4F4A

อ็อพชั่น ของคำสั่ง ps (-o THREAD, -o BND) ช่วยให้คุณสามารถแสดงรายการเเรต หรือโปรเซสทั้งหมดควบคู่ไปกับจำนวนของ CPU ที่ถูกผูกไว้ เมื่อเหมาะสม

ตัวอย่างที่ 3:

LABEL: CPU_DEALLOC_ABORTED
IDENTIFIER: 8470267F

Date/Time: Thu Sep 30 14:37:34
Sequence Number: 106
Machine Id: 00002F0E4C00
Node Id: auntbea
Class: S
Type: TEMP
Resource Name: proc2

Description
CPU DEALLOCATION ABORTED

Probable Causes
SOFTWARE PROGRAM

Failure Causes
SOFTWARE PROGRAM

Recommended Actions
MAINTENANCE IS REQUIRED BECAUSE OF CPU FAILURE

SEE USER DOCUMENTATION FOR CPU GARD

Detail Data
DEALLOCATION ABORTED CAUSE
0000 0004
DEALLOCATION ABORTED DATA
0000 0000 0000 0000

ในตัวอย่างนี้ การยกเลิกจัดสรรของ **proc2** ล้มเหลว เนื่องจากมีตัวประมวลผลที่ใช้งานอยู่จำนวนสองตัวหรือน้อยกว่านั้นในเวลาของความล้มเหลว (โค้ดเหตุผล 4)

เช็คอัปเดตแปรสภาวะแวดล้อมของระบบ

โดยหลักแล้ว สภาวะแวดล้อมของระบบคือชุดของตัวแปรที่กำหนด หรือควบคุมลักษณะบางอย่างของการดำเนินการโปรเซส

สภาวะแวดล้อมของระบบมีการตั้งค่าหรือรีเซ็ตในทุกครั้งที่เซลล์เริ่มต้นขึ้น จากมุมมองด้านการจัดการระบบ สิ่งสำคัญคือการทำให้อุ่นใจว่ามีการตั้งค่าผู้ใช้ด้วยค่าที่ถูกต้อง เมื่อล็อกอิน ตัวแปรเหล่านี้ส่วนใหญ่มีการตั้งค่าในระหว่าง การเริ่มต้นระบบ มีการอ่านนิยามของตัวแปรจากไฟล์ /etc/profile หรือตั้งค่าโดยดีฟอลต์

การทดสอบแบตเตอรี่ระบบ:

หากระบบของคุณสูญเสียการติดตามเวลา สาเหตุอาจเป็นเพราะแบตเตอรี่หมดอายุหรือไม่ได้เชื่อมต่อ

1. เมื่อต้องการกำหนดสถานะของแบตเตอรี่ระบบ ให้พิมพ์คำสั่ง **diag** ต่อไปนี้:
diag -B -c
2. เมื่อเมนูหลัก การวินิจฉัย ปรากฏขึ้น ให้เลือกอีพซัน การกำหนด ปัญหา หากแบตเตอรี่ไม่ได้เชื่อมต่อ หรือหมดอายุ เมนูปัญหาจะแสดงขึ้นพร้อมกับ service request number (SRN) บันทึก SRN บนไอเท็ม 4 ของแบบฟอร์มข้อมูลสรุปปัญหา และรายงาน ปัญหาให้หน่วยงานเซอร์วิสฮาร์ดแวร์ของคุณรับทราบ

หากแบตเตอรี่ระบบใช้งานได้ อาจเป็นเพราะรีเซ็ตเวลาระบบ ไม่ถูกต้องเนื่องจากรันคำสั่ง **date** หรือ **setclock** อย่างไม่ถูกต้องหรือไม่สำเร็จ

หลักการที่เกี่ยวข้อง:

“การตั้งค่านาฬิกากระบบ”

นาฬิกากระบบบันทึกเวลาของเหตุการณ์ระบบ อนุญาตให้คุณ จัดตารางเวลาเหตุการณ์ระบบ (เช่น การรันการวินิจฉัยฮาร์ดแวร์ในเวลา 3:00 a.m.) และบอกเวลาซึ่งคุณสร้างไฟล์ครั้งแรกหรือบันทึกไฟล์ครั้งสุดท้าย

การตั้งค่านาฬิกากระบบ:

นาฬิกากระบบบันทึกเวลาของเหตุการณ์ระบบ อนุญาตให้คุณ จัดตารางเวลาเหตุการณ์ระบบ (เช่น การรันการวินิจฉัยฮาร์ดแวร์ในเวลา 3:00 a.m.) และบอกเวลาซึ่งคุณสร้างไฟล์ครั้งแรกหรือบันทึกไฟล์ครั้งสุดท้าย

ใช้คำสั่ง **date** เพื่อตั้งค่านาฬิกากระบบของคุณ ใช้คำสั่ง **setclock** เพื่อ ตั้งค่าเวลาและวันที่โดยติดต่อกับเซิร์ฟเวอร์เวลา

งานที่เกี่ยวข้อง:

“การทดสอบแบตเตอรี่ระบบ”

หากระบบของคุณสูญเสียการติดตามเวลา สาเหตุอาจเป็นเพราะแบตเตอรี่หมดอายุหรือไม่ได้เชื่อมต่อ

คำสั่ง `date`:

คำสั่ง `date` แสดงหรือตั้งค่าวันที่และ เวลา

ป้อนคำสั่งต่อไปนี้เพื่อกำหนดสิ่งที่ระบบของคุณรับรู้ว่าเป็น วันที่และเวลาปัจจุบัน:

```
/usr/bin/date
```

ข้อควรสนใจ: อย่าเปลี่ยนวันที่เมื่อระบบกำลังรันโดยมี ผู้ใช้มากกว่าหนึ่งคน

รูปแบบต่อไปนี้สามารถถูกใช้ เมื่อทำการ ตั้งค่าวันที่ด้วยพารามิเตอร์ `Date`:

- `mmddHHMM[YYyy]` (ค่าดีฟอลต์)
- `mmddHHMM[yy]`

ตัวแปรที่ใช้กับพารามิเตอร์ `Date` ถูกกำหนดไว้ดังนี้:

ไอเท็ม	คำอธิบาย
<code>mm</code>	ระบุจำนวนเดือน
<code>dd</code>	ระบุจำนวนวันในเดือน
<code>HH</code>	ระบุชั่วโมงในหนึ่งวัน (โดยใช้นาฬิกา 24-ชั่วโมง)
<code>MM</code>	ระบุตัวเลขนาฬิกา
<code>YY</code>	ระบุตัวเลขสองตัวแรกของปีสี่ตำแหน่ง
<code>yy</code>	ระบุตัวเลขสองตัวหลังของปี

ด้วยสิทธิการใช้งานราก คุณสามารถใช้คำสั่ง `date` เพื่อตั้งค่า วันที่และเวลาปัจจุบัน ตัวอย่างเช่น:

```
date 021714252002
```

ตั้งค่า วันที่เป็น Feb. 17, 2002 และเวลาเป็น 14:25 สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ คำสั่ง `date` ให้ดูที่คำอธิบายของคำสั่งนั้นใน *Commands Reference, Volume 2*

คำสั่ง `setclock`:

คำสั่ง `setclock` แสดงหรือตั้งค่าเวลา และวันที่โดยการร้องขอเวลาปัจจุบันจากเซิร์ฟเวอร์เวลาบนเครือข่าย

เมื่อต้องการแสดงวันที่และเวลาของระบบ ให้ป้อน:

```
/usr/sbin/setclock
```

คำสั่ง `setclock` ใช้การตอบกลับครั้งแรกจาก เซิร์ฟเวอร์เวลา แปลงการอ่านนาฬิกาปฏิทินซึ่งพบที่นั่น และแสดง วันที่และเวลาโลคัล ถ้าไม่มีการตอบกลับจากเซิร์ฟเวอร์เวลา หรือถ้าเครือข่ายไม่ทำงาน คำสั่ง `setclock` แสดงข้อความแจ้งถึงผลกระทบนั้น และไม่เปลี่ยนค่าที่ตั้งวันที่และเวลา

หมายเหตุ: โฮสต์ใดๆ ที่กำลังรัน `inetd` daemon สามารถทำหน้าที่เป็นเซิร์ฟเวอร์เวลาได้

ด้วยสิทธิใช้งานราก คุณสามารถใช้คำสั่ง `setclock` เพื่อ ส่งคำร้องขอเซอร์วิสอินเทอร์เน็ท TIME ไปยังโฮสต์เซิร์ฟเวอร์เวลาและตั้งค่า วันที่และเวลาโลคัลตามนั้น ตัวอย่างเช่น:

```
setclock TimeHost
```

โดยที่ `TimeHost` คือชื่อโฮสต์หรือ IP แอดเดรสของเซิร์ฟเวอร์เวลา

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง setclock

การสนับสนุนเขตเวลา Olson และซีตอป:

เริ่มต้นด้วย AIX 6.1 มีการนำเสนอการสนับสนุนค่าเขตเวลาที่สอดคล้องกับฐานข้อมูล Olson

ข้อกำหนดคุณลักษณะเขตเวลา POSIX ที่สนับสนุนใน AIX รุ่นก่อนหน้านี้ไม่สามารถจัดการกับการเปลี่ยนแปลงในกฎเขตเวลา เช่น เวลาออมแสง ได้อย่างถูกต้อง ฐานข้อมูล Olson เก็บรักษาเรียกครัดประวัติของกฎเขตเวลา เพื่อให้กฎเปลี่ยนในที่ตั้งเฉพาะ AIX จะสามารถตีความวันที่และเวลาได้อย่างถูกต้องทั้งในปัจจุบันและในอดีต

นิยามเขตเวลาที่สอดคล้องกับข้อกำหนดคุณลักษณะ POSIX ยังคงได้รับการสนับสนุนและรับรู้โดย AIX AIX ตรวจสอบตัวแปรสถานะแวดล้อม TZ เพื่อกำหนดว่าตัวแปรสถานะแวดล้อม ตรงกับค่าเขตเวลา Olson หรือไม่ ถ้าตัวแปรสถานะแวดล้อม TZ ไม่ตรงกับ ค่าเขตเวลา Olson AIX จะ ปฏิบัติตามกฎข้อกำหนดคุณลักษณะ POSIX

สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับตัวแปรสถานะแวดล้อม TZ อ้างถึงไฟล์สถานะแวดล้อม

เมื่อต้องการตั้งค่าเขตเวลาโดยใช้ Olson กำหนดค่าให้ใช้พารามิเตอร์ต่อไปนี้: System Environments > Change / Show Date, Time and Time Zone > Change Time Zone Using System Defined Values

ข้อความของซีตอปวัน:

ข้อความของวันแสดงขึ้นในทุกครั้งที่ผู้ใช้ล็อกอินเข้าสู่ระบบ

ข้อความนี้เป็นวิธีที่สะดวกในการสื่อสารข้อมูลกับผู้ใช้ทั้งหมด เช่น หมายเลขเวอร์ชันซอฟต์แวร์ที่ติดตั้ง หรือข่าวเกี่ยวกับระบบ ปัจจุบัน เมื่อต้องการเปลี่ยนข้อความ ของวัน ให้ใช้โปรแกรมแก้ไขที่โปรดปรานเพื่อแก้ไขไฟล์ /etc/motd

AIX Runtime Expert

AIX Runtime Expert นำเสนอ ชุดของการดำเนินการแบบง่ายที่สามารถใช้กับการรวมหนึ่งรายการ สำหรับการรวบรวม การใช้ และการตรวจสอบสถานะแวดล้อมรันไทม์ ของอินสแตนซ์ AIX ตั้งแต่หนึ่งรายการขึ้นไป

มีเครื่องมือที่นำเสนอโดยคอมพิวเตอร์ AIX เช่น Reliability Availability Serviceability (RAS), ความปลอดภัย หรือ เคอร์เนล ซึ่งช่วยให้คุณเปลี่ยนค่าติดตั้งภายในแต่ละชั้นคอมพิวเตอร์ เพื่อปรับระบบปฏิบัติการตามความต้องการ หรือข้อกำหนดเฉพาะได้ AIX Runtime Expert เปิดใช้งานคอนฟิกูเรชันทั้งระบบโดยใช้กรอบงานที่สามารถขยายได้เพื่อจัดการ เมธอดคอนฟิกูเรชันที่แตกต่างกันจำนวนมากซึ่งมีอยู่ในปัจจุบันใน AIX

AIX Runtime Expert ดำเนินการคำสั่งคอนฟิกูเรชันหลายคอมพิวเตอร์ เป็นการดำเนินการเดียวโดยใช้โปรไฟล์คอนฟิกูเรชัน คุณสามารถใช้โปรไฟล์นี้เพื่อใช้ค่าติดตั้งระบบที่เหมือนกันบน หลายระบบ AIX Runtime Expert นำเสนอ ทางเลือกแบบง่าย สำหรับการจัดการรันไทม์คอนฟิกูเรชันของ หนึ่งระบบขึ้นไป แต่ไม่ให้ห้ามการใช้เมธอดอื่น เพื่อเปลี่ยนค่าติดตั้งระบบ

แนวคิด AIX Runtime Expert

คุณต้องมีความรู้พื้นฐานเกี่ยวกับ AIX Runtime Expert ก่อนคุณ เริ่มต้นใช้งาน

ความสามารถพื้นฐาน AIX Runtime Expert สนับสนุนการจัดการโปรไฟล์คอนฟิกูเรชันและแอ็พพลิเคชันสำหรับระบบ AIX เดียว เมื่อต้องการเปิดใช้งาน การใช้หลายระบบที่ปรับสเกลได้สำหรับโปรไฟล์เดียว คำอธิบายโปรไฟล์บน LDAP สามารถค้น

พบและใช้ได้โดยระบบ AIX เมื่อระบบเริ่มต้นขึ้น หรือเมื่อระบบถูกกำหนดทิศทางโดยการดำเนินการจัดการไปที่จุดปลาย AIX เป้าหมาย การจัดการรีโมตสำหรับ AIX Runtime Expert สามารถทำ ด้วยคอมพิวเตอร์ Network Install Manager (NIM) เท่านั้น โดยใช้ฟังก์ชัน NIM ที่มีอยู่ คุณสามารถรัน AIX Runtime Expert แบบรีโมต บนหลายโหนดเอ็นดี NIM แบบสแตนด์อะโลนจากเครื่องต้นแบบ NIM

โปรไฟล์ AIX Runtime Expert:

โปรไฟล์ AIX Runtime Expert ใช้เพื่อตั้งค่าบนระบบที่กำลังรัน แยกค่าของระบบที่กำลังรัน และเปรียบเทียบค่ากับระบบที่กำลังรันหรือกับโปรไฟล์อื่น

โปรไฟล์อธิบายตัวควบคุมรันไทม์คอนฟิกูเรชันตั้งแต่หนึ่งตัวขึ้นไป และค่าติดตั้งของตัวควบคุมนั้นสำหรับพื้นที่การทำงานเป้าหมาย โปรไฟล์สามารถแสดงถึงชุดทั้งหมดของตัวควบคุม หรือชุดย่อยของตัวควบคุมและ ค่าของตัวควบคุม โปรไฟล์คอนฟิกูเรชันคือไฟล์ XML มาตรฐาน โดยใช้ AIX Runtime Expert คุณสามารถจัดการโปรไฟล์ และใช้โปรไฟล์บนระบบที่กำหนดได้

โปรไฟล์สามารถประกอบด้วยพารามิเตอร์คอนฟิกูเรชันและพารามิเตอร์การปรับที่ไม่มีค่าใดๆ เช่น โปรไฟล์ตัวอย่าง วัตถุประสงค์ของโปรไฟล์ที่ไม่มีพารามิเตอร์ใดๆ คือเพื่อแยกค่าระบบปัจจุบันออกจาก โปรไฟล์ที่ระบุ โปรไฟล์ที่มีพารามิเตอร์อย่างน้อยหนึ่งตัว ซึ่งไม่มีค่าใดๆ มีข้อจำกัดต่อไปนี้:

- การใช้คำสั่ง `artexset` ล้มเหลวด้วยข้อผิดพลาด
- การใช้คำสั่ง `artexdiff` ส่งคืนข้อความคำเตือน สำหรับแต่ละพารามิเตอร์ที่ไม่มีค่า

ค่าของพารามิเตอร์ในโปรไฟล์สามารถมีข้อมูลต่อไปนี้:

- ไม่มีค่า
- ค่า blob ซึ่งเป็นข้อมูลไบนารีที่เข้ารหัส base64 เป็นไฟล์ข้อความ อินไลน์ ค่า blob ใช้เพื่อแทนที่ไฟล์ที่มีอยู่ เช่น `/etc/motd` หรือ `/etc/hosts`
- ค่า non-blob ซึ่งเป็นค่าที่กำหนดให้กับพารามิเตอร์คอนฟิกูเรชันระบบ เช่น เลขจำนวนเต็ม หรือสตริง

ในไดเรกทอรี `/etc/security/artex/samples` คุณสามารถดูโปรไฟล์ตัวอย่างที่มีอยู่ โปรไฟล์ตัวอย่างมีเฉพาะชื่อพารามิเตอร์ที่ได้รับการสนับสนุนโดยค่าติดตั้งดีฟอลต์ซึ่งติดตั้ง พร้อมกับ AIX Runtime Expert เท่านั้น พารามิเตอร์ในโปรไฟล์ตัวอย่างไม่มีค่าใดๆ โปรไฟล์ตัวอย่างเป็นไฟล์แบบอ่านอย่างเดียว ใช้โปรไฟล์ตัวอย่างเป็นเทมเพลตเพื่อสร้างโปรไฟล์ คอนฟิกูเรชันใหม่ คุณไม่สามารถใช้ตัวอย่างที่มีอยู่กับระบบที่กำลังรัน

ตัวอย่างต่อไปนี้ เป็นคำสั่งคอนฟิกูเรชันพื้นฐานบางคำสั่ง ที่สามารถควบคุมได้ผ่านทางโปรไฟล์คอนฟิกูเรชัน:

- คอนฟิกูเรชันเครือข่าย
 - no
 - mktcpip
- คอนฟิกูเรชันเคอร์เนล
 - ioo
 - schedo
- คอนฟิกูเรชัน RAS
 - alog
- คอนฟิกูเรชันความปลอดภัย

– setsecattr

ตัวอย่าง

ตัวอย่างต่อไปนี้แสดงโปรไฟล์คอนฟิกูเรชันสำหรับเค็ตตาลีอกและเค็ตตาลีอกย่อยที่ต่างกันซึ่งมีค่าที่กำหนดสำหรับพารามิเตอร์ที่ต่างกัน คุณสามารถแก้ไขโปรไฟล์นี้ด้วยโปรแกรมแก้ไข XML ใดๆ หรือใช้คำสั่ง vi และเปลี่ยนค่าที่มีอยู่สำหรับพารามิเตอร์ที่กำหนดไว้

```
<?xml version="1.0" encoding="UTF-8" ?>
<Profile origin="get" version="1.0" date="2009-04-25T15:33:37Z">
<Catalog id="vmoParam">
<Parameter name="kernel_heap_psize" value="0" applyType="nextboot" reboot="true" />
<Parameter name="maxfree" value="1088" />
</Catalog>
<Catalog id="noParam">
<SubCat id="tcp_network">
<Parameter name="tcp_recvspace" value="16384" />
<Parameter name="tcp_sendspace" value="16384" />
</SubCat>
<SubCat id="general_network">
<Parameter name="use_sndbufpool" value="1" applyType="nextboot" reboot="true" />
</SubCat>
</Catalog>
<Catalog id="lvmoParam">
<Parameter name="max_vg_pbuf_count" value="0">
<Target class="vg" instance="rootvg" />
</Parameter>
<Parameter name="pv_pbuf_count" value="512">
<Target class="vg" instance="rootvg" />
</Parameter>
</Catalog>
```

งานที่เกี่ยวข้อง:

“การแก้ไขโปรไฟล์ AIX Runtime Expert” ในหน้า 76

โปรไฟล์ AIX Runtime Expert คือไฟล์ XML และสามารถแก้ไขได้ด้วยโปรแกรมแก้ไข XML หรือโปรแกรมแก้ไขข้อความใดๆ

“การสร้างโปรไฟล์ AIX Runtime Expert” ในหน้า 75

ใช้ตัวอย่างที่มีอยู่ในไดเรกทอรี /etc/security/artex/samples เพื่อสร้างโปรไฟล์ใหม่ด้วยคำสั่ง artexget โปรไฟล์ตัวอย่างคือเพิ่มเพลตในการสร้างโปรไฟล์ซึ่งคุณสามารถแก้ไขและบันทึกไว้ในไฟล์แบบกำหนดเอง

“การเรียกค่าโปรไฟล์ AIX Runtime Expert” ในหน้า 78

ใช้คำสั่ง artexget เพื่อค้นหาข้อมูลเกี่ยวกับโปรไฟล์

“การใช้โปรไฟล์ AIX Runtime Expert” ในหน้า 79

เมื่อต้องการตั้งค่าระบบที่มีคอนฟิกูเรชันและพารามิเตอร์ที่สามารถปรับเสียงได้จากโปรไฟล์ ให้ใช้โปรไฟล์โดยใช้คำสั่ง artexset

เค็ตตาลีอก AIX Runtime Expert:

เค็ตตาลีอกเป็นกลไกที่กำหนดและระบุตัวควบคุมคอนฟิกูเรชันที่สามารถดำเนินการโดย AIX Runtime Expert

มีการนำเสนอแค็ตตาล็อกสำหรับตัวควบคุมที่ได้รับการสนับสนุนในปัจจุบัน โดย AIX Runtime Expert แค็ตตาล็อก เป็นไฟล์ การกำหนดที่แม่ค่าโปรไฟล์คอนฟิกูเรชันเข้ากับพารามิเตอร์ ที่รันคำสั่งและการดำเนินการคอนฟิกูเรชัน

AIX Runtime Expert นำเสนอแค็ตตาล็อก แบบอ่านอย่างเดียวที่มีอยู่ ซึ่งตั้งอยู่ในไดเรกทอรี /etc/security/artex/ catalogs ที่ระบุค่าที่สามารถแก้ไขได้ อย่าแก้ไข แค็ตตาล็อกเหล่านี้

แต่ละแค็ตตาล็อกประกอบด้วยพารามิเตอร์ต่างๆ สำหรับหนึ่งคอมโพเนนต์ อย่างไรก็ตาม บาง แค็ตตาล็อกสามารถประกอบด้วยพารามิเตอร์จากมากกว่าหนึ่งคอมโพเนนต์ที่เกี่ยวข้องใกล้เคียงกัน ชื่อของแค็ตตาล็อกอธิบายคอมโพเนนต์ที่มีอยู่ใน แค็ตตาล็อก อิลิเมนต์ <description> XML ในแต่ละแค็ตตาล็อกแสดงคำอธิบายของแค็ตตาล็อก

AIX Runtime Expert และ LDAP:

AIX Runtime Expert สามารถดึงโปรไฟล์จากเซิร์ฟเวอร์ Lightweight Directory Access Protocol (LDAP)

โปรไฟล์ AIX Runtime Expert ต้องมีการจัดเก็บเป็นอ็อบเจกต์ ibm-artexProfile และมีแอตทริบิวต์บังคับ ต่อไปนี้:

- Ibm-artexProfileName ชื่อโปรไฟล์ AIX Runtime Expert
- Ibm-artexProfileXMLData เนื้อหา XML ของโปรไฟล์ AIX Runtime Expert ที่จัดเก็บ เป็น octetString

ต้องติดตั้งแบบแผน AIX Runtime Expert บนเซิร์ฟเวอร์ LDAP ก่อนการจัดเก็บโปรไฟล์ AIX Runtime Expert ใดๆ การตั้งค่า เซิร์ฟเวอร์ LDAP สำหรับ AIX Runtime Expert คล้ายกับ การตั้งค่าเซิร์ฟเวอร์ LDAP สำหรับการพิสูจน์ตัวตนผู้ใช้ สำหรับ ข้อมูลเพิ่มเติมเกี่ยวกับการตั้งค่า LDAP ให้ดูที่ การตั้งค่าเซิร์ฟเวอร์ข้อมูลความปลอดภัย ITDS

การตั้งค่าไคลเอ็นต์ LDAP สำหรับ AIX Runtime Expert คล้ายกับ การตั้งค่าไคลเอ็นต์ LDAP สำหรับการพิสูจน์ตัวตนผู้ใช้ สำหรับข้อมูลเพิ่มเติม ให้ดูหัวข้อ การตั้งค่าไคลเอ็นต์ LDAP หากต้องการตั้งค่าไคลเอ็นต์ LDAP ให้ใช้คำสั่ง `mksecldap -c` เพื่อกำหนดคอนฟิก `secldapclntd` daemon AIX Runtime Expert อยู่บน `secldapclntd` daemon เพื่อเข้าถึงเซิร์ฟเวอร์ LDAP ตามค่าดีฟอลต์ AIX Runtime Expert มองหารายการโปรไฟล์ภายใต้ตัวระบุ DN: `ou=artex, cn=AIXDATA` คุณสามารถ กำหนด DN นี้เองโดยการอัปเดตคีย์ `artexbasedn` ในไฟล์คอนฟิกูเรชัน `/etc/security/ldap/ldap.cfg` `secldapclntd`

การอัปโหลดโปรไฟล์ AIX Runtime Expert

หากต้องการอัปโหลด โปรไฟล์ AIX Runtime Expert คุณสามารถสร้างการเปลี่ยนแปลงข้อมูล LDAP ที่จัดรูปแบบไฟล์ (LDIF) และใช้คำสั่ง `ldapadd` หรือใช้เครื่องมือการดูแลระบบ LDAP เช่น Tivoli® Directory Server Web Administration Tool

ต่อไปนี้เป็นตัวอย่าง ของโปรไฟล์ที่ถูกบันทึกไว้ใน LDIF:

```
dn: ou=artex,cn=AIXDATA
objectClass: organizationalUnit
objectClass: top
ou: artex
```

```
dn: ibm-artexProfileName=alogProfile.xml,ou=artex,cn=AIXDATA
objectClass: ibm-artexProfile
objectClass: top
ibm-artexProfileName: alogProfile.xml
ibm-artexProfileXMLData:< file:///etc/security/artex/samples/alogProfile.xml
```

ข้อมูล ต่อไปนี้เป็นตัวอย่างของการอัปโหลดโปรไฟล์โดยใช้คำสั่ง `ldapadd` และตัวอย่างไฟล์ LDIF ที่ชื่อ `sample.ldif`:

```
ldapadd -c -h <ldaphost> -D cn=admin -w <password> -f sample.ldif
```

งานที่เกี่ยวข้อง:

“การสร้างโปรไฟล์ AIX Runtime Expert” ในหน้า 75

ใช้ตัวอย่างที่มีอยู่ในไดเรกทอรี /etc/security/artex/samples เพื่อสร้างโปรไฟล์ใหม่ด้วยคำสั่ง `artexget` โปรไฟล์ตัวอย่างคือเท็มเพลตในการสร้างโปรไฟล์ซึ่ง คุณสามารถแก้ไขและบันทึกไว้ในไฟล์แบบกำหนดเอง

ข้อมูลที่เกี่ยวข้อง:

IBM Security Directory Server

AIX Runtime Expert และ RBAC:

Role Based Access Control (RBAC) สามารถใช้เพื่อช่วยให้ผู้ใช้ที่มีใบรอกมีความสามารถในการสั่งการคำสั่ง AIX Runtime Expert

การอนุญาต AIX Runtime Expert

ในการติดตั้งชุดไฟล์ `artex.base.rte` สามารถอนุญาตของระบบจะถูกสร้างขึ้น ซึ่งช่วยให้สามารถเข้าถึง AIX Runtime Expert functionality ในระดับที่แตกต่างกัน:

- การอนุญาต `aix.system.config.artex.read` ช่วยให้สามารถสั่งการคำสั่ง `artexlist` และ `artexmerge` คำสั่ง `artexget` และ `artexdiff` ได้รับอนุญาต แต่เฉพาะเพื่อให้ได้มาซึ่งค่าโปรไฟล์ ค่าเหล่านี้ไม่สามารถดักจับได้จากระบบ (ซึ่งก็คือคำสั่ง `artexget` ไม่สามารถรันได้ด้วยแฟล็ก `-r`, `-n` หรือ `-p` และคำสั่ง `artexdiff` สามารถดูรันได้เฉพาะระหว่างสองโปรไฟล์)
- การอนุญาต `aix.system.config.artex.get` ช่วยอนุญาตปฏิบัติการที่ได้รับอนุญาตโดยการอนุญาต `artex.system.config.read` และยังอนุญาตให้ดำเนินการสั่งการด้วยคำสั่ง `artexget` และ `artexdiff` แบบไม่จำกัด
- การอนุญาต `aix.system.config.artex.set` ช่วยอนุญาตปฏิบัติการทั้งหมดที่อนุญาตโดยการอนุญาต `artex.system.config.get` และนอกจากนี้ ยังอนุญาตการสั่งการด้วยคำสั่ง `artexset`

บทบาทของ AIX Runtime Expert

AIX Runtime Expert ไม่ได้สร้างบทบาทใหม่ แต่ชุดไฟล์ `artex.base.rte` ทำการเพิ่มการอนุญาต `aix.system.config.artex` ให้แก่บทบาท SysConfig ผู้ใช้รายใดก็ตามที่มีบทบาท SysConfig หรือบทบาทล้อมรอบใดก็ตาม (เช่น บทบาท `isso`) จะสามารถรันคำสั่ง `artexlist`, `artexmerge`, `artexdiff`, `artexget` และ `artexset` ได้

ข้อจำกัด

เนื่องด้วยเหตุผลเพื่อความปลอดภัย การใช้งานตัวแปรสภาพแวดล้อม `ARTEX_CATALOG_PATH` จะถูกจำกัดเฉพาะผู้ใช้รอก ผู้ที่มีผู้ใช้รอกที่ได้รับสิทธิ์ให้สั่งการคำสั่ง AIX Runtime Expert commands ผ่านทาง RBAC จะไม่สามารถใช้ตัวแปรสภาพแวดล้อม `ARTEX_CATALOG_PATH` ได้

การจัดการ AIX Runtime Expert

AIX Runtime Expert ใช้คำสั่ง แบบง่ายสองสามคำสั่งเพื่อสร้างโปรไฟล์ แก้ไขโปรไฟล์ รวมโปรไฟล์ และใช้โปรไฟล์

การกำหนดคอนฟิก AIX Runtime Expert:

AIX Runtime Expert ใช้ไฟล์คอนฟิกูเรชัน /etc/security/artex/config/artex.conf

รายการในไฟล์การกำหนดคอนฟิกประกอบด้วย ชื่อของตัวเลือกการกำหนดคอนฟิก ตามด้วยเว้นวรรคหนึ่งหรือสองเคาะ และค่า บรรทัดว่างและบรรทัดที่เริ่มต้นด้วยเครื่องหมาย # จะถูกละเว้น

ตัวเลือกดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 1. อีพซันการกำหนดคอนฟิก

อีพซัน	คำอธิบาย
ARTEX_CATALOG_PATH	รายการที่ค้นแยกด้วยโคลอนซึ่งแสดงไดเรกทอรีที่ถูกค้นหาไฟล์แค็ตตาล็อก ตัวเลือกนี้ถูกแทนที่โดยตัวแปรสภาพแวดล้อม ARTEX_CATALOG_PATH พาทิฟอร์มคือ /etc/security/artex/ catalogs
ARTEX_PROFILE_PATH	รายการที่ค้นแยกด้วยโคลอนซึ่งแสดงไดเรกทอรีที่ถูกค้นหาไฟล์โปรไฟล์โดย คำสั่ง <code>artexlist</code> หากไม่มีการระบุไดเรกทอรี ตัวเลือกนี้ถูกแทนที่โดยตัวแปร สภาพแวดล้อม ARTEX_PROFILE_PATH พาทิฟอร์มคือ /etc/ security/artex/samples
DEBUG_LOG_CATEGORY	หมวดหมู่ที่บันทึกสำหรับล็อกไฟล์การใช้งาน ตัวเลือกนี้สามารถดำเนินการซ้ำ เพื่อเลือกหลายหมวดหมู่ที่บันทึก
DEBUG_LOG_LEVEL	ระดับที่บันทึกสำหรับล็อกไฟล์การใช้งานระหว่าง 0 (ไม่มีการติดตามที่บันทึก) ถึง 3 (ละเอียดที่สุด)
MAX_CMDS	จำนวนสูงสุดของคำสั่งภายนอกที่สั่งการพร้อมกัน คำสั่งภายนอกเรียกใช้งาน โดย AIX Runtime Expert ถูกจัดคิวไว้ ดังนั้นจึงไม่มี คำสั่งภายนอกมากกว่า ค่า MAX_CMDS ถูกเรียกใช้งานพร้อมกัน ในเวลาใดๆ ดีฟอลต์คือ 10

การสร้างโปรไฟล์ AIX Runtime Expert:

ใช้ตัวอย่างที่มีอยู่ในไดเรกทอรี /etc/security/artex/samples เพื่อสร้างโปรไฟล์ใหม่ด้วยคำสั่ง `artexget` โปรไฟล์ตัวอย่าง
คือเพิ่มเพลตในการสร้างโปรไฟล์ซึ่ง คุณสามารถแก้ไขและบันทึกไว้ในไฟล์แบบกำหนดเอง

เมื่อต้องการสร้างโปรไฟล์ที่มีพารามิเตอร์ทั้งหมดซึ่งสนับสนุนโดย AIX Runtime Expert ให้ทำ ขั้นตอนต่อไปนี้จะสมบูรณ์:

1. กำหนดคอนฟิกและปรับระบบของคุณเพื่อให้มีค่าติดตั้งที่ต้องการ สำหรับโปรไฟล์ใหม่
2. ไปยังไดเรกทอรีตัวอย่าง: /etc/security/artex/samples
3. รันคำสั่งต่อไปนี้เพื่อสร้างโปรไฟล์ใหม่ที่ชื่อ `custom_all.xml`:

```
artexget -p all.xml > /directory_for_new_profile/custom_all.xml
```

หมายเหตุ: โปรไฟล์ `custom_all.xml` สามารถนำไปใช้เพื่อกำหนดคอนฟิกระบบอื่นที่มีคอนฟิกูเรชันระบบปัจจุบัน
คล้ายกัน

เมื่อต้องการสร้างโปรไฟล์สำหรับคอมพิวเตอร์เฉพาะ เช่น อีพซันเครือข่าย ให้ทำขั้นตอนต่อไปนี้จะสมบูรณ์:

1. กำหนดคอนฟิกและปรับระบบของคุณเพื่อให้มีค่าติดตั้งที่ต้องการ สำหรับโปรไฟล์ใหม่
2. ไปยังไดเรกทอรีตัวอย่าง: /etc/security/artex/samples
3. สร้างโปรไฟล์ใหม่ที่ชื่อ `custom_no.xml` จาก โปรไฟล์ตัวอย่างที่มีอยู่ `noProfile.xml` โดยรัน คำสั่งต่อไปนี้:

```
artexget -p noProfile.xml > /directory_for_new_profile/custom_no.xml
```

โปรไฟล์ที่สร้างขึ้นใหม่สามารถมีการกำหนดเองได้โดยเปลี่ยน หรือลบค่าของพารามิเตอร์โดยใช้โปรแกรมแก้ไข XML หรือโปรแกรมแก้ไขข้อความใดๆ

โปรไฟล์แบบกำหนดเองสามารถอัปโหลดไปยังเซิร์ฟเวอร์ LDAP เพื่อใช้จากหลายระบบ AIX เมื่อต้องการอัปโหลดโปรไฟล์ไปยังเซิร์ฟเวอร์ LDAP ให้ใช้เครื่องมือ ที่จัดเตรียมให้โดย LDAP

หลักการที่เกี่ยวข้อง:

“AIX Runtime Expert และ LDAP” ในหน้า 73

AIX Runtime Expert สามารถดึงโปรไฟล์จากเซิร์ฟเวอร์ Lightweight Directory Access Protocol (LDAP)

“โปรไฟล์ AIX Runtime Expert” ในหน้า 71

โปรไฟล์ AIX Runtime Expert ใช้เพื่อตั้งค่าบนระบบที่กำลังรัน แยกค่าของระบบที่กำลังรัน และเปรียบเทียบกับระบบที่กำลังรันหรือกับโปรไฟล์อื่น

งานที่เกี่ยวข้อง:

“การเรียกค่าโปรไฟล์ AIX Runtime Expert” ในหน้า 78

ใช้คำสั่ง `artexget` เพื่อค้นหาข้อมูล เกี่ยวกับโปรไฟล์

“การใช้โปรไฟล์ AIX Runtime Expert” ในหน้า 79

เมื่อต้องการตั้งค่าระบบที่มีคอนฟิกรูชันและพารามิเตอร์ที่สามารถปรับเสียง ได้จากโปรไฟล์ ให้ใช้โปรไฟล์โดยใช้คำสั่ง

`artexset`

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `artexget`

การแก้ไขโปรไฟล์ AIX Runtime Expert:

โปรไฟล์ AIX Runtime Expert คือไฟล์ XML และสามารถแก้ไขได้ด้วยโปรแกรมแก้ไข XML หรือโปรแกรมแก้ไขข้อความใดๆ

โปรไฟล์ที่ผู้ใช้สร้างขึ้นโดยใช้คำสั่ง `artexget` สามารถถูกกำหนดเองได้โดยการเปลี่ยนค่าของพารามิเตอร์ หรือโดยการลบบางพารามิเตอร์ที่ไม่จำเป็นต้องแก้ไขหรือมอนิเตอร์โปรไฟล์

เมื่อต้องการแก้ไขโปรไฟล์ AIX Runtime Expert ให้ทำขั้นตอนต่อไปนี้ให้สมบูรณ์:

1. จากไดเรกทอรีที่ `custom_all.xml` ตั้งอยู่ ให้รันคำสั่งต่อไปนี้เพื่อบันทึกสำเนาของโปรไฟล์:

```
cp custom_all.xml custom_all_backup.xml
```

2. จากไดเรกทอรีที่ `custom_all.xml` ตั้งอยู่ ให้รันคำสั่งต่อไปนี้เพื่อแก้ไขโปรไฟล์:

```
vi custom_all.xml
```

หมายเหตุ: คุณสามารถใช้โปรแกรมแก้ไข XML หรือโปรแกรมแก้ไขข้อความใดๆ

3. แก้ไขค่าของพารามิเตอร์หรือลบพารามิเตอร์ที่ไม่จำเป็นต้องเปลี่ยนหรือมอนิเตอร์โปรไฟล์

4. รันคำสั่งต่อไปนี้เพื่อตรวจสอบว่าการเปลี่ยนแปลงโปรไฟล์ มีการบันทึกไว้แล้วอย่างถูกต้อง โดยการเปรียบเทียบข้อมูลนี้กับค่าติดตั้งระบบปัจจุบัน:

```
artexdiff -c -r custom_all.xml custom_all_backup.xml
```

คำสั่ง `artexdiff` แสดงพารามิเตอร์ที่ถูกแก้ไขโดยโปรแกรมแก้ไข `<FirstValue>` แสดง ค่าของโปรไฟล์ และ `<SecondValue>` แสดง ค่าของระบบปัจจุบัน

หลักการที่เกี่ยวข้อง:

76 AIX เวอร์ชัน 7.2: ระบบปฏิบัติการและการจัดการกับอุปกรณ์

“โปรไฟล์ AIX Runtime Expert” ในหน้า 71

โปรไฟล์ AIX Runtime Expert ใช้เพื่อตั้งค่าระบบที่กำลังรัน แยกค่าของระบบที่กำลังรัน และเปรียบเทียบค่ากับระบบที่กำลังรันหรือกับโปรไฟล์อื่น

งานที่เกี่ยวข้อง:

“การเรียกค่าโปรไฟล์ AIX Runtime Expert” ในหน้า 78

ใช้คำสั่ง `artexget` เพื่อค้นหาข้อมูล เกี่ยวกับโปรไฟล์

“การใช้โปรไฟล์ AIX Runtime Expert” ในหน้า 79

เมื่อต้องการตั้งค่าระบบที่มีคอนพิกูเรชันและพารามิเตอร์ที่สามารถปรับเสียง ได้จากโปรไฟล์ ให้ใช้โปรไฟล์โดยใช้คำสั่ง `artexset`

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `artexdiff`

การรวมโปรไฟล์ AIX Runtime Expert:

โปรไฟล์สามารถแสดงถึงชุดทั้งหมดของตัวควบคุม หรือชุดย่อย ของตัวควบคุม อีกวิธีหนึ่งที่มีประโยชน์ในแก้ไขโปรไฟล์คือการรวม โปรไฟล์ที่แสดงถึงชุดย่อยของตัวควบคุมโดยใช้คำสั่ง `artxmerge`

คุณสามารถใช้คำสั่ง `artxmerge` เพื่อรวมหลายโปรไฟล์ เข้าเป็นโปรไฟล์เดียว

เมื่อต้องการรวมโปรไฟล์ให้ทำขั้นตอนต่อไปนี้อย่างสมบูรณ์:

1. จากไดเรกทอรีซึ่งจัดเก็บโปรไฟล์ให้รันคำสั่ง ต่อไปนี้:

```
artxmerge profile_name1.xml profile_name2.xml > new_profile_name.xml
```

2. รันคำสั่งต่อไปนี้เพื่อดูโปรไฟล์และตรวจสอบว่าเป็นโปรไฟล์ที่ถูกต้องหรือไม่:

```
artexget new_profile_name.xml
```

หมายเหตุ: หากโปรไฟล์ที่คุณกำลังรวมมีพารามิเตอร์ซ้ำ โปรเซสของการรวมโปรไฟล์จะล้มเหลว หรือถ้า คุณใช้แฟล็ก `-f` จะใช้ค่าพารามิเตอร์จาก โปรไฟล์ล่าสุด

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `artxmerge`

การค้นหาโปรไฟล์ AIX Runtime Expert:

ใช้คำสั่ง `artxlist` เพื่อค้นหาโปรไฟล์ในพาร์ที่กำหนดและจากเซิร์ฟเวอร์ LDAP

เมื่อต้องการค้นหาโปรไฟล์ให้ทำขั้นตอนต่อไปนี้อย่างสมบูรณ์:

1. ถ้าโปรไฟล์อยู่บนระบบโลคัล ให้รันคำสั่ง ต่อไปนี้:

```
artxlist
```

2. ถ้าโปรไฟล์อยู่บนเซิร์ฟเวอร์ LDAP ให้รันคำสั่ง ต่อไปนี้:

```
artxlist -l
```

โดยค่าดีฟอลต์ คำสั่งแสดงรายการโปรไฟล์ในไดเรกทอรี /etc/security/artex/samples เมื่อต้องการยกเลิกพารามิเตอร์ที่มีตัวแปรสภาวะแวดล้อม ให้ตั้งค่า ARTEX_PROFILE_PATH เป็นพารามิเตอร์ที่คั่นด้วยเครื่องหมายอัฒภาค อย่างน้อยหนึ่งตัว หรือพารามิเตอร์ที่สามารถส่งผ่านเป็นอาร์กิวเมนต์ได้

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง artexlist

การเรียกค่าโปรไฟล์ AIX Runtime Expert:

ใช้คำสั่ง artexget เพื่อค้นหาข้อมูล เกี่ยวกับโปรไฟล์

โดยใช้โปรไฟล์ คุณสามารถแสดงค่าจากโปรไฟล์ หรือจากระบบในรูปแบบอื่น (XML, CSV, หรือข้อความ) ที่มีตัวกรอง แตกต่างกันได้ เช่น พารามิเตอร์ที่ต้องรีบูตเพื่อให้มีผลบังคับใช้ และ พารามิเตอร์ที่ต้องการเซอร์วิสบางอย่างเพื่อหยุดและรีสตาร์ท

การเรียก ค่าจากระบบมีประโยชน์ในสถานการณ์ต่อไปนี้:

เมื่อต้องการใช้ snapshot ของระบบ

เมื่อระบบมีการกำหนดคอนฟิกอย่างถูกต้อง คุณสามารถบันทึกคอนฟิกูเรชัน ของระบบโดยใช้ snapshot คุณสามารถใช้ snapshot นี้ในภายหลังได้ ถ้าพารามิเตอร์ใดๆ มีการเปลี่ยนแปลงและคุณจำไม่ได้ว่า พารามิเตอร์ใดเปลี่ยนแปลงสามารถใช้โปรไฟล์ snapshot เพื่อ นำระบบกลับไปยังคอนฟิกูเรชันที่ต้องการได้

เมื่อต้องการลอกแบบคอนฟิกูเรชันของระบบเพื่อใช้บนระบบอื่น

หลังจากกำหนดคอนฟิกและปรับระบบในสภาวะแวดล้อมแล้ว คุณสามารถแยกค่าติดตั้งระบบเข้าในโปรไฟล์ AIX Runtime Expert และใช้โปรไฟล์บนระบบอื่นได้

เมื่อต้องการดีบั๊กปัญหา

เมื่อพบปัญหาบนระบบการผลิต คุณสามารถใช้ โปรไฟล์เพื่อตั้งค่าติดตั้งระบบเดียวกันบนระบบทดสอบ จากนั้นดีบั๊กปัญหาบนระบบทดสอบ

เมื่อต้องการเรียกข้อมูลเกี่ยวกับโปรไฟล์ ให้ทำ ขั้นตอนต่อไปนี้ให้สมบูรณ์:

1. ไปยังไดเรกทอรีที่จัดเก็บโปรไฟล์ซึ่งคุณต้องการ เรียกข้อมูล
2. เมื่อต้องการเรียกข้อมูลเกี่ยวกับโปรไฟล์ ให้รันคำสั่ง ต่อไปนี้:

```
artexget name_of_profile.xml
```

ข้อจำกัด: เมื่อระบบมีผู้ใช้หลายคนถูกกำหนดไว้ คำสั่ง AIX Runtime Expert artexget, artexset, และ artexdiff ใช้กับโปรไฟล์ เช่น chuserProfile.xml, coreProfile.xml หรือ all.xml profiles ต้องใช้เวลาในการทำให้เสร็จสมบูรณ์ มากกว่าปกติ

หลักการที่เกี่ยวข้อง:

“โปรไฟล์ AIX Runtime Expert” ในหน้า 71

โปรไฟล์ AIX Runtime Expert ใช้เพื่อตั้งค่าบนระบบที่กำลังรัน แยกค่าของระบบที่กำลังรัน และเปรียบเทียบค่ากับระบบที่กำลังรันหรือกับโปรไฟล์อื่น

งานที่เกี่ยวข้อง:

“การสร้างโปรไฟล์ AIX Runtime Expert” ในหน้า 75

ใช้ตัวอย่างที่มีอยู่ในไดเรกทอรี /etc/security/artex/samples เพื่อสร้างโปรไฟล์ใหม่ด้วยคำสั่ง artexget โปรไฟล์ตัวอย่างคือเพิ่มเพลตในการสร้างโปรไฟล์ซึ่ง คุณสามารถแก้ไขและบันทึกไว้ในไฟล์แบบกำหนดเอง

“การแก้ไขโปรไฟล์ AIX Runtime Expert” ในหน้า 76

โปรไฟล์ AIX Runtime Expert คือไฟล์ XML และสามารถแก้ไขได้ด้วยโปรแกรมแก้ไข XML หรือโปรแกรมแก้ไขข้อความใดๆ

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `artexget`

การใช้โปรไฟล์ AIX Runtime Expert:

เมื่อต้องการตั้งค่าระบบที่มีคอนฟิกูเรชันและพารามิเตอร์ที่สามารถปรับเสียง ได้จากโปรไฟล์ ให้ใช้โปรไฟล์โดยใช้คำสั่ง `artexset`

เมื่อต้องการใช้โปรไฟล์ที่ผู้ใช้สร้างขึ้น ให้ทำขั้นตอนต่อไปนี้จะสมบูรณ์:

1. ไปยังไดเรกทอรีที่จัดเก็บโปรไฟล์ซึ่งคุณต้องการใช้

2. เมื่อต้องการใช้โปรไฟล์กับระบบ ให้รันคำสั่งต่อไปนี้:

```
artexset -c name_of_profile.xml
```

3. ทางเลือก: ถ้าคุณต้องการใช้โปรไฟล์ในทุกครั้งที่ระบบ รีสตาร์ทเพื่อรักษาคอนฟิกูเรชันที่สอดคล้องกัน ให้รันคำสั่งต่อไปนี้:

```
artexset -b name_of_profile.xml
```

หมายเหตุ: พารามิเตอร์ที่จำกัดได้รับการสนับสนุนให้เป็นพารามิเตอร์แบบอ่านอย่างเดียว ดังนั้น ค่าของพารามิเตอร์เหล่านี้สามารถตั้งด้วยคำสั่ง `artexget` แต่สามารถตั้งค่าด้วยคำสั่ง `artexset` ได้

หลักการที่เกี่ยวข้อง:

“โปรไฟล์ AIX Runtime Expert” ในหน้า 71

โปรไฟล์ AIX Runtime Expert ใช้เพื่อตั้งค่าระบบที่กำลังรัน แยกค่าของระบบที่กำลังรัน และเปรียบเทียบค่ากับระบบที่กำลังรันหรือกับโปรไฟล์อื่น

งานที่เกี่ยวข้อง:

“การสร้างโปรไฟล์ AIX Runtime Expert” ในหน้า 75

ใช้ตัวอย่างที่มีอยู่ในไดเรกทอรี `/etc/security/artex/samples` เพื่อสร้างโปรไฟล์ใหม่ด้วยคำสั่ง `artexget` โปรไฟล์ตัวอย่างคือเพิ่มเพลตในการสร้างโปรไฟล์ซึ่ง คุณสามารถแก้ไขและบันทึกไว้ในไฟล์แบบกำหนดเอง

“การแก้ไขโปรไฟล์ AIX Runtime Expert” ในหน้า 76

โปรไฟล์ AIX Runtime Expert คือไฟล์ XML และสามารถแก้ไขได้ด้วยโปรแกรมแก้ไข XML หรือโปรแกรมแก้ไขข้อความใดๆ

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `artexset`

Rolling back AIX Runtime Expert profiles:

ใช้คำสั่ง `artexset -u` เพื่อรีเซ็ต ค่าติดตั้งคอนฟิกูเรชันเป็นค่าติดตั้งคอนฟิกูเรชันก่อนหน้าของ ระบบ คุณสามารถใช้ค่าติดตั้งระบบที่เคยใช้ก่อนการนำ โปรไฟล์มาใช้ได้

คุณไม่สามารถใช้คำสั่ง `rollback` ได้ หาก คุณไม่เปลี่ยนค่าติดตั้งระบบระหว่างเซสชันปัจจุบันของคุณ

การ Rolling back ไม่ได้ถือว่าเป็นการทำอิมเมจของระบบปฏิบัติการใหม่ เมื่อคุณใช้คำสั่ง rollback คุณไม่ได้ลบ หรือสร้างรีซอร์ส แต่คุณได้คืนค่าการกำหนดคอนฟิกที่เก่าแก่กลับ เป็นค่าที่ตั้งก่อนหน้าของระบบ ด้วยคำสั่ง rollback คุณไม่สามารถ roll back ค่าที่ตั้งแบบเจาะจงเวลาหรือวันที่ คุณสามารถทำได้เพียง roll back กลับเป็นค่าที่ตั้งก่อนหน้าก่อนที่จะทำการเปลี่ยนแปลง

คำสั่ง rollback สามารถใช้ได้ในกรณีต่อไปนี้:

- การทดสอบการเปลี่ยนแปลงคอนฟิกูเรชันในระบบ หากคอนฟิกูเรชันใหม่ทำงาน ได้ไม่ดี คุณสามารถแปลงเป็นคอนฟิกูเรชันก่อนหน้าทันทีได้อย่างรวดเร็ว
- การดีบักระบบ ถ้าระบบเริ่มรันไม่ค่อยดี การ roll back สามารถช่วยยืนยันได้หากการเปลี่ยนแปลงการกำหนดคอนฟิกมีส่วนเกี่ยวข้องกับปัญหาใหม่ที่เกิดขึ้น
- การนำโปรไฟล์ใหม่ไปใช้เพื่อให้เป็นไปตามสถานการณ์ย่อยกเว้นพิเศษ บางสถานการณ์ ตัวอย่างเช่น การดำเนินการที่ระบุ จะเกิดขึ้นเพียงหนึ่งครั้งต่อเดือนเท่านั้น บนระบบ และหลังจากที่ใช้กับระบบของคุณแล้ว คุณต้องการ เรียกคืนคอนฟิกูเรชันก่อนหน้าของระบบ

เมื่อต้องการ roll back กลับเป็นค่าที่ตั้งก่อนหน้าของระบบ ให้ทำตาม ขั้นตอนต่อไปนี้:

1. เมื่อต้องการ roll back โปรไฟล์ ให้รันคำสั่งต่อไปนี้:

```
artexset -u
```

2. เมื่อต้องการตรวจสอบว่าการ roll back ถูกต้องสมบูรณ์หรือไม่ ให้รัน คำสั่งต่อไปนี้เพื่อเปรียบเทียบค่าที่ตั้งระบบ:

```
artexdiff -f txt -r -profile_name.xml
```

หมายเหตุ: *profile_name.xml* เป็นชื่อของโปรไฟล์ที่ใช้ล่าสุดกับระบบ

ความแตกต่าง ระหว่างระบบและโปรไฟล์แสดงขึ้น

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง artexget

คำสั่ง artexlist

การเปรียบเทียบโปรไฟล์ AIX Runtime Expert:

ใช้คำสั่ง `artexdiff` เพื่อเปรียบเทียบ สองโปรไฟล์หรือค่าโปรไฟล์กับค่าระบบ

เมื่อต้องการเปรียบเทียบโปรไฟล์ของสองระบบที่แตกต่างกัน ให้ทำ ขั้นตอนต่อไปนี้ให้สมบูรณ์:

1. รันคำสั่งต่อไปนี้จากระบบ 1:

```
artexget -p all.xml > all_system1.xml
```

2. รันคำสั่งต่อไปนี้จากระบบ 2:

```
artexget -p all.xml > all_system2.xml
```

เมื่อต้องการตรวจสอบการเปลี่ยนแปลงพารามิเตอร์คอนฟิกูเรชันบนระบบ หลังจากระยะเวลาหนึ่ง ตัวอย่างเช่น ถ้าคุณหยุดงานและต้องการตรวจสอบ การเปลี่ยนแปลงใดๆ ในขณะที่คุณไม่อยู่ ให้รันคำสั่งต่อไปนี้:

- หลังจากคุณกลับมาหลังจากหยุดงาน ให้รันคำสั่งต่อไปนี้:

```
$ artexget -p all.xml > all_before_vacation.xml
```

- เมื่อต้องการดูการเปลี่ยนแปลงคอนฟิกูเรชันที่เกิดขึ้นในระหว่างที่คุณหยุดงาน ให้รันคำสั่งต่อไปนี้:

```
$ artexdiff -c -p all_before_vacation.xml
```

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง artexget

คำสั่ง artexlist

การเขียน AIX Runtime Expert โพรไฟล์

คุณสามารถขยายขอบเขตของ AIX Runtime Expert โดยการเพิ่มแค็ตตาล็อกและโพรไฟล์ที่โปรแกรมสามารถใช้ คุณต้องคุ้นเคยกับ AIX Runtime Expert แนวคิดก่อนที่จะพยายามเขียนแค็ตตาล็อกใหม่

ข้อมูลขั้นเล็กที่สุดที่จัดการโดย AIX Runtime Expert คือพารามิเตอร์ พารามิเตอร์สามารถปรับได้ ไฟล์การกำหนดคอนฟิกตัวแปรสภาพแวดล้อม คุณสมบัติของอ็อบเจกต์ เช่น ผู้ใช้ อุปกรณ์ หรือระบบย่อย (อ็อบเจกต์เหล่านี้ ถูกเรียกว่า เป้าหมายใน AIX Runtime Expert บริบท)

พารามิเตอร์ถูกรวมเข้าไว้ในโพรไฟล์ตามโดเมน ของกิจกรรม (เช่น user, tcpi) โพรไฟล์เป็นสื่อการโต้ตอบ ระหว่างผู้ใช้กับเฟรมเวิร์ก AIX Runtime Expert โพรไฟล์ต่างๆ ถูกป้อนไปยังคำสั่ง **artexget** ซึ่งเรียกใช้ ค่าพารามิเตอร์บนระบบและแสดงผลโพรไฟล์ โพรไฟล์ต่างๆ (รวมค่า ต่างๆ) เป็นอินพุตสำหรับคำสั่ง **artexset** ซึ่ง ตั้งค่าพารามิเตอร์ให้เป็นค่าที่ได้จากโพรไฟล์

แนวคิดในการเขียนโพรไฟล์ AIX Runtime Expert:

AIX Runtime Expert โพรไฟล์คือไฟล์ XML ที่ประกอบด้วยรายการพารามิเตอร์การกำหนดคอนฟิกและยังมีค่าพารามิเตอร์และแฟล็กการใช้งานเป็นทางเลือก

โพรไฟล์สามารถจัดตำแหน่งไว้ยู่บนระบบที่ถูกปรับเมื่อใช้คำสั่ง AIX Runtime Expert โดยตรงบนบรรทัดคำสั่ง.

ตำแหน่งของโพรไฟล์:

โพรไฟล์ตัวอย่าง AIX Runtime Expert มีตำแหน่งอยู่ในไดเรกทอรี `/etc/security/artex/samples`

เมื่อคุณเขียนแค็ตตาล็อกใหม่สำหรับ AIX Runtime Expert ให้สนับสนุน ขอแนะนำให้คุณเขียนโพรไฟล์ตัวอย่างที่สามารถใช้เป็นรายการสำหรับคำสั่ง **artexget** โพรไฟล์ตัวอย่างเป็นโพรไฟล์แบบอ่านอย่างเดียวโดยที่ไม่มีมออบหมายค่าให้แก่พารามิเตอร์ โพรไฟล์ตัวอย่างที่มีอยู่มีตำแหน่งอยู่ในไดเรกทอรี `/etc/security/artex/samples` ตามค่าดีฟอลต์แล้ว คำสั่ง **artexlist** จะแสดงเฉพาะโพรไฟล์ที่มีตำแหน่งอยู่ในไดเรกทอรีดีฟอลต์ แต่ไดเรกทอรีดีฟอลต์อาจได้รับการแก้ไขได้โดยการตั้งค่าตัวแปรสภาพแวดล้อม **ARTEX_PROFILE_PATH** หลายไดเรกทอรีสามารถระบุได้ในตัวแปรของสภาพแวดล้อมนี้โดยใช้เครื่องหมายค้น :

โพรไฟล์ทั้งหมดจากไดเรกทอรีตัวอย่างถูกผสมผสานเข้าด้วยกันในระหว่างการติดตั้งชุดไฟล์ **artex.base.samples** เพื่อสร้างโพรไฟล์ **default.xml** ที่ใช้งานโดยคำสั่ง **snap** โพรไฟล์ซึ่งไม่ควรเป็นส่วนหนึ่งของโพรไฟล์ **default.xml** ไม่ควรถูกจัดส่งในไดเรกทอรีตัวอย่าง ตัวอย่างของโพรไฟล์ที่ไม่ควรรวมอยู่ในโพรไฟล์ **default.xml** คือโพรไฟล์ที่มีศักยภาพในการรวมหลายพารามิเตอร์ (เช่น หากใช้ผู้ใช้เป็นคลาสเป้าหมาย) และโพรไฟล์ซึ่งควรถูกรันเฉพาะบนระบบเฉพาะ (เช่น โพรไฟล์แอดมินิวิวด์ **vios**)

การตั้งชื่อโพรไฟล์:

โพรไฟล์ AIX Runtime Expert ถูกตั้งชื่อโดยการใช้คำสั่งต่างๆ

ตามปกติแล้ว โพรไฟล์จะถูกสร้างขึ้นด้วยการใช้คำสั่งเดี่ยวหรือชุดคำสั่ง โพรไฟล์อาจประกอบด้วยหลายแค็ตตาล็อก หากแค็ตตาล็อกเหล่านั้นมีความสัมพันธ์กันอย่างใกล้ชิด วิธีปฏิบัติปกติคือ ตั้งชื่อไฟล์โดยใช้คำสั่ง `commandProfile.xml` สำหรับโพรไฟล์ตัวอย่างและ `commandParam.xml` สำหรับแค็ตตาล็อก แต่ไม่เป็นการบังคับ บังคับเฉพาะส่วนขยายไฟล์.xml

กระบวนการโพรไฟล์:

อธิบายถึงกระบวนการในการเขียนโพรไฟล์ AIX Runtime Expert ใหม่

จำเป็นต้องดำเนินการขั้นตอนต่อไปนี้เมื่อเขียนโพรไฟล์ AIX Runtime Expert ใหม่:

1. สร้างพารามิเตอร์ที่คุณต้องการในโพรไฟล์
2. สร้างอิลิเมนต์ `<Parameter name="...">` สำหรับแต่ละพารามิเตอร์ ตั้งค่าแอตทริบิวต์ `name` ตามชื่อที่ใช้ในอิลิเมนต์ `<ParameterDef>` ของโพรไฟล์แค็ตตาล็อก
3. จัดกลุ่มพารามิเตอร์ทั้งหมดที่กำหนดในไฟล์แค็ตตาล็อกเดียวกันภายในอิลิเมนต์ `<Catalog id="...">` เดียวกัน ตั้งค่าแอตทริบิวต์ `id` ตาม `id` เดียวกันที่ใช้ในอิลิเมนต์ `<Catalog>` ของไฟล์แค็ตตาล็อก
4. สำหรับแต่ละอิลิเมนต์ `<Parameter>` ให้ดำเนินการดังต่อไปนี้:
 - a. หากพารามิเตอร์ถูกกำหนดด้วย `reboot=true` ในไฟล์แค็ตตาล็อก ให้เพิ่มแอตทริบิวต์ `reboot=true` และ `applyType=nextboot`
 - b. หากพารามิเตอร์ต้องถูกดักจับเท่านั้นและไม่ถูกตั้งค่า ให้เพิ่มแอตทริบิวต์ `readOnly=true`
 - c. หากพารามิเตอร์ถูกกำหนดด้วยแอตทริบิวต์ `targetClass` ที่ไม่วางในไฟล์แค็ตตาล็อก ให้ดำเนินการดังต่อไปนี้:
 - 1) หากต้องการให้มีการค้นหาเป้าหมายสำหรับพารามิเตอร์นี้ ให้กำหนดอิลิเมนต์ `<Parameter>` เดียวสำหรับพารามิเตอร์นี้ และใช้เป้าหมาย `<Target class="" instance="">` พิเศษสำหรับอิลิเมนต์นี้
 - 2) หากเป้าหมายเฉพาะจำเป็นต้องได้รับการกำหนดสำหรับพารามิเตอร์นี้ ให้กำหนดหนึ่งอิลิเมนต์ `<Parameter>` สำหรับแต่ละเป้าหมาย ภายใต้แต่ละอิลิเมนต์ `<Parameter>` ให้กำหนดอิลิเมนต์ `<Target class="..." instance="..." />` เพื่อระบุเป้าหมายอย่างสมบูรณ์
5. ทดสอบโพรไฟล์โดยรันคำสั่ง `artexget -r`

AIX Runtime Expert อิลิเมนต์โพรไฟล์:

อิลิเมนต์ `<Profile>`:

อิลิเมนต์ `<Profile>` เป็นอิลิเมนต์รูทสำหรับไฟล์โพรไฟล์ทั้งหมด

ไวยากรณ์

แอตทริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 2. แอตทริบิวต์

แอตทริบิวต์	บังคับ	ชนิด	คำอธิบาย
<code>origin</code>	ไม่ใช่	สตริง	ที่มาของโพรไฟล์
<code>date</code>	ไม่ใช่	dateTime	วันที่สร้างหรือการแก้ไขโพรไฟล์ครั้งล่าสุด รูปแบบคือ YYYY-MM-DDThh:mm:ss.

ตารางที่ 2. แอ็ททริบิวต์ (ต่อ)

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
<i>readOnly</i>	ไม่ใช่	บูลีน	บอกว่าโปรไฟล์นี้สามารถใช้ในปฏิบัติการ set ได้หรือไม่ ค่าดีฟอลต์คือ false
<i>version</i>	ไม่ใช่	สตริง	หมายเลขเวอร์ชันของโปรไฟล์

อิลิเมนต์ข้างนี้ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 3. อิลิเมนต์ขายน

อิลิเมนต์ขายน	บังคับ	ตัวเลข	คำอธิบาย
<ShortDescription>	ไม่ใช่	0 - 1	คำอธิบายข้อความแบบสั้นของแค็ตตาล็อก
<Description>	ไม่ใช่	0 - 1	คำอธิบายข้อความแบบยาวของแค็ตตาล็อก
<Comments>	ไม่ใช่	0 - 1	คอมเมนต์ของผู้ใช้
<Catalog>	ไม่ใช่	0 - ค่าใดๆ ก็ตาม	แค็ตตาล็อกที่กำหนดเพื่อใช้จัดการปฏิบัติการต่างๆ บนโปรไฟล์

แอ็ททริบิวต์

แอ็ททริบิวต์ *origin*

แอ็ททริบิวต์ *origin* เป็นแอ็ททริบิวต์ให้ข้อมูลที่สามารถได้รับการกำหนดค่าดังต่อไปนี้ได้:

- เมื่อสร้างโปรไฟล์ตัวอย่าง แอ็ททริบิวต์ *origin* จะต้องถูกตั้งค่าเป็น reference
- เมื่อโปรไฟล์ถูกสร้างขึ้นโดยใช้คำสั่ง `artexget` แอ็ททริบิวต์ *origin* จะถูกตั้งค่าอัตโนมัติเป็น get

อิลิเมนต์ขายน

อิลิเมนต์ <Comments> เป็นสตริงทางเลือกที่สงวนไว้สำหรับวัตถุประสงค์อื่นๆ อิลิเมนต์นี้ต้องถูกใช้เมื่อโปรไฟล์ถูกสร้างขึ้นมาด้วยตนเอง และไม่ถูกใช้โดยคำสั่ง AIX Runtime Expert พื้นฐาน

ตัวอย่าง

- โปรไฟล์ตัวอย่างว่างจะมีลักษณะเหมือนเช่นดังต่อไปนี้:

```
<?xml version="1.0" encoding="UTF-8"?>
<Profile origin="reference" version="2.0.0" readOnly="true">
</Profile>
```

- คำสั่ง `artexget -r /etc/security/artex/samples/smtctmProfile.xml` จะให้เอาต์พุตโปรไฟล์ที่คล้ายกับตัวอย่างดังต่อไปนี้:

```
<?xml version="1.0" encoding="UTF-8"?>
<Profile origin="get" version="2.0.1" date="2010-09-29T07:50:56Z">
  <Catalog id="smtctlParam" version="2.0">
    <Parameter name="enableSMT" value="1"/>
  </Catalog>
</Profile>
```

ข้อมูลที่เกี่ยวข้อง

อิลิเมนต์ <Catalog>

อิลิเมนต์ <Description> and <ShortDescription>

อิลิเมนต์ <Description> and <ShortDescription>:

อิลิเมนต์ <Description> และ <ShortDescription> สามารถนำมาใช้เพื่อจัดให้มีคำอธิบายข้อความสำหรับโปรไฟล์และพารามิเตอร์

ไวยากรณ์

อิลิเมนต์พารามิเตอร์ของ <ShortDescription> คือ:

- อิลิเมนต์ <Profile>

อิลิเมนต์พารามิเตอร์ของ <Description> สามารถเป็น:

- อิลิเมนต์ <Profile>
- อิลิเมนต์ <Parameter>

ทั้งอิลิเมนต์ <Description> และ <ShortDescription> มีรูปแบบเดียวกัน ข้อความที่มีอยู่ในอิลิเมนต์ <Description> เป็นเนื้อหาของแท้จริงของแท็ก XML

การใช้งาน

คำอธิบายในไฟล์โปรไฟล์ไม่ได้ถูกใช้โดยเฟรมเวิร์ก AIX Runtime Expert ในขณะนี้ คำสั่ง AIX Runtime Expert ละเว้นคอมเมนต์ที่รวมอยู่ในโปรไฟล์อื่นๆ

ตัวอย่าง

ต่อไปนี้เป็นตัวอย่างของอิลิเมนต์ <Description> และ <ShortDescription>:

```
<ShortDescription>
  Short summary of field contents.
</ShortDescription>
<Description>
This text field can be used to display in full detail the use of the parent element.
</Description>
```

ข้อมูลที่เกี่ยวข้อง

อิลิเมนต์ <Profile>

อิลิเมนต์ <Parameter>

อิลิเมนต์ <Catalog>:

อิลิเมนต์ <Catalog> ระบุชื่อของไฟล์แค็ตตาล็อกที่มีนิยามสำหรับอิลิเมนต์ <Parameter>

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <Profile>

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 4. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	รายละเอียด
<i>id</i>	ใช่	สตริง	ระบุ <i>id</i> แค็ตตาล็อก ซึ่งนี้ควรเป็นค่าเฉพาะบนระบบ
<i>version</i>	ไม่ใช่	สตริง	ระบุเวอร์ชันของแค็ตตาล็อกที่ใช้เพื่อสร้างโปรไฟล์นี้

อิลิเมนต์ชายนัดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 5. อิลิเมนต์ชายน

อิลิเมนต์ชายน	บังคับ	ตัวเลข	คำอธิบาย
<Parameter>	ไม่ใช่	0 - ค่าใดๆก็ตาม	พารามิเตอร์ที่รวมในแค็ตตาล็อก
<SubCat>	ไม่ใช่	0 - ค่าใดๆก็ตาม	หมวดหมู่ย่อยที่รวมอยู่ในแค็ตตาล็อก
<Seed>	ไม่ใช่	0 - ค่าใดๆก็ตาม	Seed ที่รวมในแค็ตตาล็อก

แอ็ททริบิวต์

แอ็ททริบิวต์ *id*

แอ็ททริบิวต์ *id* ต้องถูกตั้งค่าเป็นชื่อของแค็ตตาล็อกที่กำหนดพารามิเตอร์ซึ่งแสดงรายการอยู่ภายใต้อิลิเมนต์ <Catalog> แอ็ททริบิวต์ *id* เป็นชื่อพื้นฐานของไฟล์แค็ตตาล็อกบนดิสก์ ซึ่งมีส่วนขยาย .xml ที่ถูกลบออก ตัวอย่างเช่น โปรไฟล์จะใช้อิลิเมนต์ <Catalog id="commandParam"> เพื่ออ้างอิงไฟล์แค็ตตาล็อก commandParam.xml

ตามค่าดีฟอลต์ ไฟล์แค็ตตาล็อกถูกค้นหาภายใต้ไดเรกทอรี /etc/security/artex/catalogs อย่างไรก็ตาม ความเป็นไปได้สำหรับผู้ใช้รากเท่านั้น ที่จะค้นไดเรกทอรีอื่นโดยการตั้งค่าตัวแปรสภาพแวดล้อม ARTEX_CATALOG_PATH หลายไดเรกทอรีสามารถระบุในตัวแปรสภาพแวดล้อมนี้โดยใช้เครื่องหมาย :

แอ็ททริบิวต์ *version*

แอ็ททริบิวต์ *version* ถูกเขียนเป็น *MM.mm* โดยที่ *MM* เป็นหมายเลขหลัก และ *mm* เป็นหมายเลขรอง

The *version* attribute must match the version of the referenced catalog file (see The <Catalog> element in the section Writing AIX Runtime Expert catalogs). หากคำสั่ง AIX Runtime Expert ถูกรันบนโปรไฟล์ที่อ้างอิงแค็ตตาล็อกที่มีเวอร์ชันไม่ถูกต้อง ข้อความเตือนต่อไปนี้จะแสดงขึ้น:

```
0590-218 เวอร์ชันแค็ตตาล็อกแตกต่างจากที่อ้างอิงในโปรไฟล์
เวอร์ชันของโลคัลแค็ตตาล็อกคือ '2.1' เวอร์ชันที่ใช้เพื่อสร้างโปรไฟล์คือ '2.0'
```

การใช้งาน

อิลิเมนต์ <Catalog> ระบุไฟล์ไฟล์ที่มีนิยามของ seed และพารามิเตอร์ที่แสดงรายการ อิลิเมนต์ seeds และพารามิเตอร์ทั้งหมดในโปรไฟล์ ต้องอยู่ในอิลิเมนต์ <Catalog> ที่เหมาะสม

โปรไฟล์อ้างอิงหลายแค็ตตาล็อก ตัวอย่างเช่น โปรไฟล์ default.xml ถูกสร้างระหว่างการติดตั้ง artex.base.sample fileset โดยการผสมชุดการเลือกของโปรไฟล์ตัวอย่างอื่นๆ

ตัวอย่าง

โปรไฟล์แอ็ททริบิวต์ความปลอดภัย secattrProfile.xml ใช้สามแค็ตตาล็อกซึ่งแต่ละอันจัดการกับแค็ตตาล็อก หนึ่งในตารางความปลอดภัยมีลักษณะดังต่อไปนี้:

```
<Profile origin="reference" readOnly="true" version="2.0.0">
  <Catalog id="privcmdParam" version="2.0">
    <Parameter name="privatecommands" />
  </Catalog>
  <Catalog id="privdevParam" version="2.0">
    <Parameter name="privatedevices"/>
  </Catalog>
  <Catalog id="privfileParam" version="2.0">
    <Parameter name="privatefiles" />
  </Catalog>
</Profile>
```

ข้อมูลที่เกี่ยวข้อง

อิลิเมนต์ <Catalog> (ในไฟล์แค็ตตาล็อก)

อิลิเมนต์ <SubCat>:

อิลิเมนต์ <SubCat> จัดให้มีวิธีการในการสร้างหมวดหมู่ย่อยโลจิคัลภายใต้อิลิเมนต์ <Catalog>

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <Catalog>, <SubCat>

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 6. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
id	ใช่	สตริง	ระบุชื่อของหมวดหมู่ย่อย ซึ่งนี้ควรจะ ไม่ซ้ำกันภายในอิลิเมนต์ <Catalog>

อิลิเมนต์ชายันดังต่อไปนี้ได้รับการสนับสนุน

ตารางที่ 7. อิลิเมนต์ชายัน

อิลิเมนต์ชายัน	บังคับ	ตัวเลข	คำอธิบาย
<Parameter>	ไม่ใช่	0 - ค่าใดๆก็ตาม	มีชื่อพารามิเตอร์
<SubCat>	ไม่ใช่	0 - ค่าใดๆก็ตาม	หมวดหมู่ย่อยแบบซ้อนกัน

แอ็ททริบิวต์

แอ็ททริบิวต์ id ระบุหมวดหมู่ย่อยภายในแค็ตตาล็อก โปรไฟล์อาจมีหลายหมวดหมู่ย่อยที่มี id เดียวกัน หากไม่ได้ถูกใช้ภายใน
อิลิเมนต์ <Catalog> เดียวกัน

อิลิเมนต์ชายัน

อิลิเมนต์ <SubCat> อาจมีอีกหนึ่ง <SubCat> เป็นอิลิเมนต์ชายัน ทั้งนี้ไม่มีการจำกัดจำนวนหมวดหมู่ย่อยแบบซ้อนกันที่คุณ
สามารถกำหนดได้

การใช้งาน

หมวดหมู่ย่อยถูกรวมเฉพาะเพื่อความสามารถในการอ่าน หมวดหมู่เหล่านี้จะไม่ส่งผลกระทบต่อวิธีการที่พารามิเตอร์ได้รับการ
การจัดการ

ตัวอย่าง

โปรไฟล์ noProfile.xml รวมหลายหมวดหมู่ย่อยไว้ด้วยกัน ต่อไปนี้เป็นตัวอย่าง:

```
<Profile origin="reference" readOnly="true" version="2.0.0">
  <Catalog id="noParam" version="2.0">
    <SubCat id="general_network"
      <Parameter name="fasttimo"/>
      <Parameter name="nbc_limit"/>
    </SubCat>
    <SubCat id="tcp_network">
      <Parameter name="clean_partial_conns"/>
      <Parameter name="delayack"/>
    </SubCat>
    <SubCat id="restricted">
      <Parameter name="extendednetstats" readOnly="true"/>
      <Parameter name="inet_stack_size" readOnly="true"/>
    </SubCat>
  </Catalog>
</Profile>
```

ข้อมูลที่เกี่ยวข้อง

อิลิเมนต์ <Parameter>

อิลิเมนต์ <Parameter>:

อิลิเมนต์ <Parameter> เป็นตัวกำหนดพารามิเตอร์ configuration

ไวยากรณ์

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 8. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
<i>name</i>	ใช่	สตริง	ระบุนามของพารามิเตอร์ ชื่อนี้ต้องไม่มีการซ้ำกันต่อแค็ตตาล็อก
<i>value</i>	ไม่ใช่	สตริง	ค่าของพารามิเตอร์ หากถูกกำหนดขึ้น
<i>applyType</i>	ไม่ใช่	สตริง	ระบุว่าจะรันใหม่หรือค่าบุทครั้งถัดไปของพารามิเตอร์ต้องถูกเรียกใช้หรือตั้งค่า หากไม่มีการระบุแฟล็ก ค่าดีฟอลต์คือ runtime
<i>reboot</i>	ไม่ใช่	บูลีน	เมื่อเป็น true แสดงว่าจำเป็นต้องรีบูทก่อน การเปลี่ยนแปลงจึงจะแสดงผล ค่าดีฟอลต์คือ false
<i>readOnly</i>	ไม่ใช่	บูลีน	ระบุว่าค่าของพารามิเตอร์ไม่สามารถกำหนดได้ ค่าดีฟอลต์: ค่าเป็น false
<i>disruptive</i>	ไม่ใช่	บูลีน	ระบุว่าวิธีการที่ใช้เพื่อตั้งค่าพารามิเตอร์แสดงถึงข้อบังคับที่เสียหายหรือไม่ ค่าดีฟอลต์คือ false
<i>setDiscover</i>	ไม่ใช่	บูลีน	ระบุว่าวิธีการที่ตั้งค่าต้องถูกตั้งค่าเป็นค่าสำหรับอินสแตนซ์ที่ค้นพบทั้งหมดของคลาสเป้าหมาย ค่าดีฟอลต์คือ false

อิลิเมนต์ชายน้ดังต่อไปนี้ได้รับการสนับสนุน

ตารางที่ 9. อิลิเมนต์ชายน้

อิลิเมนต์ชายน้	บังคับ	ตัวเลข	คำอธิบาย
<Value>	ไม่ใช่	0 - 1	ค่าของพารามิเตอร์
<Target>	ไม่ใช่	0 - ค่าใดๆ ก็ตาม	เป้าหมายที่พารามิเตอร์ใช้
<Description>	ไม่ใช่	0 - 1	คำอธิบายพารามิเตอร์
<คุณสมบัติ>	ไม่ใช่	0 - ค่าใดๆ ก็ตาม	คุณสมบัติของพารามิเตอร์

แอ็ตทริบิวต์

ตารางที่ 10. แอ็ตทริบิวต์

แอ็ตทริบิวต์	คำอธิบาย
<i>name</i>	ชื่อของพารามิเตอร์เป็นแอ็ตทริบิวต์บังคับเพียงอย่างเดียวของอิลิเมนต์ <code><Parameter></code> พร้อมๆ กับชื่อแค็ตตาล็อกที่ระบุในอิลิเมนต์พารেন্ট <code><Catalog></code> ชื่อพารามิเตอร์ระบุถึงนิยามพารามิเตอร์อย่างไม่ซ้ำอยู่ในไฟล์แค็ตตาล็อก
<i>value</i>	ค่าของพารามิเตอร์สามารถระบุเป็นแอ็ตทริบิวต์หรือเป็นอิลิเมนต์ซายน์อย่างใดอย่างหนึ่ง
<i>applyType</i>	<p>แอ็ตทริบิวต์ <i>applyType</i> สามารถใช้ค่า runtime (ค่าดีฟอลต์) หรือ nextboot แอ็ตทริบิวต์นี้ทำหน้าที่กำหนดคำสั่งที่ใช้เพื่อจัดการพารามิเตอร์</p> <p>สำหรับปฏิบัติการที่ถูกตั้งค่า <i>applyType=runtime</i> แสดงว่าคำสั่ง <code><Set type="permanent"></code> จากไฟล์แค็ตตาล็อกควรถูกใช้เพื่อตั้งค่าพารามิเตอร์ <i>applyType=nextboot</i> แสดงว่าคำสั่ง <code><Set type="nextboot"></code> ควรถูกใช้แทน</p> <p>สำหรับปฏิบัติการ <code>get</code> เมื่อแฟล็ก <code>-p</code> ถูกใช้ <i>applyType=runtime</i> จะระบุว่าคำสั่ง <code><Get type="current"></code> จากไฟล์แค็ตตาล็อกต้องถูกใช้เพื่อรับพารามิเตอร์ <code><applyType=nextboot</code> ระบุว่าคำสั่ง <code><Get type="nextboot"></code> ควรถูกใช้แทน</p> <p>แอ็ตทริบิวต์ <i>applyType</i> ต้องถูกตั้งค่าเป็น nextboot หากแอ็ตทริบิวต์ <i>reboot</i> ถูกตั้งค่าเป็น true</p>
<i>reboot</i>	<p>แอ็ตทริบิวต์นี้มีค่าดีฟอลต์เป็น false เมื่อตั้งค่าเป็น true หมายความว่า ระบบต้องถูกรีบูตก่อน การเปลี่ยนแปลงที่ดำเนินการพารามิเตอร์จึงจะแสดงผล แอ็ตทริบิวต์นี้ต้องตรงกับแอ็ตทริบิวต์ <i>reboot</i> ที่กำหนดอยู่ในอิลิเมนต์ <code><ParameterDef></code> ที่ตรงกันของไฟล์แค็ตตาล็อก</p> <p>เมื่ออิลิเมนต์นี้ถูกตั้งค่าเป็น true แอ็ตทริบิวต์ <i>applyType</i> ต้องถูกตั้งค่าเป็น nextboot</p> <p>เมื่อคุณตั้งค่าพารามิเตอร์ที่มีแอ็ตทริบิวต์ <i>reboot</i> ที่ถูกตั้งค่า ค่าเตือนจะแสดงขึ้นใหญ่ใช้เพื่อระบุว่าจำเป็นต้องปฏิบัติการรีบูท:</p> <p>0590-206 จำเป็นต้องมีการดำเนินการเองหลังปฏิบัติการ เพื่อให้การเปลี่ยนแปลงแสดงผล ไปรตรีบูทระบบ</p> <p>เพียงตั้งค่า <i>reboot</i> เป็น true การเปลี่ยนแปลงค่าพารามิเตอร์จะไม่แสดงผล จนกว่าหลังจากการรีบูทครั้งถัดไป</p>

ตารางที่ 10. แอ็ททริบิวต์ (ต่อ)

แอ็ททริบิวต์	คำอธิบาย
readOnly	<p>แอ็ททริบิวต์นี้แสดงว่า ค่าพารามิเตอร์สามารถอ่านได้โดยคำสั่ง <code>artexget</code> แต่จะไม่ถูกตั้งค่าโดยใช้คำสั่ง <code>artexset</code> หรือไม่ถูกนำมาพิจารณาในปฏิบัติการเปรียบเทียบกับค่า live โดยใช้คำสั่ง <code>artexdiff</code> ค่าดีฟอลต์เป็น false</p> <p>สถานการณ์ไม่กี่แบบที่สามารถรับประกันการตั้งค่าแอ็ททริบิวต์ <code>readOnly</code> เป็น true ดังนี้:</p> <ul style="list-style-type: none"> พารามิเตอร์นี้เป็นค่าคงที่ และค่านี้จะไม่สามารถแก้ไขได้ (เช่น พารามิเตอร์ <code>memory_frames</code> ในคำสั่ง <code>vmo</code>) เข้าถึงพารามิเตอร์ถูกจำกัดและไม่แนะนำให้ผู้ใช้ปรับเปลี่ยนพารามิเตอร์ในโพธิ์เตอร์อัตโนมัติ ในกรณีนี้วิธีการกำหนดคอนฟิกควรจะถูกกำหนดสำหรับพารามิเตอร์นี้ในไฟล์แค็ตตาล็อก แต่ผู้ดูแลระบบต้องลบแอ็ททริบิวต์ <code>readOnly</code> ออกจากโปรไฟล์ด้วยตนเอง เพื่อที่จะได้สามารถตั้งค่าพารามิเตอร์ได้
setDiscover	<p>แอ็ททริบิวต์ <code>setDiscover</code> เมื่อตั้งค่าเป็น true แสดงว่า เมื่อคำสั่ง <code>artexset</code> ถูกเรียกด้วยแฟล็ก <code>-d</code> คำสั่ง <code>discover</code> จะต้องถูกเรียกให้ดำเนินการค้นหาอินสแตนซ์ทั้งหมดของเป้าหมาย และตั้งค่าทั้งหมดให้เป็นค่าที่จัดเก็บไว้ในโปรไฟล์</p> <p>ค่าดีฟอลต์ <code>setDiscover</code> เป็น false ค่า true จะเป็นประโยชน์เฉพาะเมื่อพารามิเตอร์มีคลาสเป้าหมายระบุอยู่ในไฟล์แค็ตตาล็อก</p> <p>โปรดอย่าระบุค่าแอ็ททริบิวต์นี้เมื่อสร้างโปรไฟล์ตัวอย่าง ผู้ใช้ระดับสูงต้องเพิ่มแอ็ททริบิวต์นี้ด้วยตนเองเมื่อเห็นว่าจำเป็น</p>

แอ็ททริบิวต์อื่นๆ

แอ็ททริบิวต์ `type` และ `disruptive` เป็นแอ็ททริบิวต์ให้ข้อมูลซึ่งถูกตั้งค่าอัตโนมัติโดยคำสั่ง `artextget` เมื่อถูกเรียกโดยแฟล็ก `-i` โปรดอย่ารวมแอ็ททริบิวต์เหล่านี้เมื่อสร้างโปรไฟล์ตัวอย่าง

ตัวอย่าง

- ตัวอย่างต่อไปนี้ คัดลอกมาจากแค็ตตาล็อก `vmoProfile.xml` ตัวอย่าง เพื่อแสดงการใช้แอ็ททริบิวต์ทางเลือกที่หลากหลาย:

```
<Profile origin="reference" readOnly="true" version="2.0.0">
  <Catalog id="vmoParam" version="2.1">
    <Parameter name="nokilluid"/>
    <Parameter name="memory_frames" readOnly="true"/>
    <Parameter name="kernel_heap_psize" reboot="true" applyType="nextboot"/>
  </Catalog>
</Profile>
```

- หากคุณรันคำสั่ง `artexget -r` บนโปรไฟล์จากตัวอย่าง 1 โปรไฟล์ต่อไปนี้จะแสดงขึ้น:

```
<Profile origin="get" version="2.0.1" date="2011-03-24T13:41:01Z">
  <Catalog id="vmoParam" version="2.1">
    <Parameter name="nokilluid" value="0"/>
  </Catalog>
</Profile>
```

```

    <Parameter name="memory_frames" value="393216" readOnly="true"/>
    <Parameter name="kernel_heap_psize" value="4096" applyType="nextboot" reboot="true"/>
  </Catalog>
</Profile>

```

ข้อมูลที่เกี่ยวข้อง

หัวเรื่อง ค่าพารามิเตอร์

อิลิเมนต์ <ParameterDef>

ค่าพารามิเตอร์:

ค่าของพารามิเตอร์สามารถกำหนดอยู่ในโปรไฟล์ ไม่ว่าในฐานะที่เป็นแอตทริบิวต์ถ้าสั้นพอ หรือในฐานะที่เป็นอิลิเมนต์ชายน์ของ <Parameter>

การใช้

เมื่อเขียนโปรไฟล์ตัวอย่างต้องมีการมอบหมายให้พารามิเตอร์ ค่าของพารามิเตอร์หากมีอยู่ จะถูกรวมไว้โดยอัตโนมัติในโปรไฟล์ที่ได้โดยการรันคำสั่ง **artexget**

ค่า runtime และ nextboot

แนวคิดของค่า runtime และ nextboot เป็นส่วนสำคัญของเฟรมเวิร์ก AIX Runtime Expert

ค่า runtime ของพารามิเตอร์คือค่าปัจจุบันที่ถูกเรียกใช้บนระบบ ณ เวลาที่คำสั่ง **artexget** ถูกรัน ค่า nextboot เป็นค่าที่พารามิเตอร์จะมีหลังจากการรีบูตครั้งต่อไปของระบบ

ตัวอย่างเช่น พารามิเตอร์ *type_of_dump* ในโปรไฟล์ *sysdumpdevProfile.xml* ค่า (runtime) ปัจจุบันของพารามิเตอร์นี้อาจเป็น *traditional* หรือ *firmware-assisted* หากค่าถูกเปลี่ยน (การใช้คำสั่ง **artexset** หรือดำเนินการโดยตรงโดยใช้คำสั่ง **sysdumpdev**) ก็จะไม่แสดงผลจนกว่าจะถึงการรีบูตครั้งต่อไป จากนั้น ค่า nextboot ของพารามิเตอร์นี้ ก็จะเป็นค่าที่ถูกแก้ไข

```
<Parameter name="type_of_dump" applyType="nextboot" reboot="true" />
```

ตัวอย่าง

ตัวอย่างต่อไปนี้แสดงพารามิเตอร์ที่มีค่าที่ระบุไว้เป็นแอตทริบิวต์ และพารามิเตอร์อื่นที่มีค่าที่ระบุว่าเป็นอิลิเมนต์ชายน์:

```

<Profile origin="get" version="2.0.1" date="2010-09-28T12:30:03Z">
<Catalog id="login.cfgParam" version="2.0">
<Parameter name="shells">
<Value>
/bin/sh,/bin/bsh,/bin/csh,/bin/ksh,/bin/tsh,
/bin/ksh93,/usr/bin/sh,/usr/bin/bsh,/usr/bin/csh,
/usr/bin/ksh,/usr/bin/tsh,/usr/bin/ksh93,
/usr/bin/rksh,/usr/bin/rksh93,
/usr/sbin/uucp/uucico,/usr/sbin/sliplogin,
/usr/sbin/snappd
</Value>

```

```

</Parameter>
<Parameter name="maxlogins" value="32767"/>
</Catalog>
</Profile>

```

อิลิเมนต์ <Property>:

อิลิเมนต์ <Property> กำหนดค่าให้แก่คุณสมบัติ พารามิเตอร์

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <Parameter>

แอตทริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 11. แอ็ตทริบิวต์

แอ็ตทริบิวต์	บังคับ	ชนิด	คำอธิบาย
name	yes	สตริง	ระบุชื่อของคุณสมบัติ
value	no	สตริง	ระบุค่าของคุณสมบัติ

การใช้งาน

อิลิเมนต์ <Property> กำหนด ค่าให้แก่ชื่อคุณสมบัติของพารามิเตอร์อิลิเมนต์ คำนีใช้ เมื่อลำดับ %p[name] ถูกขยายระหว่างการสร้าง บรรทัดรับคำสั่ง

อิลิเมนต์ <Property> โดยทั่วไปไม่ถูก เพิ่มในโปรไฟล์ด้วยตนเอง อิลิเมนต์ถูกแทรกโดยอัตโนมัติ ในเอาต์พุตโปรไฟล์เมื่อคำสั่ง `artexget -r` และ `artexget -n` ถูกรัน ตามคำสั่งที่กำหนดภายใต้ อิลิเมนต์ <Property> ที่เกี่ยวข้องของไฟล์แค็ตตาล็อก

ตัวอย่าง

ตัวอย่างต่อไปนี้ตั้งค่าคุณสมบัติ `nodeId` สำหรับพารามิเตอร์ `netaddr` ค่าคุณสมบัติถูก บันทึกโดยคำสั่ง `artexget -r` และเป็นเอาต์พุตของคำสั่ง `uname -f`:

```

<Parameter name="netaddr" value="172.16.128.13">
  <Target class="device" instance="en0"/>
  <Property name="nodeId" value="8000108390E00009"/>
</Parameter>

```

ข้อมูลที่เกี่ยวข้อง

“อิลิเมนต์ <PropertyDef>” ในหน้า 131 (ใน ไฟล์แค็ตตาล็อก)

อิลิเมนต์ <Seed>:

อิลิเมนต์ <Seed> กำหนด seed ที่ถูกขยาย เป็นอย่างน้อยหนึ่งอิลิเมนต์ <ParameterDef> ระหว่างการดำเนินการ <Get>

ไวยากรณ์

อิลิเมนต์พารেন্ট: <Catalog>

แอ็ททริบิวต์ ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 12. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
name	yes	สตริง	ระบุชื่อของอิลิเมนต์ seed ที่ตรงกับอิลิเมนต์ SeedDef ในไฟล์แค็ตตาล็อก

อิลิเมนต์ชายนั้ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 13. อิลิเมนต์ชายนั้

อิลิเมนต์ชายนั้	บังคับ	ตัวเลข	คำอธิบาย
<Parameter>	no	0 - ค่าใดๆ ก็ตาม	กรองพารามิเตอร์ที่ค้นพบตามชื่อของพารามิเตอร์
<Target>	no	0 - ค่าใดๆ ก็ตาม	กรองพารามิเตอร์ที่ค้นพบตามปลายทางของพารามิเตอร์

การใช้งาน

อิลิเมนต์ <Seed> ค้นหา พารามิเตอร์แบบไดนามิกระหว่างการดำเนินการ <Get>

เมื่อ ออกคำสั่ง `artexget` แต่ละอิลิเมนต์ <Seed> ในโปรไฟล์อินพุตถูกขยายออกไปอย่างน้อยหนึ่งอิลิเมนต์ <Parameter> โปรไฟล์ถูกขยายตามกฎที่กำหนดในอิลิเมนต์ <SeedDef> ที่ตรงกัน ของไฟล์แค็ตตาล็อก กระบวนการนี้ถูกกว่าการค้นหาพารามิเตอร์ หลังจาก กระบวนการค้นหาพารามิเตอร์เสร็จสมบูรณ์ คำสั่ง `artexget` ดำเนินการตามปกติกับโปรไฟล์ที่ขยาย

ชายนั้อิลิเมนต์ <Parameter> และ <Target> ทางเลือก ถูกใช้เพื่อกรองพารามิเตอร์ที่ค้นพบ พารามิเตอร์ที่ค้นพบที่ไม่ตรงกับเกณฑ์ที่กำหนดในอิลิเมนต์ย่อย <Parameter> จะถูกละเว้น รวมทั้ง พารามิเตอร์เหล่านั้นที่ใช้กับปลายทางที่ไม่ ตรงกับเกณฑ์ที่กำหนดในอิลิเมนต์ย่อย <Target> ถูกละเว้น

ตัวอย่าง

ตัวอย่างนี้ใช้แค็ตตาล็อก `devSeed` เพื่อกำหนด seed ที่สามารถใช้เพื่อค้นหาแอ็ททริบิวต์ทั้งหมดของ อุปกรณ์ทั้งหมด:

```
<?xml version="1.0" encoding="UTF-8"?>
<Catalog id="devSeed" version="3.0">
  <SeedDef name="devAttr">
    <Discover>
      <Command>
        /usr/sbin/lsdev -F 'name class subclass type' |
        while read DEV CLASS SUBCLASS TYPE
        do
          CAT=devParam.$CLASS.$SUBCLASS.$TYPE
          /usr/sbin/lsattr -F attribute -l $DEV |
          while read PAR
```

```

do
    echo "device=$DEV $CAT $PAR"
done
done
</Command>
Mask target="1" catalog="2" name="3">(.*) (.*) (.*)</Mask>
</Discover>
</SeedDef>
</Catalog>

```

โปรไฟล์ต่อไปนี้อาจใช้เพื่อค้นหาแอตทริบิวต์ที่สนับสนุนทั้งหมดของอุปกรณ์ที่สนับสนุนทั้งหมด:

```

<?xml version="1.0" encoding="UTF-8"?>
<Profile>
  <Catalog id="devSeed" version="3.0">
    <Seed name="devAttr"/>
  </Catalog>
</Profile>

```

2. การใช้แค่ตัวเลือกเดียวกัน ตัวกรอง **<Target>** สามารถใช้เพื่อค้นหาแอตทริบิวต์ที่สนับสนุนทั้งหมดของอะแดปเตอร์ Ethernet ทั้งหมด:

```

<?xml version="1.0" encoding="UTF-8"?>
<Profile>
  <Catalog id="devSeed" version="3.0">
    <Seed name="devAttr">
      <Target class="device" match="^en[0-9]+$"/>
    </Seed>
  </Catalog>
</Profile>

```

3. ตัวกรอง **<Parameter>** สามารถถูกเพิ่มเพื่อบันทึกเฉพาะแอตทริบิวต์ **netaddr**, **netaddr6**, **alias** และ **alias6** ของอะแดปเตอร์ Ethernet ทั้งหมด:

```

<?xml version="1.0" encoding="UTF-8"?>
<Profile>
  <Catalog id="devSeed" version="3.0">
    <Seed name="devattr">
      <Parameter match="^(netaddr|alias)6?$"/>
      <Target class="device" match="^en[0-9]+$"/>
    </Seed>
  </Catalog>
</Profile>

```

ข้อมูลที่เกี่ยวข้อง

อิลิเมนต์ “อิลิเมนต์ <SeedDef>” ในหน้า 124 (ในไฟล์แค็ตตาล็อก)

อิลิเมนต์ <Target>:

อิลิเมนต์ <Target> ทำหน้าที่กำหนดอินสแตนซ์ของคลาสเป้าหมายที่นำพารามิเตอร์มาใช้ด้วย

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <Parameter>

การเกิดขึ้นหลายครั้งของพารามิเตอร์เดียวกันจากแค็ตตาล็อกเดียวกันจะได้รับอนุญาต และหากเพียงแต่การเกิดเหล่านี้นำไปใช้กับอินสแตนซ์ที่แตกต่างของเป้าหมายของตน

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 14. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
<i>class</i>	ใช่	สตริง	ระบุชื่อของคลาสเป้าหมาย
<i>instance</i>	ไม่ใช่*	สตริง	ระบุชื่อของอินสแตนซ์ของคลาส
<i>match</i>	ไม่ใช่*	สตริง	ระบุนิพจน์ปกติตามที่ใช้กับชื่ออินสแตนซ์ที่ค้นพบ

* หนึ่งเดียวและเป็นหนึ่งในแอ็ททริบิวต์ *instance* และ *match* ที่ต้องถูกระบุ

การใช้

บางพารามิเตอร์ไม่ได้ใช้กับระบบทั้งหมด แต่ใช้กับเฉพาะอ็อบเจกต์ ตัวอย่างคือ ไดร็อกทอรี *home* ของผู้ใช้ตั้งที่ระบุในโปรไฟล์ *chuserProfile.xml* พารามิเตอร์นี้ใช้กับผู้ใช้เฉพาะ (ราก ผู้เยี่ยมชม) ในโมดูลเฉพาะที่สามารถโหลดได้ (ไฟล์, LDAP) ในตัวอย่างนี้ผู้ใช้และโมดูลเป็นสองคลาสเป้าหมายหลัก พารามิเตอร์ *home* ใช้กับอินสแตนซ์เฉพาะของคลาสเป้าหมาย ตัวอย่างเช่น อินสแตนซ์ผู้เยี่ยมชมของคลาสผู้ใช้ และอินสแตนซ์ไฟล์ของคลาสโมดูล

หากแอ็ททริบิวต์ *class* และ *instance* ทั้งสองถูกตั้งค่าเป็นสตริงว่า การค้นหาจะถูกดำเนินการสำหรับพารามิเตอร์นี้เมื่อคำสั่ง *artexget* ถูกรันบนโปรไฟล์เช่นนี้วิธีการค้นหาที่ประกาศในไฟล์แค็ตตาล็อกที่ตรงกันถูกส่งการ และพารามิเตอร์ถูกสร้างขึ้นในโปรไฟล์เอาต์พุตสำหรับแต่ละอินสแตนซ์ที่ถูกค้นพบของพารามิเตอร์ ดูตัวอย่าง 1

หากทั้งแอ็ททริบิวต์ *class* และ *instance* ถูกระบุ ดังนั้น เป้าหมายจะมีคุณสมบัติครบถ้วน และพารามิเตอร์จะเฉพาะกับอินสแตนซ์ที่ถูกระบุของคลาสเป้าหมาย โปรดดูตัวอย่างที่ 2

หากทั้งแอ็ททริบิวต์ *class* และ *match* ถูกระบุ จะมีการดำเนินการค้นหาดังข้างต้น แต่เฉพาะอินสแตนซ์เป้าหมายที่มีชื่อตรงกับนิพจน์ปกติที่ระบุในแอ็ททริบิวต์ *match* เท่านั้น ที่จะถูกค้นพบ ดูตัวอย่าง 3

เมื่อเขียนโปรไฟล์ตัวอย่าง แอ็ททริบิวต์ *class* and *instance* ต้องถูกปล่อยว่าง ซึ่งหมายความว่า เมื่อพบกับคลาสเป้าหมายที่ว่างเปล่า คำสั่ง *artexget* จะค้นหารายการอินสแตนซ์ของคลาสเป้าหมาย (ผู้ใช้และระบบย่อยทั้งหมดบนระบบ) ก่อนที่จะเรียกใช้ค่า

เมื่อรันคำสั่ง *artexset* คลาสเป้าหมายที่ไม่ถูกค้นพบ จะมีการแสดงค่าเตือน:

0590-216 บางพารามิเตอร์ในโปรไฟล์ต้องการการค้นหาเป้าหมาย และจะถูกละเว้น

ตัวอย่าง

1. ตัวอย่างของโปรไฟล์ที่มีเป้าหมายก่อนการค้นพบ ได้แก่โปรไฟล์ chuserProfile.xml ที่กำหนดโฮมไดเรกทอรีของผู้ใช้ ตัวอย่างต่อไปนี้เป็นการแสดงโปรไฟล์ตัวอย่าง:

```
<Profile version="2.0.0" origin="reference" readOnly="true">
  <Catalog id="chuserParam" version="2.0">
    <Parameter name="home">
      <Target class="" instance=""/>
    </Parameter>
  </Catalog>
</Profile>
```

2. หลังการค้นพบโปรไฟล์ chuserProfile.xml จะมีสำเนาของพารามิเตอร์โฮมสำหรับแต่ละผู้ใช้ที่ถูกระบุในแต่ละโมดูลที่ถูกระบุซึ่งสามารถโหลดได้:

```
<Profile version="2.0.0" origin="get">
  <Catalog id="chuserParam" version="2.0">

    <Parameter name="home" value="/">
      <Target class="user" instance="root"/>
      <Target class="module" instance="files"/>
    </Parameter>

    <Parameter name="home" value="/etc">
      <Target class="user" instance="daemon"/>
      <Target class="module" instance="files"/>
    </Parameter>
```

...

```
</Catalog>
</Profile>
```

3. โปรไฟล์ดังต่อไปนี้ใช้แอตทริบิวต์ *match* เพื่อค้นหาโฮมไดเรกทอรีของผู้ใช้ทั้งหมดที่มีชื่อที่ขึ้นต้นด้วย *u* ในโมดูลไฟล์:

```
<Profile version="2.0.0" origin="reference" readOnly="true">
  <Catalog id="chuserParam" version="2.0">
    <Parameter name="home">
      <Target class="user" match="^u"/>
      <Target class="module" instance="files"/>
    </Parameter>
  </Catalog>
</Profile>
```

ข้อมูลที่เกี่ยวข้อง

อิลิเมนต์ <Discover> (ในไฟล์แค็ตตาล็อก)

การเขียนแค็ตตาล็อก AIX Runtime Expert

ไฟล์แค็ตตาล็อกถูกใช้เป็นการภายในโดยเฟรมเวิร์ก AIX Runtime Expert

ไฟล์แค็ตตาล็อกประกอบด้วย นิยามพารามิเตอร์และข้อมูลการโยงสำหรับวิธีการกำหนดคอนฟิกซึ่งอธิบายถึงคำสั่งที่ใช้เพื่อตั้งหรือตั้งค่าพารามิเตอร์ ไฟล์แค็ตตาล็อกเป็นแบบโลคัลสำหรับระบบที่กำลังถูกปรับและกำหนดคอนฟิก

แนวคิดแค็ตตาล็อก AIX Runtime Expert:

ไฟล์แค็ตตาล็อกประกอบด้วยข้อมูลทั้งหมดที่จำเป็นสำหรับดำเนินการปฏิบัติการบนพารามิเตอร์ ซึ่งรวมถึงนิยามเงื่อนไขการใช้งาน และวิธีการกำหนดคอนฟิก ไฟล์แค็ตตาล็อกต้องถูกจัดการโดยตรง โดยผู้ใช้และใช้ผ่านเอ็นจินคอร์ของ AIX Runtime Expert เท่านั้น

แค็ตตาล็อกถูกติดตั้งบนระบบพร้อมๆ กันกับเครื่องหลัก AIX Runtime Expert เมื่อแค็ตตาล็อกใหม่ถูกเชื่อมโยงกับคอมพิวเตอร์หรือแอ็พพลิเคชันของบุคคลภายนอกซึ่งติดตั้งอยู่บนระบบ จำเป็นที่จะต้องตรวจสอบว่า ส่วนประกอบเหล่านั้นมีความสอดคล้องกับเครื่องหลัก AIX Runtime Expert ที่ติดตั้งอยู่

ตำแหน่งแค็ตตาล็อก:

ไฟล์แค็ตตาล็อก AIX Runtime Expert catalog ถูกจัดเก็บอยู่ในไดเรกทอรี /etc/security/artex/catalogs

ชื่อไฟล์แค็ตตาล็อกต้องตรงกับแอ็ททริบิวต์ *id* ของตนเอง ซึ่งมีการต่อท้ายด้วยส่วนขยาย .xml ตัวอย่างเช่น แค็ตตาล็อกที่ชื่อ `commandParam.xml` ต้องมีค่าแอ็ททริบิวต์ *id* `commandParam`

เพื่อที่จะได้ถูกจัดวางตำแหน่งโดยโปรไฟล์ที่อ้างอิง แค็ตตาล็อกต้องมีชื่อเดียวกันกับในไฟล์ XML ของแค็ตตาล็อก และในอิลิเมนต์ `<Catalog>` ของไฟล์ XML ของโปรไฟล์ ตามค่าดีฟอลต์แล้ว เครื่องหลัก AIX Runtime Expert จะมองหาแค็ตตาล็อกในไดเรกทอรีดีฟอลต์ /etc/security/artex/catalogs ลักษณะการทำงานเช่นนี้อาจถูกเปลี่ยนแปลงสำหรับผู้ใช้รากเท่านั้น โดยการตั้งค่าตัวแปรสภาพแวดล้อม `ARTEX_CATALOG_PATH` หลายไดเรกทอรีสามารถระบุได้ในตัวแปรของสภาพแวดล้อมนี้โดยใช้เครื่องหมายค้น :

กระบวนการแค็ตตาล็อก:

ขั้นตอนในการเขียนแค็ตตาล็อก AIX Runtime Expert ใหม่

จำเป็นต้องดำเนินขั้นตอนต่อไปนี้เมื่อเขียนแค็ตตาล็อก AIX Runtime Expert ใหม่:

1. ทำรายการพารามิเตอร์ที่คุณต้องการในไฟล์แค็ตตาล็อก
2. สำหรับแต่ละพารามิเตอร์ให้สร้างอิลิเมนต์ `<ParameterDef>`
3. หากหลายพารามิเตอร์สำหรับปฏิบัติการ `<Get>`, `<Set>`, `<Discover>` หรือ `<Diff>`:
 - กำหนดอิลิเมนต์ `<CfgMethod>` ที่ส่วนบนสุดของแค็ตตาล็อก
 - ใช้แอ็ททริบิวต์ `cfgmethod` เพื่อรับจากวิธีการกำหนดคอนฟิก
4. หากหลายพารามิเตอร์ต้องอยู่ภายใต้ข้อบังคับเดียวกันให้กำหนดอิลิเมนต์ `<ConstraintDef>` ที่ส่วนบนสุดของแค็ตตาล็อก
5. สำหรับแต่ละพารามิเตอร์:
 - a. กำหนดปฏิบัติการ `<Get type="current">` และ `<Get type="nextboot">` สำหรับแต่ละพารามิเตอร์ ไม่ว่าทางตรงภายใต้อิลิเมนต์ `<ParameterDef>` หรือถูกอ้างอิงถึงภายใต้อิลิเมนต์ `<CfgMethod>` หรือใช้ผสมรวมกันในลักษณะใดๆ
 - b. กำหนดปฏิบัติการ `<Set>` ทั้งหมดที่ได้รับการสนับสนุนสำหรับแต่ละพารามิเตอร์ ไม่ว่าโดยทางตรงภายใต้อิลิเมนต์ `<ParameterDef>` หรือภายใต้อิลิเมนต์ `<CfgMethod>` ที่ถูกอ้างอิงถึง หรือใช้ความเป็นไปได้เหล่านั้นผสมรวมกัน
 - c. หากพารามิเตอร์ต้องการเป้าหมาย:
 - 1) ให้กำหนดคลาสเป้าหมายที่สนับสนุนโดยใช้แอ็ททริบิวต์ `targetClass`

2) กำหนดปฏิบัติการค้นหาไม่ว่าโดยทางตรงภายใต้อิลิเมนต์ <ParameterDef>, หรืออิลิเมนต์ <CfgMethod> ที่ถูกอ้างอิงถึงหรือใช้ความเป็นไปได้เหล่านี้ผสมรวมกัน ในกรณีส่วนใหญ่วิธีการค้นพบจะถูกกำหนดในวิธีการกำหนดคอนฟิก

- d. หากพารามิเตอร์ต้องการการรีบูทเพื่อให้การเปลี่ยนแปลงแสดงผล ให้เพิ่มแอตทริบิวต์ `reboot=true`
- e. หากพารามิเตอร์ตั้งอยู่ภายใต้ข้อบังคับไม่ว่าจะกำหนดอิลิเมนต์ <ConstraintDef> ภายใต้อิลิเมนต์ <ParameterDef> หรือใช้แอตทริบิวต์ `constraint` เพื่ออ้างอิงข้อบังคับที่มีอยู่

6. วิธีการทดสอบไฟล์แค็ตตาล็อก:

- a. สร้างโปรไฟล์ด้วยพารามิเตอร์ทั้งหมดที่กำหนดในไฟล์แค็ตตาล็อก
- b. ใช้คำสั่ง `artexget -r` เพื่อตรวจจับค่าและทดสอบปฏิบัติการ <Discover> และ <Get>
- c. ใช้คำสั่ง `artexset -c -F -R -l all` บนโปรไฟล์ผลลัพธ์เพื่อทดสอบปฏิบัติการ <Set> และ <Diff>
- d. นอกจากนี้แฟล็ก `-g 3 -g COMMANDS` ยังสามารถเพิ่มให้แก่สองคำสั่งดังกล่าวเพื่อรับข้อมูลเพิ่มเติมเกี่ยวกับบรรทัดคำสั่งที่สร้างขึ้นเพื่อดำเนินปฏิบัติการที่ถูกขอ

ข้อมูลที่เกี่ยวข้อง

โปรดดูที่หัวข้อเกี่ยวกับอิลิเมนต์รูท <Catalog>

AIX Runtime Expertอิลิเมนต์ catalog:

อิลิเมนต์ <Catalog>:

อิลิเมนต์ <Catalog> เป็นอิลิเมนต์รูทสำหรับไฟล์แค็ตตาล็อกทั้งหมด

ไวยากรณ์

แอตทริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 15. แอตทริบิวต์

แอตทริบิวต์	บังคับ	ชนิด	คำอธิบาย
<code>id</code>	ใช่	สตริง	ระบุชื่อของแค็ตตาล็อก ชื่อนี้ต้องไม่มีการซ้ำกันบนระบบ
<code>version</code>	ไม่ใช่	สตริง	ระบุหมายเลขเวอร์ชันของแค็ตตาล็อก
<code>date</code>	ไม่ใช่	dateTime	ระบุวันที่ทำการสร้างรูปแบบคือ <code>YYYY-MM-DDThh:mm:ss</code>
<code>priority</code>	ไม่ใช่	จำนวนเต็ม	ระบุลำดับการสั่งการของแค็ตตาล็อกที่สัมพันธ์กันกับแค็ตตาล็อกอื่นๆ ในวิธีการที่ตั้งค่าไว้ ค่าดีฟอลต์คือ 0
<code>inherit</code>	ไม่ใช่	สตริง	ระบุชื่อของแค็ตตาล็อกที่จะสืบทอด

อิลิเมนต์ชายนต์ต่อไปนี้ได้รับการสนับสนุน คอลัมน์ `number` กำหนดจำนวนครั้งการเปิดชายนต์ที่อนุญาต:

ตารางที่ 16. อิลิเมนต์ชายน

อิลิเมนต์ชายน	บังคับ	ตัวเลข	คำอธิบาย
<ShortDescription>	ไม่ใช่	0 - 1	คำอธิบายข้อความแบบสั้นสำหรับแค็ตตาล็อก
<Description>	ไม่ใช่	0 - 1	คำอธิบายข้อความแบบยาวสำหรับแค็ตตาล็อก
<SubCat>	ไม่ใช่	0 - ค่าใดๆก็ตาม	หมวดหมู่ย่อย
<ParameterDef>	ไม่ใช่	0 - ค่าใดๆก็ตาม	มีคุณสมบัติต่างๆของพารามิเตอร์
<ConstraintDef>	ไม่ใช่	0 - ค่าใดๆก็ตาม	นิยามข้อบังคับพารามิเตอร์ (เงื่อนไขและคำสั่ง disruptive)
<CfgMethod>	ไม่ใช่	0 - ค่าใดๆก็ตาม	นิยามวิธีการกำหนดคอนฟิก
<PrereqDef>	ไม่ใช่	0 - ค่าใดๆก็ตาม	กำหนดสิ่งที่จำเป็นต้องมี
<PropertyDef>	ไม่ใช่	0 - ค่าใดๆก็ตาม	กำหนดคุณสมบัติ
<SeedDef>	ไม่ใช่	0 - ค่าใดๆก็ตาม	กำหนด seed

แอ็ททริบิวต์

ตารางที่ 17. แอ็ททริบิวต์

แอ็ททริบิวต์	คำอธิบาย
<i>id</i>	แอ็ททริบิวต์ <i>id</i> ควรจะตรงกับชื่อไฟล์แค็ตตาล็อกที่ถอดมาจากส่วนชยาย .xml ของตนเอง id แค็ตตาล็อกถูกอ้างอิงถึงในโปรไฟล์โดยใช้อิลิเมนต์ <Catalog>
<i>Priority</i>	แอ็ททริบิวต์ <i>priority</i> ถูกใช้เมื่อจำเป็นต้องรันวิธีการที่ตั้งค่าของแค็ตตาล็อกเฉพาะ ก่อนหรือหลังวิธีการที่ตั้งค่าของแค็ตตาล็อกอื่นๆ เมื่อถูกรวมอยู่ในโปรไฟล์เดียวกัน (เช่น โปรไฟล์ผสม default.xml) ลำดับความสำคัญที่เป็นค่าดีฟอลต์ของแค็ตตาล็อกคือ 0 กฎมีอยู่ว่า เมื่อสองแค็ตตาล็อกมีลำดับความสำคัญเดียวกัน วิธีการตั้งค่าจะถูกรันอยู่ในลำดับที่ไม่มีกำหนด หากแค็ตตาล็อกมีลำดับความสำคัญที่มีค่าเป็นบวก วิธีการตั้งค่าจะถูกสั่งการก่อน ในลักษณะตามลำดับความสำคัญจากมากไปหาน้อย หากแค็ตตาล็อกมีลำดับความสำคัญที่มีค่าเป็นลบ วิธีการตั้งค่าจะถูกสั่งการทีหลัง ในลักษณะตามลำดับความสำคัญจากมากไปหาน้อย
<i>Version</i>	แอ็ททริบิวต์ <i>Version</i> มีอยู่ในทั้งโปรไฟล์และแค็ตตาล็อก version ช่วยระบุว่าโปรไฟล์และแค็ตตาล็อกเข้ากันได้กับเครื่องหลัก AIX Runtime Expert และเข้ากันได้เองหรือไม่ อ่าน แอ็ททริบิวต์ <i>Version</i> เพื่อรับทราบรายละเอียดเพิ่มเติม
<i>Date</i>	ขณะนี้ แอ็ททริบิวต์ <i>date</i> ไม่ถูกใช้สำหรับอิลิเมนต์ <Catalog> โดยจะถูกรวมไว้เพื่อการใช้งานและความสามารถในการดูแลรักษาในอนาคต
<i>inherit</i>	แอ็ททริบิวต์ <i>inherit</i> ระบุชื่อของแค็ตตาล็อกเพื่อสืบทอด โดยไม่มีส่วนชยาย .xml อิลิเมนต์ ทั้งหมดที่กำหนดในแค็ตตาล็อกที่สืบทอดจะมีอยู่ในแค็ตตาล็อกหลัก เสมือนว่าถูกกำหนดแบบโลคัล

ตัวอย่าง

ต่อไปนี้เป็นตัวอย่างของแค็ตตาล็อกที่ใช้แอ็ททริบิวต์ *priority* แค็ตตาล็อก aixpertParam.xml ตั้งค่าตัวเลือกความปลอดภัย และต้องถูกตั้งค่าหลังจากที่แค็ตตาล็อกอื่นๆ ทั้งหมดได้ถูกตั้งค่าแล้ว ด้วยเหตุนี้ ลำดับความสำคัญจึงถูกตั้งค่าเป็น ค่าติดลบสูง <Catalog id="aixpertParam" version="2.0" priority="-1000">

ข้อมูลที่เกี่ยวข้อง

<อิลิเมนต์ ConstraintDef>

<อิลิเมนต์ CfgMethod>

อิลิเมนต์<Description> และ <ShortDescription>

อิลิเมนต์ <ParameterDef>

อิลิเมนต์ <SubCat>

แอ็ททริบิวต์ *Version*:

ไวยากรณ์

เวอร์ชันของแค็ตตาล็อกถูกเขียนในฐานะที่เป็นแอ็ททริบิวต์ในรูปแบบ *MM.mm* โดยที่ *MM* เป็นหมายเลขหลัก และ *mm* เป็นหมายเลขรอง

```
<Catalog id="commandParam" version="2.0">
```

หมายเลขเวอร์ชันหลัก

หมายเลขเวอร์ชันหลักเหมือนกับสำหรับแค็ตตาล็อก AIX Runtime Expert ทั้งหมดที่ติดตั้งอยู่บนระบบ และเฟรมเวิร์ก AIX Runtime Expert ทั้งหมดซึ่งเป็นตำแหน่งที่มีการถูกอ้างอิง หมายเลขหลักนี้จะถูกเพิ่มในแต่ละการเปลี่ยนแปลงที่สำคัญของสกีมา XML ของโปรไฟล์และแค็ตตาล็อก

เมื่อสร้างแค็ตตาล็อกใหม่ ให้ตั้งค่าหมายเลขเวอร์ชันหลักเป็นเวอร์ชันเครื่องหลัก AIX Runtime Expert ปัจจุบัน ซึ่งสามารถพบได้โดยการมองเข้าไปภายในไฟล์แค็ตตาล็อกมาตรฐานใดๆ ก็ตาม ที่มาพร้อมกับชุดไฟล์ *artex.base.rte*

หากคำสั่ง *artexget* ถูกเรียกใช้บนโปรไฟล์ซึ่งมีหลายเลขเวอร์ชันหลักแตกต่างจากหมายเลขที่ถูกอ้างอิงในเครื่องหลัก AIX Runtime Expert คำสั่งจะล้มเหลว โดยมีข้อผิดพลาดดังต่อไปนี้:

0590-117 ข้อผิดพลาด เวอร์ชัน

โปรไฟล์ถูกสร้างขึ้นบนเวอร์ชันที่ไม่สนับสนุนโดย ARTEX

นอกจากนี้ยังแนะนำให้โปรไฟล์และแค็ตตาล็อกใช้หมายเลขเวอร์ชันหลักเดียวกันเพื่อให้สามารถเข้ากันได้ แค็ตตาล็อกอ้างอิงของโปรไฟล์ที่มีหมายเลขเวอร์ชันเฉพาะ หากหมายเลขเวอร์ชันหลักของโปรไฟล์ไม่เหมือนกับหมายเลขเวอร์ชันหลักของแค็ตตาล็อก คำสั่ง AIX Runtime Expert จะแสดงการเตือนเพื่อแจ้งผู้ใช้งานว่า ผลลัพธ์อาจไม่สามารถคาดเดาได้:

0590-218 เวอร์ชันแค็ตตาล็อกแตกต่างจากที่อ้างอิงในโปรไฟล์

หมายเลขเวอร์ชันรอง

หมายเลขเวอร์ชันรองมีความเฉพาะสำหรับแต่ละแค็ตตาล็อก และจะถูกทำให้เพิ่มขึ้นในแต่ละครั้งที่มีการเปลี่ยนแปลงที่สำคัญเกิดขึ้นในแค็ตตาล็อกเพื่อให้ไม่เข้ากันได้กับเวอร์ชันก่อนหน้า โปรไฟล์จะอ้างอิงแค็ตตาล็อกที่มีหมายเลขเวอร์ชันเฉพาะ หากหมายเลขเวอร์ชันรองของโปรไฟล์ไม่เหมือนกับหมายเลขเวอร์ชันรองของแค็ตตาล็อก คำสั่ง AIX Runtime Expert จะออกการเตือนให้แจ้งผู้ใช้งานว่า ผลลัพธ์อาจไม่สามารถคาดเดาได้:

0590-218 เวอร์ชันแค็ตตาล็อกแตกต่างจากที่อ้างอิงในโปรไฟล์

100 AIX เวอร์ชัน 7.2: ระบบปฏิบัติการและการจัดการกับอุปกรณ์

เมื่อสร้างโปรไฟล์ตัวอย่างหรือแค็ตตาล็อกใหม่ให้ตั้งค่าหมายเลขเวอร์ชันรองเป็น 0

อิลิเมนต์ `<Description>` และ `<ShortDescription>`:

คำอธิบายเป็นฟิลด์ข้อความให้ข้อมูลทางเลือกซึ่งสามารถเพิ่มให้แก่หลากหลายอิลิเมนต์ในไฟล์แค็ตตาล็อกได้ ฟิลด์เหล่านี้เป็นทางเลือก แต่ขอแนะนำว่า ผู้เขียนแค็ตตาล็อกควรใช้เพื่อบันทึกอิลิเมนต์พารามิเตอร์

ไวยากรณ์

อิลิเมนต์พารามิเตอร์ของอิลิเมนต์ `<ShortDescription>` อาจเป็นหนึ่งในดังต่อไปนี้:

- `<Catalog>`
- `<SubCat>`

อิลิเมนต์พารามิเตอร์ของอิลิเมนต์ `<Description>` อาจเป็นหนึ่งในดังต่อไปนี้:

- `<Catalog>`
- `<SubCat>`
- `<ParameterDef>`
- `<ConstraintDef>`

เนื้อหาของอิลิเมนต์ `<Description>` และ `<ShortDescription>` อาจเป็นสตริงที่เรียบง่ายหรือข้อความแปดที่กำหนดโดยหนึ่งในอิลิเมนต์ต่อไปนี้ `<NLSCatalog>`, `<NLSSmitHelp>` or `<NLSCCommand>` ดูที่หัวข้อ การสนับสนุนการทำให้เป็นโกลบอลสำหรับข้อมูลเพิ่มเติม

การใช้งาน

ขณะนี้เฉพาะคำอธิบายของอิลิเมนต์ของ `<ParameterDef>` เท่านั้นที่จะถูกเรียกใช้และแสดงโดยคำสั่ง `artexget` พร้อมกับแฟล็ก `-i` แนะนำว่าควรโกลบอลไลเชสชันสำหรับข้อความรวมถึงฟิลด์คำอธิบาย

ขณะนี้ ฟิลด์คำอธิบายของอิลิเมนต์อื่นๆ ไม่ได้ถูกใช้โดยเฟรมเวิร์ก AIX Runtime Expert แต่ควรจะถูกจัดให้มีขึ้นเพื่อการใช้งานในอนาคต เพื่อจุดประสงค์ในการบันทึกเอกสาร

ตัวอย่าง

1. ต่อไปนี้เป็นตัวอย่างที่เรียบง่ายของฟิลด์คำอธิบาย:

```
<ShortDescription>
  chuser parameters
</ShortDescription>
<Description>
  Parameter definition for the chuser command
</Description>
```

2. ตัวอย่างเดียวกันนี้ซึ่งใช้ข้อความแปดจากไฟล์ข้อความ `artexcat.cat`:

```
<ShortDescription>
<NLSCatalog catalog="artexcat.cat" setNum="12" msgNum="1">
  chuser parameters
</NLSCatalog>
</ShortDescription>
```

```

<Description>
<NLSCatalog catalog="artexcat.cat" setNum="12" msgNum="2">
  Parameter definition for the chuser command
</NLSCatalog>
</Description>

```

ข้อมูลที่เกี่ยวข้อง

การสนับสนุนแบบโกลบอลไลเซชัน

การสนับสนุนแบบโกลบอลไลเซชัน:

ส่วนนี้อธิบายถึงวิธีการอิมพลีเมนต์โกลบอลไลเซชัน ในฟิลด์คำอธิบายของแค็ตตาล็อก AIX Runtime Expert

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <Description>, <ShortDescription>

อิลิเมนต์พารามิเตอร์อาจมีหนึ่ง (และเพียงแค่นั้น) ในอิลิเมนต์ชาียนดังต่อไปนี้:

ตารางที่ 18. อิลิเมนต์ชาียน

อิลิเมนต์ชาียน	บังคับ	ตัวเลข	คำอธิบาย
<NLSCatalog>	ไม่ใช่	0 – 1	สตริงที่รวมอยู่ในแค็ตตาล็อกข้อความ
<NLSSmitHelp>	ไม่ใช่	0 – 1	สตริงที่รวมอยู่ในไฟล์ SMIT help HTML
<NLSCCommand>	ไม่ใช่	0 – 1	สตริงที่ออกโดยคำสั่ง AIX

NLS Catalog

รูปแบบของ NLS Catalog globalization ถูกใช้เมื่อข้อความโลคัลไลซ์ที่จะแสดงผลถูกรวมไว้ใน แค็ตตาล็อกข้อความที่มีอยู่ในรูปแบบ catgets()

อิลิเมนต์ <NLSCatalog> ประกอบด้วยแอ็ททริบิวต์ดังต่อไปนี้:

ตารางที่ 19. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
catalog	ใช่	สตริง	ชื่อของแค็ตตาล็อกที่มีข้อความอยู่
setNum	ใช่	จำนวนเต็ม	จำนวนชุดข้อความที่มีข้อความเก็บอยู่
msgNum	ใช่	จำนวนเต็ม	จำนวนข้อความในชุดข้อความ

หากแค็ตตาล็อกข้อความที่โลคัลไลซ์ไม่มีอยู่ ข้อความดีฟอลต์จะแสดงขึ้นแทน อีกทางเลือกหนึ่งก็คือ ข้อความดีฟอลต์ถูกรวมอยู่ในเนื้อหาของอิลิเมนต์ <NLSCatalog> ซึ่งเป็นแนวทางปฏิบัติที่แนะนำเพื่อจัดให้มีข้อความดีฟอลต์

NLS SMIT Help

รูปแบบของ NLS Smit Help globalization ถูกใช้เมื่อข้อความโลคัลไลซ์ที่จะแสดงผลมีอยู่แล้ว ในไฟล์ HTML วิธีใช้ของ SMIT

อิลิเมนต์ <NLSSmitHelp> มีแอตทริบิวต์ต่อไปนี้:

ตารางที่ 20. แอตทริบิวต์

แอตทริบิวต์	บังคับ	ชนิด	คำอธิบาย
<i>msgId</i>	ใช่	จำนวนเต็ม	ฟิลด์ <i>help_msg_id</i> ที่ถูกจัดให้มีอยู่ใน SMIT stanza

หากไฟล์วิธีใช้ที่โลคัลไลซ์ไม่มีอยู่ ข้อความดีฟอลต์จะแสดงขึ้นแทน อีกทางเลือกหนึ่งก็คือ ข้อความดีฟอลต์ถูกรวมอยู่ในเนื้อหาของอิลิเมนต์ <NLSSmitHelp> ซึ่งเป็นแนวทางปฏิบัติที่แนะนำเพื่อจัดให้มีข้อความดีฟอลต์

คำสั่ง NLS

รูปแบบของ NLS Command globalization ถูกใช้เมื่อข้อความโลคัลไลซ์ที่จะแสดงผลถูกส่ง โดยคำสั่ง AIX นี้เป็นกรณีสำหรับคำสั่งการปรับทั้งหมด (เช่น *no*, *vmo*) ซึ่งจัดให้มีแฟล็ก *-h* ที่จะแสดงข้อความวิธีใช้สำหรับพารามิเตอร์เฉพาะ

อิลิเมนต์ <NLSCCommand> มีแอตทริบิวต์ดังต่อไปนี้:

ตารางที่ 21. แอตทริบิวต์

แอตทริบิวต์	บังคับ	ชนิด	คำอธิบาย
<i>command</i>	คำสั่ง	สตริง	นิพจน์เชลล์ที่จะสั่งการ

ตัวอย่าง

- ตัวอย่างของอิลิเมนต์ <NLSCatalog> จากแค็ตตาล็อก *chssysParam.xml* AIX Runtime Expert ที่รวมข้อความดีฟอลต์
<Description>
<NLSCatalog catalog="artexcat.cat" setNum="10" msgNum="2">
Changes a subsystem definition in the subsystem object class.
</NLSCatalog>
</Description>
- ตัวอย่างของอิลิเมนต์ <NLSSmitHelp>:
<Description>
<NLSSmitHelp msgId="055136"/>
</Description>
- ตัวอย่างของอิลิเมนต์ <NLSCCommand> จากแค็ตตาล็อก *schedoParam.xml*:
<Description>
<NLSCCommand command="/usr/sbin/schedo -h maxspin | /usr/bin/tail -n +2"/>
</Description>

อิลิเมนต์ <SubCat>:

หมวดหมู่ย่อย, พารามิเตอร์ทางเลือก, เชื่อย่อยภายในแค็ตตาล็อก สามารถระบุได้โดยใช้อิลิเมนต์ <SubCat> ภายในไฟล์แค็ตตาล็อก

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <Catalog>, <SubCat>

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 22. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
id	ใช่	สตริง	ระบุชื่อของหมวดหมู่ย่อยของแค็ตตาล็อก ชื่อนี้ต้องไม่มีการซ้ำกันต่อแค็ตตาล็อก

อิลิเมนต์ชายนั้ดงต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 23. อิลิเมนต์ชายนั้

อิลิเมนต์ชายนั้	บังคับ	คำอธิบาย
<ShortDescription>	ไม่ใช่	คำอธิบายข้อความแบบสั้นสำหรับหมวดหมู่ย่อย
<Description>	ไม่ใช่	คำอธิบายข้อความแบบยาวสำหรับหมวดหมู่ย่อย
<SubCat>	ไม่ใช่	หมวดหมู่ย่อยแบบซ้อนกัน อิลิเมนต์นี้อาจเกิดขึ้นหลายครั้ง
<ParameterDef>	ไม่ใช่	มีคุณสมบัติต่างๆ ของพารามิเตอร์ อิลิเมนต์นี้อาจเกิดขึ้นหลายครั้ง

แอ็ททริบิวต์

หมวดหมู่ย่อยเป็นโลคัลสำหรับแค็ตตาล็อก

- id ของหมวดหมู่ย่อยมีลักษณะไม่ซ้ำกันภายในไฟล์แค็ตตาล็อก
- หลายแค็ตตาล็อกสามารถใช้ประโยชน์จากตัวตัวระบุมวุดหมู่ย่อยเดียวกัน

หมวดหมู่ย่อยที่กำหนดในแค็ตตาล็อกต้องตรงกับหมวดหมู่ย่อยที่รายงานในโปรไฟล์ตัวอย่างที่เชื่อมโยงกัน

ข้อมูลที่เกี่ยวข้อง

อิลิเมนต์<Description> และ <ShortDescription>

อิลิเมนต์ <SubCat>

อิลิเมนต์ <ParameterDef>

อิลิเมนต์ <ParameterDef>:

AIX Runtime Expert ถูกกำหนดอยู่ในไฟล์แค็ตตาล็อกโดยใช้อิลิเมนต์ <ParameterDef>

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <Catalog>, <ParameterDef>

แอตทริบิวต์ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 24. แอตทริบิวต์

แอตทริบิวต์	บังคับ	ชนิด	คำอธิบาย
<i>name</i>	ใช่	สตริง	ระบุชื่อของพารามิเตอร์ ชื่อนี้ต้องไม่มีการซ้ำกันต่อแค็ตตาล็อก
<i>type</i>	ใช่	สตริง	ระบุชนิดพารามิเตอร์ตามที่เห็นจากเครื่องหลัก
<i>targetClass</i>	ไม่ใช่	สตริง	ระบุคลาสเป้าหมายสำหรับพารามิเตอร์ หากมี
<i>reboot</i>	ไม่ใช่	บูลีน	เมื่อเป็น true แสดงว่าจำเป็นต้องรีบูต ค่าดีฟอลต์คือ false
<i>cfgmethod</i>	ไม่ใช่	สตริง	ระบุ <i>id</i> ของวิธีการกำหนดคอนฟิกที่ถูกกำหนดที่ระดับ <Catalog> ที่มีวิธีการที่ใช้สำหรับพารามิเตอร์
<i>constraint</i>	ไม่ใช่	สตริง	ระบุ <i>id</i> ของข้อบังคับที่กำหนดที่ระดับอิลิเมนต์ <Catalog> สำหรับไฟล์แค็ตตาล็อกปัจจุบัน
<i>priority</i>	ไม่ใช่	จำนวนเต็ม	ลำดับการสั่งการของพารามิเตอร์นี้ในวิธีการของชุดที่สัมพันธ์กับพารามิเตอร์อื่นๆ ในแค็ตตาล็อกนี้ ค่าดีฟอลต์คือ 0

อิลิเมนต์ชายน้ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 25. อิลิเมนต์ชายน้

อิลิเมนต์ชายน้	บังคับ	คำอธิบาย
<Description>	ไม่ใช่	คำอธิบายของพารามิเตอร์
<ConstraintDef>	ไม่ใช่	นิยามข้อบังคับพารามิเตอร์ (คำสั่งที่เป็นปัญหา)
<Get>	ไม่ใช่	นิยามวิธีการกำหนดคอนฟิกสำหรับปฏิบัติการ get อิลิเมนต์นี้อาจเกิดขึ้นหลายครั้ง
<Set>	ไม่ใช่	นิยามวิธีการกำหนดคอนฟิกสำหรับปฏิบัติการ set อิลิเมนต์นี้อาจเกิดขึ้นหลายครั้ง
<Diff>	ไม่ใช่	นิยามวิธีการกำหนดคอนฟิกสำหรับปฏิบัติการ diff
<Discover>	ไม่ใช่	นิยามวิธีการกำหนดคอนฟิกสำหรับการค้นพบเป้าหมาย

แอตทริบิวต์

ตารางที่ 26. แอ็ททริบิวต์

แอ็ททริบิวต์	คำอธิบาย
name	แอ็ททริบิวต์ name เป็นการแสดงพารามิเตอร์ภายในไฟล์แค็ตตาล็อกแบบไม่มีการซ้ำ ดูหัวข้อแอ็ททริบิวต์ชื่อเพื่อรับทราบข้อมูลเพิ่มเติม
type	<p>แอ็ททริบิวต์ type ที่บังคับ แสดงชนิดของค่าพารามิเตอร์ ค่าที่สนับสนุนได้แก่:</p> <ul style="list-style-type: none"> • string สำหรับสตริงที่เป็นตัวอักษรและตัวเลข; • integer สำหรับค่าตัวเลข; • integer-bi สำหรับค่าตัวเลขที่มีส่วนเติมท้ายเป็นตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็ก (เลือกได้) K, M, G, T, P หรือ E สำหรับ “kilo”, “mega”, “giga”, “tera”, “peta” และ “exa” ส่วนเติมท้ายเหล่านี้ถูกแปลเป็นกำลังเท่ากับ 1024; • integer-si สำหรับค่าตัวเลขที่มีส่วนต่อท้าย SI ทางเลือก ทั้งนี้ มีลักษณะที่เหมือนกับ integer-bi type, ยกเว้นส่วนต่อท้าย โดยถูกแปลเป็นพลังเท่ากับ 1000 • boolean สำหรับค่าบูลีน ค่าที่สนับสนุนได้แก่ 0 และ 1 • binary สำหรับค่าไบนารีซึ่งถอดรหัสเป็นสตริง 64 ไบโพรไฟล์
reboot	<p>ค่าดีฟอลต์สำหรับแอ็ททริบิวต์ ‘reboot’ ของบูลีนคือ “false” หากการเปลี่ยนแปลงพารามิเตอร์ต้องการการรีบูทเพื่อให้เห็นผล พารามิเตอร์นี้ต้องมีแอ็ททริบิวต์ ‘reboot’ ของตนเองตั้งค่าเป็น “true”</p> <p>AIX Runtime Expert ไม่มีการรีบูทระบบเอง ตามค่าดีฟอลต์ คำสั่ง artexset จะไม่ทำบังคับให้มีการตั้งค่าพารามิเตอร์ reboot หากไโปรไฟล์มีพารามิเตอร์ reboot คำสั่งจะล้มเหลว:</p> <p>0590-502: ไโปรไฟล์มีพารามิเตอร์ที่กำหนดให้มีการรีบูท ไโปรไฟล์ไม่ได้ถูกตั้งค่า ใช้แฟล็ก -l เพื่อบังคับการตั้งค่าพารามิเตอร์ทั้งหมด</p> <p>ถ้าเรียกด้วย แฟล็ก -l ที่เหมาะสม คำสั่ง artexset ตั้งค่าและเตือนผู้ใช้งานว่าจำเป็นต้องรีบูทเพื่อให้การเปลี่ยนแปลง มีผลใช้งาน:</p> <p>0590-206 จำเป็นต้องมีการดำเนินการเองหลังปฏิบัติการเพื่อให้การเปลี่ยนแปลงแสดงผล ไโปรครีบูทระบบ</p>
priority	<p>ตามค่าดีฟอลต์ พารามิเตอร์ถูกตั้งค่าในลำดับที่ไม่มีภาระบุโดยคำสั่ง artexset แอ็ททริบิวต์ <i>priority</i> สามารถใช้เปลี่ยนลักษณะการทำงานนี้และบังคับให้พารามิเตอร์ได้รับการตั้งค่า ก่อนหรือหลังพารามิเตอร์อื่นๆ</p> <p>ลำดับความสำคัญตามค่าดีฟอลต์คือ 0 แอ็ททริบิวต์ <i>priority</i> สามารถใช้เพื่อเปลี่ยนแปลงลำดับความสำคัญดีฟอลต์นี้เป็นค่าเต็มระหว่าง -2147483648 ถึง 2147483647 พารามิเตอร์ที่มีลำดับความสำคัญสูงกว่าจะถูกสั่งการก่อนพารามิเตอร์ที่มีลำดับความสำคัญต่ำกว่า ลำดับที่พารามิเตอร์ที่มีลำดับความสำคัญเดียวกันถูกตั้งค่านั้น ไม่ได้ถูกกำหนด</p>
targetClass	<p>บางพารามิเตอร์จะต้องเชื่อมโยงกับเป้าหมาย ดังที่อธิบายอยู่ในส่วน “อิลิเมนต์เป้าหมาย” ของไโปรไฟล์ อิลิเมนต์เหล่านั้นต้องมีแอ็ททริบิวต์ <i>targetClass</i> ที่ตั้งค่าเป็นรายการคลาสเป้าหมายที่สนับสนุนที่มีการค้นด้วยคอมมา</p>

ตารางที่ 26. แอ็ททริบิวต์ (ต่อ)

แอ็ททริบิวต์	คำอธิบาย
<i>cfgmethod</i>	อิลิเมนต์ <code><ParameterDef></code> สามารถรับอิลิเมนต์บรรทัดคำสั่งจากอิลิเมนต์ <code><CfgMethod></code> โดยการอ้างอิงแอ็ททริบิวต์ <i>id</i> ของวิธีการกำหนดคอนฟิกคำสั่งด้วยแอ็ททริบิวต์ <i>cfgmethod</i> สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีการกำหนดคอนฟิก โปรดอ้างอิงส่วน อิลิเมนต์ <code><CfgMethod></code>
<i>constraint</i>	อิลิเมนต์ <code><ParameterDef></code> สามารถใช้แอ็ททริบิวต์ <i>constraint</i> เพื่ออ้างอิงแอ็ททริบิวต์ <i>id</i> ของอิลิเมนต์ <code><ConstraintDef></code> ซึ่งเป็นการแสดงว่า พารามิเตอร์นั้นจะต้องอยู่ภายใต้ข้อบังคับนั้น สำหรับข้อมูลเพิ่มเติมเกี่ยวกับข้อบังคับ โปรดอ้างอิงส่วน “อิลิเมนต์ <code><ConstraintDef></code> ”

Examples

- ต่อไปนี้เป็นตัวอย่างของนิยามพารามิเตอร์ที่มีชนิดจำนวนเต็มทางเลือก: *kernel_heap_size* จากไฟล์แค็ตตาล็อก vmoParam.xml:

```
<ParameterDef name="kernel_heap_psize" type="integer-bi">
```

เมื่อแยกค่าของพารามิเตอร์นี้ผ่านทางคำสั่ง *artexget* ผลลัพธ์ก็คือ บางสิ่งที่มีความคล้ายคลึง (นำมาจากโปรไฟล์ผลลัพธ์)

```
<Parameter name="kernel_heap_psize" value="16M"... />
```

ค่าพารามิเตอร์จะถูกแปลในลักษณะแตกต่างออกไปขึ้นอยู่กับชนิด:

- เนื่องจากประกาศว่าเป็นชนิด *integer-bi*, ค่าจึงเป็น 16M= 16,777,216
- หากชนิดเป็น *integer-si* ค่าจะเป็น “16M”=16,000,000

- ตัวอย่างของพารามิเตอร์ *binary*: ฐานข้อมูลหลายชิ้นที่น่าเชื่อถือ *tsd.dat* ในแค็ตตาล็อก *tsdParam.xml*:

```
<ParameterDef name="tsdatabse" type="binary">
```

- ตัวอย่างของพารามิเตอร์ที่มีแอ็ททริบิวต์ *reboot* ชนิดของ พารามิเตอร์ *dump* ในแค็ตตาล็อก *sysdumpdevParam.xml*:

```
<ParameterDef name="type_of_dump" type="string" reboot="true">
```

- ตัวอย่างของพารามิเตอร์ที่มีหนึ่งคลาสเป้าหมาย: พารามิเตอร์ *addr* จากแค็ตตาล็อก *mktcpipParam.xml* ใช้กับ อินเตอร์เฟซเครือข่ายเฉพาะ:

```
<ParameterDef name="addr" type="string" cfgmethod="mktcpip" targetClass="interface">
```

- ตัวอย่างของพารามิเตอร์ที่มีหลายคลาสเป้าหมาย: พารามิเตอร์ *naming specification* จาก *coreParam.xml* ใช้กับเฉพาะ ผู้ใช้ (รูท ผู้ดูแลระบบ เกสต์ เป็นต้น) ในรีจิสทรีเฉพาะ (ไฟล์, LDAP)

```
<ParameterDef name="namingspecification" type="string" reboot="true" targetClass="user,registry"
cfgmethod="coremgt">
```

- ตัวอย่างการใช้แอ็ททริบิวต์ *cfgmethod*: สำหรับปฏิบัติการ `<Get type="current">` พารามิเตอร์คงที่ของแค็ตตาล็อก *chlicenseParam.xml* ได้รับอิลิเมนต์ `<Command>` จากวิธีการกำหนดคอนฟิก *chlicense* แต่ก็ยังกำหนด `<Filter>` และ `<Mask>` ด้วยตนเองบนเครื่องอีกด้วยสำหรับปฏิบัติการเดียวกันนี้:

```
<CfgMethod id="chlicense">
  <Get type="current">
    <Command>lslicense -c -A</Command>
  </Get>
</CfgMethod>
<ParameterDef name="fixed" cfgmethod="chlicense" type="integer">
  <Get type="current">
```

```

    <Filter>tail -n 1 | cut -d: -f3</Filter>
    <Mask value="1">(.)</Mask>
  </Get>
</ParameterDef>

```

7. ตัวอย่างการใช้แอตทริบิวต์ constraint: พารามิเตอร์ authorizations ของแค็ตตาล็อก authParam.xml จะต้องอยู่ภายใต้ข้อบังคับ setkst ซึ่งกำหนดไว้ก่อนหน้านี้ในอิลิเมนต์ <ConstraintDef>:

```

<ParameterDef name="authorizations" cfgmethod="cat" constraint="setkst" type="string">

```

ข้อมูลที่เกี่ยวข้อง

แอตทริบิวต์ name

แอตทริบิวต์ Name:

ชื่อของพารามิเตอร์มักถูกกำหนดโดยคำสั่งที่ใช้เพื่อรับหรือตั้งค่าพารามิเตอร์

พารามิเตอร์ names ต้องไม่ซ้ำกันภายในไฟล์แค็ตตาล็อก ซึ่งจำเป็นที่จะต้องตรวจสอบให้แน่ใจว่าอิลิเมนต์ <Parameter> ในโปรไฟล์สามารถเชื่อมโยงกับอิลิเมนต์ <ParameterDef> ในไฟล์แค็ตตาล็อก

- หากคำสั่ง get แสดงหลายค่าพารามิเตอร์ อิลิเมนต์ <Mask> จะสามารถใช้เพื่อแยกหลายพารามิเตอร์ออกมาจากเอาต์พุต คำสั่งเดียวได้ เรื่องนี้มีความเป็นไปได้หากชื่อของพารามิเตอร์ตรงกับชื่อที่ใช้ในเอาต์พุตของคำสั่ง get
- หากคำสั่ง set ยอมรับค่าพารามิเตอร์หลายค่า ดังนั้น ลำดับ %n และ %v1 จะสามารถใช้ในอิลิเมนต์ <Argument> ได้ เพื่อตั้งค่าหลายพารามิเตอร์ด้วยคำสั่งเดียว เรื่องนี้มีความเป็นไปได้หากชื่อของพารามิเตอร์ตรงกับชื่อที่ใช้ในเอาต์พุตของคำสั่ง set

ตัวอย่าง

1. ตัวอย่าง: คำสั่ง raso -a ที่ใช้ในแค็ตตาล็อก rasoParam.xml แสดงหนึ่งพารามิเตอร์ต่อหนึ่งบรรทัดที่แสดง:

```

kern_heap_noexec = 0
kernel_noexec = 1
mbuf_heap_noexec = 0
mtrc_commonbufsize = 485

```

ในกรณีง่ายนี้ ชื่อพารามิเตอร์จะเป็น kernel_heap_noexec, kernel_noexec, เป็นต้น

2. ตัวอย่าง: คำสั่งที่ใช้ในวิธีการกำหนดคอนฟิก get ของแค็ตตาล็อก acctctlParam.xml แสดงผลลัพธ์ที่ยากกว่าสำหรับแยกวิเคราะห์ ไม่เพียงแต่ชื่อของพารามิเตอร์จะถูกรวมอยู่ในประโยคที่ไม่ได้รับการจัดรูปแบบ แต่ทั้งชื่อพารามิเตอร์และค่าต่างๆ ก็ได้รับการโลคัลไลซ์ วิธีการกำหนดคอนฟิก get จะต้องรันคำสั่ง ในระหว่างตั้งค่าตัวแปรสภาพแวดล้อม LANG=C และในแต่ละบรรทัด ให้แทนที่ค่าหลักด้วยชื่อพารามิเตอร์ที่เกี่ยวข้อง:

```

Advanced Accounting ไม่ได้รันอยู่
การแจ้งเตือนทางอีเมลปิดทำงาน
อีเมลแอคเคาน์ตเดรสปัจจุบันที่จะถูกใช้ยังไม่ถูกตั้งค่า
Recover CPU accounting time ในโหมด turbo เป็น False

```

ในตัวอย่างข้างต้น ชื่อตัวแปรที่ถูกล็อกได้แก่ accounting, email, email_addr และ turacct

ข้อมูลที่เกี่ยวข้อง

- อิลิเมนต์ <Parameter>
- อิลิเมนต์ <Mask>

- การขยายอิลิเมนต์ของบรรทัดคำสั่ง

อิลิเมนต์ <ConstraintDef>:

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <Catalog>, <ParameterDef>

แอตทริบิวต์ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 27. แอตทริบิวต์

แอตทริบิวต์	บังคับ	ชนิด	คำอธิบาย
id	ไม่ใช่*	สตริง	ระบุชื่อข้อบังคับพารามิเตอร์

*แอตทริบิวต์นี้ต้องได้รับการระบุสำหรับอิลิเมนต์ <Constraint> ที่กำหนดในระดับแค็ตตาล็อก

อิลิเมนต์ชายนดังกล่าวต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 28. อิลิเมนต์ชายน

อิลิเมนต์ชายน	บังคับ	คำอธิบาย
<Description>	ไม่ใช่	ข้อความอธิบายของคำสั่ง disruptive
<PreOp>	ไม่ใช่	ปฏิบัติการ disruptive ที่จะรันก่อนตั้งค่าพารามิเตอร์
<PostOp>	ไม่ใช่	ปฏิบัติการ disruptive ที่จะรันหลังจากตั้งค่าพารามิเตอร์
<BuiltIn>	ไม่ใช่	ปฏิบัติการ disruptive ในตัว อิลิเมนต์นี้อาจเกิดขึ้นหลายครั้ง

การใช้งาน

บางพารามิเตอร์การปรับและกำหนดคอนฟิกอาจต้องการปฏิบัติการ disruptive สำหรับการเปลี่ยนแปลงค่าแสดงผล ปฏิบัติการ disruptive คือปฏิบัติการใดก็ตามที่อาจยับยั้งการเข้าถึงเซิร์ฟเวอร์หรืออุปกรณ์ ปฏิบัติการ disruptive ตามปกติ จะดำเนินการรีสตาร์ทเดมอน ยกเลิกการเมาต์ หรือเมาต์ระบบไฟล์ อันจะทำให้การต่อแอดปเตอร์เครือข่ายออฟไลน์หรือออนไลน์ โปรแกรม AIX Runtime Expert ใช้ข้อบังคับเพื่อแสดงให้เห็นว่า พารามิเตอร์ต้องการให้ดำเนินการปฏิบัติการ disruptive เพื่อให้การเปลี่ยนแปลงแสดงผล อิลิเมนต์ <ConstraintDef> ถูกใช้เพื่อกำหนดข้อบังคับเช่นนี้

ข้อบังคับนี้สามารถกำหนดได้เป็นอย่างใดอย่างหนึ่ง:

- ภายในอิลิเมนต์ <ParameterDef> หากข้อบังคับใช้กับเฉพาะพารามิเตอร์เดียว
- ในระดับแค็ตตาล็อก อิลิเมนต์ <ConstraintDef> มีแอตทริบิวต์ id ที่จะยอมให้ข้อบังคับถูกอ้างอิงภายหลังในอิลิเมนต์ <ParameterDef>

ข้อบังคับในตัว

อิลิเมนต์ `<BuiltIn>` ไม่มีส่วนประกอบของแอ็ททริบิวต์หรืออิลิเมนต์ชายน์

ข้อบังคับในตัวทำหน้าที่กำหนดปฏิบัติการต่างๆ ที่อยากจะเข้ารหัสในเครื่องหลัก ขณะนี้ มีเพียงหนึ่งข้อบังคับในตัวเท่านั้น: `bosboot` ความแตกต่างระหว่างข้อบังคับในตัวกับปฏิบัติการ disruptive อื่นๆ ก็คือ คำสั่ง `bosboot` ไม่เคยถูกรันโดย AIX Runtime Expert เครื่องหลักจะเพียงเตือนว่า จำเป็นต้องมี `bosboot` เพื่อให้การเปลี่ยนแปลงต่างๆ แสดงผล

0590-206 จำเป็นต้องมีการดำเนินการเองหลังปฏิบัติการ เพื่อให้การเปลี่ยนแปลงแสดงผล
โปรดดำเนินการ `bosboot`

ข้อบังคับ PreOp และ PostOp

อิลิเมนต์ `<PreOp>` เป็นตัวกำหนดคำสั่งบังคับ (นิพจน์เซลล์) เพื่อที่จะได้ถูกรันก่อนที่ค่าพารามิเตอร์จะถูกตั้งโดยวิธีการกำหนดคอนฟิก set อิลิเมนต์ `<PostOp>` กำหนดคำสั่งบังคับให้ถูกรันหลังจากสั่งการวิธีการกำหนดคอนฟิก set

อิลิเมนต์ `<ConstraintDef>` ต้องมี 0 หรือหนึ่งอิลิเมนต์ชายน์ `<PreOp>` และ 0 หรือหนึ่งอิลิเมนต์ชายน์ `<PostOp>`

ตัวอย่าง

1. ตัวอย่างของข้อบังคับในตัว (ที่ระดับหมวดหมู่)

```
<ConstraintDef id="bosboot">
  <Description>
    <NLSCatalog catalog="artexcat.cat" setNum="51" msgNum="3">
      bosboot
    </NLSCatalog>
  </Description>
  <Built>Inbosboot</BuiltIn>
</ConstraintDef>
```

2. ตัวอย่างของข้อบังคับ `<PreOp>`: ข้อบังคับ `clic` ในหมวดหมู่ `trustchkParam.xml` โปรดทราบว่าในตัวอย่างนี้ คำสั่ง `preop` ไม่ได้รันสิ่งใด เพียงแต่ตรวจสอบการมีอยู่ของส่วนขยายเคอร์เนลที่กำหนดโดยคำสั่ง `set` หากไม่มีการติดตั้งส่วนขยายเคอร์เนล ข้อบังคับที่กำหนดในอิลิเมนต์ `<PreOp>` จะล้มเหลว และคำสั่ง `set` จะไม่ถูกรัน:

```
<ConstraintDef id="clic">
  <Description>
    <NLSCatalog catalog="artexcat.cat" setNum="48" msgNum="3">
      Check that the clic.rte kernel extension is installed.
    </NLSCatalog>
  </Description>
  <Pre>Opls1pp -l "clic*"</PreOp>
</ConstraintDef>
```

3. ตัวอย่างของข้อบังคับ `<PostOp>`: ตารางข้อบังคับ `set` ยาง Kernel Security Tables ในแค็ตตาล็อก `authParam.xml` ฐานข้อมูลที่ถูกแก้ไขจำเป็นต้องถูกโหลดเฉพาะเพียงครั้งเดียวในเคอร์เนล หลังจากที่มีการแก้ไขทั้งหมดถูกดำเนินการแล้ว

```
<ConstraintDef id="setkst">
  <Description>
    <NLSCatalog catalog="artexcat.cat" setNum="5" msgNum="3">
      Send the authorizations database to the KST (Kernel Security Tables)
    </NLSCatalog></Description>
  <PostOp>/usr/sbin/setkst -t auth &gt;/dev/null</PostOp>
</ConstraintDef>
```

อิลิเมนต์ <CfgMethod>:

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <Catalog>

แอ็ททริบิวต์ ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 29. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
id	ใช่	สตริง	ระบุชื่อของวิธีการกำหนดคอนฟิก

อิลิเมนต์ชายน้ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 30. อิลิเมนต์ชายน้

อิลิเมนต์ชายน้	บังคับ	ตัวเลข	คำอธิบาย
<Get>	ไม่ใช่	0 - 1	นิยามวิธีการกำหนดคอนฟิกสำหรับปฏิบัติการ get อิลิเมนต์นี้อาจเกิดขึ้นหลายครั้ง
<Set>	ไม่ใช่	0 - 1	นิยามวิธีการกำหนดคอนฟิกสำหรับปฏิบัติการ set อิลิเมนต์นี้อาจเกิดขึ้นหลายครั้ง
<Diff>	ไม่ใช่	0 - 1	นิยามวิธีการกำหนดคอนฟิกสำหรับปฏิบัติการ diff
<Discover>	ไม่ใช่	0 - 1	นิยามวิธีการกำหนดคอนฟิกสำหรับการค้นพบเป้าหมาย
<Property>	ไม่ใช่	0 - ค่าใดๆ ก็ตาม	กำหนดคุณสมบัติแก่พารามิเตอร์โดยใช้เมธอดการกำหนดคอนฟิก

การใช้งาน

อิลิเมนต์ <CfgMethod> ทำหน้าที่กำหนดวิธีการกำหนดคอนฟิกซึ่งสามารถถูกอ้างอิงถึงภายหลังโดยพารามิเตอร์โดยใช้แอ็ททริบิวต์ *cfgmethod* ของอิลิเมนต์ <ParameterDef> จากนั้น พารามิเตอร์จะได้รับอิลิเมนต์ทั้งหมดที่ถูกกำหนดภายใต้วิธีการกำหนดคอนฟิกที่อ้างอิง

การใช้คอนฟิกูเรชันอาจมีประโยชน์เหนือกว่านิยามโลคัล ทั้งนี้ขึ้นอยู่กับพารามิเตอร์:

- ช่วยทำให้ไฟล์แค็ตตาล็อกเรียบง่าย หลีกเลี่ยงการทำซ้ำอิลิเมนต์บรรทัดคำสั่งเดียวกันสำหรับหลายพารามิเตอร์
- ช่วยให้หลายพารามิเตอร์ได้รับการดำเนินการโดยคำสั่งเดียว

ตัวอย่าง

แค็ตตาล็อก *vmParam.xml* กำหนดพารามิเตอร์เป็นจำนวนมากซึ่งทั้งหมดล้วนแต่ใช้วิธีการกำหนดคอนฟิกเดียวกัน ต่อไปนี้เป็นเวอร์ชันที่เรียบง่ายของแค็ตตาล็อกนี้:

```

<Catalog id="vmoParam" version="2.1">
  <CfgMethod id="vmo">
    <Get type="current">
      <Command>/usr/sbin/vmo -a</Command>
      <Mask name="1" value="2">[[[:space:]]*(.*) = (.*)</Mask>
    </Get>

    <Get type="nextboot">
      <Command>/usr/sbin/vmo -r -a</Command>
      <Mask name="1" value="2">[[[:space:]]*(.*) = (.*)</Mask>
    </Get>

    <Set type="permanent">
      <Command>/usr/sbin/vmo -p%a</Command>
    <Argument>%n=%v1</Argument>
  </Set>

    <Set type="nextboot">
      <Command>/usr/sbin/vmo -r%a</Command>
      <Argument>%n=%v1</Argument>
    </Set>
  </CfgMethod>

  <ParameterDef name="ame_maxfree_mem" cfgmethod="vmo" type="integer" />
  <ParameterDef name="ame_min_ucpool_size" cfgmethod="vmo" type="integer" />
  <ParameterDef name="ame_minfree_mem" cfgmethod="vmo" type="integer" />
  ...
</Catalog>

```

ข้อมูลที่เกี่ยวข้อง

- การสร้างบรรทัดคำสั่ง
- อิลิเมนต์ <Get>
- อิลิเมนต์ <Set>

อิลิเมนต์ <Get>:

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <CfgMethod>, <ParameterDef>

แอ็ททริบิวต์ ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 31. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
<i>type</i>	ใช่	สตริง	ระบุชนิดของคำสั่ง <i>get</i> (ปัจจุบันหรือ <i>nextboot</i>)

อิลิเมนต์ชายนั่งต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 32. อิลิเมนต์ชายน

อิลิเมนต์ชายน	บังคับ	ตัวเลข	คำอธิบาย
<Command>	ไม่ใช่	0 - 1	คำสั่ง
<Argument>	ไม่ใช่	0 - 1	อาร์กิวเมนต์บรรทัดคำสั่ง
<Stdin>	ไม่ใช่	0 - 1	อาร์กิวเมนต์ที่สนับสนุนโดยอิลิเมนต์ <Stdin>
<Filter>	ไม่ใช่	0 - 1	ตัวกรอง
<Mask>	ไม่ใช่	0 - 1	มาสก์ดักจับเอาต์พุต
<Prereq>	ไม่ใช่	0 - ค่าใดๆ ก็ตาม	กำหนดสิ่งที่จำเป็นต้องมีให้แกการดำเนินการ get

อิลิเมนต์ <Command> ต้องถูกกำหนดสำหรับ แต่ละพารามิเตอร์ ที่ระดับ <CfgMethod> หรือโดยตรง ที่ระดับ <ParameterDef>

การใช้งาน

อิลิเมนต์ <Get> อธิบายวิธีการที่ค่าของพารามิเตอร์เฉพาะถูกดักจับ ซึ่งสามารถใช้โดยตรงภายใต้อิลิเมนต์ <ParameterDef> หรือภายใต้อิลิเมนต์ <CfgMethod> ที่ถูกอ้างอิงถึงในอิลิเมนต์ <ParameterDef> โดยใช้แอตทริบิวต์ *cfgmethod* หรือใช้ความสามารถสองอย่างนี้รวมกัน

อิลิเมนต์ Get ทั้งสองควรถูกกำหนดสำหรับแต่ละพารามิเตอร์ หนึ่งสำหรับแต่ละค่าที่สนับสนุนของแอตทริบิวต์ *type*:

- Get *type*="current" ระบุวิธีการที่จะถูกรันเพื่อเรียกใช้ค่ารันไทม์ของพารามิเตอร์
- Get *type*="nextboot" ระบุวิธีการที่จะถูกรันเพื่อเรียกใช้ค่าที่พารามิเตอร์จะมีหลังจากการรีบูทของระบบ
- วิธีการ get ที่จะถูกรัน ขึ้นอยู่กับปฏิบัติการที่กำลังดำเนินการ:
 - หากคำสั่ง *artexget* ถูกเรียกด้วยแฟล็ก *-r* วิธีการ get ปัจจุบันก็จะถูกใช้
 - หากคำสั่ง *artexget* ถูกเรียกด้วยแฟล็ก *-n* วิธีการ *nextboot* จะถูกใช้
 - หากคำสั่ง *artexget* ถูกเรียกด้วยแฟล็ก *-p* วิธีการที่ถูกรันจะขึ้นอยู่กับอินพุตของพารามิเตอร์สำหรับแอตทริบิวต์ *applyType* วิธีการ get ปัจจุบัน ถูกใช้สำหรับพารามิเตอร์ที่มีแอตทริบิวต์ *applyType* ซึ่งถูกตั้งค่าเป็นรันไทม์และวิธีการ *nextboot get* ถูกใช้สำหรับพารามิเตอร์ที่มีแอตทริบิวต์ *applyType* ของรีบูท

ข้อมูลที่เกี่ยวข้อง

การสร้างบรรทัดคำสั่ง

<อิลิเมนต์ Mask>

อิลิเมนต์ <Set>:

อิลิเมนต์ <Set> เป็นตัวกำหนดวิธีการสร้างบรรทัดคำสั่งเพื่อตั้งค่าพารามิเตอร์

ไวยากรณ์

อิลิเมนต์พารেন্ট: <CfgMethod>, <ParameterDef>

แอ็ททริบิวต์ ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 33. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
<i>type</i>	ใช่	สตริง	ระบุชนิดคำสั่ง set เป็น current หรือ nextboot

อิลิเมนต์ขายน้ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 34. อิลิเมนต์ขายน้

อิลิเมนต์ขายน้	บังคับ	ตัวเลข	คำอธิบาย
<Command>	ไม่ใช่	0 – 1	คำสั่ง
<Argument>	ไม่ใช่	0 – 1	อาร์กิวเมนต์บรรทัดคำสั่ง
<Stdin>	ไม่ใช่	0 – 1	อาร์กิวเมนต์ Stdin
<Prereq>	ไม่ใช่	0 – ค่าใดๆ ก็ตาม	กำหนดสิ่งที่จำเป็นต้องมีให้แกการดำเนินการ <Set>

หมายเหตุ: อิลิเมนต์ <Command> ต้องถูกกำหนดสำหรับ แต่ละพารามิเตอร์ ที่ระดับ <CfgMethod> หรือโดยตรง ที่ระดับ <ParameterDef>

การใช้งาน

อิลิเมนต์ <Set> มีสามประเภทที่สามารถกำหนดสำหรับแต่ละพารามิเตอร์ ซึ่งระบุโดยแอ็ททริบิวต์ *type* ที่กำหนด:

- Set *type*=“current” เป็นตัวกำหนดปฏิบัติการ set ซึ่งจะเปลี่ยนเฉพาะค่าพารามิเตอร์สำหรับเซชันปัจจุบัน การเปลี่ยนแปลงใดๆ ก็ตามที่ทำดำเนินการโดยใช้ปฏิบัติการ set จะสูญหายไปหลังจากรีบูตระบบ
- Set *type*=“nextboot” กำหนดปฏิบัติการ set ซึ่งจะเปลี่ยนเฉพาะค่าที่พารามิเตอร์จะใช้หลังจากการรีบูตระบบครั้งถัดไป ค่าปัจจุบันไม่ถูกแก้ไข
- Set *type*=“permanent” กำหนดปฏิบัติการ set ที่จะเปลี่ยนทั้งค่า current และ nextboot ของพารามิเตอร์

ชนิดของปฏิบัติการ set ที่รัน ถูกตัดสินโดยใช้พารามิเตอร์ที่รวมอยู่เมื่อคำสั่ง *artexset* ถูกรัน โดยใช้แอ็ททริบิวต์ *applyType* ของพารามิเตอร์ในโปรไฟล์ ตารางดังต่อไปนี้ ทำหน้าที่สรุปวิธีการ set ที่ถูกรัน ทั้งนี้ขึ้นอยู่กับวิธีการ set ที่กำหนดในไฟล์ แค็ตตาล็อก ทั้งนี้ขึ้นอยู่กับแอ็ททริบิวต์ *applyType* สำหรับพารามิเตอร์:

ตารางที่ 35. เมฆอด Set - ตั้งค่าชนิดเมฆอด ที่กำหนด และพารามิเตอร์แอ็ททริบิวต์ *applyType*

current	nextboot	permanent	runtime	nextboot
0	0	0	ไม่ได้ตั้งค่า (ข้อผิดพลาด)	ไม่ได้ตั้งค่า (ข้อผิดพลาด)
0	0	1	ตั้งค่า permanent	ไม่ได้ตั้งค่า (ข้อผิดพลาด)
0	1	0	ตั้งค่า nextboot + warning	ตั้งค่า nextboot
0	1	1	ตั้งค่า permanent	ไม่ได้ตั้งค่า (ข้อผิดพลาด)
1	0	0	ตั้งค่า current + warning	ตั้งค่า nextboot
1	0	1	ตั้งค่า permanent	ไม่ได้ตั้งค่า (ข้อผิดพลาด)

ตารางที่ 35. เมธอด Set - ตั้งค่าชนิดเมธอด ที่กำหนด และพารามิเตอร์แอ็พพลิเคชัน applyType (ต่อ)

current	nextboot	permanent	runtime	nextboot
1	1	0	ตั้งค่า current ตั้งค่า nextboot	ตั้งค่า nextboot
1	1	1	ตั้งค่า permanent	ตั้งค่า nextboot

ข้อมูลที่เกี่ยวข้อง

การสร้างบรรทัดคำสั่ง

อิลิเมนต์ <Diff>:

อิลิเมนต์ <Diff> ทำหน้าที่กำหนดวิธีการสร้างบรรทัดคำสั่งเพื่อเปรียบเทียบสองค่าของพารามิเตอร์

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <CfgMethod>, <ParameterDef>

อิลิเมนต์ชายนัดต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 36. อิลิเมนต์ชายนัด

อิลิเมนต์ชายนัด	บังคับ	คำอธิบาย
<Command>	ไม่ใช่	คำสั่ง
<Argument>	ไม่ใช่	อาร์กิวเมนต์บรรทัดคำสั่ง
<Stdin>	ไม่ใช่	อาร์กิวเมนต์ Stdin
<Filter>	ไม่ใช่	ตัวกรอง
<Mask>	ไม่ใช่	มาสก์ดักจับเอาต์พุต

หมายเหตุ: อิลิเมนต์ <Command> ต้องถูกกำหนดสำหรับ แต่ละพารามิเตอร์ ที่ระดับ <CfgMethod> หรือโดยตรง ที่ระดับ <ParameterDef>

การใช้งาน

โดยปกติแล้ว อิลิเมนต์ <Diff> ไม่ได้ถูกบังคับ เนื่องจากเฟรมเวิร์กวิธีเปรียบเทียบสองค่าพารามิเตอร์เป็นการภายในโดยใช้ชนิด (string, integer, integer-bi, binar เป็นต้น). อย่างไรก็ตาม ในกรณีที่มีการเปรียบเทียบภายในไม่ได้รับการปรับสำหรับพารามิเตอร์เฉพาะ ก็มีความเป็นไปได้ที่จะใช้คำสั่งภายนอกแทน

ตัวอย่าง

อิลิเมนต์ <Diff> ต่อไปนี้ สามารถถูกใช้สำหรับพารามิเตอร์ส่วนใหญ่ แม้ว่า การใช้ฟังก์ชันการเปรียบเทียบภายในจะมีประสิทธิภาพมากกว่า อิลิเมนต์ <Diff> ใช้คำสั่ง diff เพื่อเปรียบเทียบสองไฟล์ที่มีสองค่า:

```
<Diff>
  <Command>/usr/bin/diff %f1 %f2; echo $?</Command>
</Diff>
```

ข้อมูลที่เกี่ยวข้อง

การสร้างบรรทัดคำสั่ง

<อิลิเมนต์ Mask>

อิลิเมนต์ <Discover>:

อิลิเมนต์ <Discover> กำหนดวิธีการสร้างบรรทัดคำสั่งเพื่อค้นหาเป้าหมายสำหรับพารามิเตอร์ที่สนับสนุน

ไวยากรณ์

อิลิเมนต์พารেন্ট: <CfgMethod>, <ParameterDef>

อิลิเมนต์ชายนัดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 37. อิลิเมนต์ชายนัด

อิลิเมนต์ชายนัด	บังคับ	ตัวเลข	คำอธิบาย
<Command>	no	0 - 1	คำสั่ง
<Prereq>	no	0 - ค่าใดๆ ก็ตาม	กำหนดสิ่งที่จำเป็นต้องมีให้แกการดำเนินการ discover

หมายเหตุ: อิลิเมนต์ <Command> ต้องถูกกำหนดสำหรับ แต่ละพารามิเตอร์ ที่ระดับ <CfgMethod> หรือโดยตรง ที่ระดับ <ParameterDef>

การใช้งาน

คำสั่ง discover ถูกใช้เพื่อให้ได้มาซึ่งรายการอินสแตนซ์เป้าหมายสำหรับพารามิเตอร์ที่กำหนด

เอาต์พุตของคำสั่ง discover สำหรับพารามิเตอร์ที่สนับสนุนคลาสเป้าหมาย N มีรูปแบบดังต่อไปนี้:

```
class_1=inst_1_1;class_2=inst_2_1;...;class_N=inst_N_1
class_1=inst_1_2;class_2=inst_2_2;...;
class_N=inst_N_2class_1=inst_1_3;
class_2=inst_2_3;...;class_N=inst_N_3
...
```

คำสั่ง `artexget` ทำหน้าที่สร้างและรันคำสั่ง discover สำหรับพารามิเตอร์ที่มีคุณสมบัติตรงตามหนึ่งในเกณฑ์ดังต่อไปนี้:

- มีอิลิเมนต์ <Target> ที่มีแอตทริบิวต์ `class` และ `instance` ว่าง <Target class="" instance="" />
- มีอย่างน้อยหนึ่งอิลิเมนต์ <Target> ที่มีแอตทริบิวต์ `match`: <Target class="..." match="..." />

คำสั่ง `artexset` ยังกำหนดเพิ่มเติมให้เป็นไปตามสองเกณฑ์ดังต่อไปนี้จึงจะมีคุณสมบัติครบ:

- คำสั่ง `artexset` ถูกเรียกด้วยแฟล็ก `-d`
- อิลิเมนต์ <Parameter> ในโปรไฟล์มีแอตทริบิวต์ `setDiscover` ที่ตั้งค่าเป็น true

ตัวอย่าง

1. แก้ดัดไฟล์ `mktcpipParam.xml` ใช้คำสั่ง `discover` ต่อไปนี้เพื่อจัดหารายการอินเตอร์เฟซเครือข่ายที่กำหนด บนระบบ:

```
<Discover>
  <Command>
    /usr/sbin/lsdev -C -c if -F "name" | /usr/bin/sed -e 's/^/interface=/'
  </Command>
</Discover>
```

คำสั่งนี้ให้เอาต์พุตดังต่อไปนี้:

```
interface=en0
interface=et0
interface=lo0
```

2. แก้ดัดไฟล์ chuserParam.xml ใช้คำสั่ง **discover** ดังต่อไปนี้ เพื่อไปยังรายการผู้ใช้ทั้งหมดสำหรับโมดูลการพิสูจน์ตัวตนที่สามารถโหลดได้:

```
<Discover>
  <Command>
    /usr/sbin/lsuser -a registry ALL |
    /usr/bin/sed -e "s/\\(.*) registry=\\(.*)/module=\\2;user=\\1/g"
  </Command>
</Discover>
```

คำสั่งนี้ให้เอาต์พุตดังต่อไปนี้:

```
module=LDAP;user=daemon
module=LDAP;user=bin
module=LDAP;user=sys
module=LDAP;user=adm
...
module=files;user=root
module=files;user=daemon
module=files;user=bin
module=files;user=sys
module=files;user=adm
...
```

อิลิเมนต์ <Command>:

อิลิเมนต์ <Command> กำหนดคำสั่งพื้นฐานที่ใช้เพื่อการปฏิบัติการที่กำหนดโดยอิลิเมนต์ parent

ไวยากรณ์

พารามิเตอร์อิลิเมนต์: <Get>, <Set>, <Diff>, <Discover>, <PrereqDef>, <Prereq>, <PropertyDef>, <Property>, <Command>

การใช้งาน

เนื้อหาของอิลิเมนต์ <Command> ถูกขยาย ดังที่อธิบายในส่วน การขยายอิลิเมนต์บรรทัดคำสั่ง และถูกรวมอยู่กับอิลิเมนต์บรรทัดคำสั่งอื่นๆ เพื่อสร้างเป็นบรรทัดคำสั่งที่สมบูรณ์ ดูส่วน การสร้างบรรทัดคำสั่ง เพื่อรับทราบรายละเอียดเพิ่มเติม

บางอักขระมักจะถูกพบอยู่ในนิพจน์เซลล์ เช่น <, > และ & และไม่ได้รับการอนุญาตในเอกสาร XML อักขระเหล่านี้ต้องถูกแทนที่ด้วยเอนทิตี XML ที่ตรงกัน:

ตารางที่ 38. เอนทิตี XML

อักขระ	เอนทิตี XML
<	<
>	>
&	&

อีกทางเลือกหนึ่งคือ ส่วน CDATA สามารถใช้ได้หากนิพจน์มีหลายอักขระเหล่านี้ ส่วน CDATA เริ่มต้นด้วย <![CDATA[และสิ้นสุดด้วย]]>

อิลิเมนต์ <Command> ต้องถูกกำหนดสำหรับแต่ละปฏิบัติการที่ได้รับการสนับสนุนสำหรับของแต่ละพารามิเตอร์ที่ระดับ <CfgMethod> หรือที่ระดับ <ParameterDef>

ตัวอย่าง

แค็ตตาล็อก envParam.xml กำหนดโปรไฟล์ที่เรียกเป็นพารามิเตอร์ ซึ่งเป็นตัวแทนของเนื้อหาของไฟล์ /etc/profile สำหรับพารามิเตอร์นี้ อิลิเมนต์ <Get> ใช้คำสั่ง cat เพื่อตัดกับเนื้อหาของไฟล์ /etc/profile:

```
<ParameterDef name="profile">  
  <Get type="current">  
    <Command>/usr/bin/cat /etc/environment</Command>  
  </Get>  
</ParameterDef>
```

ข้อมูลที่เกี่ยวข้อง

การสร้างบรรทัดคำสั่ง

การขยายอิลิเมนต์ของบรรทัดคำสั่ง

อิลิเมนต์ <Argument>:

ไวยากรณ์

พารেন্টอิลิเมนต์: <Get>, <Set>, <Diff>, <PrereqDef>, <Prereq>, <PropertyDef>, <Property>

การใช้งาน

เนื้อหาของอิลิเมนต์ <Argument> ถูกขยายตั้งที่ได้อธิบายไว้ในส่วน การขยายอิลิเมนต์บรรทัดคำสั่ง และถูกรวมอยู่กับ <Command> และหรืออิลิเมนต์ <Stdin> เพื่อสร้างบรรทัดคำสั่งที่สมบูรณ์ ดูส่วน การสร้างบรรทัดคำสั่ง เพื่อรับทราบรายละเอียดเพิ่มเติม

บางอักขระมักจะถูกพบอยู่ในนิพจน์เซลล์ เช่น <, > และ & และไม่ได้รับการอนุญาตในเอกสาร XML อักขระเหล่านี้ต้องถูกแทนที่ด้วยเอนทิตี XML ที่ตรงกัน:

ตารางที่ 39. เอนทิตี XML

อักขระ	เอนทิตี XML
<	<
>	>
&	&

อีกทางเลือกหนึ่งคือ ส่วน CDATA สามารถใช้ได้หากนิพจน์มีหลายอักขระเหล่านี้ ส่วน CDATA เริ่มต้นด้วย <![CDATA[และ สิ้นสุดด้วย]]>

ตัวอย่าง

แก้ดตาบล็อก vmoParam.xml ใช้เอนทิตี <Argument> เพื่อเพิ่มอาร์กิวเมนต์ในคำสั่ง vmo สำหรับแต่ละพารามิเตอร์ vmo ใน โปรไฟล์:

```
<CfgMethod id="vmo">
  <Set type="permanent">
    <Command>/usr/sbin/vmo -p%a</Command>
    <Argument> -o %n=%v1</Argument>
  </Set>
</CfgMethod>
```

ข้อมูลที่เกี่ยวข้อง

การสร้างบรรทัดคำสั่ง

การขยายเอนทิตีของบรรทัดคำสั่ง

เอนทิตี <Stdin>:

ไวยากรณ์

พารেন্টเอนทิตี: <Get>, <Set>, <Diff>, <PrereqDef>, <Prereq>, <PropertyDef>, <Property>

การใช้งาน

เนื้อหาของเอนทิตี <Stdin> ถูกขยายดังที่ได้อธิบายไว้ในส่วน เอนทิตีการขยายบรรทัดคำสั่ง และข้อมูลที่ได้อธิบายไปยังอินพุตมาตรฐานของบรรทัดคำสั่งที่สร้างขึ้นสำหรับปฏิบัติการที่กำหนดในเอนทิตีพารেন্ট

ตัวอย่าง

แก้ดตาบล็อก envParam.xml กำหนดโปรไฟล์ที่เรียกเป็นพารามิเตอร์ ซึ่งเป็นตัวแทนของเนื้อหาของไฟล์ /etc/profile สำหรับพารามิเตอร์นี้ ปฏิบัติการ set จะเขียนค่าของพารามิเตอร์ไปยังอินพุตมาตรฐานของคำสั่ง cat เพื่อเขียนทับไฟล์ /etc/profile:

```

<ParameterDef name="profile">
  <Set type="permanent">
    <Command>/usr/bin/cat &gt; /etc/profile</Command>
    <Stdin>%v1</Stdin>
  </Set>
</Get>

```

ข้อมูลที่เกี่ยวข้อง

การสร้างบรรทัดคำสั่ง

การขยายอิลิเมนต์ของบรรทัดคำสั่ง

อิลิเมนต์ <Filter>:

ไวยากรณ์

พารามิเตอร์อิลิเมนต์: <Get>, <Diff>, <PropertyDef>, <Property>

การใช้งาน

เนื้อหาของอิลิเมนต์ <Filter> เป็นคำสั่งที่เอาต์พุตของบรรทัดคำสั่งที่สร้างขึ้นสำหรับปฏิบัติการซึ่งกำหนดในอิลิเมนต์พารามิเตอร์ถูกส่งไปหาในฐานะที่เป็นอินพุต

บางอักขระซึ่งมักจะถูกพบในนิพจน์เซลล์ เช่น <, > และ & ไม่ได้รับอนุญาตในเอกสาร XML อักขระเหล่านี้จำเป็นต้องถูกแทนที่โดยเอนทิตี XML ที่ตรงกัน:

ตารางที่ 40. เอนทิตี XML

อักขระ	เอนทิตี XML
<	<
>	>
&	&

อีกทางเลือกหนึ่งคือ ส่วน CDATA สามารถใช้ได้หากนิพจน์มีหลายอักขระเหล่านี้ ส่วน CDATA เริ่มต้นด้วย <![CDATA[และสิ้นสุดด้วย]]>

ตัวอย่าง

แค็ตตาล็อก nfsParam.xml ใช้อิลิเมนต์ <Filter> สำหรับปฏิบัติการ get ของพารามิเตอร์ v4_root_node เพื่อแยกโหนดรากออกจากเอาต์พุตของคำสั่ง nfd -getnode:

```

<ParameterDef id="v4_root_node">
  <Get type="current">
    <Command>
      /usr/sbin/nfsd -getnodes
    </Command>
  <Filter>

```

```

        /usr/bin/awk -F: 'NR == 2 { printf("%s", $1) }'
    </Filter>
</Get>
</ParameterDef>

```

ข้อมูลที่เกี่ยวข้อง

การสร้างบรรทัดคำสั่ง

อิลิเมนต์ <Mask>:

ไวยากรณ์

พารามิเตอร์อิลิเมนต์: <Get>, <Diff>, <Discover> (ภายใต้ <SeedDef> เท่านั้น), <PropertyDef>, <Property>

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุนเมื่อใช้ใน อิลิเมนต์ <Get> หรือ <Diff>:

ตารางที่ 41. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
<i>name</i>	ไม่ใช่	จำนวนเต็ม	ระบุดัชนีของนิพจน์ย่อยที่ตรงกับชื่อของพารามิเตอร์ ค่าที่ใช้ได้คือ 1 และ 2
<i>value</i>	ไม่ใช่	จำนวนเต็ม	ระบุดัชนีของนิพจน์ย่อยที่ตรงกับค่าของพารามิเตอร์ ค่าที่ใช้ได้คือ 1 และ 2

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน เมื่อใช้ภายใต้อิลิเมนต์ย่อย <Discover> ของอิลิเมนต์ <SeedDef>:

ตารางที่ 42. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
<i>catalog</i>	yes	จำนวนเต็ม	ระบุดัชนีของนิพจน์ย่อยที่ตรงกับชื่อของแค็ตตาล็อก ค่าที่ใช้ได้คือ 1, 2 และ 3
<i>name</i>	yes	จำนวนเต็ม	ระบุดัชนีของนิพจน์ย่อยที่ตรงกับชื่อของพารามิเตอร์ ค่าที่ใช้ได้คือ 1, 2 และ 3
<i>target</i>	ไม่ใช่	จำนวนเต็ม	ระบุดัชนีของนิพจน์ย่อยที่ตรงกับปลายทางของพารามิเตอร์ ค่าที่ใช้ได้คือ 1, 2 และ 3

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน เมื่อใช้ภายใต้อิลิเมนต์ <PropertyDef> หรือ <Property>:

ตารางที่ 43. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
value	ไม่ใช่	จำนวนเต็ม	ระบุดัชนีของนิพจน์ย่อยที่ตรงกับชื่อของพารามิเตอร์ ต้องตั้งค่าเป็น "1" ถ้าระบุ

การใช้งาน

อิลิเมนต์ <Mask> กำหนดนิพจน์ทั่วไปที่ใช้กับแต่ละบรรทัดของเอาต์พุต คำสั่งเพื่อแยกข้อมูลจากบรรทัดเหล่านั้น ข้อมูลที่ถูกแยก จะขึ้นอยู่กับตำแหน่งที่ใช้อิลิเมนต์ <Mask>

ถ้าไม่มี แอ็ททริบิวต์ถูกระบุ บรรทัดสุดท้ายในเอาต์พุตคำสั่งที่ตรงกับนิพจน์ทั่วไปจะถูกใช้เพื่อแยกข้อมูล ข้อมูล ที่แยกเป็นส่วนหนึ่งของบรรทัดที่ตรงกับนิพจน์ทั่วไป เมื่อใช้ภายใต้อิลิเมนต์ <Get> หรือ <Diff> ข้อมูลที่แยกจะถูกใช้เป็นค่าพารามิเตอร์ เมื่อใช้ภายใต้อิลิเมนต์ <PropertyDef> หรือ <Property> ข้อมูลที่แยกถูกใช้เป็ค่าคุณสมบัติ

ถ้าระบุแอ็ททริบิวต์ value เท่านั้น ต้องถูกตั้งค่าเป็น 1 และนิพจน์ทั่วไปต้องมีหนึ่งนิพจน์ย่อยเท่านั้น บรรทัดสุดท้ายในเอาต์พุตคำสั่งที่ตรงกับนิพจน์ทั่วไปถูกใช้เพื่อแยกข้อมูล ข้อมูล ที่แยกเป็นส่วนหนึ่งของบรรทัดที่ตรงกับนิพจน์ย่อยแรก (และ ตรงกันเท่านั้น) เมื่อใช้ภายใต้อิลิเมนต์ <Get> หรือ <Diff> ข้อมูลที่แยกจะถูกใช้เป็นค่าพารามิเตอร์ เมื่อใช้ภายใต้ อิลิเมนต์ <PropertyDef> หรือ <Property> ข้อมูล ที่แยกถูกใช้เป็ค่าคุณสมบัติ

ถ้าแอ็ททริบิวต์ name และ value ถูกระบุ หนึ่งในแอ็ททริบิวต์เหล่านั้นต้องถูกตั้งค่าเป็น 1 และแอ็ททริบิวต์อื่นๆ ต้องถูกตั้งค่าเป็น 2 และนิพจน์ทั่วไปต้องมีสองนิพจน์ย่อย โดย name และ value ถูกแยกจากแต่ละบรรทัดของ เอาต์พุตคำสั่งที่ตรงกับนิพจน์ทั่วไป เมื่อใช้ในอิลิเมนต์ <Get> ชื่อถูกใช้เป็นชื่อพารามิเตอร์ และค่าเป็น ค่าพารามิเตอร์ เมื่อใช้ในอิลิเมนต์ <Diff> ชื่อ ถูกใช้เป็นชื่อพารามิเตอร์ และค่าเป็นถูกใช้เป็นผลลัพธ์ การเปรียบเทียบ การใช้ฟังก์ชันนี้ ค่าของหลายๆ พารามิเตอร์สามารถ ถูกแยกโดยใช้คำสั่ง get เดียว และ หลายพารามิเตอร์สามารถเปรียบเทียบโดยใช้คำสั่ง diff เดียว

เมื่อ ใช้ในอิลิเมนต์ย่อย <Discover> ของอิลิเมนต์ <SeedDef> แอ็ททริบิวต์ catalog และ name ต้องถูกระบุ ชื่อแค็ตตาล็อก และชื่อพารามิเตอร์ถูกแยกจากแต่ละบรรทัดของเอาต์พุตคำสั่ง ที่ตรงกับนิพจน์ทั่วไป ถ้าแค็ตตาล็อกที่ตรงกับ ชื่อแค็ตตาล็อกที่แยกถูกพบในระบบ และถ้ามี นิยามสำหรับพารามิเตอร์ที่ตรงกับชื่อพารามิเตอร์ ที่แยก พารามิเตอร์จะถูกแทรกในโปรไฟล์อาร์กิวเมนต์ปลายทาง ที่เป็นทางเลือกสามารถถูกเพิ่มเพื่อแยกนิยามปลายทางสำหรับแต่ละพารามิเตอร์ ที่ค้นพบ นิยามเป้าหมายต้องตามด้วยรายการที่ค้นด้วยเครื่องหมายคอมมาของคู่ในรูปแบบ class=instance เช่น
class1=instance1;class2=instance2;...

ตัวอย่าง

1. แค็ตตาล็อก vmoParam.xml ใช้อิลิเมนต์ <Mask> ที่มีแอ็ททริบิวต์ name และ value เพื่อแยกค่าพารามิเตอร์ทั้งหมดออกมาจากคำสั่ง vmo -a เดียว:

```
<CfgMethod id="vmo">
  <Get type="current">
    <Command>/usr/sbin/vmo -a</Command>
    <Mask name="1" value="2">[:space:]]*(.*) = (.*)</Mask>
  </Get>
</CfgMethod>
```

2. แค็ตตาล็อก vmoParam.xml ถูกเขียนในลักษณะที่หนึ่งคำสั่งแยกต่างหากถูกใช้เพื่อดักจับค่าของแต่ละพารามิเตอร์ จากนั้น อิลิเมนต์ <Mask> จะถูกใช้กับแอ็ททริบิวต์ value ที่ตั้งค่าไว้ และโดยไม่มีแอ็ททริบิวต์ name:

```

<CfgMethod id="vmo">
  <Get type="current">
    <Command>/usr/sbin/vmo -o %n</Command>
    <Mask value="1"> = (.*)</Mask>
  </Get>
</CfgMethod>

```

3. หรือโดยใช้นิพจน์ทั่วไปที่จับคู่ค่าที่ตรงกันเท่านั้น:

```

<CfgMethod id="vmo">
  <Get type="current">
    <Command>/usr/sbin/vmo -o %n</Command>
    <Mask>[^\ ]*${</Mask>
  </Get>
</CfgMethod>

```

จากสามตัวอย่างด้านบน ตัวอย่างแรก มีประสิทธิภาพสูงสุด เนื่องจากต้องการคำสั่งเดียวเพื่อ บันทึกพารามิเตอร์คำสั่ง vmo ทั้งหมดเท่านั้น ตัวอย่าง 2 และ 3 จะสร้างคำสั่งแยกต่างหากสำหรับแต่ละพารามิเตอร์คำสั่ง vmo เนื่องจากชื่อพารามิเตอร์ถูกใช้ในอิลิเมนต์ <Command>

4. อิลิเมนต์ <SeedDef> ต่อไปนี้กำหนด seed ที่สามารถใช้เพื่อค้นหาแอตทริบิวต์ทั้งหมดของอุปกรณ์ทั้งหมด ซึ่งใช้ปลายทางเพื่อกำหนดอุปกรณ์ที่ทำงานด้วย:

```

<SeedDef name="devAttr">
  <Discover>
    <Command>
      /usr/sbin/lisdev -F 'name class subclass type' |
      while read DEV CLASS SUBCLASS TYPE
      do
        /usr/sbin/lisattr -F attribute -l $DEV |
        while read PAR
        do
          echo device=$DEV devParam.$CLASS.$SUBCLASS.$TYPE $PAR
        done
      done
    </Command>
    <Mask target="1" catalog="2" name="3">(.) (.) (.) <Mask>
  </Discover>
</SeedDef>

```

คำสั่ง discovery พิมพ์แต่ละแอตทริบิวต์อุปกรณ์ที่ค้นพบ บนบรรทัดต่างหาก โดยใช้รูปแบบต่อไปนี้:

```
device=DeviceName devParam.Class.Subclass.Type AttributeName
```

ตัวอย่างเช่น

```

device=en0 devParam.if.EN.en tcp_recvspace
device=en0 devParam.if.EN.en tcp_sendspace
device=en0 devParam.adapter.vdevice.IBM,1-lan alt_addr
device=en0 devParam.adapter.vdevice.IBM,1-lan chksum_offload

```

ข้อมูลที่เกี่ยวข้อง

การสร้างบรรทัดคำสั่ง

อิลิเมนต์ <SeedDef>:

อิลิเมนต์ <SeedDef> กำหนด seed ที่สามารถใช้ในโปรไฟล์โดยใช้อิลิเมนต์ <Seed>

ไวยากรณ์

อิลิเมนต์พารามิเตอร์: <Catalog>

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 44. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
name	yes	สตริง	ระบุชื่อของ seed ชื่อนี้ต้อง เป็นค่า เฉพาะสำหรับแต่ละแค็ตตาล็อก

อิลิเมนต์ชายนต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 45. อิลิเมนต์ชายนต์

อิลิเมนต์ชายนต์	บังคับ	คำอธิบาย
<Discover>	yes	ระบุคำสั่งที่ใช้เพื่อค้นหา พารามิเตอร์

การใช้งาน

Seeds ถูกใช้เพื่อ ค้นหาพารามิเตอร์แบบไดนามิกระหว่างการดำเนินการ get

เมื่อ ออกคำสั่ง `artexget` แต่ละอิลิเมนต์ <Seed> ในโปรไฟล์อื่นจะถูกขยายเป็นอย่างน้อยหนึ่งอิลิเมนต์ <Parameter> ตามกฎที่กำหนดในอิลิเมนต์ <SeedDef> ที่ตรงกัน ของไฟล์แค็ตตาล็อก กระบวนการนี้ถูกกว่าการค้นหาพารามิเตอร์ จากนั้นคำสั่ง `artexget` ดำเนินการตามปกติกับโปรไฟล์ที่ขยาย

อิลิเมนต์ <SeedDef> มีเฉพาะอิลิเมนต์ย่อย <Discover> ที่กำหนดคำสั่ง เพื่อรัน และมาสก์เพื่อแยกชื่อพารามิเตอร์ ชื่อแค็ตตาล็อก (เป็นรายการที่ค้นด้วย โคลอน โดยไม่มีส่วนขยาย .xml) และอาจกำหนดปลายทางจากเอาต์พุตของคำสั่ง (โดยใช้รูปแบบ `class1=instance1;class2=instance2;...`) สำหรับเอาต์พุตแต่ละบรรทัด แค็ตตาล็อกแรกจากรายการที่ค้นด้วย โคลอนที่พบในระบบถูกโหลด ถ้าพบนิยามพารามิเตอร์ ในแค็ตตาล็อกนี้ จากนั้นพารามิเตอร์ถูกสร้างในโปรไฟล์ เอาต์พุตที่มีปลายทางถูกแยกจากบรรทัด บรรทัด จากเอาต์พุตคำสั่งที่ไม่ตรงกับมาสก์ หรือที่ซึ่งไม่พบ ไฟล์แค็ตตาล็อก หรือที่ไม่มีนิยามพารามิเตอร์ถ้าพบ ในไฟล์พารามิเตอร์ ถูกข้าม

ตัวอย่าง

1. แค็ตตาล็อกต่อไปนี้กำหนดอิลิเมนต์ <SeedDef> เรียกว่า `vmoTunables` ที่ ค้นหา `vmo tunables seed` ที่ไม่ถูกจำกัดทั้งหมดที่สนับสนุนโดย AIX Runtime Expert:

```
<?xml version="1.0" encoding="UTF-8"?>
<Catalog id="vmoSeed">
  <SeedDef name="vmoTunables">
    <Discover>
      <Command>/usr/sbin/vmo -x | /usr/bin/awk -F, '{ print "vmoParam:" $1 }'</Command>
```

```

    <Mask catalog="1" name="2">(.*):(.*)/Mask>
  </Discover>
</SeedDef>
</Catalog>

```

คำสั่ง discovery พิมพ์แต่ละค่าที่ปรับได้ บนบรรทัดแยก นำหน้าด้วยชื่อของแค็ตตาล็อกที่กำหนด ค่าที่ปรับได้:

```

...
vmoParam:enhanced_affinity_vmpool_limit
vmoParam:esid_allocator
vmoParam:force_realias_lite
vmoParam:kernel_heap_psize
...

```

โปรไฟล์ต่อไปนี้จะใช้ *vmo tunables seed* เพื่อบันทึก *vmo tunables seed* ที่ไม่ถูกจำกัดทั้งหมดที่ไม่สนับสนุนโดย AIX

Runtime Expert:

```

<?xml version="1.0" encoding="UTF-8"?>
<Profile>
  <Catalog id="vmoSeed">
    <Seed name="vmoTunables"/>
  </Catalog>
</Profile>

```

เมื่อรันคำสั่ง `artexget -r` บนโปรไฟล์ คำสั่งสร้างโปรไฟล์ที่คล้ายกับ ตัวอย่างต่อไปนี้:

```

<?xml version="1.0" encoding="UTF-8"?>
<Profile>
  <Catalog id="vmoParam">
    ...
    <Parameter name="enhanced_affinity_vmpool_limit" value="10"/>
    <Parameter name="esid_allocator" value="0"/>
    <Parameter name="force_realias_lite" value="0"/>
    <Parameter name="kernel_heap_psize" value="65536" applyType="nextboot" reboot="true"/>
    ...
  </Catalog>
</Profile>

```

2. อิลิเมนต์ `<SeedDef>` ต่อไปนี้กำหนด seed ที่ใช้เพื่อค้นหาแอตทริบิวต์ทั้งหมดของอุปกรณ์ทั้งหมด อิลิเมนต์ใช้ seed ปลายทางเพื่อกำหนดอุปกรณ์ที่ทำงานด้วย:

```

<SeedDef name="devAttr">
  <Discover>
    <Command>
      /usr/sbin/lsdev -F 'name class subclass type' |
      while read DEV CLASS SUBCLASS TYPE
      do
        /usr/sbin/lsattr -F attribute -l $DEV |
        while read PAR
        do
          echo device=$DEV devParam.$CLASS.$SUBCLASS.$TYPE:devParam.$CLASS
          . $SUBCLASS:devParam.$CLASS $PAR
        done
      done
    </Command>
    <Mask target="1" catalog="2" name="3">(.*) (.*) (.*)</Mask>
  </Discover>
</SeedDef>

```

คำสั่ง discovery พิมพ์แต่ละแอ็ททริบิวต์ อุปกรณ์ที่ค้นพบบนบรรทัดแยก โดยใช้รูปแบบต่อไปนี้:

```
device=DeviceName devParam.Class.Subclass.Type:devParam.Class.Subclass:devParam.Class  
AttributeName
```

ตัวอย่างเช่น:

```
device=en0 devParam.if.EN.en:devParam.if.EN:devParam.if tcp_recvspace  
device=en0 devParam.if.EN.en:devParam.if.EN:devParam.if tcp_sendspace  
device=ent0 devParam.adapter.vdevice.IBM,l-lan:devParam.adapter.vdevice:devParam.adapter  
alt_addr  
device=ent0 devParam.adapter.vdevice.IBM,l-lan:devParam.adapter.vdevice:devParam.adapter  
chksum_offload
```

อิลิเมนต์ <Prereq>:

อิลิเมนต์ <Prereq> กำหนดสิ่งที่จำเป็นต้องมีก่อน การดำเนินการ <Get>, <Set> และ <Discover>

ไวยากรณ์

พารามิเตอร์อิลิเมนต์: <Get>, <Set> และ <Discover>

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 46. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
id	no	สตริง	ระบุตัวบ่งชี้เฉพาะ

อิลิเมนต์ชายนี่ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 47. อิลิเมนต์ชายนี่

อิลิเมนต์ชายนี่	บังคับ	คำอธิบาย
<Command>	no	คำสั่ง
<Argument>	no	อาร์กิวเมนต์บรรทัดคำสั่ง
<Stdin>	no	อาร์กิวเมนต์ที่สนับสนุนโดยอิลิเมนต์ <Stdin>
<ErrMsg>	no	ข้อความที่จะพิมพ์ถ้าสิ่งที่จำเป็นต้องมีล้มเหลว

หมายเหตุ: อิลิเมนต์ <Command> ต้องถูกกำหนดสำหรับ แต่ละสิ่งที่จำเป็นต้องมี: ที่ระดับ <ParameterDef>, ที่ระดับ <CfgMethod> หรือในอิลิเมนต์ <PrereqDef>

การใช้งาน

Prereqs คือ คำสั่งที่กำหนดเงื่อนไขการประมวลผลของการดำเนินการ <Get>, <Set> และ <Discover> สำหรับพารามิเตอร์ที่ใช้ การดำเนินการ <Get>, <Set> หรือ <Discover> พารามิเตอร์ซึ่งคำสั่ง prereq ล้มเหลว (โค้ดส่งคืนไม่เท่ากับศูนย์) จะถูกข้าม และข้อความแสดงข้อผิดพลาดที่กำหนด ในสิ่งที่จำเป็นต้องมีถูกแสดงผล

อิลิเมนต์ <Prereq> กำหนดสิ่งที่จำเป็นต้องมีแก่การดำเนินการ <Get>, <Set> และ <Discover> พารามิเตอร์สิ่งที่จำเป็นต้องมีถูกกำหนดแบบโลคัลภายใต้อิลิเมนต์ <Prereq> หรือสืบทอดจาก อิลิเมนต์ <Prereq> หรือ <PrereqDef> ระดับสูงที่มีแอตทริบิวต์ *id* ที่ตรงกัน

พารามิเตอร์มีสิ่งที่จำเป็นต้องมีทั้งหมดที่กำหนดแบบโลคัลภายใต้อิลิเมนต์ <ParameterDef> รวมทั้งสิ่งที่จำเป็นต้องมีมีคุณสมบัติที่กำหนดในเมธอด การกำหนดคอนฟิกของพารามิเตอร์ ถ้าใช้เมธอดการกำหนดคอนฟิก ผลก็คือถ้าสิ่งที่จำเป็นต้องมีถูกกำหนดในอิลิเมนต์ <CfgMethod> อิลิเมนต์ <ParameterDef> ทั้งหมดที่ใช้เมธอด การกำหนดคอนฟิกจะมีสิ่งที่จำเป็นต้องมีโดยอัตโนมัติ (แม้ว่าบางส่วนของอิลิเมนต์เหล่านั้นอาจกำหนดสิ่งที่จำเป็นต้องมีใหม่แบบโลคัล)

อิลิเมนต์ <Command>, <Argument>, <Stdin> และ <ErrorMessage> ที่กำหนดสิ่งที่จำเป็นต้องมีสำหรับการดำเนินการที่กำหนดถูกค้นหาตามลำดับนี้:

- ภายใต้อิลิเมนต์ย่อย <Prereq> ของการดำเนินการที่สัมพันธ์กัน ของอิลิเมนต์ <ParameterDef>
- ถ้าอิลิเมนต์ <ParameterDef> มีแอตทริบิวต์ *cfgmethod* ภายใต้อิลิเมนต์ย่อย <Prereq> ที่มี *id* ตรงกันของ การดำเนินการที่สัมพันธ์กันของเมธอดการกำหนดคอนฟิก
- ภายใต้อิลิเมนต์ <PrereqDef> ของแค็ตตาล็อกที่มี *id* ตรงกัน

ตัวอย่าง

ตัวอย่างต่อไปนี้จะกำหนด สิ่งที่จำเป็นต้องมีที่ตรวจสอบว่าพารามิเตอร์ **netaddr** และ **netaddr6** ถูกใช้บนระบบเดียวกันกับที่ลูกบันทึก:

```
<ParameterDef name="netaddr" type="string" targetClass="device" cfgmethod="attr">
  <Set type="permanent">
    <Prereq>
      <Command>[[ `usr/bin/uname -f` = %p[nodeId] ]]</Command>
      <ErrorMessage>Parameter cannot be applied to a different node</ErrorMessage>
    </Prereq>
  </Set>
</ParameterDef>
```

```
<ParameterDef name="netaddr6" type="string" targetClass="device" cfgmethod="attr">
  <Set type="permanent">
    <Prereq>
      <Command>[[ `usr/bin/uname -f` = %p[nodeId] ]]</Command>
      <ErrorMessage>Parameter cannot be applied to a different node</ErrorMessage>
    </Prereq>
  </Set>
</ParameterDef>
```

ในตัวอย่างนี้ การทดสอบถูกรัน สองครั้ง: หนึ่งครั้งสำหรับพารามิเตอร์ **netaddr** และอีกครั้งสำหรับพารามิเตอร์ **netaddr6** การประมวลผลคู่นี้เนื่องจากแต่ละพารามิเตอร์มีสิ่งที่จำเป็นต้องมีของตนเอง ที่มีอิลิเมนต์ <Command> ของตนเอง ดูที่ “อิลิเมนต์ <PrereqDef>” ในหน้า 128 สำหรับ ตัวอย่างที่ต้องการในการรันการทดสอบหนึ่งครั้งเท่านั้น

ข้อมูลที่เกี่ยวข้อง

- “การสร้างบรรทัดรับคำสั่ง” ในหน้า 132
- “อิลิเมนต์ <PrereqDef>” ในหน้า 128

อิลิเมนต์ <PrereqDef>:

อิลิเมนต์ <PrereqDef> ที่สามารถใช้ภายหลัง ในอิลิเมนต์ <Prereq>

ไวยากรณ์

พารามิเตอร์อิลิเมนต์: <Catalog>

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 48. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
name	yes	สตริง	ระบุชื่อของคุณสมบัติ

อิลิเมนต์ชายนั้ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 49. อิลิเมนต์ชายนั้

อิลิเมนต์ชายนั้	บังคับ	คำอธิบาย
<Command>	no	คำสั่ง
<Argument>	no	อาร์กิวเมนต์บรรทัดคำสั่ง
<Stdin>	no	อาร์กิวเมนต์ที่สนับสนุนโดยอิลิเมนต์ <Stdin>
<ErrMsg>	no	ข้อความที่จะพิมพ์ถ้าสิ่งที่เป็นต้องมีล้มเหลว

หมายเหตุ: อิลิเมนต์ <Command> ต้องถูกกำหนดสำหรับ แต่ละสิ่งที่เป็นต้องมี: ที่ระดับ <ParameterDef>, ที่ระดับ <CfgMethod> หรือในอิลิเมนต์ <PrereqDef>

การใช้งาน

Prereq คือ คำสั่งที่กำหนดเงื่อนไขการรันของการดำเนินการ <Get>, <Set> และ <Discover> สำหรับพารามิเตอร์ที่ใช้การดำเนินการ <Get>, <Set> หรือ <Discover> พารามิเตอร์ซึ่งคำสั่ง **prereq** ล้มเหลว (ได้ส่งคืนไม่เท่ากับศูนย์) จะถูกข้าม และข้อความแสดงข้อผิดพลาดที่กำหนด ในสิ่งที่เป็นต้องมีถูกแสดงผล

อิลิเมนต์ <PrereqDef> กำหนดสิ่งที่เป็นต้องมีสิ่งที่เป็นต้องมีนี้สามารถเชื่อมโยงกับ การดำเนินการของพารามิเตอร์ หรือ เมธอดการกำหนดคอนฟิกภายหลังโดยใช้อิลิเมนต์ <Prereq> ที่มีแอ็ททริบิวต์ *id* เดียวกัน

ตัวอย่าง

ตัวอย่างต่อไปนี้กำหนดสิ่งที่เป็นต้องมี *nodeId* และกำหนดให้แก่พารามิเตอร์ **netaddr** และ **netaddr6**:

```
<PrereqDef id="nodeId">
  <Command>[[ `usr/bin/uname -f` = %p[nodeId] ]]</Command>
  <ErrMsg>Parameter cannot be applied to a different node</ErrMsg>
</PrereqDef>

<ParameterDef name="netaddr" type="string" targetClass="device" cfgmethod="attr">
```

```

<Set type="permanent">
  <Prereq id="nodeId"/>
</Set>
<Property name="nodeId"/>
</ParameterDef>

<ParameterDef name="netaddr6" type="string" targetClass="device" cfgmethod="attr">
  <Set type="permanent">
    <Prereq id="nodeId"/>
  </Set>
  <Property name="nodeId"/>
</ParameterDef>

```

ในตัวอย่างนี้ การทดสอบถูกเรียกใช้งาน เพียงครั้งเดียว เนื่องจากสองพารามิเตอร์ใช้อิเลเมนต์ <Command> เดียวกันสำหรับ สิ่งที่เป็นต้องมียของตน และบรรทัดรับคำสั่งที่สร้างจะเหมือนกัน สำหรับทั้งสองพารามิเตอร์

ข้อมูลที่เกี่ยวข้อง

- “การสร้างบรรทัดรับคำสั่ง” ในหน้า 132
- “อิเลเมนต์ <Prereq>” ในหน้า 126

อิเลเมนต์ <Property>:

อิเลเมนต์ <Property > กำหนดคุณสมบัติให้แก่ พารามิเตอร์หรือเมธอดการกำหนดคอนฟิก

ไวยากรณ์

อิเลเมนต์พารามิเตอร์: <CfgMethod>, <ParameterDef>

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 50. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
name	yes	สตริง	ระบุชื่อของคุณสมบัติ

อิเลเมนต์ชายนั้ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 51. อิเลเมนต์ชายนั้

อิเลเมนต์ชายนั้	บังคับ	คำอธิบาย
<Command>	no	คำสั่ง
<Argument>	no	อาร์กิวเมนต์บรรทัดคำสั่ง
<Stdin>	no	อาร์กิวเมนต์ที่สนับสนุนโดยอิเลเมนต์ <Stdin>
<Filter>	no	ตัวกรอง
<Mask>	no	มาสก์ดักจับเอาต์พุต

หมายเหตุ: อิลิเมนต์ <Command> ต้องถูกกำหนดสำหรับ แต่ละคุณสมบัติ: ที่ระดับ <ParameterDef>, ที่ระดับ <CfgMethod> หรือในอิลิเมนต์ <PropertyDef>

การใช้งาน

คุณสมบัติคือคู่คีย์-ค่า ที่สัมพันธ์กับพารามิเตอร์ ค่าของคู่คีย์-ค่า ถูกเรียกค้นโดยคำสั่ง `artexget -r` และ `artexget -n` และบันทึกในโปรไฟล์เอาต์พุต ค่าคุณสมบัติ ที่บันทึกในโปรไฟล์สามารถแทรกในบรรทัดรับคำสั่งได้โดยใช้ลำดับ
%p[property_name]

อิลิเมนต์ <Property > กำหนดคุณสมบัติให้แก่ พารามิเตอร์หรือให้แก่เมธอดการกำหนดคอนฟิก คุณสมบัติ ถูกกำหนดแบบโลคัลภายใต้อิลิเมนต์ <Property> หรือสืบทอดจากอิลิเมนต์ <Property> หรือ <PropertyDef> ระดับสูง ที่มีแอตทริบิวต์ name ที่ตรงกัน

พารามิเตอร์มีคุณสมบัติ ทั้งหมดที่กำหนดแบบโลคัลภายใต้อิลิเมนต์ <ParameterDef> รวมทั้ง พารามิเตอร์มีคุณสมบัติทั้งหมดที่กำหนดภายในเมธอด การกำหนดคอนฟิกพารามิเตอร์ ถ้าใช้เมธอดการกำหนดคอนฟิก ผลก็คือว่าคุณสมบัติถูกกำหนดภายใต้อิลิเมนต์ <CfgMethod> อิลิเมนต์ <ParameterDef> ทั้งหมดที่ใช้เมธอด การกำหนดคอนฟิกจะมีคุณสมบัตินั้นโดยอัตโนมัติ (แม้ว่าบางส่วนของอิลิเมนต์เหล่านั้นอาจกำหนดคุณสมบัติใหม่แบบโลคัล)

ค่าคุณสมบัติถูกแยก จากเอาต์พุตของบรรทัดรับคำสั่ง บรรทัดรับคำสั่งถูกสร้างโดยการรวม อิลิเมนต์ <Command>, <Argument>, <Stdin> และ <Filter> ดังอธิบายในส่วน การสร้าง บรรทัดรับคำสั่ง คุณต้องใช้หนึ่งในค่าคุณสมบัติ ต่อไปนี้: เอาต์พุตดิบของบรรทัดรับคำสั่ง หรือส่วนของ เอาต์พุตที่ตรงกับมาสก์ ถ้าอิลิเมนต์ <Mask> ถูกระบุ

อิลิเมนต์ <Command>, <Argument>, <Stdin>, <Filter> และ <Mask> ที่กำหนดคุณสมบัติจะถูกค้นหา ตามลำดับนี้:

- ภายใต้อิลิเมนต์ <Property> ที่ระดับ <ParameterDef>
- ถ้าอิลิเมนต์ <ParameterDef> มีแอตทริบิวต์ `cfgmethod` ภายใต้เมธอดการกำหนดคอนฟิกของอิลิเมนต์ <Property> ที่มีแอตทริบิวต์ `name` ตรงกัน
- ภายใต้อิลิเมนต์ <PropertyDef> ของแค็ตตาล็อกที่มีแอตทริบิวต์ `name` ตรงกัน

ตัวอย่าง

ตัวอย่างต่อไปนี้กำหนดคุณสมบัติ `nodeId` ให้แก่พารามิเตอร์ `netaddr` และ `netaddr6`:

```
<ParameterDef name="netaddr" type="string" targetClass="device" cfgmethod="attr">
  <Property name="nodeId">
    <Command>/usr/bin/uname -f</Command>
    <Mask>.*</Mask>
  </Property>
</ParameterDef>
```

```
<ParameterDef name="netaddr6" type="string" targetClass="device" cfgmethod="attr">
  <Property name="nodeId">
    <Command>/usr/bin/uname -f</Command>
    <Mask>.*</Mask>
  </Property>
</ParameterDef>
```

ในตัวอย่างนี้ มาสก์ตรงกับ ทั้งบรรทัด และใช้เพื่อแยกอักขระ `newline` ที่ท้ายของเอาต์พุตคำสั่งเท่านั้น

ในตัวอย่างนี้ คำสั่ง `uname` ถูกรันสองครั้ง: หนึ่งครั้งสำหรับพารามิเตอร์ `netaddr` และหนึ่งครั้งสำหรับ พารามิเตอร์ `netaddr6` คำสั่งถูกรันสองครั้งเนื่องจากแต่ละ พารามิเตอร์มีคุณสมบัติของตนเอง ที่มีอิลิเมนต์ `<Command>` ของตนเอง ดูที่ “อิลิเมนต์ `<PropertyDef>`” สำหรับตัวอย่างที่ต้องการ การรันคำสั่ง `uname` หนึ่งครั้งเท่านั้น

ข้อมูลที่เกี่ยวข้อง

- “การสร้างบรรทัดรับคำสั่ง” ในหน้า 132
- “การขยายอิลิเมนต์ของบรรทัดคำสั่ง” ในหน้า 135
- “อิลิเมนต์ `<PropertyDef>`”

อิลิเมนต์ `<PropertyDef>`:

อิลิเมนต์ `<PropertyDef>` กำหนดคุณสมบัติ ที่สามารถใช้ในอิลิเมนต์ `<Property>`

ไวยากรณ์

อิลิเมนต์พารেন্ট: `<Catalog>`

แอ็ททริบิวต์ต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 52. แอ็ททริบิวต์

แอ็ททริบิวต์	บังคับ	ชนิด	คำอธิบาย
<code>name</code>	yes	สตริง	ระบุชื่อของคุณสมบัติ

อิลิเมนต์ชายนั้ดังต่อไปนี้ได้รับการสนับสนุน:

ตารางที่ 53. อิลิเมนต์ชายนั้

อิลิเมนต์ชายนั้	บังคับ	คำอธิบาย
<code><Command></code>	no	คำสั่ง
<code><Argument></code>	no	อาร์กิวเมนต์บรรทัดคำสั่ง
<code><Stdin></code>	no	อาร์กิวเมนต์ที่สนับสนุนโดยอิลิเมนต์ <code><Stdin></code>
<code><Filter></code>	no	ตัวกรอง
<code><Mask></code>	no	มาสก์ดักจับเอาต์พุต

หมายเหตุ: อิลิเมนต์ `<Command>` ต้องถูกกำหนดสำหรับ แต่ละคุณสมบัติ: ที่ระดับ `<ParameterDef>`, ที่ระดับ `<CfgMethod>` หรือในอิลิเมนต์ `<PropertyDef>`

การใช้งาน

คุณสมบัติคือคู่คี่-ค่า ที่สัมพันธ์กับพารามิเตอร์ ค่าของคู่คี่-ค่า ถูกเรียกออกมาโดยคำสั่ง `artexget -r` และ `artexget -n` และบันทึกในเอาต์พุตไพล์ ค่าคุณสมบัติ ที่บันทึกในไพล์สามารถแทรกลงในบรรทัดรับคำสั่งได้โดยใช้ ลำดับ

`%p[property_name]`

อิลิเมนต์ <PropertyDef> กำหนดคุณสมบัติ คุณสมบัตินี้สามารถเชื่อมโยงกับพารามิเตอร์ หรือเมธอดการกำหนดคอนฟิกใน ภายหลังโดยใช้อิลิเมนต์ <Property> ที่มีแอตทริบิวต์ชื่อเดียวกัน

ตัวอย่าง

ตัวอย่างต่อไปนี้จะกำหนดคุณสมบัติ *nodeId* ให้แก่พารามิเตอร์ *netaddr* และ *netaddr6*:

```
<PropertyDef name="nodeId">
  <Command>/usr/bin/uname -f</Command>
  <Mask>.*</Mask>
</PropertyDef>

<ParameterDef name="netaddr" type="string" targetClass="device" cfgmethod="attr">
  <Property name="nodeId"/>
</ParameterDef>

<ParameterDef name="netaddr6" type="string" targetClass="device" cfgmethod="attr">
  <Property name="nodeId"/>
</ParameterDef>
```

ในตัวอย่างนี้ คำสั่ง *uname* ถูกรันเพียงครั้งเดียว เนื่องจากสองพารามิเตอร์ใช้อิลิเมนต์ <Command> เดียวกันสำหรับคุณสมบัติ ของตน และบรรทัดรับคำสั่งที่สร้างจะเหมือนกัน สำหรับทั้งสองพารามิเตอร์

ข้อมูลที่เกี่ยวข้อง

- “การสร้างบรรทัดรับคำสั่ง”
- “การขยายอิลิเมนต์ของบรรทัดคำสั่ง” ในหน้า 135
- “อิลิเมนต์ <Property>” ในหน้า 129

การสร้างบรรทัดรับคำสั่ง

เฟรมเวิร์ก AIX Runtime Expert ฟังพาคำสั่งภายนอกในการดักจับ ตั้งค่า และยังมีทางเลือกในการเปรียบเทียบค่าพารามิเตอร์ หัวข้อนี้อธิบายว่าบรรทัดคำสั่งถูกสร้างขึ้นโดยใช้ข้อมูลไวยากรณ์ซึ่งจัดให้มีอยู่ในไฟล์แค็ตตาล็อกได้อย่างไร

การดำเนินการ

สำหรับแต่ละพารามิเตอร์ การดำเนินการต่อไปนี้จะสามารถกำหนดได้:

- รับ *type="current"* ที่ใช้เพื่อดักจับค่าปัจจุบันของพารามิเตอร์
- รับ *type="nextboot"* ซึ่งใช้เพื่อดักจับค่าที่พารามิเตอร์จะมีขึ้นหลังจากการรีบูท
- ตั้งค่า *type="current"* ซึ่งใช้เพื่อตั้งค่าปัจจุบันของพารามิเตอร์ ค่าพารามิเตอร์นี้จะหายไปเมื่อทำการรีบูท
- ตั้งค่า *type="nextboot"* ซึ่งใช้เพื่อตั้งค่าที่พารามิเตอร์จะมีหลังจากที่ทำการบูทใหม่แล้ว
- ตั้งค่า *type="permanent"* ซึ่งใช้เพื่อตั้งค่าปัจจุบันของพารามิเตอร์ โดยที่ทราบว่า ค่านี้อาจจะคงอยู่หลังจากการรีบูท
- ปฏิบัติการ *diff* ใช้เพื่อเปรียบเทียบสองค่าของพารามิเตอร์
- ปฏิบัติการค้นหา ซึ่งใช้เพื่อค้นหาเป้าหมายสำหรับพารามิเตอร์ที่สนับสนุน
- คุณสมบัติ ใช้เพื่อบันทึกค่าคุณสมบัติสำหรับพารามิเตอร์
- สิ่งที่เป็นต้องมิใช่เพื่อกำหนดเงื่อนไขการเรียกใช้งานการดำเนินการ *get*, *set* หรือ *discover* สำหรับพารามิเตอร์ที่กำหนด

ไม่จำเป็นต้องกำหนดทุกการปฏิบัติการสำหรับพารามิเตอร์ทั้งหมด การดำเนินการ `get` สองการดำเนินการ และการดำเนินการ `set` ทั้งหมด ที่สนับสนุนโดยพารามิเตอร์ต้องถูกกำหนด การดำเนินการ `diff` เป็นทางเลือก และถ้าไม่ได้กำหนด การเปรียบเทียบค่าพารามิเตอร์จะถูกดำเนินการภายในตามชนิดพารามิเตอร์ เช่น `string` และ `integer` การดำเนินการ `discover` ต้องถูกกำหนดสำหรับพารามิเตอร์ที่มีปลายทางเท่านั้น คุณสมบัติและสิ่งที่จำเป็นต้องมีถูกกำหนด เมื่อจำเป็นเท่านั้น

อิลิเมนต์บรรทัดคำสั่ง

สำหรับแต่ละปฏิบัติการที่สนับสนุนโดยพารามิเตอร์ ทำอิลิเมนต์ที่แตกต่างสามารถนำมาใช้เพื่อกำหนดว่า บรรทัดคำสั่งจะสามารถสร้างขึ้นเพื่อให้ดำเนินการได้อย่างไร:

- อิลิเมนต์ `<Command>` ใช้เพื่อกำหนดคำสั่งพื้นฐานสำหรับจัดการพารามิเตอร์
- อิลิเมนต์ `<Stdin>` ใช้เพื่อกำหนดข้อมูลที่จะถูกเขียนไปยังอินพุตมาตรฐานของบรรทัดคำสั่ง
- อิลิเมนต์ `<Argument>` ใช้เพื่อแทรกข้อมูลพารามิเตอร์เฉพาะลงในอิลิเมนต์ `<Command>` หรือ `<Stdin>`
- อิลิเมนต์ `<Filter>` ใช้เพื่อกรองเอาต์พุตบรรทัด คำสั่งสำหรับการดำเนินการ `get` และ `diff`
- อิลิเมนต์ `<Mask>` ใช้เพื่อแยกข้อมูลจากเอาต์พุตของบรรทัดคำสั่งสำหรับการดำเนินการ `get`, `diff` และ `property`

เมื่อการดำเนินการต้องถูกดำเนินการ อิลิเมนต์ `<Command>`, `<Stdin>`, `<Argument>` และ `<Filter>` ที่กำหนดสำหรับการดำเนินการร้องขอจะถูกรวมเข้าด้วยกันเพื่อสร้าง ชุดของบรรทัดคำสั่ง ดังอธิบายในหัวข้อ “อัลกอริทึมการสร้างของบรรทัดคำสั่ง” ในหน้า 134 จากนั้นบรรทัดคำสั่งที่สร้างจะถูกรันโดยเซลล์ สำหรับการดำเนินการ `get`, `diff` และ `property` นั้นการดำเนินการ `<Mask>` ถูกใช้เพื่อแยกข้อมูลที่ร้องขอ (ค่าพารามิเตอร์, ผลลัพธ์ การเปรียบเทียบ หรือค่าคุณสมบัติ) จากเอาต์พุตคำสั่ง

วิธีการกำหนดคอนฟิก

อิลิเมนต์บรรทัดคำสั่งสามารถระบุได้บนเครื่องภายในอิลิเมนต์ `<ParameterDef>` หรือที่รับมาจากอิลิเมนต์ `<CfgMethod>` ที่ถูกอ้างอิงอยู่ในอิลิเมนต์ `<ParameterDef>` โดยใช้แอตทริบิวต์ `cfgmethod`

การรวมกันได้รับอนุญาตให้ทำได้: ชุดของอิลิเมนต์บรรทัดคำสั่งที่กำหนดสำหรับการดำเนินการที่เจาะจงของพารามิเตอร์ที่เจาะจงคือการรวมของอิลิเมนต์บรรทัดคำสั่ง ที่กำหนดแบบโลคัลภายในอิลิเมนต์ `<ParameterDef>` และอิลิเมนต์ บรรทัดคำสั่งที่กำหนดสำหรับการดำเนินการเดียวกันในอิลิเมนต์ `<CfgMethod>` ที่อ้างอิงโดยแอตทริบิวต์ `cfgmethod` ของอิลิเมนต์ `<ParameterDef>` หากอิลิเมนต์ของบรรทัดคำสั่งเดียวกันถูกกำหนดทั้งบนเครื่องและในวิธีการกำหนดคอนฟิก ข้อกำหนดโลคัลจะมีความสำคัญมาก่อน

ตัวอย่างเช่น ในไฟล์แค็ตตาล็อกที่ไม่ได้รับการปรับปรุงนี้:

```
<CfgMethod id="vmo">
  <Get type="nextboot">
    <Command>/usr/sbin/vmo -r%a</Command>
    <Mask name="1" value="2">[[ :space:]]*(.*) = (.*)</Mask>
  </Get>

  <Set type="permanent">
    <Command>/usr/sbin/vmo -p -o%a</Command>
    <Argument> -o %n=%p</Argument>
  </Set>
</CfgMethod>
```

```

<ParameterDef name="lgpg_size" cfgmethod="vmo">
  <Get type="current">
    <Command>/usr/sbin/vmo -o lgpg_size</Command>
    <Mask name="1" value="2">[[ :space:]]*(.*) = (.*)</Mask>
  </Get>

  <Get type="nextboot">
    <Argument> -o lgpg_size</Argument>
  </Get>
</ParameterDef>

```

เราจะเห็นได้ว่า:

- ปฏิบัติการ <Get type="current"> ถูกกำหนดทั้งหมดที่ระดับ <ParameterDef>
- ปฏิบัติการ <Get type="nextboot"> มีบางอิลิเมนต์ซึ่งถูกกำหนดที่ระดับ <CfgMethod> (<Command> และ <Mask>) และบางอิลิเมนต์ซึ่งถูกกำหนดที่ระดับ <ParameterDef> (<Argument>)
- ปฏิบัติการ <Get type="current"> ถูกกำหนดทั้งหมดที่ระดับ <CfgMethod>

การใช้วิธีการกำหนดคอนฟิกมีประโยชน์สองประการหลักๆ:

- ช่วยให้จัดทำแค็ตตาล็อกได้ง่าย ในหลายกรณี นิยามพารามิเตอร์จะได้รับอิลิเมนต์บรรทัดคำสั่งทั้งหมดของตนจากวิธีการกำหนดคอนฟิก และอิลิเมนต์ <ParameterDef> จะว่างเปล่า
- ซึ่งจะช่วยให้พารามิเตอร์ต่างๆ ถูกจัดกลุ่มเข้าด้วยกันในบรรทัดคำสั่งเดียวเมื่อเป็นไปได้

อัลกอริทึมการสร้างของบรรทัดคำสั่ง

บรรทัดคำสั่งถูกสร้างขึ้นโดยใช้อัลกอริทึมที่อนุญาตให้หลายพารามิเตอร์ถูกจัดกลุ่มในคำสั่งเดียว

การจัดกลุ่มพารามิเตอร์ไม่ได้เป็นที่ต้องการเฉพาะเนื่องด้วยเหตุผลด้านความสามารถในการทำงานและประสิทธิภาพเท่านั้น แต่ยังจำเป็นสิ่งจำเป็นสำหรับบางพารามิเตอร์อีกด้วย ตัวอย่างเช่น พารามิเตอร์ `vmo lgpg_regions` และ `lgpg_size` ซึ่งไม่สามารถถูกตั้งค่าแยกต่างหากได้ และจำเป็นจะต้องถูกกำหนดค่ารวมกันกับการร้องขอคำสั่ง `vmo` เดียว

อัลกอริทึมการสร้างของบรรทัดคำสั่งมีความเทียบเท่าในด้านการทำงานกับขั้นตอนดังต่อไปนี้:

1. แต่ละพารามิเตอร์ในโปรไฟล์อินพุตมีอิลิเมนต์ <Command> และ <Stdin> ซึ่งถูกขยายบางส่วน ระหว่างเฟสนี้ ลำดับ %a, %v1[name], %v2[name], %f1[name] และ %f2[name] จะได้รับการละเว้นและไม่ถูกขยาย
2. พารามิเตอร์ที่ตรวจสอบเงื่อนไขทั้งห้าด้านล่างถูกจัดกลุ่มไว้ด้วยกัน:
 - พารามิเตอร์ใช้อิลิเมนต์ <Command> เดียวกัน
 - พารามิเตอร์ใช้อิลิเมนต์ <Stdin> เดียวกัน
 - พารามิเตอร์ใช้อิลิเมนต์ <Filter> เดียวกัน
 - การขยายอิลิเมนต์ <Command> ได้ดำเนินการในระหว่างการสร้างสตริงที่เหมือนกันในขั้นตอนที่ 1
 - การขยายอิลิเมนต์ <Stdin> ดำเนินการในระหว่างการสร้างสตริงที่เหมือนกันในขั้นตอนที่ 1

ตอนนี้ กลุ่มมีอิลิเมนต์ <Command> และ <Stdin> ที่ได้รับการขยายบางส่วนเป็นของตนเอง และมีอิลิเมนต์ <Filter> ของตัวเองที่แบ่งปันโดยพารามิเตอร์ทั้งหมดในกลุ่ม

3. สำหรับแต่ละกลุ่มพารามิเตอร์ กลุ่มอิลิเมนต์ <Command> และ <Stdin> มีลำดับ %v1[name], %v2[name], %f1[name] และ %f2[name] ที่ถูกขยาย ชื่อพารามิเตอร์จะถูกค้นหาเฉพาะภายในกลุ่ม
4. สำหรับแต่ละกลุ่มพารามิเตอร์ กลุ่มอิลิเมนต์ <Command> และ <Stdin> มีลำดับ %a ที่ถูกขยาย: แต่ละพารามิเตอร์ในกลุ่มมีอิลิเมนต์ <Argument> ที่ถูกขยาย และการเรียงต่อกันของอิลิเมนต์ <Argument> ที่ถูกขยายเหล่านั้นจะแทนที่ลำดับ %a ในอิลิเมนต์ <Command> และ <Stdin>

ผลของกระบวนการนี้คือ ชุดบรรทัดคำสั่งที่มีข้อมูลเป็นทางเลือกให้เขียนลงบนอินพุตมาตรฐานและคำสั่งให้กรองเอาต์พุตของตนเอง

การขยายอิลิเมนต์ของบรรทัดคำสั่ง:

อิลิเมนต์ <Command>, <Stdin> และ <Argument> สนับสนุนลำดับพิเศษที่ถูกขยายโดยเฟรมเวิร์ก AIX Runtime Expert ให้สร้างบรรทัดคำสั่งสุดท้าย

ตารางด้านล่างเป็นข้อมูลอ้างอิงสั้นๆ ของลำดับทั้งหมดที่ได้รับการสนับสนุน สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับลำดับ โปรดดูส่วนดังต่อไปนี้

ตารางที่ 54. ลำดับ

ลำดับ	ขยายไปยัง
%%	อักขระ % จริง
%a	การเรียงต่อกันของสตริง Argument ที่ถูกขยาย สำหรับพารามิเตอร์ทั้งหมดที่สามารถประมวลผลในบรรทัดคำสั่งเดียวกัน
%n	ชื่อของพารามิเตอร์
%v1	ค่าของพารามิเตอร์
%v2	ค่าที่สองของพารามิเตอร์ เฉพาะปฏิบัติการ diff ที่ถูกต้อง
%f1	ชื่อของไฟล์ชั่วคราวที่จะมีค่าของพารามิเตอร์อยู่
%f2	ชื่อของไฟล์ชั่วคราวที่จะมีค่าที่สองของพารามิเตอร์อยู่ เฉพาะปฏิบัติการ diff ที่ถูกต้อง
%v1[name]	ค่าของชื่อพารามิเตอร์
%v2[name]	ค่าที่สองของชื่อพารามิเตอร์ เฉพาะปฏิบัติการ diff ที่ถูกต้อง
%f1[name]	ชื่อของไฟล์ชั่วคราวที่จะมีค่าของชื่อพารามิเตอร์อยู่
%f2[name]	ชื่อของไฟล์ชั่วคราวที่จะมีค่าที่สองของชื่อพารามิเตอร์อยู่ เฉพาะปฏิบัติการ diff ที่ถูกต้อง
%t[class]	ชื่อของอินสแตนซ์เป้าหมายสำหรับคลาสเป้าหมาย
%p[name]	ค่าของคุณสมบัติ <i>name</i>
%c	id แค็ตตารีอิก

การห้ล้ล้าด้บ %

ชื่อพารามิเตอร์ ค่าพารามิเตอร์ และชื่อเป้าหมายที่ถูกขยายโดย AIX Runtime Expert มีตำแหน่งอยู่ในเครื่องหมาย ' เมื่อถูกใช้ภายในอิลิเมนต์ <Command> หรือภายในอิลิเมนต์ <Argument> ที่จะต้องถูกแทรก (ผ่านล้าด้บ %a) ลงในอิลิเมนต์ <Command> ทั้งนี้ เพื่อให้แน่ใจว่า สตริงเหล่านั้นจะถูกส่งไปยังเซลล์ในฐานะที่เป็นค่าเดียว แม้ว่าจะมีส่วนประกอบของการเว้นวรรคหรืออักขระพิเศษอื่นๆ อยู่ นอกจากนี้ อักขระ ' ภายในนิพจน์ที่ถูกขยาย ยังถูกเลี้ียงอย่างเหมาะสม

ผู้เขียนเค้ดตล้าล้อกต้องระวังที่จ้ะไม่ใช้ล้าด้บ %n, %v1, %v2, %v1[name], %v2[name] หรือ %t[class] ภายในสตริงที่มีการใส่เครื่องหมาย ' ล้าด้บเหล่านั้น ต้องถูกใช้ภายในสตริง สตริงต้องถูกปิดก้อนล้าด้บ % ดังแสดงในตัวอย่างต่อไปนี้:

```
echo "Parameter "%n" is set to "%v1
```

หากไม่สามารถทำเช่นนั้นได้ จะส่งผลให้บรรทัดคำสั่งไม่ถูกต้อง และมีความเสี่ยงด้านความปลอดภัย

ล้าด้บ %%

ล้าด้บ %% ขยายไปยังอักขระ % จริง

ตัวอย่าง เช่น สตริง:

```
/bin/ps -aeF"%a"
```

ขยายไปยังสตริงต่อไปนี้:

```
/bin/ps -aeF"%a"
```

ล้าด้บ %a

ล้าด้บ %a สามารถใช้ไม่ว่าในสตริง <Command> หรือในสตริง <Stdin> ซึ่งจะถูกแทนที่ด้วยการเรียงต่อกันของสตริง <Argument> ที่ขยายทั้งหมดของพารามิเตอร์ทั้งหมดที่สามารถได้รับการปฏิบัติในคำสั่งเดียวกัน (ดูหัวข้อ การสร้างบรรทัดคำสั่ง เพื่ออ่านคำอธิบายอย่างเป็นทางการเกี่ยวกับการจัดการกลุ่มพารามิเตอร์)

ตัวอย่างเช่น เค้ดตล้าล้อกดังต่อไปนี้ (โปรดทราบว่าอาจถูกทำให้เรียบง่ายโดยการใส่ล้าด้บ %n):

```
<CfgMethod id="vmo">
  <Get type="current">
    <Command>/usr/sbin/vmo%a</Command>
  </Get>
</CfgMethod>
<ParameterDef name="lgpg_size" cfgmethod="vmo">
  <Get type="current">
    <Argument> -o lgpg_size</Argument>
  </Get>
</ParameterDef>
<ParameterDef name="lgpg_regions" cfgmethod="vmo">
  <Get type="current">
    <Argument> -o lgpg_regions</Argument>
  </Get>
</ParameterDef>
```

และโปรไฟล์ดังต่อไปนี้:

```
<Parameter name="lpgg_size" />
<Parameter name="lpgg_regions" />
```

will produce the following command line for the “get current” operation:

```
/usr/sbin/vmo -o lpgg_size -o lpgg_regions
```

ลำดับ %n

ลำดับ %n ถูกแทนที่ด้วยชื่อพารามิเตอร์

เมื่อใช้ลำดับ %n ตัวอย่างจากส่วน %a สามารถถูกทำให้เรียบง่ายได้ดังต่อไปนี้:

```
<CfgMethod id="vmo">
  <Get type="current">
    <Command>/usr/sbin/vmo%a</Command>
    <Argument> -o %n</Argument>
  </Get>
</CfgMethod>
<ParameterDef name="lpgg_size" cfgmethod="vmo" />
<ParameterDef name="lpgg_regions" cfgmethod="vmo" />
```

ด้วยโปรไฟล์ดังต่อไปนี้:

```
<Parameter name="lpgg_size" />
<Parameter name="lpgg_regions" />
```

บรรทัดคำสั่งดังต่อไปนี้จะถูกสร้างขึ้นสำหรับปฏิบัติการปัจจุบันของ get:

```
/usr/sbin/vmo -o 'lpgg_size' -o 'lpgg_regions'
```

ลำดับ %v1 และ %v2

ลำดับ %v1 ถูกแทนที่ด้วยค่าพารามิเตอร์

ลำดับ %v2 ถูกต้องเฉพาะสำหรับปฏิบัติการ <Diff> และถูกแทนที่ด้วยค่าที่สองของพารามิเตอร์

ตัวอย่างเช่น แค็ตตาล็อกดังต่อไปนี้:

```
<CfgMethod id="vmo">
  <Set type="permanent">
    <Command>/usr/sbin/vmo -p%a</Command>
    <Argument> -o %n=%v1</Argument>
  </Set>
</CfgMethod>
<ParameterDef name="lpgg_size" cfgmethod="vmo" />
<ParameterDef name="lpgg_regions" cfgmethod="vmo" />
```

with the following profile:

```
<Parameter name="lpgg_size" value="16M"/>
<Parameter name="lpgg_regions" value="128" />
```

จะสร้างบรรทัดคำสั่งดังต่อไปนี้สำหรับปฏิบัติการ set permanent:

```
/usr/sbin/vmo -p -o 'lpgg_size='16M' -o 'lpgg_regions='128'
```

ลำดับ %f1 และ %f2

ลำดับ %f1 และ %f2 ถูกแทนด้วยชื่อของไฟล์ชั่วคราวที่สร้างขึ้นก่อนที่คำสั่งจะถูกส่งการ เนื้อหาไฟล์คือค่าของพารามิเตอร์สำหรับ %f1 และค่าที่สองของพารามิเตอร์สำหรับ %f2 ลำดับ %f2 สามารถถูกใช้เฉพาะสำหรับปฏิบัติการ <Diff>

ยกตัวอย่างเช่น แค็ตตาล็อกดังต่อไปนี้:

```
<ParameterDef name="some_file">
  <Diff>
    <Command>/usr/bin/diff %f1 %f2</Command>
  </Diff>
</ParameterDef>
```

เมื่อ **artexdiff** ถูกดำเนินการระหว่างสองโปรไฟล์ ซึ่งรวมถึงพารามิเตอร์เดียวกันกับค่าที่แตกต่างกัน:

```
<Parameter name="some_file" value="foo" />
<Parameter name="some_file" value="bar" />
```

จากนั้น สองไฟล์ชั่วคราว /tmp/file1 และ /tmp/file2 (ชื่อไฟล์จริงจะต่างออกไป) ที่มีสตริง “foo” และ “bar” ตามลำดับ จะถูกสร้างขึ้น และคำสั่งต่อไปนี้จะถูกส่งการ:

```
/usr/bin/diff /tmp/file1 /tmp/file2
```

ลำดับ %v1[name] and %v2[name]

ลำดับ %v1[name] ถูกแทนที่ด้วยค่าของชื่อพารามิเตอร์

ลำดับ %v2[name] ถูกต้องเฉพาะสำหรับปฏิบัติการ <Diff> และถูกแทนที่ด้วยค่าที่สองของชื่อพารามิเตอร์

ลำดับเหล่านี้จะเป็นประโยชน์เมื่อคำสั่งกำหนดคอนฟิกยอมรับหลายพารามิเตอร์พร้อมๆ กัน แต่กำหนดว่า มีบางส่วนจะต้องจัดไว้ในตำแหน่งเฉพาะบนบรรทัดคำสั่ง นี่คือกรณีของคำสั่ง **chcons** เป็นต้น ซึ่งกำหนดว่า พาธไปยังอุปกรณ์คอนโซลหรือไฟล์จะต้องมาเป็นอันดับหลังสุดบนบรรทัดคำสั่ง เมื่อใช้ลำดับ %v1[name] แค็ตตาล็อก **chcons** จะสามารถเขียนได้ดังนี้:

```
<CfgMethod id="chcons">
  <Set type="nextboot">
    <Command>/usr/sbin/chcons%a %v1[console_device]</Command>
    <Argument> -a %n=%v1</Argument>
  </Set>
</CfgMethod>
<ParameterDef name="console_device" cfgmethod="chcons" reboot="true" />
<ParameterDef name="console_logname" cfgmethod="chcons" reboot="true" />
<ParameterDef name="console_logsize" cfgmethod="chcons" reboot="true" />
```

with the following profile:

```
<Parameter name="console_device" value="/dev/vty0"/>
<Parameter name="console_logname" value="/var/adm/ras/conslog" />
<Parameter name="console_logverb" value="9" />
```

แค็ตตาล็อกนี้จะสร้างบรรทัดคำสั่งดังต่อไปนี้สำหรับปฏิบัติการ **set nextboot** :

```
/usr/sbin/chcons -a 'console_logname='/var/adm/ras/conslog' -a 'console_logverb='9' /dev/vty0
```

ลำดับ %f1[name] และ %f2[name]

ลำดับ %f1[name] และ %f2[name] ถูกแทนที่ด้วยชื่อของไฟล์ชั่วคราวที่สร้างขึ้นก่อนที่คำสั่งจะถูกส่งการ เนื้อหาไฟล์คือค่าของชื่อพารามิเตอร์สำหรับ %f1[name] และค่าที่สองของชื่อพารามิเตอร์สำหรับ %f2[name] ลำดับ %f2[name] สามารถถูกใช้เฉพาะสำหรับปฏิบัติการ <Diff>

ลำดับ %t[class]

ลำดับ %t[class] ถูกแทนที่ด้วยชื่อของอินสแตนซ์เป้าหมายที่กำลังได้รับการดำเนินการสำหรับคลาสเป้าหมาย

ลำดับ %t[class] ถูกใช้สำหรับพารามิเตอร์ที่เข้ากับอ็อบเจกต์เฉพาะ ไม่ใช่ทั้งระบบ ตัวอย่างสำหรับกรณีนี้ได้แก่คำสั่ง `chuser` ซึ่งพารามิเตอร์ของคำสั่งดังกล่าวนำมาใช้กับเฉพาะผู้ใช้ (รูท, เกสต์) สำหรับเฉพาะรีจิสทรี (ไฟล์, LDAP) เค็ดตาลีอกสำหรับคำสั่ง `chuser` อาจเขียนได้ดังต่อไปนี้:

```
<CfgMethod id="chuser">
  <Set type="permanent">
<Command>/usr/bin/chuser -R %t[module]%a %t[user]</Command>
  <Argument> %n=%v1</Argument>
  </Set>
</CfgMethod>
<ParameterDef name="shell" cfgmethod="chuser" targetClass="module,user">
<ParameterDef name="histsize" cfgmethod="chuser" targetClass="module,user" />
```

ด้วยโปรไฟล์ดังต่อไปนี้ซึ่งตั้งค่าเชลล์และพารามิเตอร์ `histsize` สำหรับผู้ใช้ `adam` และ `bob` ใน LDAP และรีจิสทรีไฟล์:

```
<Parameter name="shell" value="/usr/bin/ksh">
  <Target class="module" instance="LDAP" />
  <Target class="user" instance="adam" />
</Parameter>
<Parameter name="histsize" value="5000">
  <Target class="module" instance="LDAP" />
  <Target class="user" instance="adam" />
</Parameter>
<Parameter name="shell" value="/usr/bin/ksh">
  <Target class="module" instance="files" />
  <Target class="user" instance="adam" />
</Parameter>
<Parameter name="histsize" value="5000">
  <Target class="module" instance="files" />
  <Target class="user" instance="adam" />
</Parameter>
<Parameter name="shell" value="/usr/bin/bash">
  <Target class="module" instance="LDAP" />
  <Target class="user" instance="bob" />
</Parameter>
<Parameter name="histsize" value="10000">
  <Target class="module" instance="LDAP" />
  <Target class="user" instance="bob" />
</Parameter>
<Parameter name="shell" value="/usr/bin/bash">
  <Target class="module" instance="files" />
  <Target class="user" instance="bob" />
</Parameter>
```

```
<Parameter name="histsize" value="10000">
  <Target class="module" instance="files" />
  <Target class="user" instance="bob" />
</Parameter>
```

ทั้งนี้ จะทำการสั่งการคำสั่งดังต่อไปนี้:

```
/usr/bin/chuser -R 'LDAP' 'shell'='/usr/bin/ksh' 'histsize'='5000' 'adam'
/usr/bin/chuser -R 'files' 'shell'='/usr/bin/ksh' 'histsize'='5000' 'adam'
/usr/bin/chuser -R 'LDAP' 'shell'='/usr/bin/bash' 'histsize'='10000' 'bob'
/usr/bin/chuser -R 'files' 'shell'='/usr/bin/bash' 'histsize'='10000' 'bob'
```

โปรดสังเกตว่าคำสั่งทั้งสี่ดังต่อไปนี้ถูกสร้างขึ้นอย่างไร เหตุผลก็คือ ลำดับ %t[module] และ %t[user] ได้ถูกใช้ในสตริง <Command> ซึ่งหมายความว่า แต่ละคำสั่งมีความเฉพาะสำหรับโมดูลและผู้ใช้เฉพาะ ด้วยเหตุนี้ เฉพาะพารามิเตอร์ที่ใช้กับโมดูลและผู้ใช้เดียวกันเท่านั้น ที่จะถูกจัดกลุ่มไว้ด้วยกัน

ลำดับ %p[name]

ลำดับ %p[name] ถูกแทนด้วยค่าที่ระบุในโปรไฟล์ อินพุตสำหรับชื่อคุณสมบัติ ตัวอย่างเช่น สิ่งที่ต้องมีต่อไปนี้ ใช้ลำดับ %p[nodeId] เพื่อตรวจสอบว่า id โหนดของระบบ โคลด (คืนค่าโดยคำสั่ง `uname -f`) ตรง กับ id โหนดที่เก็บในคุณสมบัติ nodeId ของโปรไฟล์:

```
<PrereqDef id="nodeId">
  <Command>[[ ` /usr/bin/uname -f ` = %p[nodeId] ]]</Command>
  <ErrorMessage>Parameter cannot be applied to a different node</ErrorMessage>
</PrereqDef>
```

ลำดับ %c

ลำดับ %c ถูกแทนด้วย id ของไฟล์แค็ตตาล็อกที่มีพารามิเตอร์ อยู่ นั่นคือ id แค็ตตาล็อกที่ระบุในโปรไฟล์ ซึ่ง สามารถแตกต่างจาก id ของแค็ตตาล็อกที่กำหนดพารามิเตอร์ จริงถ้าใช้การสืบทอดแค็ตตาล็อก

ตัวอย่างเช่น สิ่งที่ต้องมี ต่อไปนี้ใช้ลำดับ %c เพื่อตรวจสอบว่า *uniquetype* ของ อุปกรณ์ปลายทางตรงกับชื่อของไฟล์แค็ตตาล็อก:

```
<PrereqDef id="devUniqueType">
  <Command>[[ "devParam.` /usr/sbin/lsdev -F uniquetype -l %t[device] | /usr/bin/tr / .`" = %c ]]</Command>
  <ErrorMessage>Parameter cannot be applied to a different device type</ErrorMessage>
</PrereqDef>
```

คำสั่งและโปรเซส

คำสั่ง คือการร้องขอเพื่อทำการดำเนินงานหรือรัน โปรแกรม *โปรเซส* คือโปรแกรมหรือคำสั่งที่กำลังรันอยู่จริง บนคอมพิวเตอร์

คุณใช้คำสั่งเพื่อบอกแก่ระบบปฏิบัติการ ถึงงานที่คุณ ต้องการดำเนินการ เมื่อป้อนคำสั่ง คำสั่งจะถูกถอดรหัสโดยตัวตีความคำสั่ง (เรียกอีกอย่างว่า *เชลล์*) และภารกิจนั้นจะถูกประมวลผล

ระบบปฏิบัติการสามารถรันโปรเซสจำนวนมากที่แตกต่างกันในเวลาเดียวกันได้

ระบบปฏิบัติการอนุญาตให้คุณจัดวางข้อมูลอินพุตและเอาต์พุต (I/O) เข้าและออกจากระบบของคุณได้ โดยใช้คำสั่ง I/O และสัญลักษณ์เฉพาะ คุณสามารถควบคุมอินพุตโดยการระบุที่ตั้งซึ่งจะรวบรวมข้อมูล ตัวอย่างเช่น คุณสามารถระบุให้อ่านอิน

พุดที่บ่อนบนคีย์บอร์ด (อินพุต มาตรฐาน) หรือให้อ่านอินพุตจากไฟล์ คุณสามารถควบคุมเอาต์พุตโดยการระบุ ตำแหน่งที่จะแสดงหรือจัดเก็บข้อมูล ตัวอย่างเช่น คุณสามารถระบุให้เขียนข้อมูลเอาต์พุต ไปยังหน้าจอ (เอาต์พุตมาตรฐาน) หรือให้เขียนไปยังไฟล์

คำสั่ง

บางคำสั่งสามารถบ่อนได้โดยการพิมพ์เพียง คำเดียว และเป็นไปได้ที่จะรวมคำสั่งหลายคำสั่ง เพื่อให้เอาต์พุตจากคำสั่งหนึ่ง เป็นอินพุตสำหรับอีกคำสั่งหนึ่ง

การรวมคำสั่งเพื่อให้เอาต์พุตจากคำสั่งหนึ่งกลายเป็นอินพุต สำหรับอีกคำสั่งหนึ่งเรียกว่าเป็น *การไหล*

แฟล็กกำหนดการทำงานของคำสั่ง *แฟล็ก* คือตัวแก้ไข ที่ใช้กับชื่อคำสั่งบนบรรทัดคำสั่ง โดยปกติจะมีเครื่องหมายชี้หน้าหน้า

คำสั่งยังสามารถถูกจัดกลุ่มรวมกันและเก็บไว้ใน ไฟล์ ไฟล์เหล่านี้ เรียกว่าเป็น *เชลล์โปรแกรมเมอร์* หรือ *เชลล์สคริปต์* แทนการเรียกใช้ คำสั่งแต่ละคำสั่ง คุณสามารถรันไฟล์ที่มีคำสั่ง

เมื่อต้องการบ่อนคำสั่ง ให้พิมพ์ชื่อคำสั่งที่พร้อมต์ และกด Enter

\$ *CommandName*

หลักการที่เกี่ยวข้อง:

“คุณลักษณะเชลล์” ในหน้า 232

มีข้อดีหลายอย่างในการใช้เชลล์เป็นอินเตอร์เฟซของ ระบบ

งานที่เกี่ยวข้อง:

“การสร้างและการรันเชลล์สคริปต์” ในหน้า 235

เชลล์สคริปต์ คือไฟล์ที่มีคำสั่งตั้งแต่หนึ่งคำสั่งขึ้นไป เชลล์สคริปต์นำเสนอวิธีที่ง่ายในการดำเนินการคำสั่งที่ยุ่งยาก ลำดับของคำสั่งจำนวนมากหรือซับซ้อน และภารกิจจริง เมื่อคุณบ่อนชื่อ ของไฟล์เชลล์สคริปต์ ระบบจะดำเนินการลำดับคำสั่งที่มีอยู่ในไฟล์

ไวยากรณ์คำสั่งและชื่อคำสั่ง:

แม้ว่าบางคำสั่งสามารถบ่อนได้ง่ายโดยการพิมพ์คำเพียงคำเดียว แต่บางคำสั่งมีการใช้แฟล็กและพารามิเตอร์ แต่ละคำสั่งมีไวยากรณ์ที่กำหนด ทั้งแฟล็กและพารามิเตอร์ที่เป็นข้อบังคับและที่เป็นทางเลือก

รูปแบบทั่วไปของคำสั่งเป็นดังนี้:

CommandName flag(s) parameter(s)

ข้อมูลต่อไปนี้เป็นกฎทั่วไปบางข้อเกี่ยวกับคำสั่ง:

- ช่องว่างระหว่างคำสั่ง แฟล็ก และพารามิเตอร์เป็นสิ่งสำคัญ
- สามารถบ่อนสองคำสั่งบนบรรทัดเดียวกันได้โดยการคั่นคำสั่ง ด้วยเครื่องหมายอัฒภาค (;) ตัวอย่างเช่น:

\$ *CommandOne;CommandTwo*

เชลล์รันคำสั่งตามลำดับ

- คำสั่งเป็นแบบตรงตามตัวพิมพ์ เชลล์แยกความแตกต่างระหว่างตัวอักษรพิมพ์ใหญ่ และพิมพ์เล็ก สำหรับเชลล์ print ไม่เหมือนกับ PRINT หรือ Print

- คำสั่งที่ยาวมากสามารถป้อนบนมากกว่าหนึ่งบรรทัดได้โดยใช้อักขระ เครื่องหมายทับขวา (\) เครื่องหมายทับขวาแสดง ความต่อเนื่องของบรรทัด สำหรับเซลล์ ตัวอย่างต่อไปนี้เป็นคำสั่งหนึ่งที่ทอดข้ามสองบรรทัด:

```
$ ls Mail info temp \
(press Enter)
```

```
> diary
(the > prompt appears)
```

อักขระ > คือพร้อม์รองของคุณ (\$ คือพร้อม์หลักดีฟอลต์ของผู้ใช้ที่ไม่ใช่ผู้ใช้ ราก) ซึ่งบ่งชี้ว่าบรรทัดปัจจุบันเป็นส่วนที่ ต่อเนื่องของบรรทัด ก่อนหน้านั้น โปรดทราบว่า csh (C เซลล์) ไม่ให้พร้อม์รอง และการแบ่งต้องอยู่ที่ขอบเขตคำ และ พร้อม์หลักคือ %

คำแรกของทุกคำสั่งคือชื่อคำสั่ง บางคำสั่งมี ชื่อคำสั่งอย่างเดียว

แฟล็กคำสั่ง:

มีแฟล็กจำนวนมากที่อาจใช้ตามหลังชื่อคำสั่งได้ แฟล็กแก้ไข การดำเนินงานของคำสั่ง และบางครั้งเรียกว่า *อ็อปชัน*

แฟล็กมีการแบ่งเขตโดยช่องว่างหรือแท็บ และโดยปกติ แฟล็กขึ้นต้นด้วย เครื่องหมายขีด (-) ข้อยกเว้นคือ ps, tar, และ ar ซึ่งไม่จำเป็นต้องมีเครื่องหมายขีดนำหน้าบาง แฟล็ก ตัวอย่างเช่น ในคำสั่งต่อไปนี้:

```
ls -a -F
```

ls คือชื่อคำสั่ง และ -a -F คือแฟล็ก

เมื่อคำสั่งใช้แฟล็ก แฟล็กจะแสดงตามหลังต่อจากชื่อคำสั่ง แฟล็กที่เป็นอักขระตัวเดียว ในคำสั่งสามารถรวมได้ด้วยเครื่องหมายขีดเดียว ตัวอย่างเช่น คำสั่งก่อนหน้านี้นี้ยังสามารถเขียนได้อีกแบบหนึ่งดังนี้:

```
ls -aF
```

มีบางกรณีที่พารามิเตอร์ขึ้นต้นด้วยเครื่องหมายขีด (-) อย่างแท้จริง ในกรณีนี้ ให้ใช้เครื่องหมายขีดคั่น (-) ก่อนหน้าพารามิเตอร์—บอกให้คำสั่งทราบว่า ข้อมูลใดๆ ที่ตามมาไม่ใช่แฟล็ก แต่เป็นพารามิเตอร์

ตัวอย่างเช่น ถ้าคุณต้องการสร้างไดเรกทอรีที่ชื่อ -tmp และ คุณพิมพ์คำสั่งต่อไปนี้:

```
mkdir -tmp
```

ระบบจะแสดงข้อความแสดงข้อผิดพลาดคล้ายกับต่อไปนี้:

```
mkdir: Not a recognized flag: t
Usage: mkdir [-p] [-m mode] Directory ...
```

วิธีที่ถูกต้องในการพิมพ์คำสั่งเป็นดังนี้:

```
mkdir -- -tmp
```

ขณะนี้ไดเรกทอรีใหม่ของคุณ -tmp ถูกสร้างขึ้น

พารามิเตอร์คำสั่ง:

หลังจากชื่อคำสั่ง อาจมีแฟล็กจำนวนหนึ่ง ตามด้วย พารามิเตอร์ บางครั้งพารามิเตอร์มีการเรียกเป็น *อาร์กิวเมนต์* หรือ *ตัวถูกดำเนินการ* พารามิเตอร์ ระบุข้อมูลที่คำสั่งต้องการเพื่อรัน

หากคุณไม่ได้ระบุพารามิเตอร์ คำสั่งอาจสมมติค่า ดีฟอลต์ ตัวอย่างเช่น ในคำสั่งต่อไปนี้:

```
ls -a temp
```

ls คือชื่อคำสั่ง, -a คือแฟล็ก และ temp คือ พารามิเตอร์ คำสั่งนี้แสดงไฟล์ (-a) ทั้งหมดในไดเรกทอรี temp

ในตัวอย่างต่อไปนี้:

```
ls -a
```

ค่าดีฟอลต์ คือไดเรกทอรีปัจจุบันเนื่องจากการกำหนดพารามิเตอร์

ในตัวอย่างต่อไปนี้:

```
ls temp mail
```

ไม่มีการกำหนดแฟล็ก และ temp และ mail เป็นพารามิเตอร์ ในกรณีนี้ temp และ mail เป็นชื่อไดเรกทอรี สองชื่อที่แตกต่างกัน คำสั่ง ls แสดงไฟล์ทั้งหมด ยกเว้นไฟล์ที่ซ่อนไว้ในแต่ละไดเรกทอรีเหล่านี้

เมื่อใดก็ตามที่พารามิเตอร์หรืออ็อปชัน-อาร์กิวเมนต์เป็น หรือมีค่าตัวเลข ตัวเลขจะถูกตีความเป็นเลขจำนวนเต็มฐานสิบ ยกเว้นว่ามีการระบุเป็นอย่างอื่น ตัวเลขในช่วง 0 ถึง INT_MAX ตามที่กำหนดไว้ในไฟล์ /usr/include/sys/limits.h มีการรับรู้ทางวากยสัมพันธ์เป็นค่าตัวเลข

หากคำสั่งซึ่งคุณต้องการใช้ยอมรับตัวเลขค่าลบเป็นพารามิเตอร์หรือ อ็อปชัน-อาร์กิวเมนต์ คุณสามารถใช้ตัวเลขในช่วง INT_MIN ถึง INT_MAX ซึ่งทั้งสองค่ามีการกำหนดไว้ในไฟล์ /usr/include/sys/limits.h ซึ่งไม่จำเป็นต้องหมายความว่าตัวเลขทั้งหมดภายในช่วงนั้น ถูกต้อง บางคำสั่งมีข้อกำหนดคุณลักษณะในตัวซึ่งอนุญาตช่วงของตัวเลขที่แคบลง ตัวอย่างเช่น คำสั่ง print บางรายการ หากข้อผิดพลาดถูก สร้างขึ้น ข้อความแสดงข้อผิดพลาดช่วยให้คุณทราบว่าค่าอยู่นอกช่วง ที่สนับสนุน ไม่ใช่ว่าคำสั่งไม่ถูกต้อง

ข้อความสั่งการใช้:

ข้อความสั่งการใช้เป็นวิธีในการแสดงไวยากรณ์คำสั่งและประกอบด้วย สัญลักษณ์ เช่น วงเล็บสี่เหลี่ยม ([]), ปีกกา ({ }), และแถบแนวตั้ง (|)

ข้อมูลต่อไปนี้เป็นตัวอย่างของข้อความสั่งการใช้สำหรับคำสั่ง unget:

```
unget[ -rSID ][ -s ][ -n ] File ...
```

ระเบียบที่ใช้ในข้อความสั่งการใช้มีดังต่อไปนี้:

- รายการที่ต้องถูกบ่อนตามตัวอักษรในบรรทัดคำสั่งอยู่ใน ตัวหนา ไอเท็มเหล่านี้รวมถึงชื่อคำสั่ง แฟล็ก และอักขระตามตัวอักษร
- รายการที่แสดงตัวแปรตัวแปรที่ต้องถูกแทนที่ด้วยชื่อ เป็น *ตัวเอียง* รายการเหล่านี้รวมทั้ง พารามิเตอร์ที่ต่อจากแฟล็ก และพารามิเตอร์ที่คำสั่งอ่าน เช่น *Files* และ *Directories*
- พารามิเตอร์อยู่ในวงเล็บเหลี่ยมเป็นทางเลือก
- พารามิเตอร์ที่อยู่ในเครื่องหมายปีกกาจำเป็นต้องมี
- พารามิเตอร์ที่ไม่ได้อยู่ในวงเล็บเหลี่ยมหรือเครื่องหมายปีกกาจำเป็นต้องมี

- แท่งแนวตั้งหมายถึงคุณเลือกเพียงหนึ่งพารามิเตอร์ ตัวอย่างเช่น [a | b] บ่งชี้ว่าคุณสามารถเลือก a, b, หรือไม่เลือกเลยในลักษณะคล้ายกัน { a | b } บ่งชี้ว่าคุณต้องเลือก a หรือ b
- จุดไข่ปลา (...) แสดงว่าสามารถทำซ้ำพารามิเตอร์บนบรรทัดคำสั่งได้
- เครื่องหมายขีด (-) หมายถึงอินพุตมาตรฐาน

คำสั่ง Shutdown:

หากคุณมีสิทธิผู้ใช้ราก คุณสามารถใช้ คำสั่ง shutdown เพื่อหยุดระบบได้ หากคุณไม่ได้รับอนุญาต ให้ใช้คำสั่ง shutdown ให้คุณเพียงแต่ล็อกเอาต์ออกจาก ระบบปฏิบัติการและปล่อยให้ระบบรันต่อไป

ข้อควรสนใจ: อย่าปิดระบบโดยไม่ได้ปิดก่อน การปิดระบบจะโปรเซสทั้งหมดที่กำลังรันอยู่บนระบบ หากผู้ใช้อื่น กำลังทำงานบนระบบ หรือถ้างานกำลังรันในพื้นที่ ข้อมูลอาจสูญหายได้ ทำโพรซีเจอร์การปิดที่ถูกต้องก่อนคุณหยุด ระบบ

ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
shutdown
```

เมื่อคำสั่ง shutdown เสร็จสมบูรณ์ และระบบปฏิบัติการหยุดรัน คุณจะได้รับข้อความต่อไปนี้:

```
....Shutdown completed....
```

ให้ดูที่คำสั่ง shutdown สำหรับไวยากรณ์ทั้งหมด

การระบุตำแหน่งคำสั่งหรือโปรแกรมอื่น (คำสั่ง whereis):

คำสั่ง whereis ระบุตำแหน่งซอร์ส ไบนารี และส่วนคู่มือสำหรับไฟล์ที่ระบุ คำสั่งพยายามค้นหา โปรแกรมที่ต้องการจากรายการของที่ตั้งมาตรฐาน

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการค้นหาไฟล์ในไดเรกทอรีปัจจุบันที่ไม่มีเอกสารคู่มือ ให้พิมพ์ ดังต่อไปนี้:

```
whereis -m -u *
```

- เมื่อต้องการค้นหาไฟล์ทั้งหมดที่มีชื่อว่า Mail ให้พิมพ์ ดังต่อไปนี้:

```
whereis Mail
```

ระบบ แสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
Mail: /usr/bin/Mail /usr/lib/Mail.rc
```

ให้ดูที่คำสั่ง whereis ใน *Commands Reference, Volume 6* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงผลข้อมูลเกี่ยวกับคำสั่ง (คำสั่ง man):

คำสั่ง man แสดงข้อมูลเกี่ยวกับคำสั่ง รุทีนย่อย และไฟล์

รูปแบบทั่วไปของคำสั่ง man เป็นดังนี้:

```
man CommandName
```

เมื่อต้องการรับข้อมูลเกี่ยวกับคำสั่ง pg ให้พิมพ์ดังต่อไปนี้:

144 AIX เวอร์ชัน 7.2: ระบบปฏิบัติการและการจัดการกับอุปกรณ์

man pg

ระบบ แสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

pg Command

Purpose

Formats files to the display.

Syntax

```
pg [ - Number ] [ -c ] [ -e ] [ -f ] [ -n ] [ -p String ]  
[ -s ] [ +LineNumber | +/Pattern/ ] [ File ... ]
```

Description

The pg command reads a file name from the File parameter and writes the file to standard output one screen at a time. If you specify a - (dash) as the File parameter, or run the pg command without options, the pg command reads standard input. Each screen is followed by a prompt. If you press the Enter key, another page is displayed. Subcommands used with the pg command let you review or search in the file.

ให้ดูที่คำสั่ง **man** ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงผลฟังก์ชันของคำสั่ง (คำสั่ง **whatis**):

คำสั่ง **whatis** ค้นหาคำสั่งที่กำหนด การเรียกระบบ โลบรารีฟังก์ชัน หรือชื่อไฟล์พิเศษ ตามที่ระบุโดยพารามิเตอร์ Command จากฐานข้อมูลซึ่งคุณสร้างโดยใช้คำสั่ง **catman -w**

สำหรับข้อมูลเกี่ยวกับคำสั่ง **catman -w** ให้ดูที่ **catman -w** คำสั่ง **whatis** แสดงบรรทัดส่วนหัว จากส่วนคู่มือ จากนั้น คุณสามารถออกใช้คำสั่ง **man** เพื่อรับข้อมูลเพิ่มเติม สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง **man** ให้ดูที่ **man**

คำสั่ง **whatis** เทากับการใช้ คำสั่ง **man -f**

เมื่อต้องการ ค้นหาสิ่งที่คำสั่ง **ls** ทำให้พิมพ์ดังต่อไปนี้:

```
whatis ls
```

ระบบ แสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
ls(1) -Displays the contents of a directory.
```

ให้ดูที่คำสั่ง **whatis** ใน *Commands Reference, Volume 6* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงผลรายการคำสั่งที่ป้อนก่อนหน้านี้ (คำสั่ง **history**):

ใช้คำสั่ง **history** เพื่อแสดงผลรายการคำสั่งซึ่ง คุณป้อนก่อนหน้านี้

คำสั่ง **history** เป็นคำสั่งในตัว Korn shell ที่แสดงรายการ 16 คำสั่งล่าสุดที่ป้อน Korn shell บันทึกคำสั่งซึ่งคุณป้อนไว้ในไฟล์ประวัติคำสั่ง โดยปกติ มีชื่อว่า `$HOME/.sh_history` การใช้คำสั่งนี้ช่วยประหยัดเวลา เมื่อคุณต้องทำซ้ำคำสั่งก่อนหน้านี้

โดยค่าดีฟอลต์ Korn shell บันทึกข้อความของ 128 คำสั่งล่าสุดสำหรับผู้ใช้ที่ไม่ใช่ผู้ใช้ราก และ 512 คำสั่งสำหรับผู้ใช้ราก ขนาดของไฟล์ประวัติ (ที่ระบุโดยตัวแปรสถานะแวดล้อม *HISTSIZE*) ไม่มีขีดจำกัด แม้ว่าไฟล์ประวัติขนาดใหญ่มากอาจส่งผลให้ Korn shell เริ่มต้นได้ช้าก็ตาม

หมายเหตุ: shell เป้าหมายไม่สนับสนุน ประวัติคำสั่ง

เมื่อต้องการแสดงรายการ คำสั่งซึ่งคุณป้อนก่อนหน้านี้ ให้พิมพ์ดังต่อไปนี้:

```
history
```

คำสั่ง **history** ที่ป้อนด้วยตัวเอง แสดงรายการ 16 คำสั่งที่ป้อนก่อนหน้านี้ ระบบแสดงข้อมูลคล้ายกับตัวอย่าง ต่อไปนี้:

```
928  ls
929  mail
930  printenv MAILMSG
931  whereis Mail
932  whatis ls
933  cd /usr/include/sys
934  ls
935  man pg
936  cd
937  ls | pg
938  lscons
939  tty
940  ls *.txt
941  printenv MAILMSG
942  pwd
943  history
```

รายการแสดง ตำแหน่งของคำสั่งในไฟล์ `$HOME/.sh_history` ก่อน ตามด้วยคำสั่ง

เมื่อต้องการแสดงรายการ คำสั่ง ก่อนหน้านี้ ที่พร้อมให้พิมพ์ดังต่อไปนี้:

```
history -5
```

รายการ คล้ายกับดังต่อไปนี้แสดงขึ้น:

```
939  tty
940  ls *.txt
941  printenv MAILMSG
942  pwd
943  history
944  history -5
```

คำสั่ง **history** ที่ตามด้วย หมายเลขจะแสดงรายการคำสั่งที่ป้อนก่อนหน้านี้ทั้งหมด โดยเริ่มต้นที่ หมายเลขนั้น

เมื่อต้องการแสดงรายการ คำสั่งตั้งแต่หมายเลข 938 ที่พร้อมให้พิมพ์ดังต่อไปนี้:

```
history 938
```

รายการ คล้ายกับดังต่อไปนี้แสดงขึ้น:

```
938 lscons
939 tty
940 ls *.txt
941 printenv MAILMSG
942 pwd
943 history
944 history -5
945 history 938
```

หลักการที่เกี่ยวข้อง:

“เซลล์ระบบปฏิบัติการ” ในหน้า 229

อินเตอร์เฟซของคุณสำหรับระบบปฏิบัติการเรียกว่า *เซลล์*

“การทดแทนประวัติคำสั่ง” ในหน้า 281

ใช้คำสั่งในตัว `fc` เพื่อแสดงรายการหรือแก้ไข ส่วนต่างๆ ของไฟล์ประวัติ เมื่อต้องการเลือกส่วนของไฟล์ที่จะแก้ไขหรือแสดงรายการให้ระบุ หมายเลขหรืออักขระตัวแรกของคำสั่ง

การทำซ้ำคำสั่งโดยใช้สมนาม `r`:

ใช้สมนาม `r Korn` เซลล์เพื่อทำซ้ำคำสั่งก่อนหน้านี้

พิมพ์ `r` แล้วกด Enter และคุณสามารถระบุ หมายเลขหรือตัวอักขระตัวแรกของคำสั่ง

ถ้าคุณต้องการ แสดงรายการจอตแสดงผลที่มีอยู่บนระบบในปัจจุบัน ให้พิมพ์ `lsdisp` ที่พร้อมระบบส่งคืนข้อมูลบนหน้าจอ ถ้าคุณต้องการให้ ส่งคืนข้อมูลเดียวกันมาอีกครั้ง ที่พร้อม ให้พิมพ์ดังต่อไปนี้:

```
r
```

ระบบรันคำสั่ง ที่ป้อนล่าสุดอีกครั้ง ในตัวอย่างนี้ คำสั่ง `lsdisp` จะรัน

เมื่อต้องการทำซ้ำคำสั่ง `ls *.txt` ที่พร้อม ให้พิมพ์ดังต่อไปนี้:

```
r ls
```

สมนาม `r Korn` เซลล์ระบุตำแหน่งคำสั่งล่าสุด ที่ขึ้นต้นด้วยอักขระที่ระบุ

การทดแทนสตริงโดยใช้สมนาม `r`:

คุณสามารถใช้สมนาม `r Korn` เซลล์เพื่อ แก้ไขคำสั่งก่อนจะรัน

ในกรณีนี้ สามารถใช้พารามิเตอร์การทดแทนในรูปแบบ `Old=new` เพื่อแก้ไขคำสั่งก่อนจะรัน

ตัวอย่างต่อไปนี้แสดงวิธีการใช้สมนาม `r`:

- ถ้าบรรทัดคำสั่ง 940 เป็น `ls *.txt` และคุณต้องการรัน `ls *.exe` ที่พร้อม ให้พิมพ์ดังต่อไปนี้:

```
r txt=exe 940
```

คำสั่งนี้รันคำสั่ง 940 เพื่อทดแทน `exe` สำหรับ `txt`

- หากคำสั่งบนบรรทัด 940 เป็นคำสั่งล่าสุดที่ขึ้นต้นด้วย ตัวอักษรตัวพิมพ์เล็ก / คุณยังสามารถพิมพ์ดังต่อไปนี้:

```
r txt=exe l
```

หมายเหตุ: เฉพาะตำแหน่งแรกของสตริง *Old* เท่านั้น ถูกแทนที่ด้วยสตริง *New* การป้อนสมนาม r Korn เซลล์โดยไม่มีหมายเลขคำสั่งหรืออักขระเฉพาะ ทำการทดแทนบนคำสั่งที่ป้อนก่อนหน้านี้

การแก้ไขประวัติคำสั่ง:

ใช้คำสั่งในตัว Korn เซลล์ `fc` เพื่อแสดงรายการ หรือแก้ไขส่วนต่างๆ ของไฟล์ประวัติคำสั่ง

เมื่อต้องการเลือกส่วนของไฟล์ที่จะแก้ไขหรือแสดงรายการ ให้ระบุ หมายเลขหรืออักขระตัวแรกของคำสั่ง คุณสามารถระบุคำสั่งเดียวหรือช่วงของคำสั่ง

หากคุณไม่ได้ระบุโปรแกรมแก้ไขเป็นอาร์กิวเมนต์ของคำสั่งในตัว Korn เซลล์ `fc` จะมีการใช้โปรแกรมแก้ไขที่ระบุโดยตัวแปร `FCEDIT` ถ้าไม่ได้กำหนดตัวแปร `FCEDIT` จะใช้โปรแกรมแก้ไข `/usr/bin/ed` คำสั่งที่แก้ไขมีการพิมพ์และรันเมื่อคุณออกจากโปรแกรมแก้ไข ใช้คำสั่ง `printenv` เพื่อแสดงค่าของตัวแปร `FCEDIT`

ข้อมูลต่อไปนี้เป็นตัวอย่าง วิธีการแก้ไขประวัติคำสั่ง:

- ถ้าคุณต้องการรันคำสั่ง:

```
cd /usr/tmp
```

ซึ่ง คล้ายกับบรรทัดคำสั่ง 933 เป็นอย่างมาก ที่พร้อมทำให้พิมพ์ ดังต่อไปนี้:

```
fc 933
```

ณ จุดนี้โปรแกรมแก้ไขดีฟอลต์ของคุณปรากฏขึ้นพร้อมกับบรรทัดคำสั่ง 933 เปลี่ยน `include/sys` เป็น `tmp` และเมื่อคุณออกจากโปรแกรมแก้ไข คำสั่งที่แก้ไขจะรัน

- คุณยังสามารถระบุโปรแกรมแก้ไขซึ่งคุณต้องการใช้ในคำสั่ง `fc` ได้ด้วย ตัวอย่างเช่น ถ้าคุณต้องการแก้ไขคำสั่งโดยใช้โปรแกรมแก้ไข `/usr/bin/vi` ที่พร้อมทำให้พิมพ์ดังต่อไปนี้:

```
fc -e vi 933
```

ณ จุดนี้โปรแกรมแก้ไข `vi` ปรากฏขึ้นพร้อมกับบรรทัดคำสั่ง 933

- คุณยังสามารถระบุช่วงของคำสั่งที่จะแก้ไขได้ด้วย ตัวอย่างเช่น ถ้าคุณ ต้องการแก้ไขคำสั่ง 930 ถึง 940 ที่พร้อมทำให้พิมพ์ดังต่อไปนี้:

```
fc 930 940
```

ณ จุดนี้โปรแกรมแก้ไข ดีฟอลต์ของคุณปรากฏขึ้นพร้อมกับบรรทัดคำสั่ง 930 ถึง 940 เมื่อคุณออกจากโปรแกรมแก้ไข คำสั่งทั้งหมดซึ่งปรากฏขึ้นในโปรแกรมแก้ไขจะรันตามลำดับ

การสร้างสมนามคำสั่ง (คำสั่ง alias shell):

สมนาม อนุญาตให้คุณสร้างชื่อชื่อย่อสำหรับคำสั่ง ชื่อไฟล์ หรือข้อความเซลล์ใดๆ โดยใช้สมนาม คุณสามารถประหยัดเวลาได้มากเมื่อ ทำภารกิจซึ่งคุณทำบ่อยครั้ง คุณสามารถสร้างสมนามคำสั่งได้

ใช้คำสั่งในตัว Korn เซลล์ `alias` เพื่อกำหนดค่าเป็นสมนามของคำสั่งบางรายการ คุณสามารถใช้สมนามเพื่อกำหนดคำสั่งในตัวอีกครั้ง แต่ไม่ใช่กำหนดค่าที่สงวนไว้อีกครั้ง

อักขระตัวแรก ของชื่อสมนามสามารถเป็นอักขระที่พิมพ์ได้ใดๆ ยกเว้น เมตาอักขระ อักขระที่เหลืองต้องเหมือนกับชื่อไฟล์ที่ถูกต้อง

รูปแบบสำหรับการสร้างสมนามเป็นดังนี้:

```
alias Name=String
```

ซึ่ง พารามิเตอร์ *Name* ระบุชื่อของสมนาม และพารามิเตอร์ *String* ระบุสตริงของอักขระ หาก *String* มีพื้นที่ว่าง ให้ใส่ไว้ในเครื่องหมายอัฒประกาศ

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการสร้างสมนาม:

- เมื่อต้องการสร้างสมนามสำหรับคำสั่ง `rm -i` (พร้อมต์คุณ ก่อนการลบไฟล์) ที่พร้อมต์ให้พิมพ์ดังต่อไปนี้:

```
alias rm="/usr/bin/rm -i"
```

ในตัวอย่างนี้ เมื่อใดก็ตามที่คุณป้อนคำสั่ง `rm` คำสั่งจริงที่ทำได้คือ `/usr/bin/rm -i`

- เมื่อต้องการสร้างสมนามที่ชื่อ `dir` สำหรับคำสั่ง `ls -aF | pg` (ซึ่งแสดงข้อมูลรายละเอียดของไฟล์ทั้งหมดในไดเรกทอรีปัจจุบัน รวมถึงไฟล์ที่มองไม่เห็น โดยจะทำเครื่องหมายไฟล์ปฏิบัติการด้วย `*` และไดเรกทอรีด้วย `/` และเลื่อน แต่ละหน้าจอ) ที่พร้อมต์ให้พิมพ์ดังต่อไปนี้:

```
alias dir="/usr/bin/ls -aF | pg"
```

ในตัวอย่างนี้ เมื่อใดก็ตามที่คุณป้อนคำสั่ง `dir` คำสั่งจริงที่ทำได้คือ `/usr/bin/ls -aF | pg`

- เมื่อต้องการแสดงสมนามทั้งหมดที่คุณมี ที่พร้อมต์ให้พิมพ์ดังต่อไปนี้:

```
alias
```

ระบบ แสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
rm="/usr/bin/rm -i"
dir="/usr/bin/ls -aF | pg"
```

หลักการที่เกี่ยวข้อง:

“การกำหนดสมนามคำสั่งใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 281

Korn เซลล์หรือ POSIX เซลล์อนุญาตให้คุณสร้างสมนามให้กับ คำสั่งแบบกำหนดเองได้

การสนับสนุนอักขระสากลในการจัดรูปแบบข้อความ:

คุณสามารถใช้คำสั่ง `text formatting` เพื่อทำงานกับข้อความที่ประกอบขึ้น เป็นชุดอักขระแบบขยายสากลซึ่งใช้โดยภาษายุโรป

ชุดอักขระแบบขยายสากลนำเสนอ อักขระและสัญลักษณ์ที่ใช้ในภาษายุโรปจำนวนมาก ตลอดจนชุดย่อย ASCII ซึ่งประกอบเป็นอักขระภาษาอังกฤษ ตัวเลข และเครื่องหมายวรรคตอน

อักขระทั้งหมดในชุดอักขระแบบขยายภาษายุโรป มีรูปแบบ ASCII สามารถใช้รูปแบบเหล่านี้เพื่อแสดงถึงอักขระแบบขยายในอินพุต หรือสามารถป้อนอักขระได้โดยตรงด้วยอุปกรณ์ เช่น คีย์บอร์ดที่สนับสนุนอักขระแบบขยายภาษายุโรป

คำสั่ง `text-formatting` ต่อไปนี้สนับสนุนภาษาสากลทั้งหมดที่ใช้อักขระไบต์เดียว คำสั่งเหล่านี้ตั้ง อยู่ใน `/usr/bin` (คำสั่งที่ระบุพร้อมกับเครื่องหมายดอกจัน `*`) สนับสนุนการประมวลผลข้อความสำหรับภาษาหลายไบต์)

addbib*	hyphen	pic*	pstext
checkmm	ibm3812	ps4014	refer*
checknr*	ibm3816	ps630	roffbib*
col*	ibm5587G*	psbanne	soelim*
colcrt	ibm5585H-T*	psdit	sortbib*
deroff*	indxbib*	psplot	tbl*
enscript	lookbib*	psrev	troff*
eqn*	makedev*	psroff	vgrind
grap*	neqn*	psrv	xpreview*
hplj	nroff*		

คำสั่ง text-formatting และแมโครแพ็คเกจที่ไม่มีอยู่ใน รายการข้างบนยังไม่มีการเปิดใช้งานเพื่อประมวลผลอักขระสากล
หลักการที่เกี่ยวข้อง:

“การสนับสนุนอักขระหลายไบต์ในการจัดรูปแบบข้อความ”

สามารถใช้คำสั่งการจัดรูปแบบข้อความบางคำสั่งเพื่อประมวลผลข้อความ สำหรับภาษาหลายไบต์

การจัดรูปแบบข้อความด้วยอักขระไบต์เดี่ยวแบบขยาย:

ถ้าอุปกรณ์อินพุตของคุณสนับสนุนอักขระจากชุดอักขระแบบขยาย ภาษายุโรป คุณสามารถป้อนอักขระนั้นได้โดยตรง

มิฉะนั้น ให้ใช้รูปแบบลำดับ ASCII escape ต่อไปนี้เพื่อแสดงแทนอักขระดังกล่าว:

รูปแบบ `\[N]` โดยที่ *N* คือโค้ดฐานสิบหก 2- หรือ 4-ตำแหน่งสำหรับอักขระ

หมายเหตุ: ไม่สนับสนุนรูปแบบ `NCesc \<xx>` อีกต่อไป

ข้อความที่มีอักขระแบบขยายคือเอาต์พุตตาม ระเบียบการจัดรูปแบบของภาษาที่ใช้อักขระที่ไม่ได้กำหนดสำหรับอินเตอร์เฟซของอุปกรณ์เอาต์พุตเฉพาะไม่สร้างเอาต์พุตหรือการบ่งชี้ข้อผิดพลาด

แม้ว่าชื่อของคำร้องขอ แมโครแพ็คเกจ และ คำสั่งใช้ภาษาอังกฤษ แต่ส่วนใหญ่แล้วสามารถยอมรับอินพุต (เช่น ชื่อไฟล์ และ พารามิเตอร์) ที่มีอักขระในชุดอักขระแบบขยายภาษายุโรปได้

สำหรับคำสั่ง `nroff` และ `troff` และ preprocessors ของคำสั่งนั้น อินพุตคำสั่งต้องเป็น ASCII มิฉะนั้นจะส่งผลให้เกิดข้อผิดพลาดไวยากรณ์ที่ไม่สามารถแก้ไขได้ อักขระสากลไม่ว่าจะเป็นแบบไบต์เดี่ยวหรือหลายไบต์ สามารถป้อนได้เมื่อใส่ไว้ในเครื่องหมายอัญประกาศและอยู่ภายในข้อความอื่น ที่จะจัดรูปแบบ ตัวอย่างเช่น การใช้แมโครจากคำสั่ง `pic`:

```
define foobar % SomeText %
```

After the define directive, the specified name, foobar, must be ASCII. อย่างไรก็ตาม ข้อความที่นำมาแทนที่ซึ่งเป็น SomeText สามารถมีอักขระที่ไม่ใช่ ASCII

การสนับสนุนอักขระหลายไบต์ในการจัดรูปแบบข้อความ:

สามารถใช้คำสั่งการจัดรูปแบบข้อความบางคำสั่งเพื่อประมวลผลข้อความ สำหรับภาษาหลายไบต์

คำสั่งเหล่านี้ที่ระบุพร้อมกับเครื่องหมายดอกจัน (*) ในรายการภายใต้การสนับสนุนอักขระสากลในการจัดรูปแบบข้อความ คำสั่งการจัดรูปแบบข้อความที่ไม่ได้อยู่ในรายการยังไม่ได้เปิดใช้งานสำหรับการประมวลผลอักขระสากล

หากอุปกรณ์อินพุตของคุณสนับสนุน สามารถป้อนอักขระหลายไบต์โดยตรงได้ มิฉะนั้น คุณสามารถป้อนอักขระหลายไบต์ใดๆ ในรูปแบบ ASCII[N] โดยที่ N คือเลขฐานสิบหก 2-, 4-, 6-, 7-, หรือ 8- ตำแหน่งที่เข้ารหัสสำหรับอักขระ

แม้ว่าชื่อของคำสั่งขอ แมโคร และคำสั่ง ใช้ภาษาอังกฤษ แต่ส่วนใหญ่แล้วสามารถยอมรับอินพุต (เช่น ชื่อไฟล์และ พารามิเตอร์) ที่มีอักขระหลายไบต์ชนิดต่างๆ ได้

หากคุณคุ้นเคยกับการใช้คำสั่งการจัดรูปแบบข้อความ ที่มีข้อความไบต์เดียว รายการต่อไปนี้สรุปอักขระที่น่าจดจำหรือใช้เฉพาะในโลแคลหลายไบต์:

- ข้อความไม่มีการใช้ยัติภังค์
- ต้องใช้ชนิดรูปแบบพิเศษสำหรับเอาต์พุตตัวเลข หลายไบต์ มีชนิดรูปแบบภาษาญี่ปุ่น
- ข้อความเป็นเอาต์พุตในบรรทัดแนวนอน ที่ป้อนจากซ้าย ไปขวา
- ช่องว่างระหว่างอักขระคงที่ ดังนั้นอักขระจึงปรับแนวในคอลัมน์ โดยอัตโนมัติ
- อักขระที่ไม่ได้กำหนดสำหรับอินเตอร์เฟซของอุปกรณ์เอาต์พุตเฉพาะไม่สร้างเอาต์พุตหรือการบ่งชี้ข้อผิดพลาด

หลักการที่เกี่ยวข้อง:

“การสนับสนุนอักขระสากลในการจัดรูปแบบข้อความ” ในหน้า 149

คุณสามารถใช้คำสั่ง text formatting เพื่อทำงานกับข้อความที่ประกอบขึ้น เป็นชุดอักขระแบบขยายสากลซึ่งใช้โดยภาษายุโรป

การแสดงปฏิทิน:

คุณสามารถเขียนปฏิทินไปยังเอาต์พุตมาตรฐานได้โดยใช้คำสั่ง `cal`

พารามิเตอร์ `Month` ระบุเดือนซึ่ง คุณต้องการปฏิทิน ซึ่งสามารถเป็นตัวเลขตั้งแต่ 1 ถึง 12 สำหรับเดือนมกราคมถึง ธันวาคมตามลำดับ ถ้าไม่ระบุ `Month` คำสั่ง `cal` จะกำหนด ดีฟอลต์เป็นเดือนปัจจุบัน

พารามิเตอร์ `Year` ระบุปีซึ่ง คุณต้องการปฏิทิน เนื่องจากคำสั่ง `cal` สามารถแสดงปฏิทินปีใดก็ได้ตั้งแต่ 1 ถึง 9999 คุณต้องป้อน ค่าปีแบบเต็มแทนการป้อนเฉพาะสองหลักสุดท้าย ถ้าไม่ระบุ `Year` คำสั่ง `cal` จะกำหนดดีฟอลต์เป็นปีปัจจุบัน

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง `cal`:

1. เมื่อต้องการแสดงปฏิทินสำหรับ February 2002 ที่เวิร์กสเตชันของคุณ ให้พิมพ์:

```
cal 2 2002
```

2. กด Enter

3. เมื่อต้องการพิมพ์ปฏิทินสำหรับปี 2002 ให้พิมพ์:

```
cal 2002 | qprt
```

4. กด Enter

ให้ดูที่คำสั่ง `cal` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงข้อความบันทึกช่วยเตือน:

คุณสามารถแสดงข้อความบันทึกช่วยเตือนโดยอ่านไฟล์ที่ชื่อ `calendar` ไฟล์นี้สร้างขึ้นในไดเรกทอรีโฮมของคุณด้วยคำสั่ง `calendar` คำสั่งเขียนบรรทัดใดๆ ในไฟล์ที่มีวันที่วันนี้หรือพรุ่งนี้ไปยัง เอาต์พุตมาตรฐาน

คุณสามารถอ่านไฟล์ที่ชื่อ calendar ซึ่งคุณสร้างขึ้น ในไดเรกทอรีโฮมของคุณด้วยคำสั่ง `calendar` คำสั่งเขียนบรรทัดใดๆ ในไฟล์ที่มีวันนี้หรือพรุ่งนี้ไปยังเอาต์พุตมาตรฐาน

คำสั่ง `calendar` รู้จักรูปแบบวันที่ เช่น Dec. 7 หรือ 12/7 และยังรู้จักอักขระพิเศษเครื่องหมายดอกจัน (*) เมื่อ มีเครื่องหมายทับ (/) ตามหลัง โดยจะแปลความหมายตัวอย่างเช่น */7 เป็นการระบุวันที่เจ็ดของทุกเดือน

ในวันศุกร์ คำสั่ง `calendar` เขียนบรรทัดทั้งหมดที่มี วันที่ตรงกับวันศุกร์ วันเสาร์ วันอาทิตย์ และวันจันทร์ อย่างไรก็ตาม คำสั่งไม่ทราบถึงวันหยุด ในวันหยุด คำสั่งทำงานตามปกติและให้เฉพาะตารางเวลาของวันถัดไปเท่านั้น

การใช้ไฟล์ปฏิทินปกติ

ไฟล์ `calendar` ปกติ อาจมีลักษณะคล้ายดังต่อไปนี้:

```
*/25 - Prepare monthly report
Aug. 12 - Fly to Denver
aug 23 - board meeting
Martha out of town - 8/23, 8/24, 8/25
8/24 - Mail car payment
sat aug/25 - beach trip
August 27 - Meet with Simmons
August 28 - Meet with Wilson
```

เมื่อต้องการรันคำสั่ง `calendar` ให้พิมพ์:

```
calendar
```

หากวันนี้เป็นวันศุกร์ที่ 24 สิงหาคม คำสั่ง `calendar` จะแสดงดังต่อไปนี้:

```
*/25 - Prepare monthly report
Martha out of town - 8/23, 8/24, 8/25
8/24 - Mail car payment
sat aug/25 - beach trip
August 27 - Meet with Simmons
```

การใช้ไฟล์ปฏิทินที่มีคำสั่ง include

ไฟล์ `calendar` ที่มีคำสั่ง `include` อาจมีลักษณะคล้ายตัวอย่างต่อไปนี้:

```
#include </tmp/out>
1/21 -Annual review
1/21 -Weekly project meeting
1/22 *Meet with Harrison in Dallas*
Doctor's appointment - 1/23
1/23 -Vinh's wedding
```

เมื่อต้องการรันคำสั่ง `calendar` ให้พิมพ์:

```
calendar
```

หากวันนี้เป็นวันพุธที่ 21 มกราคม คำสั่ง `calendar` จะแสดงดังต่อไปนี้:

Jan.21 Goodbye party for David
Jan.22 Stockholder meeting in New York
1/21 -Annual review
1/21 -Weekly project meeting
1/22 *Meet with Harrison in Dallas*

ผลลัพธ์ของคำสั่ง `calendar` ระบุไฟล์ `/tmp/out` ที่มีบรรทัดต่อไปนี้:

Jan.21 Goodbye party for David
Jan.22 Stockholder meeting in New York

ให้ดูที่คำสั่ง `calendar` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

การแยกตัวประกอบตัวเลข:

คุณสามารถแยกตัวประกอบตัวเลขโดยใช้คำสั่ง `factor`

เมื่อเรียกใช้โดยไม่ได้ระบุค่าสำหรับพารามิเตอร์ `Number` คำสั่ง `factor` จะรอให้คุณป้อนตัวเลขค่าบวก ที่น้อยกว่า 1E14 (100,000,000,000,000) จากนั้นเขียนตัวประกอบจำนวนเฉพาะของ เลขจำนวนนั้นไปยังเอาต์พุตมาตรฐาน คำสั่งแสดงแต่ละตัวประกอบตามลำดับและจำนวนครั้งที่เหมาะสม ถ้ามีการใช้คำสั่งเดียวกันมากกว่าหนึ่งครั้ง เมื่อต้องการออกให้ป้อน 0 (ศูนย์) หรืออักขระที่ไม่ใช่ตัวเลขใดๆ

เมื่อเรียกใช้พร้อมอาร์กิวเมนต์ คำสั่ง `factor` จะพิจารณาตัวประกอบจำนวนเฉพาะของพารามิเตอร์ `Number` เขียนผลลัพธ์ไปยังเอาต์พุตมาตรฐานและออกจากการทำงาน

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการคำนวณตัวประกอบ:

1. เมื่อต้องการคำนวณตัวประกอบหลักของตัวเลข 123 ให้พิมพ์:

```
factor 123
```

2. กด Enter ข้อมูลต่อไปนี้จะแสดงขึ้น:

```
123 3 41
```

ให้ดูที่คำสั่ง `factor` ใน *Commands Reference, Volume 2* สำหรับไวยากรณ์ทั้งหมด

การระบุตำแหน่งคำสั่งโดยใช้คีย์เวิร์ด:

คุณสามารถแสดงส่วนแมนเพจที่มี `Keywords` ที่กำหนดใดๆ อยู่ในหัวเรื่องได้โดยใช้คำสั่ง `apropos`

คำสั่ง `apropos` จะพิจารณาแต่ละคำแยกกัน และไม่คำนึงว่าเป็นตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็ก คำที่เป็นส่วนหนึ่งของคำอื่นๆ จะแสดงขึ้นเช่นกัน ตัวอย่างเช่น เมื่อค้นหาคำ `compile` คำสั่ง `apropos` จะค้นหาในอินสแตนซ์ทั้งหมด ของคำ `compiler`

หมายเหตุ: ฐานข้อมูลที่มี คีย์เวิร์ดคือ `/usr/share/man/what is` ซึ่งอันดับแรกต้อง สร้างขึ้นด้วยคำสั่ง `catman -w`

คำสั่ง `apropos` เทียบเท่ากับ การใช้คำสั่ง `man` กับตัวเลือก `-k`

ตัวอย่างเช่น เมื่อต้องการค้นหาส่วนคู่มือที่มีคำว่า `password` อยู่ในหัวเรื่อง ให้พิมพ์:

```
apropos password
```

กด Enter

ให้ดูที่คำสั่ง **apropos** ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

โพรเซส

โปรแกรมหรือคำสั่งที่กำลังรันอยู่จริงบนคอมพิวเตอร์ มีการเรียกว่า *โพรเซส*

โพรเซสมีอยู่ในลำดับชั้นพาเรนต์-ชายน โพรเซสที่เริ่มต้นโดยโปรแกรม หรือคำสั่งคือ *โพรเซสพาเรนต์* และ *โพรเซสชายน* คือผลิตภัณฑ์ของโพรเซสพาเรนต์ โพรเซสพาเรนต์หนึ่งสามารถมีได้หลายโพรเซสชายน แต่โพรเซสชายนหนึ่งสามารถมีได้หนึ่งพาเรนต์เท่านั้น

ระบบกำหนดหมายเลขประจำตัวโพรเซส (หมายเลข PID) ให้กับแต่ละ โพรเซสเมื่อโพรเซสเริ่มต้น หากคุณเริ่มต้นโปรแกรมเดียวกันหลายครั้ง โปรแกรมนั้นจะมีหมายเลข PID ที่แตกต่างกันในแต่ละครั้ง

เมื่อโพรเซสเริ่มต้นขึ้นบนระบบ โพรเซสจะใช้บางส่วนของรีซอร์สระบบ ที่มีอยู่ เมื่อมีโพรเซสที่กำลังรันมากกว่าหนึ่งโพรเซสตัวจัดตารางเวลาที่สร้างขึ้น ภายในระบบปฏิบัติการจะมอบส่วนแบ่งใช้ของเวลาของคอมพิวเตอร์ให้กับแต่ละ โพรเซส ตามข้อมูลลำดับความสำคัญที่กำหนด ลำดับความสำคัญเหล่านี้สามารถเปลี่ยนได้ โดยใช้คำสั่ง **nice** หรือ **renice**

หมายเหตุ: เมื่อต้องการเปลี่ยนลำดับความสำคัญของโพรเซสเป็นระดับที่สูงขึ้น คุณต้องมีสิทธิ์ผู้ซากรัก ผู้ใช้ทั้งหมดสามารถลดลำดับความสำคัญของโพรเซสที่ผู้ใช้จะเริ่มต้นได้โดยใช้ คำสั่ง **nice** หรือของโพรเซสที่ผู้ใช้เริ่มต้นแล้ว โดยใช้คำสั่ง **renice**

รายการต่อไปนี้อธิบายชนิดของโพรเซส:

กระบวนการพื้นหน้าและกระบวนการส่วนหลัง

โพรเซสที่ต้องการให้ผู้ใช้เริ่มต้นหรือโต้ตอบด้วย เรียกว่า *กระบวนการพื้นหน้า* โพรเซสที่รันโดยเป็นอิสระจากผู้ใช้ เรียกว่า *กระบวนการส่วนหลัง* โปรแกรมและคำสั่ง รันเป็นกระบวนการพื้นหน้าโดยค่าดีฟอลต์ เมื่อต้องการรันโพรเซสในพื้นหลัง ให้ใส่เครื่องหมาย ampersand (&) ตามหลังชื่อคำสั่งซึ่งคุณใช้เพื่อ เริ่มต้นโพรเซส

โพรเซส Daemon

Daemons คือโพรเซสที่รันโดยไม่ได้ตั้งใจ Daemon อยู่ในพื้นหลัง เสมอและมีอยู่ตลอดเวลา โดยปกติ daemons เริ่มต้นขึ้น เมื่อระบบเริ่มต้น และรันจนกว่าระบบหยุด โดยปกติ โพรเซส daemon ทำเซอร์วิสระบบและมีอยู่ตลอดเวลา สำหรับภารกิจหรือผู้ซากรักมากกว่า หนึ่งราย โพรเซส daemon เริ่มต้นโดยผู้ซากรักหรือเชลล์ร่าก และสามารถหยุดโดยผู้ซากรักเท่านั้น ตัวอย่างเช่น โพรเซส **qdaemon** นำเสนอการเข้าถึงรีซอร์สระบบ เช่น เครื่องพิมพ์ Daemon ทั่วไปอีกอย่างหนึ่ง คือ **sendmail daemon**

กระบวนการ zombie

กระบวนการ zombie คือโพรเซสตายที่ไม่ได้ดำเนินการอีกต่อไป แต่ยังคงเป็นที่รู้จักในตารางโพรเซส (กล่าวอีกอย่างคือ มีหมายเลข PID) ไม่มีการจัดสรรพื้นที่ว่างระบบอื่นให้กับโพรเซสชนิดนี้ กระบวนการ zombie ถูกทำลาย หรือออก และมีอยู่ต่อไปในตารางโพรเซสจนกว่า โพรเซสพาเรนต์ตายหรือปิดและรีเซ็ตระบบ กระบวนการ zombie แสดงเป็น **<defunct>** เมื่อแสดงรายการโดยคำสั่ง **ps**

สตาร์ทอัพโพรเซส:

คุณเริ่มต้นกระบวนการพื้นหน้าจากสแตชันจอแสดงผลโดยการป้อน ชื่อโปรแกรมหรือชื่อคำสั่งที่พร้อมระบบ

หลังจากกระบวนการพื้นหน้าเริ่มต้นขึ้นแล้ว โพรเซสจะโต้ตอบกับคุณ ที่สแตชันจอแสดงผลของคุณจนกว่าโพรเซสเสร็จสมบูรณ์ การโต้ตอบอื่น (ตัวอย่างเช่น การป้อนคำสั่งอื่น) ไม่สามารถทำที่สแตชันจอแสดงผลได้จนกว่า โพรเซสเสร็จสิ้น หรือคุณยกเลิกโพรเซสนั้น

ผู้ขายหนึ่งสามารถรันได้มากกว่าหนึ่งโพรเซสในเวลาเดียวกัน โดยมีจำนวน ดีฟอลต์สูงสุดเป็น 40 โพรเซสต่อผู้ใช้ การเริ่มต้นโพรเซสในพื้นที่

เมื่อต้องการเริ่มต้นโพรเซสในพื้นที่ ให้ป้อนชื่อคำสั่ง พร้อมกับพารามิเตอร์และแฟล็กที่เหมาะสม:

```
$ CommandName
```

การเริ่มต้นโพรเซสในพื้นที่

เมื่อต้องการรันโพรเซสในพื้นที่ ให้พิมพ์ชื่อคำสั่งพร้อมกับ พารามิเตอร์และแฟล็กที่เหมาะสม ตามด้วยเครื่องหมาย ampersand (&):

```
$ CommandName&
```

เมื่อ โพรเซสกำลังรันอยู่ในพื้นที่ คุณสามารถทำภารกิจเพิ่มเติมโดยการป้อน คำสั่งอื่นที่สแตชันจอแสดงผลของคุณ โดยทั่วไป กระบวนการส่วนหลัง มีประโยชน์มากที่สุดสำหรับคำสั่งที่ใช้เวลารันนาน อย่างไรก็ตาม เนื่องจากกระบวนการส่วนหลังเพิ่มจำนวนงานทั้งหมดที่ตัวประมวลผลกำลังทำ ดังนั้น กระบวนการส่วนหลังจึงอาจทำให้ส่วนที่เหลือของระบบทำงานช้าลงได้

โพรเซสส่วนใหญ่กำหนดทิศทาง เอาต์พุตไปยังเอาต์พุตมาตรฐาน แม้ว่าโพรเซสรันในพื้นที่ ยกเว้นว่ามี การ เปลี่ยนทิศทาง เอาต์พุตมาตรฐานจะไปยังอุปกรณ์แสดงผล เนื่องจากเอาต์พุต จากกระบวนการส่วนหลังอาจรบกวนงานอื่นของคุณบนระบบ โดยปกติแล้ว วิธีปฏิบัติที่ดีคือการเปลี่ยนทิศทางเอาต์พุตของกระบวนการส่วนหลัง ไปยังไฟล์หรือเครื่องพิมพ์ จากนั้น คุณสามารถดูเอาต์พุตในเวลาใดก็ตามที่คุณพร้อม

หมายเหตุ: ภายใต้สถานการณ์บางอย่าง เมื่อรันในพื้นที่ โพรเซสอาจสร้างเอาต์พุตในลำดับที่แตกต่างจากเมื่อรันในพื้นที่ โปรแกรมเมอร์ อาจต้องการใช้รูทีนย่อย fflush เพื่อให้มั่นใจว่าเอาต์พุตเกิดขึ้นในลำดับที่ถูกต้อง โดยไม่คำนึงว่า โพรเซสรันในพื้นที่หรือพื้นที่

ในขณะที่กระบวนการส่วนหลัง กำลังรัน คุณสามารถตรวจสอบสถานะได้โดยใช้คำสั่ง ps

คำสั่งการตรวจสอบสถานะโพรเซส (คำสั่ง ps):

ในทุกเวลาที่ระบบกำลังรัน โพรเซสจะกำลังรันด้วย คุณ สามารถใช้คำสั่ง ps เพื่อค้นหาโพรเซสที่กำลังรัน และแสดงข้อมูลเกี่ยวกับโพรเซสเหล่านั้น

คำสั่ง ps มีหลายแฟล็กที่ช่วยให้คุณ สามารถ ระบุโพรเซสซึ่งจะแสดงรายการและข้อมูลที่จะแสดงขึ้นเกี่ยวกับแต่ละ โพรเซส

เมื่อต้องการแสดงโพรเซสทั้งหมดที่กำลังรันบนระบบของคุณ ที่พร้อมให้พิมพ์ดังต่อไปนี้:

```
ps -ef
```

ระบบแสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
USER  PID  PPID  C   STIME  TTY  TIME CMD
root   1    0     0   Jun 28  -   3:23 /etc/init
root  1588  6963  0   Jun 28  -   0:02 /usr/etc/biod 6
root  2280  1     0   Jun 28  -   1:39 /etc/syncd 60
mary  2413  16998  2  07:57:30  -   0:05 aixterm
mary  11632 16998  0  07:57:31  lft/1 0:01 xbiff
mary  16260 2413  1  07:57:35  pts/1 0:00 /bin/ksh
mary  16469  1     0  07:57:12  lft/1 0:00 ksh /usr/lpp/X11/bin/xinit
mary  19402 16260  20 09:37:21  pts/1 0:00 ps -ef
```

คอลัมน์ในเอาต์พุตก่อนหน้านี้มีการกำหนดไว้ดังนี้:

ไอเท็ม	คำอธิบาย
USER	ชื่อล็อกอินของผู้ใช้
PID	ID โพรเซส
PPID	ID โพรเซสพารেন্ট
C	การใช้ CPU ของโปรเซส
STIME	เวลาเริ่มต้นของโปรเซส
TTY	เวิร์กสเตชันควบคุมของโปรเซส
TIME	เวลาดำเนินการทั้งหมดของโปรเซส
CMD	คำสั่ง

ในตัวอย่างก่อนหน้า ID โพรเซสของคำสั่ง `ps -ef` คือ 19402 ID โพรเซสพารেন্টคือ 16260 คำสั่ง `/bin/ksh`

ถ้ารายการยาวเกินไป ส่วนบนสุดจะหลุดจากหน้าจอ เมื่อต้องการแสดง รายการหนึ่งหน้า (หน้าจอ) ในแต่ละครั้ง ให้พิมพ์คำสั่ง `ps` เข้ากับคำสั่ง `pg` ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
ps -ef | pg
```

เมื่อต้องการแสดงข้อมูลสถานะของโปรเซสทั้งหมดที่กำลังรันบนระบบของคุณ ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
ps gv
```

คำสั่งรูปแบบนี้แสดงตัวเลขสถิติของแต่ละโปรเซสที่ใช้งานอยู่เอาต์พุตจากคำสั่งนี้มีลักษณะคล้ายกับตัวอย่างต่อไปนี้:

```
PID  TTY  STAT  TIME  PGIN  SIZE  RSS  LIM  TSIZ  TRS  %CPU  %MEM  COMMAND
0    -   A     0:44  7     8     8    xx   0     0  0.0  0.0  swapper
1    -   A     1:29  518   244   140  xx   21    24  0.1  1.0  /etc/init
771  -   A     1:22  0     16    16   xx   0     0  0.0  0.0  kproc
1028 -   A     0:00  10    16    8     xx   0     0  0.0  0.0  kproc
1503 -   A     0:33  127   16    8     xx   0     0  0.0  0.0  kproc
1679 -   A     1:03  282   192   12 32768 130  0  0.7  0.0  pcidosvr
2089 -   A     0:22  918   72    28   xx   1     4  0.0  0.0  /etc/sync
2784 -   A     0:00  9     16    8     xx   0     0  0.0  0.0  kproc
2816 -   A     5:59  6436  2664  616   8    852  156  0.4  4.0  /usr/lpp/
3115 -   A     0:27  955   264   128  xx   39    36  0.0  1.0  /usr/lib/
3451 -   A     0:00  0     16    8     xx   0     0  0.0  0.0  kproc
3812 -   A     0:00  21    128   12 32768 34  0  0.0  0.0  usr/lib/lpd/
3970 -   A     0:00  0     16    8     xx   0     0  0.0  0.0  kproc
4267 -   A     0:01  169   132   72 32768 16  16  0.0  0.0  /etc/sysl
4514 lft/0 A     0:00  60    200   72   xx   39    60  0.0  0.0  /etc/gett
4776 pts/3 A     0:02  250   108   280   8    303  268  0.0  2.0  -ksh
5050 -   A     0:09  1200  424   132 32768 243  56  0.0  1.0  /usr/sbin
```

```

5322 - A 0:27 1299 156 192 xx 24 24 0.0 1.0 /etc/cron
5590 - A 0:00 2 100 12 32768 11 0 0.0 0.0 /etc/writ
5749 - A 0:00 0 208 12 xx 13 0 0.0 0.0 /usr/lpp/
6111 - T 0:00 66 108 12 32768 47 0 0.0 0.0 /usr/lpp/

```

ให้ดูที่คำสั่ง `ps` ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

การตั้งค่าลำดับความสำคัญแรกเริ่มของโปรเซส (คำสั่ง `nice`):

คุณสามารถตั้งค่าลำดับความสำคัญแรกเริ่มของโปรเซสเป็นค่าที่ต่ำกว่า ลำดับความสำคัญพื้นฐานที่จัดตารางเวลาได้

เมื่อต้องการตั้งค่าลำดับความสำคัญแรกเริ่มของโปรเซสเป็นค่าที่ต่ำกว่า ลำดับความสำคัญพื้นฐานที่จัดตารางเวลา ให้ใช้คำสั่ง `nice` เพื่อเริ่มต้นโปรเซส

หมายเหตุ: เมื่อต้องการรันโปรเซสที่ลำดับความสำคัญสูงกว่าลำดับความสำคัญพื้นฐานที่จัดตาราง เวลา คุณต้องมีสิทธิ์ผู้ใช้ ราก

เมื่อต้องการตั้งค่าลำดับความสำคัญแรกเริ่มของโปรเซส ให้พิมพ์ดังต่อไปนี้:

```
nice -n Number CommandString
```

โดยที่ *Number* คือ ค่าในช่วงตั้งแต่ 0 ถึง 39 โดย 39 เป็นลำดับความสำคัญต่ำสุด *nice value* คือค่าฐานสิบของลำดับความ สำคัญที่จัดตารางเวลาโดยระบบ ของโปรเซส ตัวเลขยิ่งสูง ลำดับความสำคัญยิ่งต่ำ ถ้าคุณใช้ศูนย์ โปรเซสจะรันที่ลำดับความ สำคัญพื้นฐานที่จัดตารางเวลาของโปรเซส *CommandString* คือ คำสั่งและแฟล็กและพารามิเตอร์ซึ่งคุณต้องการรัน

ให้ดูที่คำสั่ง `nice` ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

คุณยังสามารถใช้คำสั่ง `smit nice` เพื่อทำภารกิจนี้ได้ด้วย

การเปลี่ยนลำดับความสำคัญของโปรเซสที่กำลังรัน (คำสั่ง `renice`):

คุณสามารถเปลี่ยนลำดับความสำคัญที่จัดตารางเวลาของโปรเซสที่กำลังรัน เป็นค่าที่ต่ำกว่าหรือสูงกว่าลำดับความสำคัญพื้นฐานที่จัดตารางเวลาได้ โดยใช้คำสั่ง `renice` จากบรรทัดคำสั่ง คำสั่งนี้เปลี่ยนค่าที่ละเอียดของโปรเซส

หมายเหตุ: เมื่อต้องการรันโปรเซสที่ลำดับความสำคัญสูงขึ้น หรือเมื่อต้องการเปลี่ยน ลำดับความสำคัญของโปรเซสที่คุณไม่ได้เริ่มต้น คุณต้องมีสิทธิ์ผู้ใช้ราก

เมื่อต้องการเปลี่ยนลำดับความสำคัญของโปรเซสที่กำลังรัน ให้พิมพ์ดังต่อไปนี้:

```
renice Priority -p ProcessID
```

โดยที่ *Priority* คือตัวเลข ในช่วงตั้งแต่ -20 ถึง 20 ตัวเลขยิ่งสูง ลำดับความสำคัญยิ่งต่ำ ถ้าคุณใช้ศูนย์ โปรเซสจะรันที่ลำดับ ความสำคัญพื้นฐานที่จัดตารางเวลาของโปรเซส *ProcessID* คือ PID ซึ่งคุณต้องการเปลี่ยนลำดับความสำคัญ

คุณยังสามารถใช้คำสั่ง `smit renice` เพื่อทำภารกิจนี้ได้ด้วย

การยกเลิกกระบวนการพื้นหน้า:

หากคุณเริ่มต้นกระบวนการพื้นหน้าแล้วตัดสินใจว่าคุณไม่ต้องการให้กระบวนการนั้นเสร็จสิ้น คุณสามารถยกเลิกได้โดยกด INTERRUPT ซึ่งโดยปกติเป็น Ctrl-C หรือ Ctrl-Backspace

หมายเหตุ: INTERRUPT (Ctrl-C) ไม่ได้ยกเลิกกระบวนการส่วนหลัง เมื่อต้องการยกเลิกกระบวนการส่วนหลัง คุณต้องใช้คำสั่ง kill

คำสั่งแบบง่ายส่วนใหญ่รันอย่างรวดเร็วและเสร็จสิ้นก่อนคุณมีเวลาที่จะยกเลิก ดังนั้น ตัวอย่างในส่วนนี้จึงใช้คำสั่งที่ใช้เวลาในการรันนานกว่าสองสามวินาที: `find / -type f` คำสั่งนี้แสดงชื่อพาธสำหรับไฟล์ทั้งหมดบนระบบของคุณ คุณไม่จำเป็นต้องศึกษาคำสั่ง `find` เพื่อทำส่วนนี้ให้สมบูรณ์ เนื่องจากมีการใช้คำสั่งเพื่อสาธิตวิธีการทำงานกับโปรเซสเท่านั้น

ในตัวอย่างต่อไปนี้ คำสั่ง `find` จะเริ่มต้นโปรเซส หลังจากโปรเซสรันนานสองสามวินาที คุณสามารถยกเลิกโปรเซสได้โดยกดคีย์ INTERRUPT:

```
$ find / -type f
/usr/sbin/acct/lastlogin
/usr/sbin/acct/prctmp
/usr/sbin/acct/prdaily
/usr/sbin/acct/runacct
/usr/sbin/acct/sdisk
/usr/sbin/acct/shutacct INTERRUPT (Ctrl-C)
$ _
```

ระบบกลับไปยังพร้อมต้นหน้าจอ ขณะนี้ คุณสามารถป้อนคำสั่งอื่นได้

งานที่เกี่ยวข้อง:

“รายการของการกำหนดคีย์ควบคุมสำหรับเทอร์มินัลของคุณ (คำสั่ง `stty`)” ในหน้า 341

เมื่อต้องการแสดงค่าติดตั้งเทอร์มินัล ให้ใช้คำสั่ง `stty` โปรดสังเกตเป็นพิเศษว่าเทอร์มินัลใช้คีย์ใดเป็นคีย์ควบคุม

คำสั่งคีย์บอร์ดเพื่อหยุดกระบวนการพื้นหน้า:

สามารถหยุดโปรเซสได้ แต่ไม่สามารถลบ ID โปรเซส (PID) ออกจากตารางโปรเซส คุณสามารถหยุดกระบวนการพื้นหน้าโดยกด Ctrl-Z จากคีย์บอร์ด

หมายเหตุ: Ctrl-Z ทำงานใน Korn เชลล์ (`ksh`) และ C เชลล์ (`csch`) แต่ไม่ทำงานในเชลล์เป้าหมาย (`bsh`)

การรีสตาร์ทโปรเซสที่หยุด:

โปรซีเดอร์นี้อธิบายวิธีการรีสตาร์ทโปรเซสที่ถูกหยุด ด้วย Ctrl-Z

หมายเหตุ: Ctrl-Z ทำงานใน Korn เชลล์ (`ksh`) และ C เชลล์ (`csch`) แต่ไม่ทำงานในเชลล์เป้าหมาย (`bsh`) เพื่อรีสตาร์ทโปรเซสที่หยุด คุณต้องเป็นผู้ใช้ที่เริ่มต้น โปรเซสหรือมีสิทธิ์ผู้ใช้ราก

1. เมื่อต้องการแสดงโปรเซสทั้งหมดที่กำลังรันหรือหยุด แต่ไม่ใช่โปรเซสที่ถูก ทำลายบนระบบของคุณ ให้พิมพ์ดังต่อไปนี้:

```
ps -ef
```

คุณอาจต้องการโพ้คำสั่งนี้โดยใช้คำสั่ง `grep` เพื่อจำกัดให้แสดงรายการเฉพาะโปรเซสที่น่าจะเป็นโปรเซสซึ่งคุณต้องการรีสตาร์ท ตัวอย่างเช่น ถ้าคุณต้องการรีสตาร์ทเชสชัน `vi` คุณสามารถพิมพ์ดังต่อไปนี้:

```
ps -ef | grep vi
```

คำสั่งนี้จะแสดงเฉพาะบรรทัดจากเอาต์พุตคำสั่ง `ps` ที่มีคำว่า `vi` เท่านั้น เอาต์พุตจะมีลักษณะคล้าย ดังนี้:

```
UID    PID  PPID  C    STIME     TTY  TIME  COMMAND
root   1234  13682  0    00:59:53  -    0:01  vi test
root   14277 13682  1    01:00:34  -    0:00  grep vi
```

2. ในเอาต์พุตคำสั่ง `ps` ให้ค้นหาโปรเซสซึ่งคุณต้องการ รีเซ็ตและจดบันทึกหมายเลข PID ของโปรเซสนั้น ในตัวอย่าง PID คือ 1234

3. เมื่อต้องการส่งสัญญาณ CONTINUE ไปยังโปรเซสที่หยุด ให้พิมพ์ดังต่อไปนี้:

```
kill -19 1234
```

ทดแทน PID ของโปรเซสของคุณด้วย 1234 - 19 บ่งชี้ สัญญาณ CONTINUE คำสั่งนี้รีเซ็ตโปรเซสในพื้นหลัง หากโปรเซสสามารถรันในพื้นหลังได้ ถือว่าคุณทำโพริเตอร์เสร็จสิ้นแล้ว หากโปรเซสต้องรันในพื้นหน้า (อาจเป็นเชสชัน `vi`) คุณต้องทำขั้นตอนถัดไป

4. เมื่อต้องการนำโปรเซสไปไว้ในพื้นหน้า ให้พิมพ์ดังต่อไปนี้:

```
fg 1234
```

ทดแทน PID ของโปรเซสของคุณด้วย 1234 อีกครั้ง ขณะนี้ โปรเซสของคุณควรจะ กำลังรันในพื้นหน้า (ขณะนี้ คุณอยู่ในเชสชันแก้ไข `vi`)

การจัดตารางเวลาโปรเซสสำหรับการดำเนินงานในภายหลัง:

คุณสามารถตั้งค่าโปรเซสเป็น *แบตช์โปรเซส* เพื่อรันใน พื้นหลังในเวลาที่จะจัดตารางเวลาไว้

คำสั่ง `at` และ `smit` อนุญาตให้คุณป้อนชื่อของคำสั่งที่จะรันในภายหลัง และช่วยให้คุณระบุเวลาที่ควรจะรันคำสั่งได้

หมายเหตุ: ไฟล์ `/var/adm/cron/at.allow` และ `/var/adm/cron/at.deny` ควบคุมว่าคุณสามารถใช้คำสั่ง `at` ได้หรือไม่ บุคคลที่มีสิทธิ์ผู้ใช้ระดับรากสามารถ สร้าง แก้ไข หรือลบไฟล์เหล่านี้ รายการในไฟล์เหล่านี้คือชื่อล็อกอินผู้ใช้ที่มีหนึ่งชื่อ ต่อหนึ่งบรรทัด ต่อไปนี้คือตัวอย่างของไฟล์ `at.allow`:

```
root
nick
dee
sarah
```

ถ้ามีไฟล์ `at.allow` อยู่ เฉพาะผู้ใช้ที่มีชื่อล็อกอินปรากฏในไฟล์เท่านั้นสามารถใช้คำสั่ง `at` ได้ ผู้ดูแลระบบสามารถหยุดมิให้ผู้ใช้ใช้คำสั่ง `at` โดยการแสดงชื่อล็อกอินของผู้ใช้ในไฟล์ `at.deny` หากมีอยู่เฉพาะไฟล์ `at.deny` ผู้ใช้ใดๆ ที่ไม่มีชื่อปรากฏในไฟล์สามารถใช้คำสั่ง `at`

คุณ ไม่สามารถใช้คำสั่ง `at` ถ้าเงื่อนไขอย่างใดอย่างหนึ่งต่อไปนี้ เป็นจริง:

- ไฟล์ `at.allow` และไฟล์ `at.deny` ไม่มีอยู่ (อนุญาตผู้้รากเท่านั้น)
- ไฟล์ `at.allow` มีอยู่แต่ชื่อล็อกอินของผู้ใช้ ไม่แสดงอยู่ในไฟล์
- ไฟล์ `at.deny` มีอยู่และชื่อล็อกอินของผู้ใช้ มีแสดงอยู่ในไฟล์

หากไฟล์ `at.allow` ไม่มีอยู่ และไฟล์ `at.deny` ไม่มีอยู่หรือว่าง ผู้ใช้ที่มีสิทธิ์รากเท่านั้นที่สามารถส่ง งานด้วยคำสั่ง `at`

ไวยากรณ์คำสั่ง `at` อนุญาตให้คุณระบุสตริงวันที่ สตริงเวลาและวัน หรือสตริงส่วนเพิ่ม สำหรับเวลาซึ่งคุณต้องการให้โปรเซสรัน คุณยังสามารถระบุ เซลล์หรือคิวที่จะใช้ได้ด้วย ตัวอย่างต่อไปนี้แสดงการใช้งานทั่วไปบางอย่าง ของคำสั่ง

ตัวอย่างเช่น ถ้าชื่อล็อกอินของคุณคือ `joyce` และ คุณมีสคริปต์ชื่อว่า `WorkReport` ซึ่งคุณต้องการรันในตอนเที่ยงคืน ให้ทำดังต่อไปนี้:

1. พิมพ์เวลาซึ่งคุณต้องการให้โปรแกรมเริ่มต้นการรัน:

```
at midnight
```

2. พิมพ์ชื่อของโปรแกรมที่จะรัน โดยกด `Enter` หลังจากแต่ละชื่อ After typing the surname, press the end-of-file character (`Ctrl-D`) to signal the end of the list.

```
WorkReport^D
```

หลังจาก คุณกด `Ctrl-D` ระบบแสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
job joyce.741502800.a at Fri Jul 6 00:00:00 CDT 2002.
```

โปรแกรม `WorkReport` ได้รับการกำหนดงานหมายเลข `joyce.741502800.a` และจะรันตอนเที่ยงคืนวันที่ 6 กรกฎาคม

3. เมื่อต้องการแสดงรายการโปรแกรมซึ่งคุณได้ส่งเพื่อรันในภายหลัง ให้พิมพ์ดังต่อไปนี้:

```
at -l
```

ระบบแสดงข้อมูลคล้ายกับตัวอย่าง ต่อไปนี้:

```
joyce.741502800.a Fri Jul 6 00:00:00 CDT 2002
```

ให้ดูที่คำสั่ง `at` สำหรับไวยากรณ์ทั้งหมด

งานที่เกี่ยวข้อง:

“การแสดงผลการโปรเซสที่จัดตารางเวลาทั้งหมด (คำสั่ง `at` หรือ `atq`)”

ใช้แฟล็ก `-l` พร้อมกับคำสั่ง `at` หรือ พร้อมกับคำสั่ง `atq` เพื่อแสดงผลการโปรเซสที่จัดตารางเวลาไว้ทั้งหมด

“การลบโปรเซสออกจากรายการเวลา” ในหน้า 161

คุณสามารถลบโปรเซสที่จัดตารางเวลาโดยใช้คำสั่ง `at` พร้อมกับแฟล็ก `-r`

การแสดงผลการโปรเซสที่จัดตารางเวลาทั้งหมด (คำสั่ง `at` หรือ `atq`):

ใช้แฟล็ก `-l` พร้อมกับคำสั่ง `at` หรือ พร้อมกับคำสั่ง `atq` เพื่อแสดงผลการโปรเซสที่จัดตารางเวลาไว้ทั้งหมด

ทั้งสองคำสั่งให้เอาต์พุตเหมือนกัน อย่างไรก็ตาม คำสั่ง `atq` ยังสามารถจัดลำดับโปรเซสโดยใช้เวลาเท่ากับการออกใช้คำสั่ง `at` และแสดงเฉพาะจำนวนโปรเซสในคิวเท่านั้น

คุณสามารถ แสดงรายการโปรเซสที่จัดตารางเวลาไว้ทั้งหมดในวิธีต่อไปนี้:

- โดยใช้คำสั่ง `at` จากบรรทัดคำสั่ง
- โดยใช้คำสั่ง `atq`

คำสั่ง `at`

เมื่อต้องการแสดงผลการโปรเซสที่จัดตารางเวลา ให้พิมพ์ดังต่อไปนี้:

at -l

คำสั่งนี้แสดงรายการโปรเซสที่จัดตารางเวลาทั้งหมดในคิวของคุณ หากคุณเป็นผู้ใช้ราก คำสั่งนี้จะแสดงรายการโปรเซสที่จัดตารางเวลาทั้งหมดสำหรับผู้ใช้ทุกราช หากต้องการรายละเอียดทั้งหมดของไวยากรณ์ ให้ดูที่คำสั่ง **at**

คำสั่ง **atq**

ดูตัวอย่างวิธีการใช้คำสั่ง **atq** ต่อไปนี้:

- เมื่อต้องการแสดงรายการโปรเซสที่จัดตารางเวลาทั้งหมดในคิว ให้พิมพ์ดังต่อไปนี้:

```
atq
```

- หากคุณเป็นผู้ใช้ราก คุณสามารถแสดงรายการโปรเซสที่จัดตารางเวลาในคิวของผู้ใช้เฉพาะได้โดยพิมพ์:

```
atq UserName
```

- เมื่อต้องการแสดงรายการจำนวนของโปรเซสที่จัดตารางเวลาในคิว ให้พิมพ์ดังต่อไปนี้:

```
atq -n
```

งานที่เกี่ยวข้อง:

“การจัดตารางเวลาโปรเซสสำหรับการดำเนินงานในภายหลัง” ในหน้า 159

คุณสามารถตั้งค่าโปรเซสเป็น *แบตช์โปรเซส* เพื่อรันใน พื้นหลังในเวลาที่จะจัดตารางเวลาไว้

“การลบโปรเซสออกจากตารางเวลา”

คุณสามารถลบโปรเซสที่จัดตารางเวลาโดยใช้คำสั่ง **at** พร้อมกับแฟล็ก **-r**

การลบโปรเซสออกจากตารางเวลา:

คุณสามารถลบโปรเซสที่จัดตารางเวลาโดยใช้คำสั่ง **at** พร้อมกับแฟล็ก **-r**

ดูตัวอย่างต่อไปนี้เกี่ยวกับวิธีการใช้คำสั่ง **at** หรือ **atq**:

1. เพื่อลบโปรเซสที่จัดตารางเวลา คุณต้องทราบหมายเลขโปรเซส คุณสามารถดูหมายเลขโปรเซสได้โดยใช้คำสั่ง **at -l** หรือคำสั่ง **atq**
2. เมื่อคุณทราบหมายเลขของโปรเซสซึ่งคุณต้องการลบ ให้พิมพ์ ดังต่อไปนี้:

```
at -r ProcessNumber
```

คุณยังสามารถใช้คำสั่ง **smit rmat** เพื่อทำภารกิจนี้ได้ด้วย

งานที่เกี่ยวข้อง:

“การแสดงผลรายการโปรเซสที่จัดตารางเวลาทั้งหมด (คำสั่ง **at** หรือ **atq**)” ในหน้า 160

ใช้แฟล็ก **-l** พร้อมกับคำสั่ง **at** หรือ พร้อมกับคำสั่ง **atq** เพื่อแสดงผลรายการโปรเซสที่จัดตารางเวลาไว้ทั้งหมด

“การจัดตารางเวลาโปรเซสสำหรับการดำเนินงานในภายหลัง” ในหน้า 159

คุณสามารถตั้งค่าโปรเซสเป็น *แบตช์โปรเซส* เพื่อรันใน พื้นหลังในเวลาที่จะจัดตารางเวลาไว้

การลบกระบวนการส่วนหลัง (คำสั่ง **kill**):

ถ้า **INTERRUPT** ไม่ได้หยุดกระบวนการพื้นหน้า หรือถ้าคุณตัดสินใจ หลังจากเริ่มต้นกระบวนการส่วนหลังว่าคุณไม่ต้องการให้โปรเซสเสร็จสิ้น คุณสามารถยกเลิกโปรเซสได้โดยใช้คำสั่ง **kill**

ก่อนคุณสามารถยกเลิกโปรเซสโดยใช้คำสั่ง `kill` คุณต้องทราบหมายเลข PID ของโปรเซสนั้น รูปแบบทั่วไปของคำสั่ง `kill` เป็นดังนี้:

```
kill ProcessID
```

หมายเหตุ:

- เพื่อย้ายโปรเซส คุณต้องมีสิทธิ์ผู้ใช้รากหรือเป็นผู้ใช้ที่ เริ่มต้นโปรเซส สัญญาณดีฟอลต์ไปยังโปรเซสจากคำสั่ง `kill` คือ -15 (SIGTERM)
 - เมื่อต้องการลบกระบวนการ zombie คุณต้องลบพาเรนต์โปรเซส
1. ใช้คำสั่ง `ps` เพื่อกำหนด ID โปรเซสของ โปรเซสซึ่งคุณต้องการลบ คุณอาจต้องการพิมพ์คำสั่งนี้โดยใช้คำสั่ง `grep` เพื่อแสดงรายการเฉพาะโปรเซสซึ่งคุณต้องการ ตัวอย่างเช่น ถ้าคุณต้องการ ID โปรเซสของ `vi` เซสชัน คุณสามารถพิมพ์ดังต่อไปนี้:

```
ps -l | grep vi
```

2. ในตัวอย่างต่อไปนี้ คุณออกใช้คำสั่ง `find` เพื่อรันในพื้นที่หลังจากนั้น คุณตัดสินใจที่จะยกเลิกโปรเซส ออกใช้คำสั่ง `ps` เพื่อแสดงรายการหมายเลข PID

```
$ find / -type f > dir.paths &
```

```
[1] 21593
```

```
$ ps
```

PID	TTY	TIME	COMMAND
1627	pts3	0:00	ps
5461	pts3	0:00	ksh
17565	pts3	0:00	-ksh
21593	pts3	0:00	find / -type f

```
$ kill 21593
```

```
$ ps
```

PID	TTY	TIME	COMMAND
1627	pts3	0:00	ps
5461	pts3	0:00	ksh
17565	pts3	0:00	-ksh

```
[1] + Terminated 21593 find / -type f > dir.paths &
```

คำสั่ง `kill 21593` สิ้นสุดโปรเซส `find` ในพื้นที่ และคำสั่ง `ps` ที่สองไม่ส่งคืนข้อมูลสถานะเกี่ยวกับ PID 21593 ระบบไม่แสดงข้อความการยุติจนกว่า คุณป้อนคำสั่งถัดไป ยกเว้นว่าคำสั่งนั้นเป็น `cd`

คำสั่ง `kill` ช่วยให้คุณสามารถยกเลิกกระบวนการส่วนหลังได้ คุณอาจต้องการทำเช่นนี้ถ้าคุณรู้ว่า คุณวางโปรเซสไว้ในพื้นที่โดยผิดพลาด หรือโปรเซสกำลังใช้เวลารันนานเกินไป

ให้ดูที่คำสั่ง `kill` ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

นอกจากนี้ยังสามารถใช้คำสั่ง `kill` ใน `smit` โดยพิมพ์:

```
smit kill
```

ข้อมูลสรุปคำสั่งสำหรับคำสั่งและโปรเซส

ข้อมูลต่อไปนี้เป็นคำสั่งสำหรับคำสั่งและโปรเซส

ตารางที่ 55. ข้อมูลสรุปคำสั่งสำหรับคำสั่ง

ไอเท็ม	คำอธิบาย
alias	คำสั่งเซลล์ที่พิมพ์รายการของสมนามไปยังเอาต์พุตมาตรฐาน
history	คำสั่งเซลล์ที่แสดงรายการเหตุการณ์ประวัติ
man	แสดงข้อมูลเกี่ยวกับคำสั่ง รูทีนย่อย และ ไฟล์ออนไลน์
whatis	อธิบายฟังก์ชันที่คำสั่งทำ
whereis	ระบุตำแหน่งแหล่งที่มา ไบนารี หรือคู่มือสำหรับโปรแกรม ที่ติดตั้ง

ตารางที่ 56. ข้อมูลสรุปคำสั่งสำหรับโปรเซส

ไอเท็ม	คำอธิบาย
at	รันคำสั่งในภายหลัง แสดงรายการโปรเซสที่จัดตารางเวลาทั้งหมด หรือลบโปรเซสออกจากตารางเวลา
atq	แสดงคิวของงานที่กำลังรอการรัน
kill	ส่งสัญญาณไปยังโปรเซสที่กำลังรัน
nice	รันคำสั่งที่ลำดับความสำคัญต่ำกว่าหรือสูงกว่า
ps	แสดงสถานะปัจจุบันของโปรเซส
renice	เปลี่ยนลำดับความสำคัญของโปรเซสที่กำลังรัน

การจัดการ hang ระบบ

การจัดการ hang ระบบช่วยให้ผู้ใช้สามารถรันแอปพลิเคชันที่มีเป้าหมายสำคัญมากได้อย่างต่อเนื่องในขณะที่ปรับปรุงการมีอยู่ของแอปพลิเคชัน การตรวจพบ hang ระบบ จะแจ้งเตือนผู้ดูแลระบบถึงปัญหาที่เป็นไปได้ จากนั้นอนุญาตให้ผู้ดูแลระบบล็อกอินเป็นราก หรือรีบูตระบบเพื่อแก้ไขปัญหา

คำสั่ง shconf

คำสั่ง shconf มีการเรียกใช้เมื่อเปิดใช้งาน การตรวจพบ Hang ระบบ คำสั่ง shconf กำหนดคอนฟิก เหตุการณ์ที่จะสำรวจและการดำเนินการที่จะใช้ถ้าเหตุการณ์นั้น เกิดขึ้น คุณสามารถระบุการดำเนินการใดๆ ต่อไปนี้ ลำดับความสำคัญที่จะ ตรวจสอบไทม์เอาต์ในขณะที่ไม่มีโปรเซสหรือเธรดดำเนินการที่ลำดับความสำคัญ ต่ำกว่าหรือเท่ากัน อุปกรณ์เทอร์มินัลสำหรับการดำเนินการเตือน และการดำเนินการคำสั่ง getty :

- บันทึกข้อผิดพลาดในไฟล์ errlog
- แสดงข้อความค่าเตือนบนคอนโซลระบบ (คอนโซลตัวอักษรและตัวเลข) หรือบน TTY ที่ระบุ
- รีบูตระบบ
- ให้ getty พิเศษเพื่อให้ผู้ใช้สามารถล็อกอินเป็นราก และออกใช้คำสั่งได้
- ออกใช้คำสั่ง

สำหรับอ็อปชัน ออกใช้คำสั่ง และ ให้ getty พิเศษ การตรวจพบ hang ระบบจะออกใช้คำสั่ง getty พิเศษ หรือคำสั่งที่ระบุที่ลำดับความสำคัญสูงสุด คำสั่ง getty พิเศษ พิมพ์ข้อความค่าเตือนว่าเป็น getty ภูคินซึ่งกำลังรัน ที่ลำดับความสำคัญ 0 ตารางต่อไปนี้จะแสดงการดำเนินการต่างๆ และพารามิเตอร์ ดีฟอลต์ที่เชื่อมโยงสำหรับการตรวจพบ hang ลำดับความสำคัญ มีการเปิดใช้งานเพียงหนึ่งการดำเนินการ เท่านั้นสำหรับการตรวจพบแต่ละชนิด

อ็พชั่น	การเปิดใช้งาน	ลำดับความสำคัญ	ไทม์เอาต์ (วินาที)
บันทึกข้อผิดพลาดในไฟล์errlog	ปิดใช้งาน	60	120
แสดงข้อความค่าเตือน	ปิดใช้งาน	60	120
ให้getty ที่กึ่งคืน	เปิดใช้งาน	60	120
ออกใช้คำสั่ง	ปิดใช้งาน	60	120
รีบูตระบบ	ปิดใช้งาน	39	300

หมายเหตุ: เมื่อเปิดใช้งาน ออกใช้getty ที่กึ่งคืนบนคอนโซล คำสั่งshconfจะเพิ่มแฟล็ก -u ลงในคำสั่งgetty ในinittabซึ่งเชื่อมโยงกับคอนโซลล็อกอิน

สำหรับการตรวจพบ IO ที่หายไป คุณสามารถตั้งค่าไทม์เอาต์และเปิดใช้งาน การดำเนินการต่อไปนี้:

อ็พชั่น	การเปิดใช้งาน
แสดงข้อความค่าเตือน	ปิดใช้งาน
รีบูตระบบ	ปิดใช้งาน

shdaemon daemon

shdaemon daemon คือโปรเซสที่ออกใช้โดยinit และรันที่ลำดับความสำคัญ 0 (ศูนย์) โปรเซสนี้รับผิดชอบในการจัดการการตรวจพบhang ระบบ โดยการดึงข้อมูลคอนฟิกูเรชัน เริ่มต้นโครงสร้างการทำงาน และเริ่มต้นเวลาการตรวจพบที่ตั้งค่าโดยผู้ใช้ หลักการที่เกี่ยวข้อง:

“การตรวจพบhang ลำดับความสำคัญ” ในหน้า 165

AIX สามารถตรวจสอบเงื่อนไขhang ระบบและพยายามกู้คืนจากสถานการณ์ดังกล่าว ตามข้อมูลการดำเนินการที่ผู้ใช้กำหนด

“การตรวจพบ I/O hang ที่หายไป” ในหน้า 166

AIX สามารถตรวจพบสภาพhang ระบบและพยายามแก้ไขจากสถานการณ์ดังกล่าว ตามการดำเนินการที่ผู้ใช้กำหนด

การกำหนดคอนฟิกการตรวจพบhang ระบบ

คุณสามารถจัดการคอนฟิกูเรชันการตรวจพบhang ระบบจาก เครื่องมือการจัดการ SMIT

อ็พชั่นเมนู SMIT ช่วยให้คุณสามารถเปิดใช้งานหรือปิดใช้งานกลไกการตรวจพบ แสดงสถานะปัจจุบันของคุณลักษณะ และเปลี่ยนหรือแสดงคอนฟิกูเรชันปัจจุบัน พาด่วนสำหรับเมนูการตรวจพบhang ระบบคือ:

smit shd

จัดการกับการตรวจพบ Hang ระบบ

smit shstatus

สถานะการตรวจพบ Hang ระบบ

smit shprioCfg

เปลี่ยน/แสดงลักษณะของการตรวจพบปัญหาลำดับความสำคัญ

smit shreset

เรียกคืนคอนฟิกูเรชันปัญหาลำดับความสำคัญดีพอลต์

smit shliocfg

เปลี่ยน/แสดงลักษณะของการตรวจพบ IO ที่หายไป

smit shlioreset

เรียกคืนคอนฟิกูเรชันการตรวจพบ I/O ที่หายไปดีพอลต์

คุณยังสามารถจัดการกับการตรวจพบ hang ระบบโดยใช้คำสั่ง `shconf` ได้ด้วย

การตรวจพบ hang ลำดับความสำคัญ

AIX สามารถตรวจสอบเงื่อนไข hang ระบบและพยายามกู้คืนจากสถานการณ์ดังกล่าว ตามข้อมูลการดำเนินการที่ผู้กำหนด

โปรเซสทั้งหมด (รู้จักในอีกชื่อว่าเซรต) รันตามลำดับความสำคัญ ลำดับความสำคัญนี้เป็นตัวเลขผกผันในช่วงตั้งแต่ 0-126 ศูนย์คือลำดับความสำคัญสูงสุดและ 126 คือลำดับความสำคัญต่ำสุด ลำดับความสำคัญดีพอลต์สำหรับเซรตทั้งหมดคือ 60 ผู้ใช้ใดๆ สามารถลด ลำดับความสำคัญของโปรเซสได้โดยใช้คำสั่ง `nice` บุคคลใดก็ตามที่มีสิทธิ์การใช้งานราก ยังสามารถเพิ่ม ลำดับความสำคัญของโปรเซสได้ด้วย

ตัวจัดตารางเวลาเคอร์เนลเลือกเซรตที่รันได้และมีลำดับความสำคัญสูงสุดเสมอ เพื่อวาง CPU ดังนั้นจึงมีความเป็นไปได้ที่เซรตลำดับความสำคัญสูงจะใช้รีซอร์สทั้งหมดของเครื่องจนเซรตลำดับความสำคัญต่ำ ไม่มีโอกาสได้รับ หากเซรตที่กำลังรันมีลำดับความสำคัญสูงกว่าดีพอลต์ 60 เซรตเหล่านี้สามารถล็อกเอาต์เซลล์ปกติทั้งหมดและล็อกอินจนถึงจุดที่ทำให้ระบบ hung

คุณลักษณะ การตรวจพบ Hang ระบบ นำเสนอกลไกการตรวจพบสถานการณ์นี้ และช่วยให้ผู้ดูแลระบบสามารถกู้คืนได้ คุณลักษณะนี้มีการนำไปใช้ เป็น daemon (`shdaemon`) ซึ่งจะรันบนโปรเซสที่มีระดับความสำคัญสูงสุด Daemon นี้จะเคียวรีเคอร์เนลเพื่อหาเซรตลำดับความสำคัญต่ำสุดที่รัน ในเวลาที่ระบุ หากลำดับความสำคัญสูงกว่า threshold ที่กำหนดคอนฟิกไว้ daemon สามารถดำเนินการอย่างใดอย่างหนึ่ง แต่ละการดำเนินการเหล่านี้สามารถเปิดใช้งาน อย่างเป็นอิสระจากกัน และสามารถกำหนดคอนฟิกแต่ละการดำเนินการเพื่อทริกเกอร์ที่ลำดับความสำคัญใดๆ และในช่วงเวลาใดก็ได้ การดำเนินการและค่าดีพอลต์เป็นดังนี้:

Action	Default Enabled	Default Priority	Default Timeout	Default Device
1) Log an error	no	60	2	
2) Console message	no	60	2	/dev/console
3) High priority login shell	yes	60	2	/dev/tty0
4) Run a command at high priority	no	60	2	
5) Crash and reboot	no	39	5	

หลักการที่เกี่ยวข้อง:

“การจัดการ hang ระบบ” ในหน้า 163

การจัดการ hang ระบบช่วยให้ผู้ใช้สามารถรันแอปพลิเคชันที่มีเป้าหมายสำคัญมากได้อย่างต่อเนื่องในขณะที่ปรับปรุงการมีอยู่ของแอปพลิเคชัน การตรวจพบ hang ระบบ จะแจ้งเตือนผู้ดูแลระบบถึงปัญหาที่เป็นไปได้ จากนั้นอนุญาตให้ผู้ดูแลระบบล็อกอินเป็นราก หรือรีบูตระบบเพื่อแก้ไขปัญหา

การตรวจพบ I/O hang ที่หายไป

AIX สามารถตรวจพบสภาพ hang ระบบและพยายามแก้ไขจากสถานการณ์ดังกล่าว ตามการดำเนินการที่ผู้ใช้กำหนด

เนื่องจากข้อผิดพลาด I/O พาร I/O จึงอาจถูกบล็อกและส่งผลกระทบต่อ I/O เพิ่มเติมบนพาร์นั้น ในสถานการณ์เหล่านี้จำเป็นที่ระบบปฏิบัติการ ต้องแจ้งเตือนผู้ใช้และทำการดำเนินการที่ผู้ใช้กำหนด ในส่วนหนึ่งของกระบวนการตรวจพบ I/O ที่หายไปและการแจ้งเตือน shdaemon ที่ได้รับความช่วยเหลือจาก Logical Volume Manager จะมอนิเตอร์บัพเพอร์ I/O ในรอบระยะเวลาหนึ่งและตรวจสอบว่ามี I/O ไต่ค้างอยู่นานเกินไป หรือไม่ หากเวลารอเกินกว่าเวลารอ threshold ที่กำหนดโดยไฟล์ shconf จะตรวจพบ I/O ที่หายไปและทำการดำเนินการเพิ่มเติม ข้อมูลเกี่ยวกับ I/O ที่สูญหายมีการบันทึกไว้ในบันทึกข้อผิดพลาด นอกจากนี้ตามข้อมูลค่าติดตั้งใน ไฟล์ shconf ระบบอาจถูกรีบูตเพื่อกู้คืนจาก สถานการณ์ I/O ที่หายไป

สำหรับการตรวจพบ I/O ที่หายไป คุณสามารถตั้งค่าใหม่เอาต์และเปิดใช้งาน การดำเนินการต่อไปนี้:

การดำเนินการ	เปิดใช้งานดีฟอลต์	อุปกรณ์ดีฟอลต์
ข้อความคอนโซล	ไม่	/dev/console
ความเสียหายและรีบูต	ไม่	-

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการตรวจพบ hang ระบบ ให้ดูที่ “การจัดการ hang ระบบ” ในหน้า 163

หลักการที่เกี่ยวข้อง:

“การจัดการ hang ระบบ” ในหน้า 163

การจัดการ hang ระบบช่วยให้ผู้ใช้สามารถรันแอ็พพลิเคชันที่มีเป้าหมายสำคัญมากได้อย่างต่อเนื่องในขณะที่ปรับปรุงการมีอยู่ของแอ็พพลิเคชัน การตรวจพบ hang ระบบ จะแจ้งเตือนผู้ดูแลระบบถึงปัญหาที่เป็นไปได้ จากนั้นอนุญาตให้ผู้ดูแลระบบ ล็อกอินเป็นราก หรือรีบูตระบบเพื่อแก้ไขปัญหา

การจัดการโปรเซส

โปรเซสคือเอนทิตีที่ระบบปฏิบัติการใช้เพื่อควบคุม การใช้รีซอร์สระบบ เทรด สามารถควบคุมการใช้เวลาตัวประมวลผล แต่เครื่องมือการจัดการระบบส่วนใหญ่ยังต้องการให้คุณอ้างอิงโปรเซส ซึ่งเทรดกำลังรันอยู่ แทนการอ้างอิงตัวเทรดเอง

เครื่องมือมีอยู่สำหรับ:

- สังเกตการสร้าง การยกเลิก เอกลักษณะ และการใช้รีซอร์สของโปรเซส
 - คำสั่ง ps ใช้เพื่อรายงาน IDs โปรเซส, ผู้ใช้, การใช้เวลา CPU, และแอ็ททริบิวต์อื่น
 - คำสั่ง who -u รายงาน ID โปรเซสเซลล์ของผู้ใช้ที่ล็อกออน
 - คำสั่ง svmon ใช้เพื่อรายงานปริมาณการใช้ หน่วยความจำแบบจริงของโปรเซส (โปรดดูที่ *Performance Toolbox Version 3: Guide and Reference* สำหรับ ข้อมูลเกี่ยวกับคำสั่ง svmon)
 - กลไกคำสั่ง acct เขียนเร็กคอร์ดเมื่อโปรเซสยุติลง โดยสรุปการใช้รีซอร์สของโปรเซส
- ควบคุมลำดับความสำคัญซึ่งโปรเซสช่วงชิง CPU
 - คำสั่ง nice ทำให้อันดับความสำคัญโปรเซสที่ระบุ
 - คำสั่ง renice เปลี่ยนลำดับความสำคัญ ของโปรเซสที่กำหนด
- ยุติโปรเซสที่อยู่นอกเหนือการควบคุม
 - คำสั่ง kill ส่งสัญญาณการยุติไปยังหนึ่งโปรเซสขึ้นไป

หลักการที่เกี่ยวข้อง:

“การทำแอดเคาต์ระบบ” ในหน้า 174

ยูนิตีการทำการแอดเคาต์ระบบอนุญาตให้คุณรวบรวมและรายงาน ในการใช้งานแต่ละคน หรือกลุ่มของรีซอร์สระบบต่างๆ

การมอนิเตอร์โปรเซส

คุณในฐานะผู้ดูแลระบบ สามารถจัดการกับโปรเซสได้

คำสั่ง ps เป็นเครื่องมือหลักสำหรับการสังเกต โปรเซสในระบบ แฟล็กส่วนใหญ่ของคำสั่ง ps อยู่ในหมวดหมู่ อย่างใดอย่างหนึ่ง จากสองหมวดหมู่นี้:

- แฟล็กที่ระบุชนิดของโปรเซสที่จะรวมไว้ในเอาต์พุต
- แฟล็กที่ระบุแอตทริบิวต์ที่จะแสดงขึ้นของโปรเซสดังกล่าว

ความแตกต่างที่มีประโยชน์มากที่สุดของ ps สำหรับวัตถุประสงค์ การจัดการระบบคือ:

ไอเท็ม

ps -ef

คำอธิบาย

แสดงรายการโปรเซสที่ไม่ใช่เคอร์เนลทั้งหมด พร้อมกับ ID ผู้ใช้, ID โปรเซส, การใช้ CPU ล่าสุด, การใช้ CPU ทั้งหมด, และคำสั่งที่เริ่มต้นโปรเซส (รวมถึง พารามิเตอร์ของคำสั่งนั้น)

ps -fu UserID

แสดงรายการโปรเซสทั้งหมดที่เป็นของ UserID พร้อมกับ ID โปรเซส, การใช้ CPU ล่าสุด, การใช้ CPU ทั้งหมด, และคำสั่งที่เริ่มต้น โปรเซส (รวมถึงพารามิเตอร์ของคำสั่งนั้น)

เมื่อต้องการระบุผู้ใช้ที่ใช้เวลา CPU สูงสุดในปัจจุบัน คุณควรป้อน:

```
ps -ef | egrep -v "STIME|$LOGNAME" | sort +3 -r | head -n 15
```

คำสั่งนี้แสดงรายการโปรเซสที่ใช้ CPU สูงสุด 15 โปรเซสซึ่งไม่ใช่ของคุณ ในลำดับจากมากไปน้อย

สำหรับการใช้งานเฉพาะด้านมากขึ้น มีการจัดทำสองตารางต่อไปนี้เพื่อช่วยให้ ภารกิจการเลือกแฟล็ก ps ง่ายขึ้นโดยการสรุป ผลกระทบ ของแฟล็กต่างๆ

แฟล็กการระบุโปรเซส

	-A	-a	-d	-e	-G -g	-k	-p	-t	-U -u	a	g	t	x
โปรเซสทั้งหมด	Y	-	-	-	-	-	-	-	-	-	Y	-	-
ไม่ใช่ผู้นำกลุ่ม โปรเซสและไม่ เชื่อมโยงกับ เทอร์มินัล	-	Y	-	-	-	-	-	-	-	-	-	-	-
ไม่ใช่ผู้นำกลุ่ม โปรเซส	-	-	Y	-	-	-	-	-	-	-	-	-	-
ไม่ใช่โปรเซส เคอร์เนล	-	-	-	Y	-	-	-	-	-	-	-	-	-
สมาชิกของกลุ่ม โปรเซสที่ระบุ	-	-	-	-	Y	-	-	-	-	-	-	-	-
โปรเซสเคอร์เนล	-	-	-	-	-	Y	-	-	-	-	-	-	-

แฟล็กการระบุโปรเซส

	-A	-a	-d	-e	-G-g	-k	-p	-t	-U-u	a	g	t	x
ที่ระบุในรายการหมายเลขโปรเซส	-	-	-	-	-	-	Y	-	-	-	-	-	-
ที่เชื่อมโยงกับtty(s) ในรายการ	-	-	-	-	-	-	-	Y (n ttys)	-	-	-	Y (1 tty)	-
โปรเซสผู้ใช้ที่ระบุ	-	-	-	-	-	-	-	-	Y	-	-	-	-
โปรเซสเกี่ยวกับเทอร์มินัล	-	-	-	-	-	-	-	-	-	Y	-	-	-
ไม่เชื่อมโยงกับtty	-	-	-	-	-	-	-	-	-	-	-	-	Y

แฟล็กการเลือกคอลัมน์

ดีฟอลต์1	-f	-l	-U-u	ดีฟอลต์2	e	l	s	u	v	
PID	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
TTY	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
TIME	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CMD	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
USER	-	Y	-	-	-	-	-	-	Y	-
UID	-	-	Y	Y	-	-	Y	-	-	-
PPID	-	Y	Y	-	-	-	Y	-	-	-
C	-	Y	Y	-	-	-	Y	-	-	-
STIME	-	Y	-	-	-	-	-	-	Y	-
F	-	-	Y	-	-	-	-	-	-	-
S/STAT	-	-	Y	-	Y	Y	Y	Y	Y	Y
PIR	-	-	Y	-	-	-	Y	-	-	-
NI/NICE	-	-	Y	-	-	-	Y	-	-	-
ADDR	-	-	Y	-	-	-	Y	-	-	-
SIZE	-	-	-	-	-	-	-	-	Y	-
SZ	-	Y	-	-	-	Y	-	Y	-	-
WCHAN	-	-	Y	-	-	-	Y	-	-	-
RSS	-	-	-	-	-	-	Y	-	Y	Y
SSIZ	-	-	-	-	-	-	-	Y	-	-

แฟล็กการเลือกคอลัมน์

ดีฟอลต์1	-f	-l	-U -u	ดีฟอลต์2	e	l	s	u	v	
%CPU	-	-	-	-	-	-	-	-	Y	Y
%MEM	-	-	-	-	-	-	-	-	Y	Y
PGIN	-	-	-	-	-	-	-	-	-	Y
LIM	-	-	-	-	-	-	-	-	-	Y
TSIZ	-	-	-	-	-	-	-	-	-	Y
TRS	-	-	-	-	-	-	-	-	-	Y
สภาวะแวดล้อม (ตามคำสั่ง)	-	-	-	-	-	Y	-	-	-	-

ถ้ากำหนด ps โดยไม่มีแฟล็กหรือพร้อมกับแฟล็กการระบุโปรเซส ที่ขึ้นต้นด้วยเครื่องหมายลบ คอลัมน์ที่แสดงขึ้นคือคอลัมน์ที่แสดงสำหรับดีฟอลต์1 ถ้ากำหนดคำสั่งพร้อมกับแฟล็กการระบุโปรเซสที่ไม่ได้ขึ้นต้นด้วยเครื่องหมายลบ คอลัมน์ดีฟอลต์2 จะแสดงขึ้น แฟล็ก -u หรือ -U เป็นทั้งแฟล็กการระบุโปรเซสและแฟล็กการเลือกคอลัมน์

ข้อมูลต่อไปนี้เป็นคำอธิบายย่อเกี่ยวกับเนื้อหาของคอลัมน์:

ไอเท็ม	คำอธิบาย
PID	ID โปรเซส
TTY	เทอร์มินัลหรือเทอร์มินัลจำลองที่เชื่อมโยงกับโปรเซส
TIME	เวลา CPU สะสมที่ใช้ในหน่วยนาทีและวินาที
CMD	คำสั่งของโปรเซสที่กำลังรัน
USER	ชื่อล็อกอินของผู้ใช้ที่เป็นเจ้าของโปรเซส
UID	ID ผู้ใช้ที่เป็นตัวเลขของผู้ใช้ที่เป็นเจ้าของโปรเซส
PPID	ID ของโปรเซสพารেন্টของโปรเซสนี้
C	เวลา CPU ที่ใช้ล่าสุด
STIME	เวลาที่โปรเซสเริ่มต้น ถ้าน้อยกว่า 24 ชั่วโมง มิฉะนั้น วันที่ซึ่ง โปรเซสเริ่มต้น
F	ค่าเลขฐานสิบหกแปดอักขระที่อธิบายแฟล็กซึ่งเชื่อมโยงกับ โปรเซส (โปรดดูที่คำอธิบายโดยละเอียดของคำสั่ง ps)
S/STAT	สถานะของโปรเซส (โปรดดูที่คำอธิบายโดยละเอียดของคำสั่ง ps)
PRI	ค่าลำดับความสำคัญปัจจุบันของโปรเซส
NI/NICE	ค่าที่ละเอียดของโปรเซส
ADDR	หมายเลขเซ็กเมนต์ของโปรเซสสแต็ก
SIZE	(แฟล็ก -v) ขนาดเสมือนของส่วนข้อมูลของโปรเซส (ในหน่วยกิโลไบต์)
SZ	(แฟล็ก -l และ l) ขนาดในหน่วยกิโลไบต์ของรูปภาพหลัก ของโปรเซส
WCHAN	เหตุการณ์ซึ่งโปรเซสกำลังรอ
RSS	ผลรวมของจำนวนของหน้าเซ็กเมนต์การทำงานและโค้ดเซ็กเมนต์ใน หน่วยความจำคูณด้วย 4
SSIZ	ขนาดของเคอร์เนลสแต็ก
%CPU	เปอร์เซ็นต์ของเวลาที่ใช้ CPU นับตั้งแต่โปรเซส เริ่มต้นขึ้น
%MEM	โดยปกติ เปอร์เซ็นต์ของหน่วยความจำแบบจริงที่โปรเซสกำลังใช้อยู่ การประเมินนี้ไม่ได้สัมพันธ์กับสถิติหน่วยความจำอื่นใด
PGIN	จำนวนหน้า ins ที่เกิดขึ้นโดยความผิดพลาดของเพจ เนื่องจาก I/O ทั้งหมดถูกจัดประเภท เป็นความผิดพลาดของเพจ นี่จึงเป็นการประเมินพื้นฐานของ I/O วอลุ่ม
LIM	xx เสมอ
TSIZ	ขนาดของส่วนข้อความของไฟล์ปฏิบัติการ
TRS	จำนวนของหน้าโค้ดเซ็กเมนต์คูณด้วย 4
สภาวะแวดล้อม	ค่าของตัวแปรสภาวะแวดล้อมทั้งหมดของโปรเซส

การเปลี่ยนลำดับความสำคัญของโปรเซส

โดยพื้นฐาน ถ้าคุณทราบโปรเซสที่กำลังใช้เวลา CPU มากเกินไป คุณสามารถลดลำดับความสำคัญที่ใช้อยู่ของโปรเซสนั้นได้ โดยการเพิ่มค่าที่ละเอียด โดยใช้คำสั่ง `renice`

ตัวอย่างเช่น:

```
renice +5 ProcID
```

ค่าที่ละเอียดของ `ProcID` จะเพิ่มโปรเซสจากค่าปกติ 20 ของกระบวนการพื้นหน้าเป็น 25 คุณต้องมีสิทธิ์การใช้งานรากเพื่อรีเซ็ต ค่าที่ละเอียดของ `ProcID` เป็น 20 โดยพิมพ์:

```
renice -5 ProcID
```

การยุติโปรเซส

โดยปกติ คุณใช้คำสั่ง `kill` เพื่อสิ้นสุดโปรเซส

คำสั่ง `kill` ส่งสัญญาณไปยังโปรเซสที่กำหนด ขึ้นอยู่กับชนิดของสัญญาณและธรรมชาติของโปรแกรมที่กำลังรันในโปรเซส โปรเซสอาจสิ้นสุดหรืออาจรันต่อไป สัญญาณที่คุณส่งคือ:

ไอเท็ม	คำอธิบาย
SIGTERM	(สัญญาณ 15) เป็นคำร้องขอไปยังโปรแกรมให้ยุติ หากโปรแกรม มีตัวจัดการสัญญาณสำหรับ SIGTERM ที่ไม่ได้ยุติแอปพลิเคชันอย่างแท้จริง <code>kill</code> นี้ อาจไม่มีผล นี่เป็นสัญญาณดีฟอลต์ ที่ส่งโดย <code>kill</code>
SIGKILL	(สัญญาณ 9) เป็นคำสั่งให้ทำลายโปรเซสในทันที ไม่สามารถดักจับ หรือละเว้นสัญญาณนี้

โดยปกติ ควรออกใช้ SIGTERM มากกว่า SIGKILL หากโปรแกรมมีตัวจัดการ สำหรับ SIGTERM โปรแกรมสามารถทำความสะอาดและยุติตามลำดับได้ โดยพิมพ์:

```
kill -term ProcessID
```

(`-term` สามารถข้ามไปได้) หากโปรเซสไม่ตอบสนองต่อ SIGTERM ให้พิมพ์:

```
kill -kill ProcessID
```

คุณอาจสังเกตเห็นโปรเซสร้างในบางโอกาสซึ่งเรียกอีกอย่างว่า *zombies* ในตารางโปรเซสของคุณ โปรเซสเหล่านี้ไม่มีการดำเนินการอีกต่อไป ไม่มีพื้นที่ว่างระบบ ที่จัดสรร แต่ยังคงหมายเลข PID ของโปรเซสไว้ คุณสามารถทราบว่าเป็นกระบวนการ zombie ในตารางโปรเซสได้ เนื่องจากมีการแสดงคำว่า `<defunct>` ในคอลัมน์ CMD ตัวอย่างเช่น:

```
UID  PID  PPID  C   STIME  TTY  TIME CMD
.
.
.
lee  22392 20682  0   Jul 10   -   0:05 xclock
lee  22536 21188  0   Jul 10  pts/0  0:00 /bin/ksh
lee  22918 24334  0   Jul 10  pts/1  0:00 /bin/ksh
lee  23526 22536 22                0:00 <defunct>
lee  24334 20682  0   Jul 10   ?   0:00 aixterm
lee  24700   1   0   Jul 16   ?   0:00 aixterm
root 25394 26792  2   Jul 16  pts/2  0:00 ksh
lee  26070 24700  0   Jul 16  pts/3  0:00 /bin/ksh
lee  26792 20082  0   Jul 10  pts/2  0:00 /bin/ksh
root 27024 25394  2 17:10:44 pts/2  0:00 ps -ef
```

กระบวนการ zombie ยังคงมีอยู่ต่อไปในตารางโปรเซสจนกว่าโปรเซสพารেন্টตาย หรือปิดและรีสตาร์ทระบบ ในตัวอย่างที่แสดงข้างบน โปรเซส พารেন্ট (PPID) คือคำสั่ง `ksh` เมื่อออกจาก Korn shell โปรเซสสร้างจะถูกลบออกจากตารางโปรเซส

ในบางครั้ง โปรเซสสร้างเหล่านี้มีอยู่เป็นจำนวนมากในตารางโปรเซส เนื่องจากแอปพลิเคชันแตกแขนงหลายโปรเซสซายน์และยังไม่ได้ออก หากนี่กลายเป็นปัญหาวิธีแก้ไขที่ง่ายที่สุดคือการแก้ไขแอปพลิเคชัน เพื่อให้รู้ที่น้อย sigaction ละเว้นสัญญาณ

SIGCHLD

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `sigaction`

การผูกหรือการยกเลิกผูกโปรเซส

คุณสามารถผูกโปรเซสเข้ากับตัวประมวลผล หรือยกเลิกผูกโปรเซสที่ผูกไว้ก่อนหน้านี้

คุณต้องมีสิทธิ์ผู้ใช้รากเพื่อผูกหรือยกเลิกผูกโปรเซสซึ่งคุณ ไม่ได้เป็นเจ้าของ

บนระบบหลายตัวประมวลผล คุณสามารถผูกโปรเซสเข้ากับตัวประมวลผล หรือ ยกเลิกผูกโปรเซสที่ผูกไว้ก่อนหน้านี้จาก:

- SMIT
- บรรทัดคำสั่ง

หมายเหตุ: แม้ว่าการผูกโปรเซสเข้ากับตัวประมวลผลอาจช่วยปรับปรุง ประสิทธิภาพของโปรเซสที่ผูก (โดยการลด hardware-cache misses) แต่การใช้สิ่งอำนวยความสะดวกนี้มากเกินไปอาจส่งผลให้ตัวประมวลผล บางตัวโอเวอร์โหลด ในขณะที่ตัวประมวลผลอื่นมีการใช้งานน้อยเกินไป ปัญหาคอขวดที่เกิดขึ้น อาจส่งผลให้ปริมาณงานและประสิทธิภาพโดยรวมลดลง ในระหว่างการดำเนินงานปกติ ควรปล่อยให้ระบบปฏิบัติการกำหนดโปรเซสให้กับตัวประมวลผลโดยอัตโนมัติ เพื่อกระจายโหลดระบบบนตัวประมวลผลทั้งหมด ควรผูกเฉพาะโปรเซส ซึ่งคุณทราบว่าสามารถได้รับประโยชน์จากการรันบนตัวประมวลผลเดียว

ภารกิจการผูกหรือการยกเลิกผูก โปรเซส

ภารกิจ	พาด่วน SMIT	คำสั่งหรือไฟล์
การผูกโปรเซส	<code>smit bindproc</code>	<code>bindprocessor -q</code>
การยกเลิกผูกโปรเซส	<code>smit ubindproc</code>	<code>bindprocessor -u</code>

การแก้ไขโปรเซสที่เก่าหรือที่ไม่ต้องการ:

โปรเซสที่เก่าหรือที่ไม่ต้องการสามารถทำให้เกิดปัญหา เกี่ยวกับเทอร์มินัลของคุณ บางปัญหาจัดทำข้อความบนหน้าจอที่ให้อข้อมูลเกี่ยวกับสาเหตุที่เป็นไปได้

ในการทำโปรซีเตอร์ต่อไปนี้ คุณต้องมี เทอร์มินัลเครื่องที่สอง โมเด็ม หรือล็อกอินเครือข่าย อย่างใดอย่างหนึ่ง หากคุณไม่มีอุปกรณ์เหล่านี้ ให้แก้ไขปัญหาเทอร์มินัลโดยการรีบูตเครื่อง

เลือกโปรซีเตอร์ที่เหมาะสมสำหรับการแก้ไขปัญหา เทอร์มินัล:

การทำเทอร์มินัลที่ใช้โดยโปรเซสในว่าง:

คุณสามารถหยุดโปรเซสที่เก่าหรือที่ไม่ต้องการได้

ระบุและหยุดโปรเซสที่เก่าหรือที่ไม่ต้องการโดยทำ ดังต่อไปนี้:

1. กำหนดโปรเซสที่ใช้งานอยู่และกำลังรันบนหน้าจอโดยพิมพ์ คำสั่ง ps ต่อไปนี้:

```
ps -ef | pg
```

คำสั่ง ps แสดงสถานะของโปรเซส แฟล็ก -e เขียนข้อมูลเกี่ยวกับโปรเซสทั้งหมด (ยกเว้นโปรเซสเคอร์เนล) และแฟล็ก f สร้างรายการฉบับสมบูรณ์ของ โปรเซส รวมถึงชื่อคำสั่งและพารามิเตอร์ เมื่อโปรเซสถูกสร้างขึ้น คำสั่ง pg จำกัดเอาต์พุตเพียงหน้าเดียวในแต่ละครั้ง เพื่อไม่ให้ข้อมูลหลุดจากหน้าจอเร็วนัก

โปรเซสที่น่าสงสัย รวมถึงโปรเซสระบบหรือผู้ใช้ที่ใช้รีซอร์สระบบ เช่น CPU หรือพื้นที่ว่างดิสก์ จำนวนมากเกินไป โปรเซสระบบ เช่น **sendmail**, **routed**, และ **lpd** มัก runaways บ่อยครั้ง ใช้คำสั่ง **ps -u** เพื่อตรวจสอบการใช้ CPU

2. กำหนดบุคคลที่กำลังรันโปรเซสบนเครื่องนี้โดยใช้คำสั่ง who:

```
who
```

คำสั่ง who แสดงข้อมูลเกี่ยวกับผู้ใช้ทั้งหมดบนระบบนี้ในปัจจุบัน เช่น ชื่อล็อกอิน ชื่อเวิร์กสเตชัน วันที่ และเวลาการล็อกอิน

3. กำหนดว่าคุณต้องการหยุด พักไว้ หรือเปลี่ยนลำดับความสำคัญ ของโปรเซสผู้ใช้หรือไม่

หมายเหตุ: คุณต้องมีสิทธิ์การใช้งานรากเพื่อหยุด โปรเซสที่คุณไม่ได้เป็นเจ้าของ หากคุณยุติหรือเปลี่ยนลำดับความสำคัญของโปรเซสผู้ใช้โปรดติดต่อเจ้าของโปรเซสและอธิบายสิ่งที่คุณสามารถทำได้

- หยุดโปรเซสโดยใช้คำสั่ง **kill** ตัวอย่างเช่น:

```
kill 1883
```

คำสั่ง **kill** ส่งสัญญาณไปยังโปรเซสที่กำลังรัน เมื่อต้องการหยุดโปรเซส ให้ระบุ ID โปรเซส (PID) ซึ่งเป็น 1883 ในตัวอย่างนี้ ใช้คำสั่ง **ps** เพื่อกำหนดหมายเลข PID ของคำสั่ง

- พักโปรเซสไว้และรันในพื้นหลังโดยใช้เครื่องหมาย ampersand (&) ตัวอย่างเช่น:

```
/u/bin1/prog1 &
```

& ส่งสัญญาณว่า คุณต้องการให้โปรเซสนี้รันในพื้นหลัง ในกระบวนการส่วนหลัง เซลล์ไม่รอให้คำสั่งเสร็จสมบูรณ์ก่อนกลับไปยัง เซลล์พร้อมต์ เมื่อโปรเซสใช้เวลามากกว่าสองสามวินาทีในการทำให้เสร็จสมบูรณ์ ให้รันคำสั่งในพื้นหลังโดยพิมพ์ & ที่ตอนท้ายของ บรรทัดคำสั่ง งานที่กำลังรันในพื้นหลังปรากฏขึ้นในคำสั่ง ps ปกติ

- เปลี่ยนลำดับความสำคัญของโปรเซสที่ taken over โดยใช้คำสั่ง **renice** ต่อไปนี้:

```
renice 20 1883
```

คำสั่ง **renice** เปลี่ยนลำดับความสำคัญที่จัดตารางเวลาของโปรเซสที่กำลังรันตั้งแต่หนึ่งโปรเซสขึ้นไป ตัวเลขยิ่งสูงลำดับความสำคัญยิ่งต่ำ โดยที่ 20 เป็นลำดับความสำคัญต่ำสุด

ใน ตัวอย่างก่อนหน้านี้ **renice** จัดตารางเวลาโปรเซสหมายเลข 1883 อีกครั้งเป็นลำดับความสำคัญต่ำสุด โปรเซสนี้จะรันเมื่อมีเวลาตัวประมวลผลที่ไม่ได้ใช้จำนวนเล็กน้อย

การตอบกลับไปยังข้อความหน้าจอ:

ใช้โปรแกรมนี้เพื่อตอบกลับไปยังและกู้คืนจากข้อความหน้าจอ

1. ตรวจสอบให้แน่ใจว่ามีการตั้งค่าตัวแปรสถานะแวดล้อม **DISPLAY** อย่างถูกต้อง ใช้เมธอดอย่างใดอย่างหนึ่งต่อไปนี้เพื่อตรวจสอบสถานะแวดล้อม **DISPLAY**:

- ใช้คำสั่ง **setenv** เพื่อแสดงตัวแปรสถานะแวดล้อม

```
setenv
```

คำสั่ง **setenv** แสดง สถานะแวดล้อมในสภาพที่ป้องกันไว้เมื่อคุณล็อกอิน

กำหนดว่ามีการตั้งค่า ตัวแปร **DISPLAY** แล้วหรือไม่ ในตัวอย่าง ต่อไปนี้ ตัวแปร **DISPLAY** ไม่ปรากฏขึ้น ซึ่งบ่งชี้ว่า ไม่มีการตั้งค่าตัวแปร **DISPLAY** เป็น ค่าเฉพาะ

```
SYSENVIRON:  
NAME=casey  
TTY=/dev/pts/5  
LOGNAME=casey  
LOGIN=casey
```

หรือ

- เปลี่ยนค่าของตัวแปร **DISPLAY** ตัวอย่าง เช่น เมื่อต้องการตั้งค่าเป็นเครื่องที่ชื่อ **bastet** และเทอร์มินัล 0 ให้ป้อน:

```
DISPLAY=bastet:0  
export DISPLAY
```

หากไม่ได้ตั้งค่าโดยเฉพาะ ตัวแปรสถานะแวดล้อม **DISPLAY** จะใช้ค่าดีฟอลต์ **unix:0** (คอนโซล) ค่าของตัวแปร อยู่ในรูปแบบ **name:number** โดยที่ **name** คือ ชื่อโฮสต์ของเครื่องเฉพาะ และ **number** คือ หมายเลขเซิร์ฟเวอร์ X บนระบบที่ระบุ

2. รีเซ็ตเทอร์มินัลเป็นค่าดีฟอลต์โดยใช้คำสั่ง **stty** ต่อไปนี้:

```
stty sane
```

คำสั่ง **stty sane** เรียกคืน “ภาวะปกติ” ของเทอร์มินัลไดรเวอร์ คำสั่ง ให้เอาต์พุตเป็นโค้ดการรีเซ็ตเทอร์มินัลที่เหมาะสม จากไฟล์ **/etc/termcap** (หรือ **/usr/share/lib/terminfo** ถ้ามีอยู่)

3. หากปุ่ม Return ทำงานไม่ถูกต้อง ให้รีเซ็ตโดยพิมพ์:

```
^J stty sane ^J
```

^J แสดงถึงลำดับคีย์ **Ctrl-J**

การรันหลายคิวโดยใช้ตัวแปรสถานะแวดล้อม **RT_MPC และ **RT_GRQ**:**

การใช้หลายคิวเพิ่มความสัมพันธ์ของตัวประมวลผลของเรด แต่มีสถานการณ์พิเศษซึ่งคุณอาจต้องดำเนินการกับ ผลกระทบนี้

เมื่อมีคิวที่รันอยู่เพียงคิวเดียว เรดที่เรียกใช้แล้ว (เรดที่เรียกใช้) โดยเรดที่กำลังรันอื่น (เรดผู้เรียก) โดยปกติแล้ว จะสามารถ ใช้ CPU ซึ่งเรดผู้เรียกกำลังรันได้ในทันที หากมีคิวที่รันอยู่หลายคิว เรดที่เรียกใช้อาจอยู่บนคิวการรันของ CPU อื่นซึ่งไม่สามารถสังเกตเรดที่เรียกใช้ได้จนกว่าจะมีการตัดสินใจจัดตารางเวลาครั้งถัดไป ซึ่งอาจส่งผลให้เกิดเวลาหน่วงสูงสุดถึง 10 ms

กรณีนี้คล้ายกับสถานการณ์จำลองในรีลีสก่อนหน้าของระบบปฏิบัติการนี้ ซึ่งอาจไม่เกิดขึ้นเมื่อใช้อ็พชัน **bindprocessor** หาก CPUs ทั้งหมดอยู่ อย่างต่อเนื่อง และมีเรดที่เรียกใช้ซึ่งเชื่อมโยงอยู่เป็นจำนวนมาก สามารถเลือกใช้ได้สองอ็พชัน

- อ็พชันแรกซึ่งใช้คิวที่รันอยู่เพียงคิวเดียว คือการตั้งค่าตัวแปรสถานะแวดล้อม **RT_GRQ=ON** ซึ่งจะบังคับให้ยกเลิกการผูกเรดที่เลือก ไว้เพื่อจัดส่งไปยังคิวที่รันอยู่แบบสากล

- หรือผู้ใช้สามารถเลือกเรียลไทม์เคอร์เนลโอเพชัน (พิมพ์คำสั่ง bosdebug -R on จากนั้น bosboot) และตัวแปรสภาวะแวดล้อม RT_MPC=ON สำหรับโปรเซสที่เลือกจำเป็นต้องเก็บรักษาข้อมูลบันทึกประสิทธิภาพของระบบของคุณเพื่อมอนิเตอร์ผลกระทบของการปรับใดๆ ที่คุณพยายามทำอย่างใกล้ชิด

การทำแอคเคาต์ระบบ

ยูทิลิตี้การทำแอคเคาต์ระบบอนุญาตให้คุณรวบรวมและรายงานในการใช้งานแต่ละคน หรือกลุ่มของรีซอร์สระบบต่างๆ

ข้อมูลแอคเคาต์นี้สามารถใช้เพื่อแจ้งผู้ใช้ ทราบถึงการใช้งานรีซอร์สระบบที่ใช้งาน และเพื่อมอนิเตอร์แนวทางที่เลือกของการดำเนินการระบบ เมื่อต้องการช่วยในการเรียงเก็บเงิน ระบบแอคเคาต์ยังจัดให้มี ผลรวมการใช้งานรีซอร์สที่กำหนดโดยสมาชิกของกลุ่ม adm และถ้าคำสั่ง chargefee ถูกรวม ปัจจัยในค่าธรรมเนียมการเรียกเก็บเงิน

ระบบแอคเคาต์ยังจัดให้มีข้อมูลเพื่อเข้าถึง ความพอเพียงของการกำหนดค่ารีซอร์สปัจจุบัน ตั้งค่าขีดจำกัดรีซอร์ส และโควตา คาดการณ์ความต้องการในอนาคต และสั่งซื้ออุปกรณ์สิ้นเปลืองสำหรับเครื่องพิมพ์และอุปกรณ์อื่นๆ

ข้อมูลต่อไปนี้จะช่วยให้คุณเข้าใจ วิธีประยุกต์ใช้ยูทิลิตี้ระบบแอคเคาต์ในระบบของคุณ

หลักการที่เกี่ยวข้อง:

“การจัดการโปรเซส” ในหน้า 166

โปรเซสคือเอนทิตีที่ระบบปฏิบัติการใช้เพื่อควบคุม การใช้รีซอร์สระบบ *เรด* สามารถควบคุมการใช้เวลาตัวประมวลผล แต่เครื่องมือการจัดการระบบส่วนใหญ่ยังต้องการให้คุณอ้างอิงโปรเซส ซึ่งเรดกำลังรันอยู่ แทนการอ้างอิงตัวเรดเอง

“ผู้จัดการเวิร์กโหลด” ในหน้า 525

Workload Manager (WLM) ออบแบบมาเพื่อให้ ผู้ดูแลระบบสามารถควบคุมวิธีการที่ตัวจัดตารางเวลา virtual memory manager (VMM) และระบบย่อยดิสก์ I/O จัดสรรรีซอร์ส ให้กับโปรเซสได้มากขึ้น

“การบัญชีต่อคลาส” ในหน้า 532

ยูทิลิตี้ระบบการบัญชี AIX ช่วยให้คุณสามารถรวบรวมและรายงานการใช้รีซอร์สระบบหลายชนิด โดยเรียงตามผู้ใช้กลุ่ม หรือ คลาส WLM

งานที่เกี่ยวข้อง:

“การแก้ไขโอเวอร์ไฟล์ในระบบไฟล์ /var” ในหน้า 491

ตรวจสอบข้อมูลต่อไปนี้เมื่อระบบไฟล์ /var เต็ม

ข้อมูลที่เกี่ยวข้อง:

AIX เวอร์ชัน 6.1 ระบบย่อยการบัญชีขั้นสูง

รายงานข้อมูลการบัญชี

หลังจากรวบรวมข้อมูลการบัญชีชนิดต่างๆ แล้ว เร็กคอร์ด จะถูกประมวลผลและแปลงเป็นรายงาน

คำสั่งการบัญชีแปลงเร็กคอร์ดเป็นสัญญาณทางวิทยาศาสตร์โดยอัตโนมัติเมื่อตัวเลขเริ่มมากขึ้น มีการใช้สัญญาณทางวิทยาศาสตร์ แสดงแทนตัวเลขในรูปแบบต่อไปนี้:

Base+Exp

Base-Exp

ซึ่งเป็นตัวเลขเท่ากับตัวเลข *Base* คูณด้วย 10 เป็นกำลัง *+Exp* หรือ *-Exp* ตัวอย่างเช่น สัญกรณ์ทางวิทยาศาสตร์ 1.345e+9 เท่ากับ 1.345×10^9 หรือ 1,345,000,000 และสัญกรณ์ทางวิทยาศาสตร์ 1.345e-9 เท่ากับ 1.345×10^{-9} หรือ 0.000000001345

หลักการที่เกี่ยวข้อง:

“ข้อมูลบัญชีกระบวนการ” ในหน้า 195

ระบบการบัญชีรวบรวมข้อมูลเกี่ยวกับการใช้รีซอร์สของแต่ละโปรเซสในขณะที่โปรเซสรัน

รายงานการบัญชีรายวัน:

เมื่อต้องการสร้างรายงานรายวัน ให้ใช้คำสั่ง `runacct`

คำสั่งนี้สรุปข้อมูลไว้ในไฟล์ ASCII ที่ชื่อ `/var/adm/acct/sum(x)/rprtMMDDMMDD` ระบุเดือนและวันซึ่งรันรายงาน รายงาน ครอบคลุมข้อมูลต่อไปนี้:

- รายงานรายวัน
- รายงานการใช้งานรายวัน
- สรุปคำสั่งรายวัน
- สรุปคำสั่งทั้งหมดรายเดือน
- ล็อกอินครั้งล่าสุด

รายงานรายวัน:

รายงานการบัญชีรายวันมีข้อมูลเวลาเชื่อมต่อ โปรเซส การใช้ดิสก์ การใช้เครื่องพิมพ์ และค่าธรรมเนียมที่จะเรียกเก็บ

คำสั่ง `acctmrg` รวมข้อมูลการบัญชีดิบ เกี่ยวกับเวลาเชื่อมต่อ โปรเซส การใช้ดิสก์ การใช้เครื่องพิมพ์ และ ค่าธรรมเนียมที่จะเรียกเก็บเข้าในรายงานรายวัน โดยถูกเรียกโดยคำสั่ง `runacct` เป็นส่วนหนึ่งของการดำเนินงานรายวัน คำสั่ง `acctmrg` จัดทำข้อมูลต่อไปนี้:

`/var/adm/acct/nite(x)/dacct`

รายงานระหว่างกลางที่จัดทำขึ้นเมื่อไฟล์อินพุตไฟล์ใดไฟล์หนึ่ง เต็ม

`/var/adm/acct/sum(x)/tacct`

รายงานทั้งหมดสะสมในรูปแบบ `tacct` ไฟล์นี้ใช้โดยคำสั่ง `monacct` เพื่อจัดทำข้อมูลสรุปรายเดือน ASCII

คำสั่ง `acctmrg` สามารถแปลง เร็กคอร์ดระหว่างรูปแบบ ASCII และไบนารี และรวมเร็กคอร์ดจากแหล่งข้อมูลที่แตกต่างกัน เข้าเป็นเร็กคอร์ดเดียวสำหรับแต่ละผู้ใช้ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `acctmrg` ให้ดูที่ `acctmrg`

บรรทัดแรกของรายงานรายวันเริ่มต้นด้วยเวลาเริ่มต้นและเวลาสิ้นสุด ของข้อมูลที่รวบรวมในรายงาน รายการของเหตุการณ์ ระดับระบบ รวมถึง การปิด การรีบูต และการเปลี่ยนแปลงระดับการรันใดๆ ที่มีอยู่นอกจากนี้ ยังแสดง ช่วงเวลาทั้งหมดเพื่อบ่งชี้จำนวนนาฬิกาทั้งหมดที่รวมไว้ภายใน รอบระยะเวลาบัญชี (โดยปกติคือ 1440 นาที ถ้ารันรายงานทุก 24 ชั่วโมง) รายงานมีข้อมูลต่อไปนี้:

ไอเท็ม	คำอธิบาย
LINE	คอนโซล, tty, หรือ pty ที่ใช้อยู่
MINUTES	จำนวนนาทีทั้งหมดที่บรรทัดถูกใช้งาน
PERCENT	เปอร์เซ็นต์ของเวลาในรอบระยะเวลาบัญชีที่บรรทัดถูกใช้งาน
# SESS	จำนวนของล็อกอินเซสชันใหม่ที่เริ่มต้น
# ON	เหมือนกับ # SESS
# OFF	จำนวนของล็อกเอาต์บวกการขัดจังหวะที่ทำบนบรรทัด

รายงานการบัญชีการใช้งานรายวัน:

รายงานการใช้งานรายวันคือรายงานสรุปของการใช้งานระบบต่อ ID ผู้ใช้ในระหว่างรอบระยะเวลาบัญชี

บางฟิลด์มีการแบ่งเป็นเวลาหลักและเวลาที่ไม่ใช่เวลาหลัก ตามที่กำหนด โดยผู้ดูแลระบบการบัญชีในไดเรกทอรี /usr/lib/acct/holidays รายงานมีข้อมูลต่อไปนี้:

ไอเท็ม	คำอธิบาย
UID	ID ผู้ใช้
LOGIN NAME	ชื่อผู้ใช้
CPU (PRIME/NPRIME)	เวลา CPU ทั้งหมดของกระบวนการทั้งหมดของผู้ใช้ในหน่วยนาที
KCORE (PRIME/NPRIME)	หน่วยความจำทั้งหมดที่ใช้โดยการรันกระบวนการในหน่วยกิโลไบต์ต่อนาที
CONNECT (PRIME/NPRIME)	เวลาเชื่อมต่อทั้งหมด (ระยะเวลาที่ผู้ใช้ล็อกอิน) ในหน่วยนาที
DISK BLOCKS	จำนวนทั้งหมดโดยเฉลี่ยของพื้นที่วางดิสก์ซึ่งใช้โดยผู้ใช้บนระบบไฟล์ทั้งหมด ซึ่งเปิดใช้งานการบัญชี
FEES	ค่าธรรมเนียมทั้งหมดที่ป้อนด้วยคำสั่ง chargefee
# OF PROCS	จำนวนโปรเซสทั้งหมดที่เป็นของผู้ใช้รายนี้
# OF SESS	จำนวน distinct ล็อกอินเซสชันของผู้ใช้รายนี้
# DISK SAMPLES	จำนวนครั้งที่รันตัวอย่างดิสก์ในระหว่างรอบระยะเวลาบัญชี หากไม่มี DISK BLOCKS ค่าจะเป็นศูนย์

รายงานการบัญชีสรุปคำสั่งรายวัน:

รายงานสรุปคำสั่งรายวันแสดงคำสั่งแต่ละรายการที่ดำเนินการในระหว่าง รอบระยะเวลาบัญชี โดยแสดงหนึ่งบรรทัดสำหรับแต่ละชื่อคำสั่งที่ไม่ซ้ำกัน

ตารางมีการเรียงลำดับตาม TOTAL KCOREMIN (ที่อธิบายด้านล่าง) โดยที่บรรทัดแรกแสดงข้อมูลรวมสำหรับคำสั่งทั้งหมด ข้อมูลที่แสดงรายการสำหรับแต่ละคำสั่งเป็นข้อมูลสะสมของการดำเนินการทั้งหมดของ คำสั่งในระหว่างรอบระยะเวลาบัญชี คอลัมน์ในตารางนี้มีข้อมูล ต่อไปนี้:

ไอเท็ม	คำอธิบาย
COMMAND NAME	คำสั่งที่ถูกดำเนินการ
NUMBER CMDS	จำนวนครั้งที่ดำเนินการคำสั่ง
TOTAL KCOREMIN	หน่วยความจำทั้งหมดที่ใช้โดยการรันคำสั่ง ในหน่วยกิโลไบต์ต่อนาที
TOTAL CPU-MIN	เวลา CPU ทั้งหมดที่ใช้โดยคำสั่งในหน่วยนาที
TOTAL REAL-MIN	เวลาจริงทั้งหมดที่ผ่านไปสำหรับคำสั่งในหน่วยนาที
MEAN SIZE-K	ขนาดเฉลี่ยของหน่วยความจำที่ใช้โดยคำสั่งต่อนาที CPU
MEAN CPU-MIN	จำนวนเฉลี่ยของนาที CPU ต่อการดำเนินการของคำสั่ง
HOG FACTOR	หน่วยวัดจำนวนคำสั่งที่ hogs CPU ในขณะที่คำสั่งใช้งานอยู่ นี้เป็นอัตราส่วนของ TOTAL CPU-MIN ต่อ TOTAL REAL-MIN
CHARS TRNSFD	จำนวนอักขระที่โอนย้ายโดยคำสั่งด้วยการอ่านและการเขียน ระบบ
BLOCKS READ	จำนวนของการอ่านและการเขียนฟิลิคลบล็อกที่ทำโดยคำสั่ง

รายงานการบัญชีสรุปคำสั่งทั้งหมดรายเดือน:

สรุปคำสั่งทั้งหมดรายเดือนที่สร้างขึ้นโดยคำสั่ง `monacct` แสดงข้อมูลเกี่ยวกับคำสั่งทั้งหมดซึ่งดำเนินการ ตั้งแต่รายงานรายเดือนฉบับ ก่อนหน้านี้

ฟิลด์และข้อมูลมีความหมายเหมือนกับฟิลด์และข้อมูลใน สรุปคำสั่งรายวัน

ล็อกอินครั้งสุดท้าย:

รายงานล็อกอินครั้งสุดท้ายแสดงสองฟิลด์สำหรับแต่ละ ID ผู้ใช้ ฟิลด์แรกคือ `YY-MM-DD` และบ่งชี้ล็อกอินครั้งสุดท้ายของผู้ใช้ที่ระบุ ฟิลด์ที่สองคือชื่อของบัญชีผู้ใช้

ฟิลด์วันที่ `00-00-00` บ่งชี้ว่า ID ผู้ใช้ ไม่เคยล็อกอิน

สรุปรายงานการบัญชี:

คุณสามารถสร้างรายงานที่สรุปข้อมูลการบัญชีดิบ

เมื่อต้องการสรุปข้อมูลการบัญชีดิบ ให้ใช้คำสั่ง `sa` คำสั่งนี้อ่านข้อมูลการบัญชีดิบ ซึ่งโดยปกติ มีการรวบรวมในไฟล์ `/var/adm/pacct` และข้อมูลสรุปการใช้งานปัจจุบันในไฟล์ `/var/adm/savacct` ถ้ามีข้อมูลสรุป คำสั่งรวมข้อมูลนี้เข้าในรายงานสรุปการใช้งานใหม่ และล้างไฟล์ข้อมูลดิบเพื่อให้มีพื้นที่ว่างสำหรับการรวบรวมข้อมูลเพิ่มเติม

สิ่งที่จำเป็นต้องมี

คำสั่ง `sa` ต้องการไฟล์อินพุตของข้อมูลการบัญชีดิบ เช่น ไฟล์ `pacct` (ไฟล์บัญชีกระบวนการ) เพื่อรวบรวมข้อมูลการบัญชีดิบ คุณต้องมีระบบการบัญชีที่ตั้งค่า และกำลังรัน

โพธิ์เตอร์

วัตถุประสงค์ของคำสั่ง `sa` คือการสรุปข้อมูลการบัญชีกระบวนการ และการแสดงหรือจัดเก็บข้อมูลนั้น การใช้ที่ง่ายที่สุดของคำสั่งแสดงรายการของสถิติเกี่ยวกับทุกโปรเซสที่ รันในระหว่างอายุการใช้งานของไฟล์ `pacct` ที่อ่านอยู่ เมื่อต้องการจัดทำรายการดังกล่าว ให้พิมพ์:

```
/usr/sbin/sa
```

เมื่อต้องการสรุปข้อมูลบัญชีและรวมข้อมูลนั้นเข้าในไฟล์สรุปให้พิมพ์:

```
/usr/sbin/sa -s
```

คำสั่ง `sa` นำเสนอแฟล็กเพิ่มเติม จำนวนมากที่ระบุวิธีการประมวลผลและแสดงข้อมูลการบัญชี โปรดดูที่คำอธิบายคำสั่ง `sa` สำหรับข้อมูลเพิ่มเติม

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

รายงานรายเดือน:

คุณสามารถสร้างรายงานการบัญชีรายเดือนได้

โดยถูกเรียกโดย cron daemon คำสั่ง **monacct** จัดทำข้อมูลต่อไปนี้:

ไอเท็ม

/var/adm/acct/fiscal

คำอธิบาย

รายงานสรุปเป็นครั้งคราวที่จัดทำขึ้นจากรายงาน /var/adm/acct/sum/tacct โดยคำสั่ง **monacct** คำสั่ง **monacct** สามารถมีการกำหนดคอนฟิกเพื่อรันทุกเดือนหรือเมื่อสิ้นสุดรอบระยะเวลาบางประมาณ

รายงานเวลาเชื่อมต่อ:

เรีกคอร์ดการบัญชีมีเรีกคอร์ดล็อกอิน ล็อกเอาต์ การปิดระบบ และ เรีกคอร์ดล็อกอินครั้งล่าสุด

คำสั่ง **runacct** เรียกสองคำสั่งคือ **acctcon1** และ **acctcon2** เพื่อประมวลผลเรีกคอร์ดล็อกอิน ล็อกเอาต์ และการปิดระบบซึ่งรวบรวมในไฟล์ /var/adm/wtmp คำสั่ง **acctcon1** จะแปลงเรีกคอร์ดเหล่านี้เป็นเรีกคอร์ดเซสชันและเขียนเรีกคอร์ดเหล่านั้นไปยังไฟล์ /var/adm/acct/nite(x)/lineuse จากนั้น คำสั่ง **acctcon2** จะแปลงเรีกคอร์ดเซสชัน เป็นเรีกคอร์ดการบัญชีทั้งหมด /var/adm/logacct ซึ่ง คำสั่ง **acctmerg** เพิ่มลงในรายงานรายวัน สำหรับข้อมูล เกี่ยวกับคำสั่งเหล่านี้ ให้ดูที่ **runacct**, **acctcon1**, และ **acctcon2**

หากคุณรันคำสั่ง **acctcon1** จากบรรทัดคำสั่ง คุณต้องรวมแฟล็ก **-I** เพื่อจัดทำ รายงานการใช้บรรทัด /var/adm/acct/nite(x)/lineuse เมื่อต้องการจัดทำ รายงานเซสชันโดยรวมสำหรับรอบระยะเวลาบัญชี /var/adm/acct/nite(x)/reboots ให้ใช้คำสั่ง **acctcon1** พร้อมด้วยแฟล็ก **-o**

คำสั่ง **lastlogin** จัดทำ รายงานซึ่งแสดงวันที่ล่าสุดซึ่งผู้ใช้แต่ละรายล็อกอิน สำหรับข้อมูลเกี่ยวกับคำสั่ง **lastlogin** ให้ดูที่ **lastlogin** หลักการที่เกี่ยวข้อง:

“ข้อมูลการบัญชีเวลาเชื่อมต่อ” ในหน้า 195

ข้อมูลเวลาเชื่อมต่อมีการรวบรวมโดยคำสั่ง **init** และคำสั่ง **login**

“ข้อมูลบัญชีการใช้ดิสก์” ในหน้า 196

ข้อมูลการบัญชีจำนวนมากมีการรวบรวมเมื่อมีการใช้รีซอร์ส คำสั่ง **ddisk** ซึ่งรันตามที่ระบุโดย cron daemon เขียนเรีกคอร์ดการใช้ดิสก์เป็นครั้งคราวสำหรับผู้ใช้แต่ละรายไปยังไฟล์ /var/adm/acct/nite(x)/dacct

รายงานบัญชีการใช้ดิสก์:

เรีกคอร์ดการใช้ดิสก์ที่รวบรวมไว้ในไฟล์ /var/adm/acct/nite(x)/dacct มีการรวมเข้าในรายงานการบัญชีรายวันโดยคำสั่ง **acctmerg**

สำหรับข้อมูลเกี่ยวกับคำสั่ง **acctmerg** ให้ดูที่ **acctmerg**

รายงานบัญชีการใช้เครื่องพิมพ์:

เรีกคอร์ด ASCII ในไฟล์ /var/adm/qacct สามารถ มีการแปลงเป็นเรีกคอร์ดการบัญชีทั้งหมดที่จะถูกเพิ่มลงในรายงานรายวัน โดยคำสั่ง **acctmerg**

สำหรับข้อมูลเกี่ยวกับคำสั่ง **acctmerg** ให้ดูที่ **acctmerg**

หลักการที่เกี่ยวข้อง:

“ข้อมูลบัญชีการใช้เครื่องพิมพ์” ในหน้า 197

การรวบรวมข้อมูลการใช้เครื่องพิมพ์เป็นความพยายามร่วมกันระหว่าง คำสั่ง **enq** และ queuing daemon

รายงานการบัญชีค่าธรรมเนียม:

ถ้าคุณใช้คำสั่ง `chargefee` เพื่อเรียกเก็บเงินจากผู้ชำระค่าบริการ เช่น การเรียกคืนไฟล์ การให้คำปรึกษา หรือเอกสารประกอบ จะมีการเขียนเรกคอร์ดการบัญชีทั้งหมด ASCII ขึ้นในไฟล์ `/var/adm/fee` ไฟล์นี้จะถูกเพิ่มลงในรายงานรายวันโดยคำสั่ง `acctmerg`

สำหรับข้อมูลเกี่ยวกับคำสั่ง `chargefee` และ `acctmerg` ให้ดูที่ `chargefee` และ `acctmerg`

หลักการที่เกี่ยวข้อง:

“ข้อมูลการบัญชีค่าธรรมเนียม” ในหน้า 197

คุณสามารถจัดทำเรกคอร์ดการบัญชีทั้งหมด ASCII ในไฟล์ `/var/adm/fee`

รายงานการบัญชีงบประมาณ:

โดยทั่วไปแล้ว รายงานการบัญชีงบประมาณมีการจัดทำทุกเดือนโดยใช้คำสั่ง `monacct`

รายงานมีการจัดเก็บไว้ใน `/var/adm/acct/fiscal(x)/fiscrptMM` โดยที่ `MM` คือ เดือนที่ดำเนินการคำสั่ง `monacct` รายงานนี้มีข้อมูลคล้ายกับรายงานประจำวันที่สรุปสำหรับ ทั้งเดือน

รายงานกิจกรรมระบบการบัญชี:

คุณสามารถสร้างรายงานที่แสดงกิจกรรมของระบบการบัญชี

เมื่อต้องการสร้างรายงานเกี่ยวกับกิจกรรมระบบ ให้ใช้คำสั่ง `prtacct` คำสั่งนี้อ่าน ข้อมูลในไฟล์การบัญชีทั้งหมด (รูปแบบไฟล์ `tacct`) และจัดทำเอาต์พุตที่จัดรูปแบบ ไฟล์การบัญชีทั้งหมดมี รายงานประจำวันเกี่ยวกับเวลาเชื่อมต่อ เวลาประมวลผล การใช้ดิสก์ และการใช้เครื่องพิมพ์

สิ่งที่จำเป็นต้องมี

คำสั่ง `prtacct` ต้องการไฟล์อินพุตในรูปแบบไฟล์ `tacct` คำสั่งนี้สมมติว่าคุณมีระบบการบัญชีที่ตั้งค่าและกำลังรัน หรือคุณได้รันระบบการบัญชีแล้วในอดีต

โพธิ์เตอร์

สร้างรายงานเกี่ยวกับกิจกรรมระบบโดยการป้อน:

```
prtacct -f Specification -v Heading File
```

ข้อกำหนดคุณลักษณะ คือรายการที่ค้นด้วยเครื่องหมายคอมมาของ หมายเลขฟิลด์หรือช่วงที่ใช้โดยคำสั่ง `acctmerg` แฟล็ก `-v` ที่เป็นทางเลือก จัดทำเอาต์พุต `verbose` โดยที่หมายเลขจุดลอยตัวแสดงขึ้นในสัญกรณ์ ความแม่นยำสูงขึ้น *ส่วนหัว* คือหัวเรื่อง ซึ่งคุณต้องการให้ปรากฏขึ้นบนรายงาน และเป็นทางเลือก *ไฟล์* คือ ชื่อพารามิเตอร์เพิ่มเติมของไฟล์การบัญชีทั้งหมดที่จะใช้สำหรับอินพุต คุณสามารถ ระบุได้มากกว่าหนึ่งไฟล์

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

การสนับสนุนชื่อผู้ใช้งานมากกว่าแปดอักขระ:

เพื่อรักษาความเข้ากันได้กับสคริปต์รุ่นก่อนหน้าทั้งหมด การสนับสนุนชื่อผู้ใช้งานที่ยาวจึงไม่มีการเปิดใช้งานโดยค่าดีฟอลต์ภายในการบัญชี ID ผู้ใช้ ทั้งหมดจะถูกตัดให้เหลือแปดอักขระตัวแรกแทน

เพื่อเปิดใช้งานการสนับสนุนชื่อผู้ใช้งานที่ยาว คำสั่งส่วนใหญ่จึงมี แฟล็ก -X เพิ่มเติมซึ่งอนุญาตให้คำสั่งยอมรับและส่งเอาต์พุต ID ผู้ใช้ที่มากกว่าแปดอักขระได้ (ในทั้งรูปแบบ ASCII และไบนารี) นอกจากนี้ เมื่อเปิดใช้งานการสนับสนุนชื่อผู้ใช้งานที่ยาว คำสั่งและสคริปต์จะประมวลผลไฟล์ ในไดเรกทอรี /var/adm/acct/sumx, /var/adm/acct/nitex, และ /var/adm/acct/fiscalx แทน การใช้ /var/adm/acct/sum, /var/adm/acct/nite, และ /var/adm/acct/fiscal

คำสั่งการบัญชี

คำสั่งการบัญชีทำงานในหลายวิธีที่แตกต่างกัน

บางคำสั่ง:

- รวบรวมข้อมูลหรือจัดทำรายงานสำหรับการบัญชีชนิดเฉพาะ: เวลาเชื่อมต่อ โปรเซส การใช้ดิสก์ การใช้เครื่องพิมพ์ หรือการใช้คำสั่ง
- เรียกคำสั่งอื่น ตัวอย่างเช่น คำสั่ง **runacct** ซึ่งโดยปกติแล้ว รันโดยอัตโนมัติโดย **cron daemon** จะเรียกคำสั่งจำนวนมากซึ่งรวบรวมและประมวลผลข้อมูลการบัญชีและจัดเตรียม รายงาน เมื่อต้องการได้รับการบัญชีอัตโนมัติ อันดับแรก คุณต้องกำหนดคอนฟิก **cron daemon** เพื่อรันคำสั่ง **runacct** ให้ดูที่คำสั่ง **crontab** สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีการกำหนดคอนฟิก **cron daemon** เพื่อส่งคำสั่งในช่วงเวลาสม่ำเสมอที่จัดตารางเวลาไว้ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ คำสั่งเหล่านี้ ให้ดูที่ **runacct**, **cron daemon**, และ **crontab**
- ทำฟังก์ชันการบำรุงรักษาและตรวจสอบให้แน่ใจถึงคุณภาพของไฟล์ข้อมูล ที่ใช้งานอยู่
- เปิดใช้งานสมาชิกของกลุ่ม adm เพื่อทำภารกิจเป็นบางโอกาส เช่น การแสดง เร็กคอร์ดเฉพาะ โดยการป้อนคำสั่งที่คีย์บอร์ด
- เปิดใช้งานผู้ใช้เพื่อแสดงข้อมูลเฉพาะ มีคำสั่งผู้ใช้เพียงรายการเดียวเท่านั้น คือคำสั่ง **acctcom** ซึ่งแสดงข้อมูลสรุป บัญชีกระบวนการ

คำสั่งที่รันโดยอัตโนมัติ:

มีหลายคำสั่งรวบรวมข้อมูลการบัญชีโดยอัตโนมัติ

โดยปกติแล้ว หลายคำสั่งที่รันโดย **cron daemon** จะรวบรวมข้อมูลการบัญชีโดยอัตโนมัติ คำสั่งเหล่านี้คือ:

runacct

จัดการโปรซีเดเจอร์การบัญชีรายวันที่สำคัญ โดยปกติ ถูกเริ่มต้นโดย **cron daemon** ในระหว่างชั่วโมงที่ใช้งานไม่มากนัก คำสั่ง **runacct** จะเรียกคำสั่ง การบัญชีอื่นหลายคำสั่งเพื่อประมวลผลไฟล์ข้อมูล ที่ใช้งานอยู่ และจัดทำข้อมูลสรุป การใช้คำสั่งและรีซอร์สซึ่งเรียงลำดับตามชื่อผู้ใช้งาน นอกจากนี้ คำสั่งนี้ยังเรียกคำสั่ง **acctmrg** เพื่อจัดทำไฟล์รายงานสรุปรายวัน และคำสั่ง **ckpacct** เพื่อรักษาคุณภาพของไฟล์ข้อมูล ที่ใช้งานอยู่

ckpacct

จัดการขนาดไฟล์ **pacct** การมีไฟล์ **pacct** ขนาดเล็ก ลงหลายไฟล์เป็นสิ่งที่มีความจำเป็นถ้าคุณต้องรีสตาร์ทโปรซีเดเจอร์ **runacct** หลังจากความล้มเหลวในการประมวลผลเร็กคอร์ดเหล่านี้ คำสั่ง **ckpacct** ตรวจสอบขนาดของไฟล์ข้อมูล ที่ใช้งานอยู่ /var/adm/pacct และถ้าไฟล์มีขนาดใหญ่กว่า 500 บล็อก คำสั่งจะเรียกใช้คำสั่ง **turnacct switch** เพื่อปิด

บัญชีกระบวนการชั่วคราว ข้อมูล มีการโอนย้ายไปยังไฟล์ `pacct` ใหม่, `/var/adm/pacct.x` (x คือเลขจำนวนเต็มที่เพิ่มขึ้นในแต่ละครั้งที่มีการสร้างไฟล์ `pacct` ใหม่) หากจำนวนของดิสก์บล็อกที่ว่างลดลงต่ำกว่า 500 คำสั่ง `ckpacct` จะเรียกคำสั่ง `turnacct off` เพื่อปิดบัญชีกระบวนการ

dodisk เรียกคำสั่ง `acctdisk` และคำสั่ง `diskusg` หรือ คำสั่ง `acctdusg` อย่างใดอย่างหนึ่ง เพื่อเขียนเรีกคอร์ดการใช้ดิสก์ไปยังไฟล์ `/var/adm/acct/nite/dacct` ข้อมูลนี้มีการรวมเข้าใน รายงานรายวันในภายหลัง

dodisk เรียกคำสั่ง `acctdisk` และคำสั่ง `diskusg` หรือ คำสั่ง `acctdusg` อย่างใดอย่างหนึ่ง เพื่อเขียนเรีกคอร์ดการใช้ดิสก์ไปยังไฟล์ `/var/adm/acct/nite/dacct` ข้อมูลนี้มีการรวมเข้าใน รายงานรายวันในภายหลัง

monacct

จัดทำข้อมูลสรุปเป็นครั้งคราวจากรายงานรายวัน

sa1 รวบรวมและจัดเก็บข้อมูลไบนารีไว้ในไฟล์ `/var/adm/sa/sa dd` โดยที่ `dd` คือวันของเดือน

sa2 เขียนรายงานรายวันในไฟล์ `/var/adm/sa/sadd` โดยที่ `dd` คือวันของเดือน คำสั่งลบ รายงานซึ่งอยู่ในไฟล์ `/var/adm/sa/sadd` นานกว่าหนึ่งสัปดาห์

คำสั่งอื่นที่มีการรันโดยอัตโนมัติโดยโพธิ์เตอร์อื่น ที่ไม่ใช่ `cron daemon`:

startup เมื่อถูกเพิ่มลงในไฟล์ `/etc/rc` คำสั่ง `startup` จะเริ่มต้นโพธิ์เตอร์สตาร์ทอัพสำหรับระบบการบัญชี

shutacct

เรีกคอร์ดการบัญชีเวลาถูกปิดโดยการเรียกคำสั่ง `acctwtmp` เพื่อเขียนบรรทัดไปยังไฟล์ `/var/adm/wtmp` จากนั้นเรียก คำสั่ง `turnacct off` เพื่อปิดบัญชีกระบวนการ

คำสั่งคีย์บอร์ด:

สมาชิกของกลุ่ม `adm` สามารถป้อนคำสั่งต่อไปนี้จาก คีย์บอร์ดได้

ac พิมพ์เรีกคอร์ดเวลาเชื่อมต่อ คำสั่งนี้แนะนำเสนอเพื่อให้เข้ากันได้กับ ระบบการกระจายซอฟต์แวร์ของ Berkeley (BSD)

acctcom

แสดงข้อมูลสรุปบัญชีกระบวนการ คำสั่งนี้ยังมีอยู่สำหรับ ผู้ใช้ด้วย

acctcon1

แสดงข้อมูลสรุปเวลาเชื่อมต่อ ต้องใช้แฟล็ก `-l` หรือแฟล็ก `-o` อย่างใดอย่างหนึ่ง

accton เปิดหรือปิดบัญชีกระบวนการ

chargefee

เรียกเก็บเงินจากผู้ใช้สำหรับค่าธรรมเนียมที่กำหนดไว้ล่วงหน้าของหน่วยของงานที่ทำ ค่าธรรมเนียม จะถูกเพิ่มลงใน รายงานรายวันโดยคำสั่ง `acctmerg`

fwtmp แปลงไฟล์ระหว่างรูปแบบไบนารีและ ASCII

last แสดงข้อมูลเกี่ยวกับบล็อกอินก่อนหน้า คำสั่งนี้แนะนำเสนอเพื่อให้เข้ากันได้กับ ระบบ BSD

lastcomm

แสดงข้อมูลเกี่ยวกับคำสั่งล่าสุดที่ดำเนินการ คำสั่งนี้ นำเสนอเพื่อให้เข้ากันได้กับระบบ BSD

lastlogin

แสดงเวลาที่ผู้ใช้แต่ละรายล็อกอินครั้งล่าสุด

- pac** จัดเตรียมเรีกคอร์ดการบัญชีเครื่องพิมพ์/พล็อตเตอร์ คำสั่งนี้ นำเสนอเพื่อให้เข้ากันได้กับระบบ BSD
- prctmp** แสดงเรีกคอร์ดเซสชัน
- prtacct** แสดงไฟล์การบัญชีทั้งหมด
- sa** สรุปข้อมูลการบัญชีดิบเพื่อช่วยจัดการวอลุ่มขนาดใหญ่ของ ข้อมูลการบัญชี คำสั่งนี้ นำเสนอเพื่อให้เข้ากันได้กับระบบ BSD
- sadc** รายงานเกี่ยวกับการดำเนินการระบบโลคัลต่างๆ เช่น การใช้บัฟเฟอร์ กิจกรรม ดิสก์และเทป I/O, ตัวนับกิจกรรม อุปกรณ์ TTY, และตัวนับการเข้าถึงไฟล์
- sar** เขียนไปยังเอาต์พุตมาตรฐานเกี่ยวกับเนื้อหาของตัวนับกิจกรรมสะสมที่เลือกไว้ในระบบปฏิบัติการ คำสั่ง **sar** รายงานเฉพาะ กิจกรรมโลคัลเท่านั้น
- time** พิมพ์เวลาจริง เวลาผู้ใช้ และเวลาระบบที่ต้องการเพื่อรันคำสั่ง
- timex** รายงานจำนวนวินาทีของเวลาที่ผ่านไป เวลาผู้ใช้ และเวลารัน

หลักการที่เกี่ยวข้อง:

“การรวบรวมข้อมูลระบบและการรายงาน” ในหน้า 194
 คุณสามารถตั้งค่าระบบเพื่อรวบรวมข้อมูลและสร้าง รายงานโดยอัตโนมัติ

ไฟล์การบัญชี

ไดเรกทอรีการบัญชีหลักสองไดเรกทอรีคือไดเรกทอรี `/usr/sbin/acct` ซึ่งโปรแกรมภาษา C ทั้งหมดและเซลล์โพรซีเจอร์ ต้องการเพื่อรันการจับเก็บ ระบบการบัญชี และไดเรกทอรี `/var/adm` ซึ่งมีไฟล์ข้อมูล รายงาน และไฟล์ข้อมูลสรุป

ไฟล์ข้อมูลการบัญชีเป็นของ สมาชิกของกลุ่ม `adm` และไฟล์ข้อมูลที่ใช้งานอยู่ทั้งหมด (เช่น `wtmp` และ `pacct`) ตั้งอยู่ในไดเรกทอรีโฮม `adm /var/adm`

ไฟล์ข้อมูลการบัญชี:

ไฟล์ต่อไปนี้อยู่ในไดเรกทอรี `/var/adm`

<p>ไอเอ็ม <code>/var/adm/diskdiag</code> <code>/var/adm/dtmp</code> <code>/var/adm/fee</code> <code>/var/adm/pacct</code> <code>/var/adm/wtmp</code> <code>/var/adm/Spacct.mmd</code></p>	<p>คำอธิบาย เอาต์พุตการวิเคราะห์ในระหว่างการรันโปรแกรมการบัญชีดิสก์ เอาต์พุตจากคำสั่ง <code>acctdusg</code> เอาต์พุตจากคำสั่ง <code>chargefee</code> ในเรีกคอร์ด ASCII <code>tacct</code> ไฟล์บัญชีกระบวนการใช้งานอยู่ ไฟล์บัญชีกระบวนการใช้งานอยู่ ไฟล์บัญชีกระบวนการใช้สำหรับ <code>mmd</code> ในระหว่าง การดำเนินการคำสั่ง <code>runacct</code></p>
--	---

รายงานการบัญชีและไฟล์ข้อมูลสรุป:

ต้องการบางไดเรกทอรีย่อยก่อนคุณสามารถใช้งานระบบการบัญชีได้

รายงานและไฟล์ข้อมูลสรุปตั้งอยู่ในไดเรกทอรีย่อย `/var/adm/acct` คุณต้องสร้างไดเรกทอรีย่อยต่อไปนี้ก่อนสามารถใช้งานระบบการบัญชีได้

`/var/adm/acct/nite(x)`
 มีไฟล์ที่คำสั่ง `runacct` ใช้ซ้ำรายวัน

/var/adm/acct/sum(x)

มีไฟล์ข้อมูลสรุปสะสมที่คำสั่ง **runacct** อัปเดตรายวัน

/var/adm/acct/fiscal(x)

มีไฟล์ข้อมูลสรุปรายเดือนที่คำสั่ง **monacct** สร้างขึ้น

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

การเริ่มต้นคำสั่ง runacct สำหรับการบัญชี:

คุณสามารถเริ่มต้นคำสั่ง **runacct** ได้

สิ่งที่จำเป็นต้องมี

1. คุณต้องมีระบบการบัญชีที่ติดตั้งไว้แล้ว
2. คุณต้องมีสิทธิ์ผู้ใช้รากหรือกลุ่ม adm

หมายเหตุ:

1. หากคุณเรียกคำสั่ง **runacct** โดยไม่มีพารามิเตอร์ คำสั่งจะสมมติว่าเป็นครั้งแรกที่รันคำสั่งในวันนี้ ด้วยเหตุนี้ คุณต้องรวมพารามิเตอร์ **mddd** เมื่อคุณรีสตาร์ทโปรแกรม **runacct** เพื่อให้เดือน และวันถูกต้อง หากคุณไม่ได้รับสุขภาพโปรแกรม **runacct** จะอ่านไฟล์ `/var/adm/acct/nite(x)/statefile` เพื่อกำหนด จดรายการสำหรับการประมวลผล เมื่อต้องการยกเลิกไฟล์ `/var/adm/acct/nite(x)/statefile` ให้ระบุสุขภาพที่ต้องการบนบรรทัดคำสั่ง
2. เมื่อคุณทำภารกิจต่อไปนี้อย่างใดอย่างหนึ่ง คุณอาจต้องใช้ชื่อพารามิเตอร์เพิ่มเติม `/usr/sbin/acct/runacct` แทนชื่อคำสั่ง แบบง่าย **runacct**

โพรซีเจอร์

เมื่อต้องการเริ่มต้นคำสั่ง **runacct** ให้พิมพ์ดังต่อไปนี้:

```
nohup runacct 2> \  
/var/adm/acct/nite/accterr &
```

รายการนี้ส่งผลให้ คำสั่งละเว้นสัญญาณ **INTR** และ **QUIT** ทั้งหมดในขณะที่ทำการประมวลผลส่วนหลัง และเปลี่ยนทิศทางเอาต์พุตข้อผิดพลาดมาตรฐานทั้งหมดไปยังไฟล์ `/var/adm/acct/nite/accterr`

การรีสตาร์ทคำสั่ง runacct สำหรับการบัญชี:

หากคำสั่ง **runacct** ไม่สำเร็จ คุณสามารถรีสตาร์ทคำสั่งนั้นได้

ข้อกำหนดเบื้องต้นสำหรับโพรซีเจอร์นี้คือ:

- คุณต้องมีระบบการบัญชีที่ติดตั้งไว้แล้ว
- คุณต้องมีสิทธิ์ผู้ใช้รากหรือกลุ่ม adm

หมายเหตุ: เหตุผลที่พบได้บ่อยที่สุดซึ่งทำให้คำสั่ง **runacct** ล้มเหลวคือเนื่องจาก:

- ระบบดาวนั้
- ระบบไฟล์ `/usr` ขาดพื้นที่ว่าง

- ไฟล์ /var/adm/wtmp มีเร็กคอร์ดที่มีการประทับเวลา ไม่สอดคล้องกัน

หากคำสั่ง **runacct** ไม่สำเร็จ ให้ทำดังต่อไปนี้:

1. ตรวจสอบไฟล์ /var/adm/acct/nite(x)/active *mmdd* เพื่อดูข้อความแสดงข้อผิดพลาด
2. ถ้ามีทั้งไฟล์แอคทีฟและไฟล์ล็อกอยู่ใน acct/nite ให้ตรวจสอบไฟล์ accterr ซึ่งข้อความแสดงข้อผิดพลาดจะถูกเปลี่ยนทิศทางเมื่อ cron daemon เรียกคำสั่ง **runacct**
3. ปฏิบัติตามการดำเนินการใดๆ ที่จำเป็นเพื่อแก้ไขข้อผิดพลาด
4. รีสตาร์ทคำสั่ง **runacct**
5. เมื่อต้องการรีสตาร์ทคำสั่ง **runacct** ในวันที่ เฉพาะ ให้พิมพ์ดังต่อไปนี้:

```
nohup runacct 0601 2>> \
/var/adm/acct/nite/accterr &
```

คำสั่งนี้รีสตาร์ทโปรแกรม **runacct** ในวันที่ 1 มิถุนายน (0601) โปรแกรม **runacct** อ่านไฟล์ /var/adm/acct/nite/statefile เพื่อค้นหาว่าจะเริ่มต้นในสถานะใด เอาต์พุตข้อผิดพลาดมาตรฐานทั้งหมดถูกผนวกเข้ากับไฟล์ /var/adm/acct/nite/accterr

6. เมื่อต้องการรีสตาร์ทโปรแกรม **runacct** ในสถานะที่ระบุ ตัวอย่างเช่น สถานะ MERGE ให้พิมพ์ดังต่อไปนี้:

```
nohup runacct 0601 MERGE 2>> \
/var/adm/acct/nite/accterr &
```

ไฟล์คำสั่ง **runacct**:

คำสั่ง **runacct** จัดทำรายงานและไฟล์ ข้อมูลสรุป

รายงานและไฟล์ข้อมูลสรุปต่อไปนี้ซึ่งจัดทำขึ้นโดย คำสั่ง **runacct** เป็นข้อมูลที่น่าสนใจเป็นพิเศษ:

ไอเท็ม	คำอธิบาย
/var/adm/acct/nite(x)/lineuse	มีสถิติการใช้งานสำหรับแต่ละบรรทัดเทอร์มินัลบนระบบ รายงานนี้มีประโยชน์อย่างยิ่งสำหรับการตรวจพบบรรทัดที่ไม่ดี ถ้าอัตราส่วนระหว่าง จำนวนของล็อกเอาต์และล็อกอินเกินกว่าประมาณ 3 ต่อ 1 อาจมีโอกาasเป็นไปได้สูงที่บรรทัดล้มเหลวอยู่
/var/adm/acct/nite(x)/dayacct	มีไฟล์การบัญชีทั้งหมดสำหรับวันก่อนหน้า
/var/adm/acct/sum(x)/tacct	มีค่าสะสมของไฟล์ nite/dayacct ของแต่ละวัน และสามารถใช้ในการเรียกเก็บเงินได้ คำสั่ง monacct รีสตาร์ทไฟล์ในแต่ละเดือนหรือรอบระยะเวลาบางประมาณ
/var/adm/acct/sum(x)/cms	มีค่าสะสมข้อมูลสรุปคำสั่งของแต่ละวัน คำสั่ง monacct อ่านเวอร์ชันไบนารีนี้ของไฟล์และล้างออก เวอร์ชัน ASCII คือ nite/cms
/var/adm/acct/sum(x)/daycms	มีข้อมูลสรุปคำสั่งรายวัน เวอร์ชัน ASCII มีการจัดเก็บไว้ใน nite/daycms
/var/adm/acct/sum(x)/loginlog	มีเร็กคอร์ดของเวลาล่าสุดที่ใช้แต่ละ ID ผู้ใช้
/var/adm/acct/sum(x)/rprt <i>mmdd</i>	ไฟล์นี้มีสำเนาของรายงานรายวันซึ่งบันทึกโดยคำสั่ง runacct

ไฟล์ในไดเร็กทอรี /var/adm/acct/nite(x):

ไฟล์ต่อไปนี้อยู่ในไดเร็กทอรี /var/adm/acct/nite(x)

ไอเท็ม	คำอธิบาย
active	ใช้โดยคำสั่ง runacct เพื่อบันทึกความคืบหน้าและ พิมพ์ค่าเตือนและข้อความแสดงข้อผิดพลาด ไฟล์ active mmdd คือ สำเนาของไฟล์ active ที่จัดทำโดยโปรแกรม runacct หลังจากโปรแกรมตรวจพบข้อผิดพลาด
cms	ข้อมูลสรุปคำสั่งทั้งหมด ASCII ที่ใช้โดยคำสั่ง prdaily
ctacct.mmdd	เรียกคอร์ดการบัญชีทั้งหมดของการเชื่อมต่อ
ctmp	เรียกคอร์ดเซสชันเชื่อมต่อ
daycms	ข้อมูลสรุปคำสั่งรายวัน ASCII ที่ใช้โดยคำสั่ง prdaily
daytacct	เรียกคอร์ดการบัญชีทั้งหมดสำหรับหนึ่งวัน
dacct	เรียกคอร์ดการบัญชีทั้งหมดของดิสก์ที่สร้างขึ้นโดยคำสั่ง dodisk
accterr	เอาต์พุตการวิเคราะห์ที่จัดทำขึ้นในระหว่างการดำเนินการคำสั่ง runacct
lastdate	วันล่าสุดที่ดำเนินการ runacct ในรูปแบบ date +%m%d
lock1	ใช้เพื่อควบคุมการใช้ตามลำดับของคำสั่ง runacct
lineuse	รายงานการใช้บรรทัด tty ที่ใช้โดยคำสั่ง prdaily
log	เอาต์พุตการวิเคราะห์จากคำสั่ง acctcon1
logmmdd	เหมือนกับ บันทึก หลังจากคำสั่ง runacct ตรวจพบ ข้อผิดพลาด
reboots	มีวันที่เริ่มต้นและวันที่สิ้นสุดจาก wtmp และรายการของรีสตาร์ทระบบ
statefile	ใช้เพื่อบันทึกสถานะปัจจุบันในระหว่างการดำเนินการคำสั่ง runacct
tmpwtmp	ไฟล์ wtmp ที่แก้ไขโดยคำสั่ง wtmpfix
wtmperror	มีข้อความแสดงข้อผิดพลาด wtmpfix
wtmperrmmdd	เหมือนกับ wtmperror หลังจากคำสั่ง runacct ตรวจ พบข้อผิดพลาด
wtmp.mmdd	มีไฟล์ wtmp ของวันก่อนหน้านี้ ไฟล์นี้ถูกลบออกในระหว่างการทำความสะอาดของคำสั่ง runacct

ไฟล์ในไดเรกทอรี /var/adm/acct/sum(x):

ไฟล์ต่อไปนี้อยู่ในไดเรกทอรี /var/adm/acct/sum(x)

ไอเท็ม	คำอธิบาย
cms	ไฟล์ข้อมูลสรุปคำสั่งทั้งหมดสำหรับรอบระยะเวลาประมาณปัจจุบัน ในรูปแบบไบนารี
cmsprev	ไฟล์ข้อมูลสรุปคำสั่งที่ไม่มีอัปเดตล่าสุด
daycms	ไฟล์ข้อมูลสรุปคำสั่งสำหรับวันก่อนหน้านี้นี้ในรูปแบบไบนารี
lastlogin	ไฟล์ที่สร้างขึ้นโดยคำสั่ง lastlogin
pacct.mmdd	เวอร์ชันที่ต่อกันของไฟล์ pacct ทั้งหมดสำหรับ mmdd ไฟล์นี้ถูกลบออกหลังจากสแตทท์พระบอบโดยคำสั่ง remove สำหรับข้อมูลเกี่ยวกับคำสั่ง remove ให้อูที่ remove
rprtmmdd	เอาต์พุตที่บันทึกไว้ของคำสั่ง prdaily
tacct	ไฟล์การบัญชีทั้งหมดสะสมสำหรับรอบระยะเวลาประมาณปัจจุบัน
tacctprev	เหมือนกับ tacct ที่ไม่มีอัปเดตล่าสุด
tacctmmdd	ไฟล์การบัญชีทั้งหมดสำหรับ mmdd

ไฟล์ในไดเรกทอรี /var/adm/acct/fiscal(x):

ไฟล์ต่อไปนี้อยู่ในไดเรกทอรี /var/adm/acct/fiscal(x)

ไอเท็ม	คำอธิบาย
cms?	ไฟล์ข้อมูลสรุปคำสั่งทั้งหมดสำหรับรอบระยะเวลาประมาณที่ระบุโดย ? ในรูปแบบไบนารี
fiscrpt?	รายงานคล้ายกับรายงานของคำสั่ง prdaily สำหรับรอบระยะเวลาประมาณที่ระบุโดย ? ในรูปแบบไบนารี
tacct?	ไฟล์การบัญชีทั้งหมดสำหรับรอบระยะเวลาประมาณที่ระบุโดย ? ในรูปแบบไบนารี

รูปแบบไฟล์การบัญชี:

ตารางต่อไปนี้นำสรุปเอาต์พุตและรูปแบบของไฟล์การบัญชี

ไอเท็ม	คำอธิบาย
wtmp	จัดทำไฟล์บัญชีกระบวนการที่ใช้งานอยู่ รูปแบบของไฟล์ wtmp มีการกำหนดไว้ในไฟล์ utmp.h สำหรับข้อมูลเกี่ยวกับไฟล์ utmp.h ให้อยู่ที่ utmp.h
ctmp	จัดทำเรกคอร์ดเซสชันเชื่อมต่อ รูปแบบมีการอธิบายอยู่ในไฟล์ ctmp.h
pacct*	จัดทำเรกคอร์ดบัญชีกระบวนการที่ใช้งานอยู่ รูปแบบของเอาต์พุต มีการกำหนดไว้ในไฟล์ /usr/include/sys/acct.h
Spacct*	จัดทำไฟล์บัญชีกระบวนการสำหรับ mmdf ในระหว่าง การรันคำสั่ง runacct รูปแบบของไฟล์เหล่านี้ มีการกำหนดไว้ในไฟล์ sys/acct.h
dayacct	จัดทำเรกคอร์ดการบัญชีทั้งหมดสำหรับหนึ่งวัน รูปแบบของไฟล์ มีการกำหนดไว้ในรูปแบบไฟล์ tacct
sum/tacct	จัดทำไฟล์ไบนารีที่สะสมข้อมูลสรุปคำสั่งของแต่ละวัน รูปแบบของไฟล์นี้มีการกำหนดไว้ในไฟล์ส่วนหัว /usr/include/sys/acct.h
ptacct	จัดทำเวอร์ชันที่ต่อกันของไฟล์ pacct รูปแบบของไฟล์เหล่านี้มีการกำหนดไว้ในไฟล์ tacct
ctacct	จัดทำเรกคอร์ดการบัญชีทั้งหมดของการเชื่อมต่อ เอาต์พุตของไฟล์นี้ มีการกำหนดไว้ในไฟล์ tacct
cms	จัดทำข้อมูลสรุปคำสั่งการบัญชีทั้งหมดที่ใช้โดยคำสั่ง prdaily ในรูปแบบไบนารี เวอร์ชัน ASCII คือ nite/cms
daycms	ข้อมูลสรุปคำสั่งรายวันที่ใช้โดยคำสั่ง prdaily ในรูปแบบไบนารี เวอร์ชัน ASCII คือ nite/daycms

การจัดการบัญชีระบบ

มีภารกิจหลายอย่างซึ่งคุณสามารถทำให้เสร็จสมบูรณ์ได้สำหรับบัญชีระบบ ภารกิจเหล่านี้รวมถึง การตั้งค่าระบบการบัญชี การแสดงการใช้ CPU และการแสดงโปรเซสการบัญชี

การตั้งค่าระบบการบัญชี:

คุณสามารถตั้งค่าระบบการบัญชี

คุณต้องมีสิทธิ์การใช้งานรากเพื่อทำโปรซีเดอร์นี้ให้สมบูรณ์

ข้อมูลข้างล่างเป็นภาพรวมของขั้นตอนซึ่งคุณต้องทำ เพื่อตั้งค่าระบบการบัญชี โปรดอ้างอิงคำสั่งและไฟล์ที่ระบุในขั้นตอนเหล่านี้เพื่อดูข้อมูลเพิ่มเติม

1. ใช้คำสั่ง `nulladm` เพื่อให้แน่ใจว่าแต่ละไฟล์มีสิทธิ์การเข้าถึงที่ถูกต้อง: สิทธิ์การอ่าน (r) และเขียน (w) สำหรับเจ้าของไฟล์ และกลุ่ม และสิทธิ์การอ่าน (r) สำหรับบุคคลอื่น โดยพิมพ์:

```
/usr/sbin/acct/nulladm wtmp pacct
```

คำสั่งนี้ให้สิทธิ์เข้าถึงไฟล์ `pacct` และ `wtmp`

2. อัปเดตไฟล์ `/etc/acct/holidays` เพื่อรวมจำนวนชั่วโมงซึ่งคุณกำหนดเป็นเวลาหลัก และเพื่อสะท้อนถึงตารางเวลาวันหยุด ประจำปีของคุณ

หมายเหตุ: บรรทัดข้อคิดเห็นสามารถปรากฏขึ้นในตำแหน่งใดก็ได้ในไฟล์ ตรวจจับที่อักขระแรกในบรรทัดคือเครื่องหมายดอกจัน (*)

- a. เมื่อต้องการกำหนดเวลาหลัก ให้กรอกข้อมูลในฟิลด์ต่างๆ บนบรรทัดข้อมูลแรก (บรรทัดแรกที่ไม่ใช่ข้อคิดเห็น) โดยใช้หน้าพีกา 24-ชั่วโมง บรรทัดนี้ประกอบด้วย ฟิลด์ 4-ตำแหน่งสามฟิลด์ ในลำดับต่อไปนี้:

- 1) Current year
- 2) จุดเริ่มต้นของเวลาหลัก (hhmm)
- 3) จุดสิ้นสุดของเวลาหลัก (hhmm)

โดยละเว้นพื้นที่ว่างเปล่า นำหน้า คุณสามารถป้อนเวลาเที่ยงคืนเป็น 0000 หรือ 2400 อย่างใดอย่างหนึ่ง ตัวอย่างเช่น เมื่อต้องการระบุปี 2000 ที่มีเวลาหลักเริ่มต้น ในเวลา 8:00 a.m. และสิ้นสุดในเวลา 5:00 p.m. ให้ป้อน:

```
2000 0800 1700
```

- b. เมื่อต้องการกำหนดวันหยุดของบริษัทประจำปี ให้กรอกข้อมูลในบรรทัดข้อมูลถัดไป แต่ละบรรทัดมีสี่ฟิลด์ในลำดับต่อไปนี้:

- 1) วันของปี
- 2) เดือน
- 3) วันของเดือน
- 4) คำอธิบายของวันหยุด

ฟิลด์วันของปีประกอบด้วยตัวเลขของวันซึ่งเป็นวันหยุด และต้องเป็นตัวเลขตั้งแต่ 1 ถึง 365 (366 ในปีอธิกสุรทิน) ตัวอย่างเช่น วันที่ 1 กุมภาพันธ์คือวัน 32 อีกสามฟิลด์คือฟิลด์สำหรับข้อมูล เท่านั้น และถือว่าเป็นข้อคิดเห็น ตัวอย่างสองบรรทัดมีดังนี้:

```
1 Jan 1 New Year's Day
332 Nov 28 Thanksgiving Day
```

3. เปิดบัญชีกระบวนการโดยการเพิ่มบรรทัดต่อไปนี้ลงในไฟล์ /etc/rc หรือโดยการลบสัญลักษณ์ข้อคิดเห็น (#) ที่ด้านหน้าของบรรทัดออกถ้ามีอยู่:

```
/usr/bin/su - adm -c /usr/sbin/acct/startup
```

โปรดใช้สตาร์ทอัพ บันทึกเวลาที่เปิดการบัญชี และทำความสะอาดไฟล์การบัญชีของวันก่อนหน้า

4. ระบุแต่ละระบบไฟล์ซึ่งคุณต้องการรวมไว้ในการบัญชีดิสก์ โดยการเพิ่มบรรทัดต่อไปนี้ลงใน stanza ของระบบไฟล์ในไฟล์ /etc/filesystems:

```
account = true
```

5. ระบุไฟล์ข้อมูลที่จะใช้สำหรับข้อมูลเครื่องพิมพ์โดยการเพิ่มบรรทัดต่อไปนี้ลงใน stanza คิวในไฟล์ /etc/qconfig:

```
acctfile = /var/adm/qacct
```

6. ในฐานะผู้ใช้ adm ให้สร้างไดเรกทอรี /var/adm/acct/nite, /var/adm/acct/fiscal, และ /var/adm/acct/sum เพื่อรวบรวมเรีกคอร์ดประจำวันและรอบระยะเวลางบประมาณ:

```
su - adm
cd /var/adm/acct
mkdir nite fiscal sum
exit
```

สำหรับชื่อผู้ที่ยาว ให้ใช้คำสั่งต่อไปนี้แทน:

```
su - adm
cd /var/adm/acct
mkdir nitex fiscalx sumx
exit
```

7. ตั้งค่าโพรซีเจอร์การบัญชีประจำวันเพื่อรันโดยอัตโนมัติโดยการแก้ไขไฟล์ /var/spool/cron/crontabs/adm เพื่อรวมคำสั่ง **dodisk**, **ckpacct**, และ **runacct** ตัวอย่างเช่น:

```
0 2 * * 4 /usr/sbin/acct/dodisk
5 * * * * /usr/sbin/acct/ckpacct
0 4 * * 1-6 /usr/sbin/acct/runacct
                2>/var/adm/acct/nite/accterr
```

สำหรับชื่อผู้ที่ยาว ให้เพิ่มบรรทัดต่อไปนี้แทน:

```
0 2 * * 4 /usr/sbin/acct/dodisk -X
5 * * * * /usr/sbin/acct/ckpacct
0 4 * * 1-6 /usr/sbin/acct/runacct -X
2>/var/adm/acct/nitex/accterr
```

บรรทัดแรก เริ่มต้นการบัญชีดิสก์ในเวลา 2:00 a.m. (0 2) แต่ละวันพฤหัสบดี (4) บรรทัดที่สอง เริ่มต้นการตรวจสอบ
คุณภาพของไฟล์ข้อมูลที่ใช้งานอยู่ เมื่อผ่านไป 5 นาทีในแต่ละชั่วโมง (5 *) ทุกวัน (*) บรรทัดที่สามรันโปรแกรมการ
บัญชีส่วนใหญ่ และประมวลผล ไฟล์ข้อมูลที่ใช้งานอยู่ในเวลา 4:00 a.m. (0 4) ทุกวันจันทร์ถึง วันเสาร์ (1-6) หากเวลา
เหล่านี้ไม่เหมาะสมกับชั่วโมงที่ระบบของคุณ ดำเนินงาน ให้ปรับรายการของคุณ

หมายเหตุ: คุณต้องมีสิทธิ์ผู้ใช้ราก เพื่อแก้ไขไฟล์ /var/spool/cron/crontabs/adm

8. ตั้งค่าสรุปการบัญชีรายเดือนเพื่อรันโดยอัตโนมัติโดยการรวม คำสั่ง **monacct** ไว้ในไฟล์ /var/spool/cron/crontabs/
adm ตัวอย่าง เช่น พิมพ์:

```
15 5 1 * * /usr/sbin/acct/monacct
```

สำหรับชื่อผู้ใช้ที่ยาว ให้เพิ่มบรรทัดต่อไปนี้แทน:

```
15 5 1 * * /usr/sbin/acct/monacct -X
```

ต้องแน่ใจว่า จัดตารางเวลาโปรแกรมนี้ล่วงหน้าเพียงพอให้รายงานเสร็จสิ้นได้ ตัวอย่างนี้เริ่มต้นโปรแกรมในเวลา 5:15
a.m. ในวันแรกของแต่ละเดือน

9. เมื่อต้องการส่งไฟล์ cron ที่แก้ไข ให้พิมพ์:

```
crontab /var/spool/cron/crontabs/adm
```

หลักการที่เกี่ยวข้อง:

“คำสั่งสำหรับการทำความสะอาดระบบไฟล์โดยอัตโนมัติ” ในหน้า 409

ใช้คำสั่ง **skulker** เพื่อทำความสะอาดระบบไฟล์โดยการลบไฟล์ที่ไม่ต้องการออก

“การรวบรวมข้อมูลระบบและการรายงาน” ในหน้า 194

คุณสามารถตั้งค่าระบบเพื่อรวบรวมข้อมูลและสร้าง รายงานโดยอัตโนมัติ

“รายงานกิจกรรมระบบการบัญชี” ในหน้า 179

คุณสามารถสร้างรายงานที่แสดงกิจกรรมของระบบการบัญชี

“สรุปรายงานการบัญชี” ในหน้า 177

คุณสามารถสร้างรายงานที่สรุปข้อมูลการบัญชีดิบ

งานที่เกี่ยวข้อง:

“การจำกัดผู้ใช้จากบางไดเร็กทอรี” ในหน้า 409

คุณสามารถรีเซ็ตพื้นที่ว่างดิสก์และอาจปล่อยให้ว่างไว้ โดยการจำกัด สิทธิเข้าถึงในไดเร็กทอรีและการมอนิเตอร์การใช้ดิสก์

“การแก้ไขระบบไฟล์ที่โอเวอร์โฟลว์และกำหนดโดยผู้ใช้” ในหน้า 486

ใช้โปรแกรมนี้เพื่อแก้ไขระบบไฟล์ที่โอเวอร์โฟลว์และกำหนดโดยผู้ใช้

“การแสดงผลประมวลผลของโปรเซสการบัญชีที่ใช้งานอยู่” ในหน้า 190

คุณสามารถแสดงผลประมวลผลของโปรเซสที่ใช้งานอยู่

“การแสดงผลประมวลผลของโปรเซสการบัญชีที่เสร็จสิ้น” ในหน้า 190

คุณสามารถแสดงผลประมวลผลของโปรเซสที่เสร็จสิ้นแล้ว

“การแสดงผลการใช้ CPU สำหรับแต่ละโปรเซสการบัญชี” ในหน้า 191

คุณสามารถแสดงรายงานที่จัดรูปแบบเกี่ยวกับการใช้ CPU โดยโปรเซส โดยใช้คำสั่ง **acctprc1**

“การแสดงผลการใช้การบัญชี CPU สำหรับผู้ใช้แต่ละราย” ในหน้า 191
คุณสามารถแสดงรายงานที่จัดรูปแบบเกี่ยวกับการใช้ CPU โดยผู้ใช้โดยใช้ชุดของคำสั่ง `acctprc1` และ `prtacct`

“การแสดงเรียกคอร์ตการบัญชีการใช้เครื่องพิมพ์หรือพล็อตเตอร์” ในหน้า 193
คุณสามารถแสดงเรียกคอร์ตการบัญชีการใช้เครื่องพิมพ์หรือพล็อตเตอร์ได้โดยใช้ คำสั่ง `pac`
สิ่งอ้างอิงที่เกี่ยวข้อง:

“รายงานการบัญชีและไฟล์ข้อมูลสรุป” ในหน้า 182
ต้องการบางได้เรียกทอเรีย้อยก่อนคุณสามารถใช้งานระบบการบัญชีได้

การแสดงผลกิจกรรมระบบการบัญชี:

คุณสามารถแสดงข้อมูลที่จัดรูปแบบเกี่ยวกับกิจกรรมระบบได้โดยใช้ คำสั่ง `sar`

เมื่อต้องการแสดงสถิติกิจกรรมระบบ คำสั่ง `sadc` ต้องกำลังรันอยู่

หมายเหตุ: วิธีการปกติในการรันคำสั่ง `sadc` คือการวางรายการของคำสั่ง `sa1` ไว้ในไฟล์ `crontab` ราก คำสั่ง `sa1` คือตัวแปรเชลล์โพรซีเตอร์ของคำสั่ง `sadc` ที่ออกแบบมาเพื่อทำงานกับ `cron daemon`

เมื่อต้องการแสดงข้อมูลกิจกรรมระบบพื้นฐาน ให้พิมพ์:

```
sar 2 6
```

โดยที่ ตัวเลขแรกคือจำนวนวินาทีระหว่างช่วงเวลารวมตัวอย่าง และตัวเลขที่สองคือจำนวนช่วงเวลาที่จะแสดงขึ้น เอาต์พุตของคำสั่งนี้ คล้ายกับตัวอย่างนี้:

```
arthurd 2 3 000166021000 05/28/92
14:03:40 %usr %sys %wio %idle
14:03:42 4 9 0 88
14:03:43 1 10 0 89
14:03:44 1 11 0 88
14:03:45 1 11 0 88
14:03:46 3 9 0 88
14:03:47 2 10 0 88
Average 2 10 0 88
```

คำสั่ง `sar` ยัง นำเสนอแฟล็กจำนวนมากสำหรับการแสดงอาร์เรย์แบบต่างๆ ของสถิติ ระบบ เมื่อต้องการดูสถิติทั้งหมดที่มีอยู่ ให้ใช้แฟล็ก `-A` สำหรับ รายการของสถิติที่มีอยู่และแฟล็กสำหรับการแสดงรายการดังกล่าว ให้ดูที่คำสั่ง `sar`

หมายเหตุ: หาก ต้องการให้มีการเขียนรายงานกิจกรรมระบบรายวันลงใน `/var/adm/sa/sadd` ให้รวมรายการในไฟล์ราก `crontab` ของคำสั่ง `sa2` คำสั่ง `sa2` คือตัวแปรเชลล์โพรซีเตอร์ของคำสั่ง `sar` ที่ออกแบบมาเพื่อทำงานกับ `cron daemon`

การแสดงผลกิจกรรมระบบการบัญชีในขณะรันคำสั่ง:

คุณสามารถแสดงข้อมูลที่จัดรูปแบบเกี่ยวกับกิจกรรม ระบบในขณะที่คำสั่งเฉพาะกำลังรันอยู่

แฟล็ก `-o` และ `-p` ของคำสั่ง `timex` กำหนดว่าบัญชีระบบต้องเปิดอยู่

คุณสามารถใช้คำสั่ง `time` และ `timex` เพื่อแสดงข้อมูลที่จัดรูปแบบเกี่ยวกับกิจกรรมระบบในขณะที่คำสั่งเฉพาะกำลังรันอยู่

เมื่อต้องการ แสดงเวลาที่ผ่านไป เวลาผู้ใช้ และเวลาดำเนินการระบบของคำสั่ง เฉพาะ ให้พิมพ์:

```
time CommandName
```

หรือ

```
timex CommandName
```

เมื่อต้องการแสดงกิจกรรมระบบทั้งหมด (ไอเท็มข้อมูลทั้งหมดที่รายงาน โดยคำสั่ง sar) ในระหว่างการดำเนินการของคำสั่ง เฉพาะ ให้พิมพ์:

```
timex -s CommandName
```

คำสั่ง **timex** มีแฟล็กเพิ่มเติมสองแฟล็ก แฟล็ก **-o** รายงานจำนวนบล็อกทั้งหมด ที่อ่านหรือเขียนโดยคำสั่งและชาชนทั้งหมด แฟล็ก **-p** แสดง รายการเรียกคอร์ตบัญชีกระบวนการทั้งหมดสำหรับคำสั่งและชาชนทั้งหมด

การแสดงผลของโปรเซสการบัญชีที่ใช้งานอยู่:

คุณสามารถแสดงผลของโปรเซสที่ใช้งานอยู่

คำสั่ง **acctcom** อ่านอินพุตในรูปแบบเรียกคอร์ตการบัญชี ทั้งหมด (รูปแบบไฟล์ acct) คำสั่งนี้สมมติว่าคุณ มีบัญชีกระบวนการที่เปิดอยู่หรือคุณได้รับบัญชีกระบวนการแล้วในอดีต

คำสั่ง **ps** นำเสนอแฟล็กจำนวนมากเพื่อ ปรับแต่งข้อมูลที่แสดงขึ้น

เมื่อต้องการจัดทำรายการทั้งหมดของโปรเซสทั้งหมด ที่ใช้งานอยู่ ยกเว้นโปรเซสเคอร์เนล ให้พิมพ์:

```
ps -ef
```

คุณยังสามารถแสดงรายการของโปรเซสทั้งหมดซึ่งเชื่อมโยงกับเทอร์มินัลได้ด้วย เมื่อต้องการทำ เช่นนี้ ให้พิมพ์:

```
ps -al
```

การใช้งานทั้งสองอย่างเหล่านี้แสดง คอลัมน์จำนวนหนึ่งสำหรับแต่ละโปรเซส รวมถึงเวลา CPU ปัจจุบันของโปรเซสในหน่วยนาที่และวินาที

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

การแสดงผลของโปรเซสการบัญชีที่เสร็จสิ้น:

คุณสามารถแสดงผลของโปรเซสที่เสร็จสิ้นแล้ว

คำสั่ง **acctcom** อ่านอินพุตในรูปแบบเรียกคอร์ตการบัญชี ทั้งหมด (รูปแบบไฟล์ acct) คำสั่งนี้สมมติว่าคุณ มีบัญชีกระบวนการที่เปิดอยู่หรือคุณได้รับบัญชีกระบวนการแล้วในอดีต

ฟังก์ชันบัญชีกระบวนการเปิดด้วยคำสั่ง **startup** ซึ่งโดยปกติแล้ว เริ่มต้นขึ้นที่การเริ่มต้นระบบด้วยการเรียกในไฟล์ /etc/rc ในขณะที่ฟังก์ชันบัญชีกระบวนการกำลังรัน จะมีการเขียนเรียกคอร์ตลงใน /var/adm/pacct (ไฟล์ เรียกคอร์ตการบัญชีรวม) สำหรับทุกโปรเซสที่เสร็จสิ้นซึ่งรวมถึง เวลาเริ่มต้นและเวลาหยุดของโปรเซส คุณสามารถแสดงข้อมูลเวลาประมวลผลจากไฟล์ pacct ได้โดยใช้คำสั่ง **acctcom** คำสั่งนี้มีแฟล็กจำนวนมากซึ่งช่วยเพิ่มความยืดหยุ่นในการระบุ โปรเซสที่จะแสดง

ตัวอย่างเช่น เมื่อต้องการดูโปรเซสทั้งหมดที่รันนานเท่ากับ จำนวนวินาที CPU ต่ำสุดหรือมากกว่านั้น ให้ใช้แฟล็ก -O และ พิมพ์:

```
acctcom -O 2
```

คำสั่งนี้ แสดงเรีกคอร์ดของทุกโปรเซสที่รันเป็นเวลาอย่างน้อย 2 วินาที หากคุณ ไม่ได้ระบุไฟล์อินพุต คำสั่ง acctcom จะอ่าน อินพุตจากไดเรกทอรี /var/adm/pacct

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

การแสดงผลการใช้ CPU สำหรับแต่ละโปรเซสการบัญชี:

คุณสามารถแสดงรายงานที่จัดรูปแบบเกี่ยวกับการใช้ CPU โดยโปรเซสโดยใช้คำสั่ง acctprc1

คำสั่ง acctprc1 ต้องการอินพุตในรูปแบบเรีกคอร์ดการบัญชีทั้งหมด (รูปแบบไฟล์ acct) คำสั่งนี้สมมติว่าคุณมีบัญชีกระบวนการที่เปิดอยู่หรือคุณได้รับ บัญชีกระบวนการแล้วในอดีต

เมื่อต้องการจัดทำรายงานที่จัดรูปแบบเกี่ยวกับการใช้ CPU โดยโปรเซส ให้พิมพ์:

```
acctprc1 </var/adm/pacct
```

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

การแสดงผลการใช้การบัญชี CPU สำหรับผู้ใช้แต่ละราย:

คุณสามารถแสดงรายงานที่จัดรูปแบบเกี่ยวกับการใช้ CPU โดยผู้ใช้ โดยใช้ชุดของคำสั่ง acctprc1 และ prtacct

คำสั่ง ../com.ibm.aix.cmds1/acctprc1.htm ต้องการอินพุตในรูปแบบเรีกคอร์ดการบัญชีทั้งหมด (รูปแบบไฟล์ acct) คำสั่งนี้สมมติว่าคุณ มีบัญชีกระบวนการที่เปิดอยู่หรือคุณได้รับบัญชีกระบวนการแล้วในอดีต

เมื่อต้องการแสดงผลการใช้ CPU สำหรับผู้ใช้แต่ละราย ให้ทำขั้นตอนต่อไปนี้:

1. จัดทำไฟล์เอาต์พุตของการใช้ CPU โดยเรียงตามโปรเซสโดยพิมพ์:

```
acctprc1 </var/adm/pacct >out.file
```

ไฟล์ /var/adm/pacct เป็นดีฟอลต์เอาต์พุตสำหรับ เรีกคอร์ดบัญชีกระบวนการ คุณอาจต้องการระบุไฟล์ pacct ที่เก็บถาวรแทน

2. จัดทำไฟล์เรีกคอร์ดการบัญชีทั้งหมดในรูปแบบไบนารีจากเอาต์พุตของ ขั้นตอนก่อนหน้านี้นี้โดยพิมพ์:

```
acctprc2 <out.file >/var/adm/acct/nite/daytacct
```

หมายเหตุ: ไฟล์ daytacct ถูกผสมเข้ากับเรีกคอร์ดการบัญชีทั้งหมดอื่นๆ โดยใช้คำสั่ง acctmerge เพื่อจัดทำเรีกคอร์ดสรุปรายวัน /var/adm/acct/sum(x)/tacct

3. ใช้คำสั่ง ../com.ibm.aix.cmds4/prtacct.htm เพื่อแสดงรายงานที่จัดรูปแบบของการใช้ CPU ซึ่งสรุปตามผู้ใช้โดยพิมพ์:

```
prtacct </var/adm/acct/nite/daytacct
```

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186
คุณสามารถตั้งค่าระบบการบัญชี

การแสดงการใช้เวลาเชื่อมต่อสำหรับการบัญชี:

คุณสามารถแสดงเวลาเชื่อมต่อของผู้ใช้ทั้งหมด ของผู้ใช้แต่ละราย และของการล็อกอินแต่ละรายการได้โดยใช้คำสั่ง `ac`

เนื่องจากคำสั่ง `ac` ดึงข้อมูลล็อกอินมาจากไฟล์ `/var/adm/wtmp` ดังนั้นไฟล์นี้จึงต้องมีอยู่ หากยังไม่ได้สร้างไฟล์ จะมีข้อความแสดงความผิดพลาดต่อไปนี้ส่งคืน:

```
No /var/adm/wtmp
```

หากไฟล์เต็มเกินไปไฟล์ `wtmp` เพิ่มเติมจะถูกสร้างขึ้น คุณสามารถแสดงข้อมูลเวลาเชื่อมต่อจากไฟล์เหล่านี้ได้โดยระบุไฟล์พร้อมกับแฟล็ก `-w` สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `ac` ให้ดูที่ `ac`

เมื่อต้องการแสดงเวลาเชื่อมต่อทั้งหมดของผู้ใช้ทั้งหมด ให้พิมพ์:

```
/usr/sbin/acct/ac
```

คำสั่งนี้แสดงตัวเลขฐานสิบตัวเดียวที่เป็นผลรวมเวลาเชื่อมต่อทั้งหมด ในหน่วยนาที ของผู้ใช้ทุกรายที่ล็อกอินในระหว่างอายุการใช้งานของไฟล์ `wtmp` ปัจจุบัน

เมื่อ ต้องการแสดงเวลาเชื่อมต่อทั้งหมดของผู้ใช้เฉพาะตั้งแต่หนึ่งรายขึ้นไป ให้พิมพ์:

```
/usr/sbin/acct/ac User1 User2 ...
```

คำสั่งนี้แสดงตัวเลขฐานสิบตัวเดียวที่เป็นผลรวมเวลาเชื่อมต่อทั้งหมด ในหน่วยนาที ของผู้ใช้ซึ่งคุณระบุสำหรับ ล็อกอินใดๆ ในระหว่างอายุการใช้งานของไฟล์ `wtmp` ปัจจุบัน

เมื่อต้องการ แสดงเวลาเชื่อมต่อของผู้ใช้แต่ละรายบวกเวลาเชื่อมต่อทั้งหมด ให้พิมพ์:

```
/usr/sbin/acct/ac -p User1 User2 ...
```

คำสั่งนี้แสดงตัวเลขฐานสิบตัวเดียวสำหรับผู้ใช้แต่ละรายที่ระบุ ซึ่งเท่ากับเวลาเชื่อมต่อทั้งหมดในหน่วยนาที ของใช้นั้นในระหว่างอายุการใช้งานของไฟล์ `wtmp` ปัจจุบัน และยังแสดงตัวเลขฐานสิบที่เป็นผลรวมเวลาเชื่อมต่อทั้งหมดของ ผู้ใช้ทั้งหมดที่ระบุด้วย ถ้าไม่ระบุผู้ใช้ในคำสั่ง รายการจะรวม ผู้ใช้ทุกรายที่ล็อกอินในระหว่างอายุการใช้งานของไฟล์ `wtmp`

การแสดงผลการใช้พื้นที่ว่างดิสก์สำหรับการบัญชี:

คุณสามารถแสดงข้อมูลการใช้พื้นที่ว่างดิสก์โดยใช้คำสั่ง `acctmrg`

เพื่อแสดงข้อมูลการใช้พื้นที่ว่างดิสก์ คำสั่ง `acctmrg` ต้องการอินพุตจากไฟล์ `dacct` (การบัญชีดิสก์) การรวบรวม เรียกว่าการบัญชีการใช้ดิสก์มีการทำโดยคำสั่ง `dodisk`

เมื่อต้องการแสดงข้อมูลการใช้พื้นที่ว่างดิสก์ ให้พิมพ์:

```
acctmrg -a1 -2,13 -h </var/adm/acct/nite(x)/dacct
```

คำสั่งนี้แสดงเรียกคอร์ตการบัญชีดิสก์ ซึ่งมีจำนวนของบล็อกขนาด 1 KB ที่ใช้โดยผู้ใช้แต่ละราย

หมายเหตุ: คำสั่ง `acctmerge` อ่านจากอินพุตมาตรฐานเสมอ และสามารถอ่านไฟล์เพิ่มเติมได้สูงสุดเก้าไฟล์ หากคุณไม่ได้พิมพ์อินพุตกับคำสั่ง คุณต้องเปลี่ยนทิศทางอินพุตจากไฟล์หนึ่ง ส่วนไฟล์ที่เหลือสามารถระบุได้โดยไม่ต้องเปลี่ยนทิศทาง

การแสดงเรีกคอร์ดการบัญชีการใช้เครื่องพิมพ์หรือพล็อตเตอร์:

คุณสามารถแสดงเรีกคอร์ดการบัญชีการใช้เครื่องพิมพ์หรือพล็อตเตอร์ได้โดยใช้ คำสั่ง `pac`

- เพื่อรวบรวมข้อมูลการใช้เครื่องพิมพ์ คุณต้องมีระบบการบัญชีที่ตั้งค่า และกำลังรัน โปรดดู “การตั้งค่าระบบการบัญชี” ในหน้า 186 สำหรับแนวทาง
- เครื่องพิมพ์หรือพล็อตเตอร์ซึ่งคุณต้องการเรีกคอร์ดการบัญชีต้องมี ส่วนคำสั่ง `acctfile=` อยู่ใน stanza เครื่องพิมพ์ของไฟล์ `/etc/qconfig` ไฟล์ที่ระบุในส่วนคำสั่ง `acctfile=` ต้องให้สิทธิ์การอ่าน และการเขียนแก่ผู้ใช้รากหรือกลุ่ม `printq`
- หากมีการระบุแฟล็ก `-s` ของคำสั่ง `pac` คำสั่งจะเขียน ชื่อไฟล์สรุปอีกครั้งโดยการผนวก `_sum` เข้ากับชื่อพาร์ที่ระบุโดยส่วนคำสั่ง `acctfile=` ในไฟล์ `/etc/qconfig` ไฟล์นี้ต้องมีอยู่และมอบสิทธิ์การอ่านและการเขียนให้แก่ผู้ใช้ราก หรือกลุ่ม `printq`

เมื่อต้องการแสดงข้อมูลการใช้เครื่องพิมพ์สำหรับผู้ใช้ทั้งหมดของเครื่องพิมพ์ เฉพาะให้พิมพ์:

```
/usr/sbin/pac -PPrinter
```

หากคุณไม่ได้ระบุเครื่องพิมพ์ เครื่องพิมพ์ดีฟอลต์จะมีการระบุโดยตัวแปรสถานะแวดล้อม `PRINTER` ถ้าไม่มีการกำหนดตัวแปร `PRINTER` ค่าดีฟอลต์คือ `lp0`

เมื่อต้องการแสดงข้อมูลการใช้เครื่องพิมพ์ สำหรับผู้ใช้เฉพาะรายของเครื่องพิมพ์เฉพาะให้พิมพ์:

```
/usr/sbin/pac -PPrinter User1 User2 ...
```

คำสั่ง `pac` นำเสนอแฟล็กอื่นสำหรับการควบคุมข้อมูลที่จะแสดงขึ้น

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

การอัปเดตไฟล์วันหยุด:

ไฟล์วันหยุดกลายเป็นล้าสมัยหลังผ่านพ้นวันหยุดสุดท้ายที่แสดงรายการ หรือปีเปลี่ยนแปลง คุณสามารถอัปเดตไฟล์วันหยุดได้

คำสั่ง `acctcon1` (เริ่มต้นจากคำสั่ง `runacct`) ส่งเมลไปยังแอดเคาต์ ราก และ `adm` เมื่อไฟล์ `/usr/lib/acct/holidays` ล้าสมัย

อัปเดตไฟล์วันหยุดที่ล้าสมัยโดยการแก้ไขไฟล์ `/var/adm/acct/holidays` เพื่อให้แตกต่างกันระหว่างเวลาที่ใช้งานสูงสุด และเวลาที่ใช้งานไม่มากนัก

เวลาที่ใช้งานสูงสุดมีการสมมติว่าเป็นรอบระยะเวลาที่ระบบของคุณมีการใช้งาน มากที่สุด เช่น วันทำงาน วันเสาร์และวันอาทิตย์เป็นเวลาที่ใช้งานไม่มากนักเสมอสำหรับ ระบบการบัญชี เนื่องจากเป็นวันหยุดซึ่งคุณแสดงรายการ

ไฟล์วันหยุดประกอบด้วยรายการสามชนิดคือ: ข้อคิดเห็น ปี และ รอบเวลาที่ใช้งานสูงสุด และรายการของวันหยุดดังเช่นในตัวอย่างต่อไปนี้:

* Prime/Non-Prime Time Table for Accounting System

*

* Curr	Prime	Non-Prime
* Year	Start	Start
1992	0830	1700

*

* Day of	Calendar	Company
* Year	Date	Holiday

*

* 1	Jan 1	New Year's Day
* 20	Jan 20	Martin Luther King Day
* 46	Feb 15	President's Day
* 143	May 28	Memorial Day
* 186	Jul 3	4th of July
* 248	Sep 7	Labor Day
* 329	Nov 24	Thanksgiving
* 330	Nov 25	Friday after
* 359	Dec 24	Christmas Eve
* 360	Dec 25	Christmas Day
* 361	Dec 26	Day after Christmas

บรรทัดแรกที่ไม่ใช่ข้อคิดเห็นต้องระบุปีปัจจุบัน (เป็นสี่ตำแหน่ง) และจุดเริ่มต้นและจุดสิ้นสุดของเวลาที่ใช้งานสูงสุด ซึ่งแต่ละค่าต้องเป็นสี่ตำแหน่งเช่นกัน หลักการของเวลาที่ใช้งานสูงสุดและเวลาที่ใช้งานไม่มากก็นักมีผลกระทบต่อวิธีการที่โปรแกรมการบัญชีประมวลผลเรียกคอร์ตการบัญชีเท่านั้น

หากรายการวันหยุดยาวเกินไป คำสั่ง `acctcon1` จะสร้างข้อผิดพลาด และคุณจะต้องตัดรายการให้สั้นลง จำนวนวันหยุดที่ปลอดภัยคือ 20 วันหรือน้อยกว่านั้น ถ้าคุณต้องการเพิ่มวันหยุดเพิ่มเติมให้เพียงแต่แก้ไขไฟล์วันหยุด สำหรับแต่ละเดือน

การรวบรวมข้อมูลการบัญชี

หลังจากคุณตั้งค่าบัญชีระบบแล้ว คุณจะต้องการเริ่มต้นรวบรวม และประมวลผลข้อมูลการบัญชีชนิดอื่น

การรวบรวมข้อมูลระบบและการรายงาน:

คุณสามารถตั้งค่าระบบเพื่อรวบรวมข้อมูลและสร้างรายงานโดยอัตโนมัติ

สำหรับข้อมูลที่จะรวบรวมโดยอัตโนมัติ สมาชิกของ กลุ่ม `adm` ต้องมีการตั้งค่าเป็นระบบการบัญชี การตั้งค่า ระบบการบัญชีช่วยให้ `cron daemon` สามารถรัน คำสั่งที่สร้างข้อมูลเกี่ยวกับ:

- จำนวนเวลาที่ผู้ใช้แต่ละรายใช้ล็อกอินเข้าสู่ระบบ
- การใช้หน่วยการประมวลผล หน่วยความจำ และรีซอร์ส I/O
- จำนวนพื้นที่ว่างที่ใช้โดยไฟล์ของผู้ใช้แต่ละราย
- การใช้เครื่องพิมพ์และพล็อตเตอร์
- จำนวนครั้งที่ให้คำสั่งเฉพาะ

ระบบเขียนเรียกคอร์ตของแต่ละเซสชันและโปรเซส หลังจากทีเซสชันและโปรเซสเสร็จสมบูรณ์แล้ว เรียกคอร์ตเหล่านี้มีการแปลงเป็นเรียกคอร์ตการบัญชีทั้งหมด (`tacct`) ที่จัดเรียงตามผู้ใช้และถูกรวมเข้าในรายงาน รายวัน รายงานรายวันจะถูกรวมเข้าด้วยกันเป็นครั้งคราวเพื่อจัดทำผลรวมสำหรับ รอบระยะเวลาบางประมาณที่กำหนด เมธอดการรวบรวมและรายงานข้อมูลและ คำสั่งการบัญชีและไฟล์ต่างๆ มีการอธิบายไว้ในส่วนภายหลัง

แม้ว่าข้อมูลการบัญชีส่วนใหญ่มีการรวบรวมและประมวลผล โดยอัตโนมัติ แต่สมาชิกของกลุ่ม adm สามารถป้อนบางคำสั่ง จากคีย์บอร์ดเพื่อรับข้อมูลเฉพาะได้

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

สิ่งอ้างอิงที่เกี่ยวข้อง:

“คำสั่งคีย์บอร์ด” ในหน้า 181

สมาชิกของกลุ่ม adm สามารถป้อนคำสั่งต่อไปนี้จาก คีย์บอร์ดได้

ข้อมูลการบัญชีเวลาเชื่อมต่อ:

ข้อมูลเวลาเชื่อมต่อมีการรวบรวมโดยคำสั่ง `init` และคำสั่ง `login`

เมื่อคุณล็อกอิน โปรแกรม `login` จะเขียนเรกคอร์ดในไฟล์ `/etc/utmp` เรกคอร์ดนี้มี ชื่อผู้ใช้ของคุณ วันที่และเวลาของล็อกอิน และล็อกอินพอร์ต คำสั่ง เช่น `who` ใช้ไฟล์นี้เพื่อค้นหาผู้ใช้ที่ล็อกอินเข้าในสถานีแสดงผลต่างๆ ถ้ามีไฟล์การบัญชีเวลาเชื่อมต่อ `/var/adm/wtmp` คำสั่ง `login` จะเพิ่มสำเนาของ เรกคอร์ดล็อกอินนี้ลงในไฟล์ สำหรับข้อมูลเกี่ยวกับคำสั่ง `init` และ `login` ให้ดูที่ `init` และ `login`

เมื่อโปรแกรม `login` ของคุณสิ้นสุดลง (โดยปกติ เมื่อคุณล็อกเอาต์) คำสั่ง `init` จะบันทึกการสิ้นสุดของเซสชันโดยการเขียนเรกคอร์ดอื่นในไฟล์ `/var/adm/wtmp` เรกคอร์ด ล็อกเอาต์แตกต่างจากเรกคอร์ดล็อกอินตรงที่ไม่มีชื่อผู้ใช้ ทั้งเรกคอร์ด ล็อกอิน และล็อกเอาต์มีรูปแบบดังอธิบายไว้ในไฟล์ `utmp.h` สำหรับข้อมูลเกี่ยวกับไฟล์ `utmp.h` ให้ดูที่ `utmp.h`

คำสั่ง `acctwtmp` ยังเขียน รายการพิเศษในไฟล์ `/var/adm/wtmp` ซึ่งเกี่ยวข้องกับ การปิดและสตาร์ทอ็อประบบด้วย

หลักการที่เกี่ยวข้อง:

“รายงานเวลาเชื่อมต่อ” ในหน้า 178

เรกคอร์ดการบัญชีมีเรกคอร์ดล็อกอิน ล็อกเอาต์ การปิดระบบ และ เรกคอร์ดล็อกอินครั้งล่าสุด

ข้อมูลบัญชีกระบวนการ:

ระบบการบัญชีรวบรวมข้อมูลเกี่ยวกับการใช้รีซอร์สของแต่ละโปรเซสในขณะที่โปรเซสรัน

ข้อมูลนี้รวมถึง:

- หมายเลขผู้ใช้และกลุ่มซึ่งโปรเซสรันอยู่ภายใต้มัน
- อักขระแปดตัวแรกของชื่อคำสั่ง
- คีย์ตัวเลข 64-บิตที่แสดงถึง Workload Manager คลาสซึ่ง โปรเซสเป็นสมาชิก
- เวลาที่ผ่านไปและเวลาตัวประมวลผลที่ใช้โดยโปรเซส
- การใช้หน่วยความจำ
- จำนวนอักขระที่โอนย้าย
- จำนวนของดิสก์บล็อกที่อ่านหรือเขียนในนามของโปรเซส

คำสั่ง `accton` บันทึกข้อมูลเหล่านี้ไว้ในไฟล์ที่ระบุ ซึ่งโดยปกติแล้วคือไฟล์ `/var/adm/pacct` สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `accton` ให้ดูที่ `accton`

คำสั่งที่เกี่ยวข้องคือคำสั่ง `startup`, คำสั่ง `shutacct`, คำสั่ง `dodisk`, คำสั่ง `ckpacct`, และคำสั่ง `turnacct` สำหรับข้อมูลเกี่ยวกับคำสั่งเหล่านี้ให้ดูที่ `startup`, `shutacct`, `dodisk`, `ckpacct`, และ `turnacct`

หลักการที่เกี่ยวข้อง:

“รายงานข้อมูลการบัญชี” ในหน้า 174

หลังจากรวบรวมข้อมูลการบัญชีชนิดต่างๆ แล้วเรีกคอร์ด จะถูกประมวลผลและแปลงเป็นรายงาน

รายงานบัญชีกระบวนการ:

มีสองคำสั่งที่ประมวลผลข้อมูลซึ่งเกี่ยวข้องกับการเรียกเก็บเงินที่มีการรวบรวมไว้ใน `/var/adm/pacct` หรือไฟล์ที่ระบุอื่น

คำสั่ง `acctprc1` แปลง ID ผู้ใช้ เป็นชื่อผู้ใช้และเขียนเรีกคอร์ด ASCII ที่มีไอเท็มซึ่งเรียกเก็บเงินได้ (เวลา CPU ที่ใช้งานสูงสุด และที่ใช้งานไม่มากนัก ขนาดหน่วยความจำเฉลี่ย และข้อมูล I/O) คำสั่ง `acctprc2` แปลง เรีกคอร์ดเหล่านี้เป็นเรีกคอร์ดการบัญชีทั้งหมดซึ่งจะถูกเพิ่มลงในรายงานรายวัน โดยคำสั่ง `acctmerg` สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `acctmerg` ให้ดูที่ [acctmerg](#)

ข้อมูลบัญชีกระบวนการยังแสดงข้อมูลที่ คุณสามารถใช้เพื่อมอนิเตอร์การใช้รีซอร์สระบบได้ คำสั่ง `acctcms` สรุป การใช้รีซอร์สตามชื่อคำสั่ง รายงานนี้แสดงข้อมูลจำนวนครั้ง ที่รันแต่ละคำสั่ง จำนวนเวลาตัวประมวลผลและหน่วยความจำที่ใช้ และมีการใช้รีซอร์สมากเพียงใด (เรียกอีกอย่างว่า *hog factor*) คำสั่ง `acctcms` จัดทำสถิติในระยะยาวเกี่ยวกับการใช้ประโยชน์ระบบ เพื่อแสดงข้อมูลการใช้ระบบทั้งหมด และความถี่ของการใช้คำสั่ง สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `acctcms` ให้ดูที่ `acctcms`

คำสั่ง `acctcom` จัดการข้อมูลเดียวกัน กับคำสั่ง `acctcms` แต่แสดงข้อมูลรายละเอียด เกี่ยวกับแต่ละกระบวนการ คุณสามารถแสดงเรีกคอร์ดบัญชีกระบวนการทั้งหมด หรือเลือกเรีกคอร์ดที่สนใจเป็นพิเศษ เกณฑ์การเลือกรวมถึงโหนด ที่เกิดจากโปรเซสรอบระยะเวลาเมื่อโปรเซสสิ้นสุด ชื่อของ คำสั่ง ผู้ใช้หรือกลุ่มซึ่งเรียกใช้โปรเซส ชื่อของ WLM คลาสที่โปรเซสเป็นสมาชิก และพอร์ตซึ่งโปรเซสรัน สิ่งที่แตกต่างกัน คำสั่งการบัญชีอื่นคือ `acctcom` สามารถรันได้โดยผู้ใช้ทั้งหมด สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `acctcom` ให้ดูที่ `acctcom`

ข้อมูลบัญชีการใช้ดิสก์:

ข้อมูลการบัญชีจำนวนมากมีการรวบรวมเมื่อมีการใช้รีซอร์ส คำสั่ง `dodisk` ซึ่งรันตามที่ระบุโดย `cron` daemon เขียนเรีกคอร์ดการใช้ดิสก์เป็นครั้งคราวสำหรับผู้ใช้แต่ละรายไปยังไฟล์ `/var/adm/acct/nite(x)/dacct`

เพื่อจัดทำบัญชีนี้ให้สำเร็จ คำสั่ง `dodisk` เรียกคำสั่งอื่น ขึ้นอยู่กับความยากในการค้นหาบัญชี สามารถใช้คำสั่ง `diskusg` หรือคำสั่ง `acctdusg` เพื่อรวบรวมข้อมูล คำสั่ง `acctdisk` ถูกใช้เพื่อ เขียนเรีกคอร์ดการบัญชีทั้งหมด ในทางกลับกัน เรีกคอร์ดการบัญชีทั้งหมด ถูกใช้โดยคำสั่ง `acctmerg` เพื่อเปรียบเทียบรายงานการบัญชี รายวัน

คำสั่ง `dodisk` เรียกเก็บเงินจากผู้ใช้ สำหรับลิงก์ไปยังไฟล์ที่พบในไดเร็กทอรีล็อกอินของผู้ใช้ และแบ่งค่าธรรมเนียม อย่างเท่าเทียมกันสำหรับแต่ละไฟล์ระหว่างลิงก์ คำสั่งนี้กระจายต้นทุนของการใช้ไฟล์ให้กับทุกคนที่ใช้ไฟล์ และลบค่าธรรมเนียมออกจากผู้ใช้เมื่อผู้ใช้ยกเลิก การเข้าถึงไฟล์ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `dodisk` และ `cron` daemon ให้ดูที่ `dodisk` และ `cron`

หลักการที่เกี่ยวข้อง:

“รายงานเวลาเชื่อมต่อ” ในหน้า 178

เรีกคอร์ดการบัญชีมีเรีกคอร์ดล็อกอิน ล็อกเอาต์ การปิดระบบ และ เรีกคอร์ดล็อกอินครั้งล่าสุด

ข้อมูลบัญชีการใช้เครื่องพิมพ์:

การรวบรวมข้อมูลการใช้เครื่องพิมพ์เป็นความพยายามร่วมกันระหว่าง คำสั่ง `enq` และ `queuing daemon`

คำสั่ง `enq` จัดคิวชื่อผู้ใช้ หมายเลขงาน และชื่อของไฟล์ที่จะพิมพ์ หลังจากพิมพ์ไฟล์แล้ว คำสั่ง `qdaemon` จะเขียนเรกคอร์ด ASCII ไปยังไฟล์ซึ่งโดยปกติแล้วคือไฟล์ `/var/adm/qacct` ที่ประกอบด้วย ชื่อผู้ใช้ หมายเลขผู้ใช้ และจำนวนหน้าที่พิมพ์แล้ว คุณสามารถเรียงลำดับเรกคอร์ด เหล่านี้และแปลงเป็นเรกคอร์ดการบัญชีทั้งหมดได้ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งเหล่านี้ให้ดูที่ `enq` และ `qdaemon`

หลักการที่เกี่ยวข้อง:

“รายงานบัญชีการใช้เครื่องพิมพ์” ในหน้า 178

เรกคอร์ด ASCII ในไฟล์ `/var/adm/qacct` สามารถ มีการแปลงเป็นเรกคอร์ดการบัญชีทั้งหมดที่จะถูกเพิ่มลงในรายงานรายวัน โดยคำสั่ง `acctmerge`

ข้อมูลการบัญชีค่าธรรมเนียม:

คุณสามารถจัดทำเรกคอร์ดการบัญชีทั้งหมด ASCII ในไฟล์ `/var/adm/fee`

คุณสามารถป้อนคำสั่ง `chargefee` เพื่อจัดทำเรกคอร์ดการบัญชีทั้งหมด ASCII ในไฟล์ `/var/adm/fee` ไฟล์นี้จะถูกเพิ่มลงในรายงานรายวันโดยคำสั่ง `acctmerge`

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `chargefee` และ `acctmerge` ให้ดูที่ `chargefee` และ `acctmerge`

หลักการที่เกี่ยวข้อง:

“รายงานการบัญชีค่าธรรมเนียม” ในหน้า 179

ถ้าคุณใช้คำสั่ง `chargefee` เพื่อเรียกเก็บเงินจาก ผู้ใช้สำหรับค่าเซอร์วิส เช่น การเรียกคืนไฟล์ การให้คำปรึกษา หรือเอกสารประกอบ จะมีการเขียนเรกคอร์ดการบัญชีทั้งหมด ASCII ขึ้นในไฟล์ `/var/adm/fee` ไฟล์นี้จะถูกเพิ่มลงในรายงานรายวันโดยคำสั่ง `acctmerge`

การแก้ไขปัญหาบัญชีระบบ

ใช้เมธอดการแก้ไขปัญหาเหล่านี้เพื่อจัดการกับปัญหาพื้นฐานบางอย่าง ที่อาจเกิดขึ้นเมื่อใช้บัญชีระบบ หากข้อมูลการแก้ไขปัญหา ไม่ได้ช่วยแก้ไขปัญหา โปรดติดต่อผู้แทนเซอร์วิสของคุณ

การแก้ไขข้อผิดพลาด `tacct`:

หากคุณกำลังใช้ระบบการบัญชีเพื่อเรียกเก็บค่าธรรมเนียมรีซอร์สระบบ จากผู้ใช้ บุรณภาพของไฟล์ `/var/adm/acct/sum/tacct` ค่อนข้าง เป็นสิ่งสำคัญในบางโอกาส เรกคอร์ด `tacct` ที่น่าสับสนอาจมี ตัวเลขค่าลบ หมายเลขผู้ใช้ซ้ำ หรือหมายเลขผู้ใช้เป็น 65,535 คุณสามารถแก้ไขปัญหาเหล่านี้ได้

คุณต้องมีสิทธิ์ผู้ใช้รากหรือกลุ่ม `adm`

เมื่อต้องการจัดส่งไฟล์ `tacct` ให้ปฏิบัติตามขั้นตอนต่อไปนี้:

1. ย้ายไปยังไดเรกทอรี `/var/adm/acct/sum` โดย พิมพ์:

```
cd /var/adm/acct/sum
```

2. ใช้คำสั่ง `prtacct` เพื่อตรวจสอบไฟล์การบัญชีทั้งหมดที่ชื่อ `tacctprev` โดยพิมพ์:

```
prtacct tacctprev
```

คำสั่ง **prtacct** จัดรูปแบบและแสดงไฟล์ **tacctprev** เพื่อให้คุณสามารถตรวจสอบเวลาเชื่อมต่อ เวลาประมวลผล การใช้ดิสก์ และการใช้เครื่องพิมพ์

3. หากไฟล์ **tacctprev** มีลักษณะถูกต้อง ให้เปลี่ยนไฟล์ **tacct.mmd** ล่าสุดจากไฟล์ไบนารีเป็นไฟล์ ASCII ในตัวอย่างต่อไป นี้ คำสั่ง **acctmerg** แปลงไฟล์ **tacct.mmd** เป็นไฟล์ ASCII ที่ชื่อ **tacct.new**:

```
acctmerg -v < tacct.mmd > tacct.new
```

หมายเหตุ: คำสั่ง **acctmerg** ที่มีแฟล็ก **-a** ยังจัดทำเอาต์พุต ASCII เช่นกัน แฟล็ก **-v** จัดทำสำเนาที่มีความแม่นยำมากขึ้นสำหรับตัวเลขจุดลอยตัว

คำสั่ง **acctmerg** ใช้เพื่อรวมรายงานเรีกคอร์ดการบัญชีระหว่างกาลเข้าในรายงาน สะสมทั้งหมด (**tacct**) รายงานสะสมทั้งหมดนี้เป็นแหล่งข้อมูลซึ่งคำสั่ง **monacct** ใช้สร้างรายงานสรุปรายเดือนในรูปแบบ ASCII เนื่องจากโปรซีเดอร์ของคำสั่ง **monacct** ลบไฟล์ **tacct.mmd** ทั้งหมด คุณจึงสร้างไฟล์ **tacct** อีกครั้งโดยการรวมไฟล์เหล่านี้

4. แก้ไขไฟล์ **tacct.new** เพื่อลบเรีกคอร์ดที่ไม่ดี และเขียนเรีกคอร์ดหมายเลขผู้ใช้เข้าไปยังไฟล์อื่นโดยพิมพ์:

```
acctmerg -i < tacct.new > tacct.mmd
```

5. สร้างไฟล์ **tacct** อีกครั้งโดยพิมพ์:

```
acctmerg tacctprev < tacct.mmd > tacct
```

การแก้ไขข้อผิดพลาด **wtmp**:

/var/adm/wtmp หรือไฟล์ "who temp" อาจทำให้เกิดปัญหาในการดำเนินงานประจำวันของระบบการบัญชีได้ คุณสามารถแก้ไขข้อผิดพลาด **wtmp** ได้

คุณต้องมีสิทธิ์ผู้ใช้รากหรือกลุ่ม **adm** เพื่อทำโปรซีเดอร์นี้

เมื่อวันที่ถูกเปลี่ยนแปลงและระบบอยู่ในโหมดหลายผู้ใช้ เรีกคอร์ดการเปลี่ยนแปลงวันที่จะถูกเขียนลงในไฟล์ **/var/adm/wtmp** เมื่อพบการเปลี่ยนแปลงวันที่ คำสั่ง **wtmpfix** จะปรับการประทับเวลาในเรีกคอร์ด **wtmp** การเปลี่ยนแปลงวันที่ และการรีสตาร์ทระบบบางชุดอาจแจ้งหลังจากคำสั่ง **wtmpfix** และส่งผลให้คำสั่ง **acctcon1** ล้มเหลวและคำสั่ง **runacct** ส่งเมลไปยังแอดแคต์ ราก และ **adm** ที่แสดงรายการวันที่ซึ่งไม่ถูกต้อง

เมื่อต้องการแก้ไขข้อผิดพลาด **wtmp** ให้ปฏิบัติตาม โปรซีเดอร์ต่อไปนี้:

1. ย้ายไปยังไดเรกทอรี **/var/adm/acct/nite** โดย พิมพ์:

```
cd /var/adm/acct/nite
```

2. แปลงไฟล์ **wtmp** ไบนารีเป็นไฟล์ ASCII ซึ่งคุณสามารถแก้ไขได้โดยพิมพ์:

```
fwtmp < wtmp.mmd > wtmp.new
```

คำสั่ง **fwtmp** แปลง **wtmp** จากรูปแบบไบนารีเป็น ASCII

3. แก้ไขไฟล์ ASCII **wtmp.new** เพื่อลบเรีกคอร์ดที่เสียหาย หรือเรีกคอร์ดทั้งหมดออก ตั้งแต่จุดเริ่มต้นของไฟล์ไปจนถึงการเปลี่ยนแปลงวันที่ ซึ่งต้องการโดยพิมพ์:

```
vi wtmp.new
```

4. แปลงไฟล์ ASCII **wtmp.new** กลับเป็นรูปแบบ ไบนารีโดยพิมพ์:

```
fwtmp -ic < wtmp.new > wtmp.mmd
```

5. หากไฟล์ wtmp เสียหายเกินกว่าจะซ่อมแซมได้ ให้ใช้คำสั่ง `nulladm` เพื่อสร้างไฟล์ wtmp ที่ว่าง การทำเช่นนี้ช่วยป้องกันค่าธรรมเนียมใดๆ ในเวลาเชื่อมต่อ

```
nulladm wtmp
```

คำสั่ง `nulladm` สร้างไฟล์ที่ระบุด้วยสิทธิการอ่านและการเขียนสำหรับเจ้าของไฟล์ และกลุ่ม และสิทธิการอ่านสำหรับผู้ใช้อื่น และคำสั่งนี้ทำให้แน่ใจว่าเจ้าของไฟล์ และกลุ่มคือ `adm`

งานที่เกี่ยวข้อง:

“การแก้ไขข้อผิดพลาดการบัญชี” ในหน้า 200

คุณสามารถแก้ไขความไม่สอดคล้องของวันที่และการประทับเวลาได้

การแก้ไขสิทธิอนุญาตไฟล์การบัญชีที่ไม่ถูกต้อง:

เพื่อใช้ระบบการบัญชี ความเป็นเจ้าของไฟล์และสิทธิอนุญาตต้อง ถูกต้อง

คุณต้องมีสิทธิผู้ใช้รากหรือกลุ่ม `adm` เพื่อทำโปรซีเดอร์นี้

แอคเคาต์การจัดการ `adm` เป็นเจ้าของคำสั่งและสคริปต์ การบัญชี ยกเว้นสำหรับ `/var/adm/acct/accton` ซึ่งเป็น ของราก

เมื่อต้องการแก้ไขสิทธิอนุญาตไฟล์การบัญชีที่ไม่ถูกต้อง ให้ปฏิบัติตาม โปรซีเดอร์ต่อไปนี้:

1. เมื่อต้องการตรวจสอบสิทธิอนุญาตไฟล์โดยใช้คำสั่ง `ls` ให้พิมพ์:

```
ls -l /var/adm/acct  
  
-rws--x--- 1 adm adm 14628 Mar 19 08:11 /var/adm/acct/fiscal  
-rws--x--- 1 adm adm 14628 Mar 19 08:11 /var/adm/acct/nite  
-rws--x--- 1 adm adm 14628 Mar 19 08:11 /var/adm/acct/sum
```

2. ปรับสิทธิอนุญาตไฟล์ด้วยคำสั่ง `chown` ถ้าจำเป็น สิทธิอนุญาตเป็น 755 (สิทธิอนุญาตทั้งหมด สำหรับเจ้าของ และสิทธิการอ่านและสิทธิการดำเนินการสำหรับบุคคลอื่นทั้งหมด) นอกจากนี้ ตัวไดเรกทอรีเอง ควรมีการป้องกันการเขียนจากบุคคลอื่นด้วย ตัวอย่างเช่น:

- a. ย้ายไปยังไดเรกทอรี `/var/adm/acct` โดยพิมพ์:

```
cd /var/adm/acct
```

- b. เปลี่ยนความเป็นเจ้าของของไดเรกทอรี `sum`, `nite`, และ `fiscal` เป็นสิทธิกลุ่ม `adm` โดยพิมพ์:

```
chown adm sum/* nite/* fiscal/*
```

เพื่อเป็นการป้องกัน ไม่ยั่วให้ผู้ใช้งานพยายามหลีกเลี่ยงค่าธรรมเนียม ให้ปฏิเสธสิทธิการเขียนสำหรับ บุคคลอื่นบนไฟล์เหล่านี้ เปลี่ยนเจ้าของกลุ่มคำสั่ง `accton` เป็น `adm` และสิทธิอนุญาตเป็น 710 นั่นคือไม่มีสิทธิอนุญาตสำหรับบุคคลอื่น โปรเซสที่เป็นของ `adm` สามารถดำเนินการคำสั่ง `accton` แต่ผู้ใช้ทั่วไปไม่สามารถทำได้

3. ไฟล์ `/var/adm/wtmp` ต้องเป็นของ `adm` ด้วย ถ้า `/var/adm/wtmp` เป็นของราก คุณจะเห็น ข้อความต่อไปนี้ในระหว่างสตาร์ทอัพ:

```
/var/adm/acct/startup: /var/adm/wtmp: Permission denied
```

เมื่อต้องการ แก้ไขความเป็นเจ้าของ `/var/adm/wtmp` ให้เปลี่ยนความเป็นเจ้าของ เป็นกลุ่ม `adm` โดยพิมพ์คำสั่งต่อไปนี้:

```
chown adm /var/adm/wtmp
```

การแก้ไขข้อผิดพลาดการบัญชี:

คุณสามารถแก้ไขความไม่สอดคล้องของวันที่และการประทับเวลาได้

คุณต้องมีสิทธิ์ผู้ใช้รากหรือกลุ่ม adm เพื่อทำโปรซีเดอร์นี้

การประมวลผลไฟล์ /var/adm/wtmp อาจส่งผลให้มีการเมลค่าเตือนบางรายการไปยังรากไฟล์ wtmp ประกอบด้วย ข้อมูลที่รวบรวมโดย /etc/init และ /bin/login และใช้โดยสคริปต์การบัญชีเป็นหลักสำหรับการคำนวณเวลาเชื่อมต่อ (ระยะเวลาที่ผู้ใช้ล็อกอิน) โชคไม่ดีที่การเปลี่ยนแปลงวันที่สร้างความสับสนให้กับ โปรแกรมที่ประมวลผลไฟล์ wtmp ผลที่ตามมาคือ คำสั่ง `runacct` จะส่งเมลไปยังรากและ adm เพื่อแจ้งถึงข้อผิดพลาดใดๆ หลังจากการเปลี่ยนแปลง วันที่นับตั้งแต่วันที่การบัญชีครั้งล่าสุด

1. พิจารณาว่าคุณได้รับข้อผิดพลาดใดๆ หรือไม่ คำสั่ง `acctcon1` ส่งเอาต์พุต ข้อความแสดงข้อผิดพลาดที่จะเมลไปยัง adm และรากโดยคำสั่ง `runacct` ตัวอย่างเช่น ถ้าคำสั่ง `acctcon1` สะดุด หลังจากการเปลี่ยนแปลงวันที่และล้มเหลวในการรวบรวมเวลาเชื่อมต่อ adm อาจ ได้รับเมลคล้ายกับข้อความเมลต่อไปนี้:

```
Mon Jan 6 11:58:40 CST 1992
acctcon1: bad times: old: Tue Jan 7 00:57:14 1992
new: Mon Jan 6 11:57:59 1992
acctcon1: bad times: old: Tue Jan 7 00:57:14 1992
new: Mon Jan 6 11:57:59 1992
acctcon1: bad times: old: Tue Jan 7 00:57:14 1992
new: Mon Jan 6 11:57:59 1992
```

2. ปรับไฟล์ wtmp โดยพิมพ์:

```
/usr/sbin/acct/wtmpfix wtmp
```

คำสั่ง `wtmpfix` ตรวจสอบไฟล์ wtmp เพื่อหา ความไม่สอดคล้องกันของวันที่และการประทับเวลา และแก้ไขปัญหาที่อาจทำให้ `acctcon1` ล้มเหลว อย่างไรก็ตาม การเปลี่ยนแปลงวันที่บางรายการมีการแจ้งโดย `wtmpfix`

3. รันการบัญชีทันทีก่อนหน้าการปิดเครื่อง หรือทันทีหลังจากสแตร์ทอัพ การใช้คำสั่ง `runacct` ในเวลาเหล่านี้ช่วยลดจำนวนรายการที่มีเวลาไม่ดีที่เหลือน้อยที่สุด คำสั่ง `runacct` ยังคง ส่งเมลไปยังแอดเคาต์รากและ adm ต่อไปจนกระทั่งคุณแก้ไขสคริปต์ `runacct` ค้นหาส่วน WTMPFIX และแสดงข้อคิดเห็นบรรทัดซึ่ง บันทึกไฟล์ได้รับเมลที่ส่งไปยังแอดเคาต์รากและ adm

งานที่เกี่ยวข้อง:

“การแก้ไขข้อผิดพลาด wtmp” ในหน้า 198

/var/adm/wtmp หรือไฟล์ "who temp" อาจทำให้เกิดปัญหาในการดำเนินงานประจำวันของระบบการบัญชีได้ คุณสามารถแก้ไขข้อผิดพลาด wtmp ได้

ข้อผิดพลาดการบัญชีที่พบในขณะรันคำสั่ง runacct:

คุณอาจพบข้อผิดพลาดในขณะรันคำสั่ง `runacct`

หมายเหตุ: คุณต้องมีสิทธิ์ผู้ใช้รากหรือกลุ่ม adm เพื่อรันคำสั่ง `runacct`

คำสั่ง `runacct` ประมวลผลไฟล์ที่มีขนาดใหญ่มาก โปรซีเดอร์เกี่ยวข้องกับ การส่งผ่านไฟล์หลายครั้งและใช้รีซอร์สระบบเป็นจำนวนมาก ในขณะที่โปรซีเดอร์เกิดขึ้น เนื่องจากคำสั่ง `runacct` ใช้รีซอร์ส เป็นจำนวนมาก ดังนั้น โดยปกติแล้ว คำสั่งนี้จึงรันในช่วงเช้าเมื่อคำสั่งสามารถใช้เครื่องได้เต็มที่และไม่รบกวนคำสั่งอื่นๆ

คำสั่ง **runacct** คือสคริปต์ที่แบ่งออกเป็นระยะต่างๆ ระยะช่วยให้คุณสามารถรีเซ็ตคำสั่งในตำแหน่งที่หยุดได้โดยไม่ต้องรีรันทั้งสคริปต์

เมื่อ **runacct** พบปัญหา คำสั่งจะส่งข้อความแสดงข้อผิดพลาด ไปยังปลายทางอื่น ขึ้นอยู่กับว่าข้อผิดพลาดเกิดขึ้นที่ใด โดยปกติแล้ว คำสั่ง ส่งวันที่และข้อความไปยังคอนโซลที่แนะนำให้คุณค้นหาในไฟล์ **activeMMDD** (เช่น **active0621** สำหรับ 21 มิถุนายน) ซึ่งอยู่ในไดเรกทอรี **/usr/adm/acct/nite** เมื่อคำสั่ง **runacct** เลิกกลางคัน คำสั่งจะย้ายไฟล์ **active** ทั้ง ไฟล์ไปยัง **activeMMDD** และผนวก ข้อความที่อธิบายปัญหา

ตรวจทานตารางข้อความแสดงข้อผิดพลาดต่อไปนี้เพื่อดูข้อผิดพลาดซึ่งคุณอาจพบในขณะรันคำสั่ง **runacct**

หมายเหตุ:

- คำย่อ **MMDD** ย่อมาจาก เดือนและวัน เช่น 0102 หมายถึง 2 มกราคม ตัวอย่างเช่น ข้อผิดพลาดที่ไม่สามารถกู้คืนได้ขณะที่ **CONNECT1** ประมวลผลเมื่อวันที่ 2 มกราคมสร้างไฟล์ **active0102** ที่มีข้อความแสดงข้อผิดพลาด
- คำย่อ "SE message" หมายถึงข้อความแสดงข้อผิดพลาดมาตรฐาน เช่น:

```
***** ACCT ERRORS : see active0102 *****
```

สถานะเบื้องต้นและข้อความแสดงข้อผิดพลาดจากคำสั่ง **runacct**

สถานะ	คำสั่ง	ไม่สามารถกู้คืนได้?	ข้อความแสดงข้อผิดพลาด	ปลายทาง
pre	runacct	ใช่	* 2 CRONS or ACCT PROBLEMS * ERROR: locks found, run aborted	คอนโซล, เมล, active
pre	runacct	ใช่	runacct: Insufficient space in /usr (nnn blks) ; Terminating procedure	คอนโซล, เมล, active
pre	runacct	ใช่	SE message; ERROR: acctg already run for 'date': check lastdate	คอนโซล, เมล, activeMMDD
pre	runacct	ไม่	* SYSTEM ACCOUNTING STARTED *	คอนโซล
pre	runacct	ไม่	restarting acctg for 'date' at STATE	console active, คอนโซล
pre	runacct	ไม่	restarting acctg for 'date' at state (argument \$2) previous state was STATE	active
pre	runacct	ใช่	SE message; Error: runacct called with invalid arguments	คอนโซล, เมล, activeMMDD

สถานะและข้อความแสดงข้อผิดพลาด จากคำสั่ง **runacct**

สถานะ	คำสั่ง	ไม่สามารถกู้คืนได้?	ข้อความแสดงข้อผิดพลาด	ปลายทาง
SETUP	runacct	ไม่	ls -l fee pacct* /var/ adm/wtmp	active
SETUP	runacct	ใช่	SE message; ERROR: turnacct switch returned rc=error	คอนโซล, เมล, activeMMDD
SETUP	runacct	ใช่	SE message; ERROR: SpacctMMDD already exists file setups probably already run	activeMMDD
SETUP	runacct	ใช่	SE message; ERROR: wtmpMMDD already exists: run setup manually	คอนโซล, เมล, activeMMDD
WTMPFIX	wtmpfix	ไม่	SE message; ERROR: wtmpfix errors see xtmperrorMMDD	activeMMDD, wtmperrorMMDD
WTMPFIX	wtmpfix	ไม่	wtmp processing complete	active
CONNECT1	acctcon1	ไม่	SE message; (errors from acctcon1 log)	คอนโซล, เมล, activeMMDD
CONNECT2	acctcon2	ไม่	connect acctg complete	active
PROCESS	runacct	ไม่	WARNING: accounting already run for pacctN	active
PROCESS	acctprc1 acctprc2	ไม่	process acctg complete for SpacctNMDD	active
PROCESS	runacct	ไม่	all process actg complete for date	active
MERGE	acctmerg	ไม่	tacct merge to create dayacct complete	active
FEES	acctmerg	ไม่	merged fees OR no fees	active
DISK	acctmerg	ไม่	merged disk records OR no disk records	active
MERGEACCT	acctmerg	ไม่	WARNING: recreating sum/tacct	active
MERGEACCT	acctmerg	ไม่	updated sum/tacct	active
CMS	runacct	ไม่	WARNING: recreating sum/cms	active
CMS	acctems	ไม่	command summaries complete	active

สถานะและข้อความแสดงข้อผิดพลาด จากคำสั่ง **runacct**

สถานะ	คำสั่ง	ไม่สามารถกู้คืนได้?	ข้อความแสดงข้อผิดพลาด	ปลายทาง
CLEANUP	runacct	ไม่	system accounting completed at 'date'	active
CLEANUP	runacct	ไม่	*SYSTEM ACCOUNTING COMPLETED*	คอนโซล
<wrong>	runacct	ใช่	SE message; ERROR: invalid state, check STATE	คอนโซล, เมล, activeMMDD

หมายเหตุ: เลเบล <wrong> ในตารางก่อนหน้านี้ไม่ได้แสดงถึงสถานะ แต่เป็นสถานะอื่นที่ไม่ใช่สถานะที่ถูกต้อง ซึ่งถูกเขียนขึ้นในไฟล์สถานะ /usr/adm/acct/nite/statefile

ข้อมูลสรุปของปลายทางข้อความ

ปลายทาง	คำอธิบาย
คอนโซล	อุปกรณ์ /dev/console
เมล	ข้อความถูกเมลไปยังแอดเดสส์ ราก และ adm
active	ไฟล์ /usr/adm/acct/nite/active
activeMMDD	ไฟล์ /usr/adm/acct/nite/activeMMDD
wtmperrMMDD	ไฟล์ /usr/adm/acct/nite/wtmperrorMMDD
STATE	Current state in /usr/adm/acct/nite/statefile file
fd2log	ข้อความแสดงข้อผิดพลาดอื่นใด

System Resource Controller

System Resource Controller (SRC) นำเสนอชุดของคำสั่ง และรูทีนย่อยที่ทำให้ผู้จัดการระบบและโปรแกรมเมอร์สามารถสร้างและควบคุมระบบย่อยได้ง่ายขึ้น

ระบบย่อย คือโปรแกรมหรือโปรเซสใดๆ หรือชุดของโปรแกรมหรือ โปรเซสซึ่งโดยปกติแล้ว สามารถดำเนินงานได้โดยอิสระ หรือใช้พร้อมๆ กับ ระบบการควบคุม ระบบย่อยมีการออกแบบมาเป็นยูนิทเพื่อนำเสนอฟังก์ชัน ที่กำหนดไว้

SRC ได้รับการออกแบบมาเพื่อลดความต้องการการดำเนินการ ของผู้ใช้งานให้เหลือน้อยที่สุด SRC นำเสนอกลไกการควบคุม โปรเซสระบบย่อยโดยใช้ บรรทัดคำสั่งทั่วไปและอินเตอร์เฟซ C กลไกนี้ประกอบด้วยต่อไปนี้:

- อินเตอร์เฟซผู้ใช้ที่สอดคล้องกันสำหรับการเริ่มต้น หยุด และสอบถามสถานะ
- การบันทึกการสิ้นสุดที่ผิดปกติของระบบย่อย
- โปรแกรมการแจ้งเตือนที่เรียกเมื่อโปรเซสที่เกี่ยวข้องสิ้นสุดลงอย่าง ผิดปกติ
- การติดตามระบบย่อย กลุ่มระบบย่อย หรือเซิร์ฟเวอร์ย่อย
- การสนับสนุนการควบคุมการดำเนินงานบนระบบรีโมต
- การรีเฟรชระบบย่อย (เช่น หลังจากการเปลี่ยนข้อมูลคอนฟิกูเรชัน)

SRC มีประโยชน์ถ้าคุณต้องการวิธีทั่วไปในการเริ่มต้น หยุด และรวบรวม ข้อมูลสถานะเกี่ยวกับโปรเซส
หลักการที่เกี่ยวข้อง:

“บทนำเกี่ยวกับ AIX for BSD system managers” ในหน้า 353

ข้อมูลต่อไปนี้เป็นเคล็ดลับที่ช่วยผู้จัดการระบบการกระจายซอฟต์แวร์ของ Berkeley (BSD) เริ่มต้นจัดการ AIX

คอมโพเนนต์ระบบย่อย

ข้อมูลต่อไปนี้เป็นคุณสมบัติและคอมโพเนนต์ของระบบย่อย

ระบบย่อยสามารถมีคุณสมบัติต่อไปนี้ตั้งแต่หนึ่งรายการขึ้นไป:

- เป็นที่รู้จักของระบบโดยใช้ชื่อ
- ต้องการสถานะแวดล้อมการดำเนินการที่ซับซ้อนมากขึ้นกว่ารูทีนย่อยหรือ โปรแกรมที่ไม่มีเอกลักษณ์
- มีแอพลิเคชันโปรแกรมและไลบรารี ตลอดจนโค้ดระบบย่อย
- ควบคุมรีซอร์สที่สามารถเริ่มต้นและหยุดได้โดยใช้ชื่อ
- ต้องการการแจ้งเตือนถ้าโปรเซสที่เกี่ยวข้องไม่สำเร็จเพื่อทำความสะอาด หรือเพื่อกู้คืนรีซอร์ส
- ต้องการการควบคุมในการดำเนินงานมากกว่าโปรเซส daemon แบบง่าย
- ต้องควบคุมโดยผู้ใช้งานรีโมต
- นำเซิร์ฟเวอร์ย่อยไปใช้เพื่อจัดการกับรีซอร์สเฉพาะ
- ไม่วางตัวเองไว้ในพื้นหลัง

ตัวอย่างบางส่วนของระบบย่อยคือ ypsserv, ntsd, qdaemon, inetd, syslogd, และ sendmail

หมายเหตุ: โปรดดูที่ระบบย่อยเฉพาะแต่ละระบบสำหรับรายละเอียดความสามารถ SRC

ใช้คำสั่ง `lssrc -a` เพื่อแสดงรายการระบบย่อยที่ใช้งานอยู่และที่ไม่ได้ใช้งานอยู่บนระบบของคุณ

ข้อมูลต่อไปนี้จะกำหนดกลุ่มระบบย่อยและเซิร์ฟเวอร์ย่อย:

กลุ่มระบบย่อย

กลุ่มระบบย่อย คือกลุ่มของระบบย่อยที่ระบุใดๆ การจัดกลุ่ม ระบบย่อยเข้าด้วยกันช่วยให้สามารถควบคุมระบบย่อย
ได้หลายระบบในครั้งเดียว ตัวอย่างบางส่วนของกลุ่มระบบย่อยคือ TCP/IP, SNA Services, Network Information
System (NIS), และระบบไฟล์เน็ตเวิร์ก (NFS)

เซิร์ฟเวอร์ย่อย

เซิร์ฟเวอร์ย่อย คือโปรแกรมหรือโปรเซสที่เป็นสมาชิกของระบบย่อย ระบบย่อยหนึ่งสามารถมีเซิร์ฟเวอร์ย่อยได้
หลายรายการ และรับผิดชอบในการเริ่มต้น การหยุด และการแสดงสถานะของเซิร์ฟเวอร์ย่อย เซิร์ฟเวอร์ย่อย
สามารถกำหนดได้เฉพาะสำหรับ ระบบย่อยที่มีชนิดการสื่อสารเป็นคิวข้อความและซ็อกเก็ต IPC เท่านั้น ระบบย่อยที่
ใช้การสื่อสารสัญญาณไม่สนับสนุนเซิร์ฟเวอร์ย่อย

เซิร์ฟเวอร์ย่อย เริ่มต้นขึ้นเมื่อระบบย่อยพาเรนต์เริ่มต้นขึ้น หากคุณพยายามเริ่มต้น เซิร์ฟเวอร์ย่อยและระบบย่อย
พาเรนต์ไม่ได้ใช้งานอยู่ คำสั่ง `startsrc` จะเริ่มต้นระบบย่อยด้วย

ลำดับชั้น SRC

ลำดับชั้น System Resource Controller เริ่มต้นด้วยระบบปฏิบัติการ ตามด้วยกลุ่มระบบย่อย (เช่น tcpip) ซึ่งประกอบด้วยระบบย่อยต่างๆ (เช่น inetd daemon) ซึ่งสามารถเป็นเจ้าของ หลายเซิร์ฟเวอร์ย่อย (เช่น ftp daemon และคำสั่ง finger)

คำสั่งการจัดการ SRC

คุณสามารถจัดการ SRC จากบรรทัดคำสั่ง

คำสั่งการจัดการ SRC มีดังนี้:

ไอเท็ม	คำอธิบาย
srcmstr daemon	เริ่มต้น System Resource Controller
คำสั่ง startsrc	เริ่มต้นระบบย่อย กลุ่มระบบย่อย หรือเซิร์ฟเวอร์ย่อย
คำสั่ง stopsrc	หยุดระบบย่อย กลุ่มระบบย่อย หรือเซิร์ฟเวอร์ย่อย
คำสั่ง refresh	รีเฟรชระบบย่อย
คำสั่ง traceson	เปิดการติดตามระบบย่อย กลุ่มระบบย่อย หรือเซิร์ฟเวอร์ย่อย
คำสั่ง tracesoff	ปิดการติดตามระบบย่อย กลุ่มระบบย่อย หรือเซิร์ฟเวอร์ย่อย
คำสั่ง lssrc	เรียกดูสถานะบนระบบย่อย

การเริ่มต้น System Resource Controller

System Resource Controller (SRC) เริ่มต้นขึ้นในระหว่างการเริ่มต้น ระบบด้วยเร็กคอร์ด /usr/sbin/srcmstr daemon ในไฟล์ /etc/inittab

ข้อกำหนดเบื้องต้นสำหรับการเริ่มต้น SRC มีดังต่อไปนี้:

- การอ่านและการเขียนไฟล์ /etc/inittab ต้องการ สิทธิผู้ไชราก
- คำสั่ง mkitab ต้องการสิทธิผู้ไชราก
- ต้องมีเร็กคอร์ด srcmstr daemon อยู่ในไฟล์ /etc/inittab

ไฟล์ /etc/inittab ดีฟอลต์มีเร็กคอร์ด ดังกล่าวอยู่แล้ว ดังนั้นโปรซีเดอร์นี้อาจไม่จำเป็น คุณยังสามารถเริ่มต้น SRC จากบรรทัดคำสั่ง โพรไฟล์ หรือเซลล์สคริปต์ แต่มีหลายเหตุผลที่ควร เริ่มต้นในระหว่างการเริ่มต้นระบบ:

- การเริ่มต้น SRC จากไฟล์ /etc/inittab ช่วยให้คำสั่ง init สามารถรีสตาร์ท SRC ได้ถ้า SRC หยุดเนื่องจากเหตุผลใดๆ
- SRC ได้รับการออกแบบมาเพื่อช่วยให้ง่ายและลดจำนวนการดำเนินการของผู้ใช้ ที่จำเป็นในการควบคุมระบบย่อย การเริ่มต้น SRC จากซอร์สใดๆ ที่ไม่ใช่ไฟล์ /etc/inittab ไม่เป็นไปตามเป้าหมายนั้น
- ไฟล์ /etc/inittab ดีฟอลต์มีเร็กคอร์ดสำหรับ การเริ่มต้นระบบย่อยการจัดการเวลาการพิมพ์ (qdaemon) โดยใช้คำสั่ง startsrc การติดตั้งปกติมีระบบย่อยอื่นที่เริ่มต้นด้วยคำสั่ง startsrc ในไฟล์ /etc/inittab เช่นกัน เนื่องจากคำสั่ง srcmstr กำหนดว่า SRC ต้องกำลังรัน การลบ srcmstr daemon ออกจากไฟล์ /etc/inittab ส่งผลให้คำสั่ง startsrc เหล่านี้ ล้มเหลว

หมายเหตุ: โปรซีเดอร์นี้จำเป็นเฉพาะถ้าไฟล์ /etc/inittab ยังไม่มี เร็กคอร์ดสำหรับ srcmstr daemon

1. จัดทำเร็กคอร์ดสำหรับ srcmstr daemon ในไฟล์ /etc/inittab โดยใช้คำสั่ง mkitab ตัวอย่างเช่น เมื่อต้องการ จัดทำเร็กคอร์ดซึ่งเหมือนกับที่ปรากฏขึ้นในไฟล์ /etc/inittab ดีฟอลต์ ให้พิมพ์:

```
mkkitab -i fbcheck srcmstr:2:respawn:/usr/sbin/srcmstr
```

แฟล็ก -i fbcheck ทำให้มั่นใจว่าการแทรกเร็กคอร์ดไว้ ก่อนหน้าเร็กคอร์ดระบบย่อยทั้งหมด

2. บอกให้คำสั่ง init ประมวลผลไฟล์ /etc/inittab อีกครั้ง โดยพิมพ์:

```
telinit q
```

เมื่อคำสั่ง `init` เข้าไปที่ไฟล์ `/etc/inittab` อีกครั้ง คำสั่งจะประมวลผลเรีกคอร์ดที่บ่อนใหม่ สำหรับ `srcmstr` daemon และเริ่มต้น SRC

หลักการที่เกี่ยวข้อง:

“การควบคุมระบบย่อย” ในหน้า 208

สามารถใช้คำสั่ง `traceson` เพื่อเปิด และสามารถใช้คำสั่ง `traceoff` เพื่อปิดการติดตามรีซอร์ส System Resource Controller (SRC) เช่น ระบบย่อย กลุ่มของ ระบบย่อย หรือเซิร์ฟเวอร์ย่อย

งานที่เกี่ยวข้อง:

“การรีเฟรชระบบย่อยหรือกลุ่มระบบย่อย” ในหน้า 207

ใช้คำสั่ง `refresh` เพื่อบอกให้รีซอร์ส System Resource Controller (SRC) เช่น ระบบย่อย หรือกลุ่มของระบบย่อย รีเฟรชตัวเอง

การเริ่มต้นหรือการหยุดระบบย่อย กลุ่มระบบย่อย หรือเซิร์ฟเวอร์ย่อย

ใช้คำสั่ง `startsrc` เพื่อเริ่มต้นรีซอร์ส System Resource Controller (SRC) เช่น ระบบย่อย กลุ่มของระบบย่อย หรือ เซิร์ฟเวอร์ย่อย ใช้คำสั่ง `stopsrc` เพื่อหยุดรีซอร์ส SRC เช่น ระบบย่อย กลุ่มของระบบย่อย หรือเซิร์ฟเวอร์ย่อย

ข้อกำหนดเบื้องต้นสำหรับการเริ่มต้นหรือการหยุด ระบบย่อย กลุ่มระบบย่อย หรือเซิร์ฟเวอร์ย่อย มีดังต่อไปนี้:

- เพื่อเริ่มต้นหรือหยุดรีซอร์ส SRC SRC ต้องกำลังรัน โดยปกติ SRC มีการเริ่มต้น ในระหว่างการเริ่มต้นระบบ ไฟล์ `/etc/inittab` ดีฟอลต์ ซึ่งกำหนดโปรเซสที่จะเริ่มต้นในระหว่างการเริ่มต้น มีเรีกคอร์ดสำหรับ `srcmstr` daemon (SRC) เมื่อต้องการดูว่า SRC กำลังรันหรือไม่ให้พิมพ์ `ps -A` และมองหาโปรเซสที่ชื่อ `srcmstr`
- ผู้ใช้หรือโปรเซสที่เริ่มต้นรีซอร์ส SRC ต้องมีสิทธิผู้ใช้ราก โปรเซสที่เริ่มต้นระบบ (คำสั่ง `init`) มีสิทธิผู้ใช้ราก
- ผู้ใช้หรือโปรเซสที่หยุดรีซอร์ส SRC ต้องมีสิทธิผู้ใช้ราก

สามารถใช้คำสั่ง `startsrc`:

- จากไฟล์ `/etc/inittab` เพื่อให้เริ่มต้นรีซอร์ส ในระหว่างการเริ่มต้นระบบ
- จากบรรทัดคำสั่ง
- With SMIT

เมื่อคุณเริ่มต้นกลุ่มระบบย่อย ระบบย่อยทั้งหมดของกลุ่มนั้นจะถูก เริ่มต้นขึ้นด้วย เมื่อคุณเริ่มต้นระบบย่อย เซิร์ฟเวอร์ย่อยทั้งหมดของระบบย่อยนั้นจะถูกเริ่มต้นขึ้นด้วย เมื่อคุณเริ่มต้นเซิร์ฟเวอร์ย่อย ระบบย่อยพาเรนต์ของเซิร์ฟเวอร์ย่อยนั้นจะถูกเริ่มต้นขึ้นด้วยถ้ายังไม่ได้ รันอยู่

เมื่อคุณหยุดระบบย่อย เซิร์ฟเวอร์ย่อยทั้งหมดของระบบย่อยนั้น จะหยุดด้วย อย่างไรก็ตาม เมื่อคุณหยุดเซิร์ฟเวอร์ย่อย สถานะของระบบย่อยพาเรนต์ของ เซิร์ฟเวอร์ย่อยนั้นจะไม่เปลี่ยน

ทั้งคำสั่ง `startsrc` และ `stopsrc` มีแฟล็กที่ช่วยให้สามารถจัดทำคำร้องขอบนโลคัลหรือรีโมตโฮสต์ได้ โปรดดูที่คำสั่ง `srcmstr` สำหรับข้อกำหนดคอนฟิกูเรชันเพื่อสนับสนุนคำร้องขอ SRC รีโมต

การเริ่ม หรือการหยุดภารกิจของระบบย่อย

ภารกิจ	SMIT fast path	คำสั่งหรือไฟล์
เริ่มระบบย่อย	smit startssys	/bin/startsrc -s <i>SubsystemName</i> , หรือแก้ไข /etc/inittab
หยุดระบบย่อย	smit stopssys	/bin/stopsrc -s <i>SubsystemName</i>

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง stopsrc

คำสั่ง startsrc

คำสั่ง srcmstr

การแสดงผลสถานะของระบบย่อย

ใช้คำสั่ง **Issrc** เพื่อแสดงผลสถานะของรีซอร์ส System Resource Controller (SRC) เช่น ระบบย่อย กลุ่มของระบบย่อย หรือ เซิร์ฟเวอร์ย่อย

ระบบย่อยทั้งหมดสามารถส่งคืนรายงานสถานะแบบย่อที่รวมถึงข้อมูล กลุ่มซึ่งระบบย่อยเป็นสมาชิก ระบบย่อยใช้งานอยู่หรือไม่ และ ID โป้รเซส (PID) หากระบบย่อยไม่ได้ใช้เมธอดการสื่อสารด้วยสัญญาณ สามารถเขียนโปรแกรมให้ระบบย่อยส่งคืนรายงานสถานะแบบเต็มที่มีข้อมูลสถานะเพิ่มเติมได้

คำสั่ง **Issrc** จัดเตรียมแฟล็กและ พารามิเตอร์สำหรับการระบุระบบย่อยโดยใช้ชื่อหรือ PID สำหรับการแสดงรายการ ระบบย่อยทั้งหมด สำหรับการร้องขอรายงานสถานะแบบย่อหรือแบบเต็ม และสำหรับการร้องขอสถานะของรีซอร์ส SRC บนโหนดหรือรีโมตโฮสต์

โปรดดูที่คำสั่ง **srcmstr** สำหรับข้อกำหนดคอนฟิกูเรชันเพื่อสนับสนุนคำร้องขอ SRC รีโมต

ภารกิจการแสดงผลสถานะของระบบย่อย

ภารกิจ	พาทด่วน SMIT	คำสั่งหรือไฟล์
แสดงผลสถานะของระบบย่อย (รูปแบบเต็ม)	smit qssys	Issrc -l -s <i>SubsystemName</i>
แสดงผลสถานะของระบบย่อยทั้งหมด	smit lssys	Issrc -a
แสดงผลสถานะของระบบย่อยทั้งหมดบนโฮสต์ เฉพาะ		Issrc -h <i>HostName</i> -a

การรีเฟรชระบบย่อยหรือกลุ่มระบบย่อย

ใช้คำสั่ง **refresh** เพื่อบอกให้รีซอร์ส System Resource Controller (SRC) เช่น ระบบย่อย หรือกลุ่มของระบบย่อย รีเฟรชตัวเอง

ข้อกำหนดเบื้องต้นสำหรับการรีเฟรชระบบย่อยหรือ กลุ่มระบบย่อยมีดังต่อไปนี้:

- SRC ต้องกำลังรัน
- รีซอร์สซึ่งคุณต้องการรีเฟรชต้องไม่ใช่เมธอดการสื่อสารด้วย สัญญาณ
- รีซอร์สซึ่งคุณต้องการรีเฟรชต้องมีการเขียนโปรแกรมให้ตอบสนองต่อ คำร้องขอการรีเฟรช

คำสั่ง **refresh** จัดเตรียมแฟล็กและพารามิเตอร์ สำหรับการระบุระบบย่อยโดยใช้ชื่อหรือ PID คุณยังสามารถใช้แฟล็กและพารามิเตอร์ดังกล่าวเพื่อร้องขอ ระบบย่อยหรือกลุ่มของระบบย่อยที่รีเฟรช บนโหนดหรือรีโมต โฮสต์ โปรดดูที่คำสั่ง **srcmstr**

สำหรับข้อกำหนดคอนฟิกูเรชันเพื่อสนับสนุนคำร้องขอ SRC รีโมต

การรีเฟรชระบบย่อยหรือกลุ่มระบบย่อย

ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
รีเฟรชระบบย่อย	smit refresh	refresh -s Subsystem

งานที่เกี่ยวข้อง:

“การเริ่มต้น System Resource Controller” ในหน้า 205

System Resource Controller (SRC) เริ่มต้นขึ้นในระหว่างการเริ่มต้น ระบบด้วยเร็กคอร์ด /usr/sbin/srcmstr daemon ในไฟล์ /etc/inittab

การควบคุมระบบย่อย

สามารถใช้คำสั่ง **traceson** เพื่อเปิด และสามารถใช้คำสั่ง **tracessoff** เพื่อปิดการติดตามรีซอร์ส System Resource Controller (SRC) เช่น ระบบย่อย กลุ่มของ ระบบย่อย หรือเซิร์ฟเวอร์ย่อย

ใช้คำสั่ง **traceson** เพื่อเปิดการติดตามรีซอร์ส System Resource Controller (SRC) เช่น ระบบย่อย กลุ่มของระบบย่อย หรือเซิร์ฟเวอร์ย่อย

ใช้คำสั่ง **tracessoff** เพื่อปิดการติดตามรีซอร์ส System Resource Controller (SRC) เช่น ระบบย่อย กลุ่มของระบบย่อย หรือเซิร์ฟเวอร์ย่อย

คำสั่ง **traceson** และ **tracessoff** สามารถใช้ เพื่อเปิดหรือปิดการติดตามแบบรีโมตบนโฮสต์เฉพาะ โปรดดูที่ คำสั่ง **srcmstr** สำหรับข้อกำหนดคอนฟิกูเรชัน เพื่อสนับสนุนคำร้องขอ SRC รีโมต

ข้อกำหนดเบื้องต้น

- เมื่อต้องการเปิดหรือปิดการติดตามรีซอร์ส SRC SRC ต้องกำลัง รันอยู่
- รีซอร์สซึ่งคุณต้องการติดตามต้องไม่ใช่เมธอดการสื่อสารด้วย สัญญาณ
- รีซอร์สซึ่งคุณต้องการติดตามต้องมีการเขียนโปรแกรมให้ตอบสนองต่อ คำร้องขอการติดตาม

ภารกิจการเปิด/ปิดระบบย่อย กลุ่มระบบย่อย หรือ เซิร์ฟเวอร์ย่อย

ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
เปิดการติดตามระบบย่อย (รูปแบบย่อ)	smit tracessysyn	traceson -s Subsystem
เปิดการติดตามระบบย่อย (รูปแบบเต็ม)	smit tracessysyn	traceson -l -s Subsystem
ปิดการติดตามระบบย่อย	smit tracessyssoff	tracessoff -s Subsystem

งานที่เกี่ยวข้อง:

“การเริ่มต้น System Resource Controller” ในหน้า 205

System Resource Controller (SRC) เริ่มต้นขึ้นในระหว่างการเริ่มต้น ระบบด้วยเร็กคอร์ด /usr/sbin/srcmstr daemon ในไฟล์ /etc/inittab

ไฟล์ระบบปฏิบัติการ

ไฟล์ถูกใช้สำหรับอินพุตและเอาต์พุต (I/O) ทั้งหมดของข้อมูลใน ระบบปฏิบัติการ เพื่อให้การเข้าถึงทั้งซอฟต์แวร์และฮาร์ดแวร์เป็นมาตรฐาน

Input เกิดขึ้นเมื่อเนื้อหาของไฟล์ถูก แก่ไขหรือเขียนลง *Output* เกิดขึ้นเมื่อเนื้อหาของไฟล์หนึ่ง ถูกอ่านหรือถ่ายโอนไปยังไฟล์อื่น ตัวอย่างเช่น เมื่อต้องการสร้างสำเนาพิมพ์ของไฟล์ ระบบอ่านข้อมูลจากไฟล์ข้อความ และเขียน ข้อมูลนั้นไปยังไฟล์ที่แทนเครื่องพิมพ์

ชนิดของไฟล์

ชนิดของไฟล์ที่ระบบรู้จักคือ ปกติ, ไตเร็กทอรี, หรือ พิเศษ อย่างไรก็ตาม อย่างไม่อย่างหนึ่ง อย่างไรก็ตาม ระบบปฏิบัติการใช้ชนิดพื้นฐานเหล่านี้ในหลาย รูปแบบที่แตกต่างกันไป

ชนิดพื้นฐานของไฟล์มีดังต่อไปนี้:

ไอเท็ม	คำอธิบาย
ปกติ	จัดเก็บข้อมูล (ข้อความ ไบนารี และปฏิบัติการ)
ไตเร็กทอรี	มีข้อมูลที่ชี้เพื่อเข้าถึงไฟล์อื่น
พิเศษ	กำหนดไฟล์ไฟฟ์ FIFO (first-in, first-out) หรืออุปกรณ์ฟิลิซัล

ชนิดของไฟล์ทั้งหมดที่ระบบรู้จักอยู่ภายใต้หมวดหมู่ อย่างไม่อย่างหนึ่ง อย่างไรก็ตาม ระบบปฏิบัติการใช้ชนิดพื้นฐานเหล่านี้ในหลาย รูปแบบที่แตกต่างกันไป

ไฟล์ปกติ

ไฟล์ปกติคือไฟล์ทั่วไปที่สุดและใช้เพื่อจัดเก็บข้อมูล ไฟล์ปกติอยู่ในรูปแบบของไฟล์ข้อความหรือไฟล์ ไบนารี:

ไฟล์ข้อความ

ไฟล์ข้อความคือไฟล์ปกติที่มีข้อมูล ซึ่งจัดเก็บไว้ในข้อความรูปแบบ ASCII และสามารถอ่านได้โดยผู้ใช้งาน คุณสามารถแสดง และพิมพ์ไฟล์เหล่านี้ได้ บรรทัดของไฟล์ข้อความต้องไม่มีอักขระ NUL และต้องยาวไม่เกิน {LINE_MAX} ไบต์ รวมถึง อักขระบรรทัดใหม่

คำว่า *ไฟล์ข้อความ* ไม่ได้ป้องกัน การรวมอักขระควบคุมหรืออักขระที่พิมพ์ไม่ได้อื่น (นอกเหนือ จาก NUL) ด้วยเหตุนี้ ยูทิลิตี้มาตรฐานที่แสดงรายการไฟล์ข้อความ เป็นอินพุตหรือเอาต์พุตจึงสามารถประมวลผล อักขระพิเศษ หรือ สามารถอธิบายข้อจำกัดของตนภายในแต่ละส่วนของตนได้อย่างชัดเจน

ไฟล์ไบนารี

ไฟล์ไบนารีคือไฟล์ปกติที่มีข้อมูล ซึ่งอ่านได้โดยคอมพิวเตอร์ ไฟล์ไบนารีอาจเป็นไฟล์ปฏิบัติการที่สั่งให้ระบบทำงานจนบรรลุผลสำเร็จ คำสั่งและโปรแกรมต่างๆ มีการจัดเก็บไว้ในไฟล์ไบนารี ปฏิบัติการโปรแกรมคอมไพล์พิเศษจะแปลงข้อความ ASCII เป็นไบนารีไค้ด

ไฟล์ข้อความและไบนารีแตกต่างกันเพียงอย่างเดียวคือ ไฟล์ข้อความมีบรรทัดน้อยกว่า {LINE_MAX} ไบต์ และไม่มี อักขระ NUL โดยแต่ละบรรทัดสิ้นสุดโดยอักขระบรรทัดใหม่

ไฟล์ไตเร็กทอรี

ไฟล์ไตเร็กทอรีมีข้อมูลที่ระบบ ต้องการเพื่อเข้าถึงไฟล์ทุกชนิด แต่ไฟล์ไตเร็กทอรีไม่มีข้อมูล ไฟล์ที่แท้จริง ส่งผลให้ไฟล์ไตเร็กทอรีใช้พื้นที่ว่างน้อยกว่าไฟล์ปกติ และช่วยให้โครงสร้างระบบไฟล์มีความยืดหยุ่นและความลึก แต่ละรายการ ไตเร็กทอรีแสดงถึงไฟล์ หรือไตเร็กทอรีย่อย อย่างไม่อย่างหนึ่ง แต่ละรายการมี ชื่อไฟล์และหมายเลขอ้างอิง

โหนดดัชนีของไฟล์ (หมายเลขไอ-โหนด) หมายเลขไอ-โหนดชี้ไปยังโหนดดัชนีเฉพาะที่กำหนดให้กับไฟล์ หมายเลขไอ-โหนด อธิบายที่ตั้งของข้อมูลซึ่งเชื่อมโยงกับไฟล์ ไดรฟ์ทอริถูกสร้างขึ้นและควบคุมโดยชุดคำสั่งที่แยกต่างหาก

ไฟล์พิเศษ

ไฟล์พิเศษกำหนดอุปกรณ์สำหรับระบบ หรือ เป็นไฟล์ชั่วคราวที่สร้างขึ้นโดยโปรเซส ชนิดพื้นฐานของไฟล์พิเศษ คือ FIFO (first-in, first-out), บล็อก, และอักขระ ไฟล์ FIFO เรียกอีกอย่างว่า *ไพพ์* ไฟล์ถูกสร้างขึ้นโดยโปรเซสหนึ่ง เพื่ออนุญาตการสื่อสารกับโปรเซสอื่น เป็นการชั่วคราว ไฟล์เหล่านี้สิ้นสุดลงเพื่อออกเมื่อโปรเซส แรกเสร็จสิ้น ไฟล์ บล็อกและอักขระกำหนดอุปกรณ์

ทุกไฟล์มีชุดของสิทธิอนุญาต (เรียกว่า *โหมด การเข้าถึง*) ที่กำหนดว่าใครสามารถอ่าน แก้ไข หรือดำเนินการไฟล์ได้ **หลักการที่เกี่ยวข้อง:**

“โหมดการเข้าถึงไฟล์และไดเรกทอรี” ในหน้า 323

ทุกไฟล์มีเจ้าของ สำหรับไฟล์ใหม่ ผู้ใช้ที่ สร้างไฟล์เป็นเจ้าของของไฟล์นั้น เจ้าของกำหนด *โหมดการเข้าถึง* ให้กับไฟล์ โหมดการเข้าถึงมอบสิทธิการอ่าน การแก้ไข หรือการดำเนินการในไฟล์ ให้แก่ผู้ใช้ระบบรายอื่น เฉพาะเจ้าของไฟล์หรือผู้ใช้ที่มีสิทธิการใช้งานรากเท่านั้น สามารถเปลี่ยนโหมดการเข้าถึงของไฟล์ได้

ระเบียบการตั้งชื่อไฟล์:

ชื่อของแต่ละไฟล์ต้องไม่ซ้ำกันภายในไดเรกทอรี ซึ่งจัดเก็บ ลักษณะเช่นนี้ทำให้มั่นใจว่าไฟล์มีชื่อพาทที่ไม่ซ้ำกัน ในระบบไฟล์

แนวทางการตั้งชื่อไฟล์มีดังนี้:

- ชื่อไฟล์สามารถยาวได้สูงสุด 255 อักขระและสามารถมีตัวอักษร ตัวเลข และขีดเส้นใต้
- ระบบปฏิบัติการคำนึงถึงตัวพิมพ์ ซึ่งหมายความว่ามีการแยกความแตกต่าง ระหว่างตัวอักษรพิมพ์ใหญ่และพิมพ์เล็กในชื่อไฟล์ ดังนั้น FILEA, FiLea, และ filea จึงเป็นชื่อไฟล์สามชื่อที่แตกต่างกัน แม้ว่า ตั้งอยู่ในไดเรกทอรีเดียวกันก็ตาม
- ชื่อไฟล์ควรเป็นคำอธิบายและมีความหมายเท่าที่เป็นไปได้
- ไดเรกทอรีใช้ระเบียบการตั้งชื่อเดียวกันกับไฟล์
- อักขระบางตัวมีความหมายพิเศษในระบบปฏิบัติการ หลีกเลี่ยงการใช้อักขระเหล่านี้เมื่อคุณตั้งชื่อไฟล์ อักขระเหล่านี้ รวมถึงต่อไปนี้:

/ \ " ' * ; - ? [] () ~ ! \$ { } < > # @ & | space tab newline

- ชื่อไฟล์ถูกซ่อนไว้จากรายการไดเรกทอรีปกติถ้าชื่อไฟล์ขึ้นต้น ด้วยจุด (.). เมื่อป้อนคำสั่ง ls พร้อมด้วยแฟล็ก -a ไฟล์ที่ซ่อนไว้จะแสดงขึ้นพร้อมกับ ไฟล์และไดเรกทอรีปกติ

ชื่อพาทไฟล์:

ชื่อพาทสำหรับแต่ละไฟล์และไดเรกทอรีในระบบไฟล์ประกอบด้วย ชื่อของทุกไดเรกทอรีที่อยู่ก่อนหน้าไฟล์และไดเรกทอรีนั้น ในโครงสร้างแผนผัง

เนื่องจากทุกพาทในระบบไฟล์มีต้นกำเนิดมาจากไดเรกทอรี / (root) แต่ละไฟล์ในระบบไฟล์จึงมีความสัมพันธ์ที่ไม่ซ้ำกันกับไดเรกทอรีราก ซึ่งรู้จักกันว่าเป็น *ชื่อพาทสมบูรณ์* ชื่อพาทสมบูรณ์ขึ้นต้นด้วยสัญลักษณ์ทับ (/) ตัวอย่างเช่น ชื่อพาทสมบูรณ์ของไฟล์ h อาจเป็น /B/C/h โปรดสังเกตว่าสามารถมีไฟล์ที่ชื่อ h สองไฟล์อยู่ในระบบ เนื่องจากพาทสมบูรณ์ไปยังสองไฟล์แตก

การจับคู่รูปแบบโดยใช้เมตาอักขระเซลล์ []:

เมตาอักขระนำเสนอสัญลักษณ์ wildcard อีกชนิดหนึ่งโดยการใส่อักขระที่ต้องการไว้ภายใน [] ซึ่งคล้ายกับการใช้ ? แต่คุณสามารถเลือกอักขระเฉพาะที่จะจับคู่ได้

[] ยังช่วยให้คุณระบุช่วงของค่าได้ โดยใช้ตัวยึด (-) เมื่อต้องการระบุตัวอักษรทั้งหมดในตัวอักษร ให้ใช้ [[:alpha:]] เมื่อต้องการระบุตัวอักษรพิมพ์เล็กทั้งหมดในตัวอักษร ให้ใช้ [[:lower:]]

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการอ้างอิงเฉพาะไฟล์ที่ลงท้ายด้วย 1 หรือ 2 ให้ใช้:

```
*file[12]
```

ไฟล์ที่เลือกจะเป็น: afile1, afile2, file1, และ file2

- เมื่อต้องการอ้างอิงเฉพาะไฟล์ที่ขึ้นต้นด้วยตัวเลขใดๆ ให้ใช้:

```
[0123456789]* or [0-9]*
```

ไฟล์ที่เลือกจะเป็น: 1test และ 2test

- เมื่อต้องการอ้างอิงเฉพาะไฟล์ที่ไม่ได้ขึ้นต้นด้วย a ให้ใช้:

```
[!a]*
```

ไฟล์ที่เลือกจะเป็น: 1test, 2test, bfile1, file, file1, file10, file2, และ file3

การจับคู่รูปแบบกับนิพจน์ปกติ:

นิพจน์ปกติช่วยให้คุณเลือกสตริงเฉพาะ จากชุดของสตริงอักขระ โดยทั่วไป การใช้นิพจน์ปกติ เชื่อมโยงกับการประมวลผลข้อความ

นิพจน์ปกติสามารถแสดงแทนสตริงที่เป็นไปได้จำนวนมาก ในขณะที่นิพจน์ปกติจำนวนมากสามารถมีการตีความในลักษณะที่แตกต่างกัน ขึ้นอยู่กับไคลเอนต์ปัจจุบัน คุณลักษณะความเป็นสากลนำเสนอความคล้ายกันของ คอนเท็กซ์ระหว่างไคลเอนต์ต่างๆ

โปรดดูตัวอย่างในการเปรียบเทียบต่อไปนี้:

Pattern Matching	Regular Expression
*	.*
?	.
[!a]	[^a]
[abc]	[abc]
[[:alpha:]]	[[:alpha:]]

ให้ดูที่คำสั่ง awk ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

การจัดการไฟล์

มีหลายวิธีในการทำงานกับไฟล์บนระบบของคุณ โดยปกติ คุณสร้างไฟล์ข้อความด้วยโปรแกรมแก้ไขข้อความ

โปรแกรมแก้ไขที่ใช้กันทั่วไปในสถานะแวดล้อม UNIX คือ vi และ ed เนื่องจากมีโปรแกรมแก้ไขข้อความ อยู่หลายโปรแกรม คุณจึงสามารถเลือกโปรแกรมแก้ไขที่คุณต้องการใช้ได้

คุณยังสามารถสร้างไฟล์โดยใช้การเปลี่ยนทิศทางการพิมพ์และเอาต์พุตได้ด้วย คุณสามารถ ส่งเอาต์พุตของคำสั่งไปยังไฟล์ใหม่ หรือผนวกเข้ากับไฟล์ที่มีอยู่

หลังจากสร้างและแก้ไขไฟล์แล้ว คุณอาจต้องคัดลอกหรือย้ายไฟล์ จากไดเรกทอรีหนึ่งไปยังอีกไดเรกทอรีหนึ่ง เปลี่ยนชื่อไฟล์ เพื่อให้แตกต่างจาก เวอร์ชันอื่นของไฟล์ หรือตั้งชื่ออื่นให้กับไฟล์เดียวกัน คุณยังอาจต้อง สร้างไดเรกทอรีเมื่อทำงานบนโครงการอื่นด้วย

นอกจากนี้ คุณอาจต้องลบบางไฟล์ ไดเรกทอรีของคุณอาจยุ่งเหยิงได้อย่างรวดเร็ว ด้วยไฟล์ที่มีข้อมูลเก่าหรือที่ไม่มีประโยชน์ ในการรีลีส พื้นที่ทางหน่วยเก็บระบบของคุณ ตรวจสอบให้แน่ใจว่าคุณลบไฟล์ที่ไม่ต้องการ อีกต่อไป

หลักการที่เกี่ยวข้อง:

“การเปลี่ยนทิศทางการพิมพ์และเอาต์พุต” ในหน้า 375

ระบบปฏิบัติการ AIX อนุญาตให้คุณจัดวางอินพุตและเอาต์พุต (I/O) ของข้อมูลเข้าและ ออกจากระบบของคุณได้ โดยใช้คำสั่ง I/O และสัญลักษณ์เฉพาะ

การลบไฟล์ (คำสั่ง rm):

ใช้คำสั่ง `rm` เพื่อลบไฟล์ซึ่งคุณไม่ต้องการ อีกต่อไป

คำสั่ง `rm` ลบรายการของไฟล์ที่ระบุ กลุ่มของไฟล์ หรือ ไฟล์ที่เลือกบางไฟล์ออกจากรายการภายในไดเรกทอรี ไม่จำเป็นต้องมีการยืนยันจากผู้ใช้ สิทธิการอ่าน และสิทธิการเขียน ก่อนไฟล์ถูกลบออกเมื่อคุณใช้คำสั่ง `rm` อย่างไรก็ตาม คุณต้องมีสิทธิการเขียนในไดเรกทอรีที่มีไฟล์

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง `rm`:

- เมื่อต้องการลบไฟล์ที่ชื่อ `myfile` ให้พิมพ์ดังต่อไปนี้:
`rm myfile`
- เมื่อต้องการลบไฟล์ทั้งหมดในไดเรกทอรี `mydir` ทีละหนึ่งไฟล์ ให้พิมพ์ดังต่อไปนี้:
`rm -i mydir/*`

หลังจาก ชื่อไฟล์แต่ละชื่อแสดงขึ้น ให้พิมพ์ `y` และกด `Enter` เพื่อ ลบไฟล์ หรือถ้าต้องการเก็บไฟล์ไว้ ให้เพียงแต่กด `Enter`

ให้ดูที่คำสั่ง `rm` ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

การย้ายและการเปลี่ยนชื่อไฟล์ (คำสั่ง mv):

ใช้คำสั่ง `mv` เพื่อย้ายไฟล์และไดเรกทอรี จากไดเรกทอรีหนึ่งไปยังอีกไดเรกทอรีหนึ่ง หรือเพื่อเปลี่ยนชื่อไฟล์หรือไดเรกทอรี หากคุณย้ายไฟล์หรือไดเรกทอรีไปยังไดเรกทอรีใหม่โดยไม่ได้รับชื่อใหม่ ไฟล์หรือไดเรกทอรีจะยังคงชื่อเดิมไว้

ข้อควรสนใจ: คำสั่ง `mv` สามารถเขียนทับ ไฟล์ที่มีอยู่จำนวนมากได้ ยกเว้นว่าคุณระบุแฟล็ก `-i` แฟล็ก `-i` พร้อมต์ให้คุณยืนยันก่อนจะเขียนทับไฟล์ แฟล็ก `-f` ไม่ได้ พร้อมต์คุณ หากระบุทั้งแฟล็ก `-f` และ `-i` แฟล็กล่าสุดที่ระบุมีความสำคัญมากกว่า

การย้ายไฟล์โดยใช้คำสั่ง `mv`

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง mv:

- เมื่อต้องการย้ายไฟล์ไปยังไดเรกทอรีอื่นและตั้งชื่อใหม่ให้กับไฟล์ให้พิมพ์ดังต่อไปนี้:

```
mv intro manual/chap1
```

คำสั่งนี้ย้ายไฟล์ intro ไปยังไดเรกทอรี manual/chap1 ชื่อ intro ถูกลบออกจากไดเรกทอรีปัจจุบัน และไฟล์เดียวกันปรากฏขึ้นเป็น chap1 ในไดเรกทอรี manual

- เมื่อต้องการย้ายไฟล์ไปยังไดเรกทอรีอื่นและรักษาชื่อเดิมไว้ให้พิมพ์ดังต่อไปนี้:

```
mv chap3 manual
```

คำสั่งนี้ย้าย chap3 ไปยัง manual/chap3

การเปลี่ยนชื่อไฟล์โดยใช้คำสั่ง mv

ใช้คำสั่ง mv เพื่อเปลี่ยนชื่อของไฟล์โดยไม่ย้ายไฟล์ไปยังไดเรกทอรีอื่น

เมื่อต้องการเปลี่ยนชื่อไฟล์ให้พิมพ์ดังต่อไปนี้:

```
mv appendix apndx.a
```

คำสั่งนี้เปลี่ยนชื่อไฟล์ appendix เป็น apndx.a ถ้ามีไฟล์ชื่อ apndx.a อยู่แล้ว เนื้อหาเก่าจะถูกแทนที่ด้วยเนื้อหาของไฟล์ appendix

ให้ดูที่คำสั่ง mv ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

การคัดลอกไฟล์ (คำสั่ง cp):

ใช้คำสั่ง cp เพื่อสร้างสำเนาของเนื้อหาไฟล์ หรือไดเรกทอรีที่ระบุโดยพารามิเตอร์ *SourceFile* หรือ *SourceDirectory* เข้าในไฟล์ หรือไดเรกทอรีที่ระบุโดยพารามิเตอร์ *TargetFile* หรือ *TargetDirectory*

หากมีไฟล์ที่ระบุเป็น *TargetFile* สำเนาจะเขียนทับเนื้อหาเดิมของไฟล์โดยไม่มีคำเตือน ถ้าคุณกำลังคัดลอกมากกว่าหนึ่ง *SourceFile* เป้าหมายต้องเป็นไดเรกทอรี

หากไฟล์ที่มีชื่อเดียวกันมีอยู่ที่ปลายทางใหม่ ไฟล์ที่คัดลอกจะเขียนทับไฟล์ที่ปลายทางใหม่ ด้วยเหตุนี้วิธีปฏิบัติที่ดีคือกำหนดชื่อ *ใหม่* ให้กับสำเนาของไฟล์เพื่อให้แน่ใจว่าไม่มีไฟล์ที่มีชื่อเดียวกัน อยู่ในไดเรกทอรีปลายทาง

เพื่อวางสำเนาของ *SourceFile* เข้าในไดเรกทอรีให้ระบุพารามิเตอร์ไปยังไดเรกทอรีที่มีอยู่สำหรับพารามิเตอร์ *TargetDirectory* ไฟล์รักษาชื่อตามลำดับของไฟล์ไว้ เมื่อถูกคัดลอกไปยังไดเรกทอรี ยกเว้นว่าคุณระบุชื่อไฟล์ใหม่ที่ ตอนท้ายของพารามิเตอร์ คำสั่ง cp ยังคัดลอกทั้งไดเรกทอรี เข้าในไดเรกทอรีอื่นถ้าคุณระบุแฟล็ก -r หรือ -R

คุณยังสามารถคัดลอกไฟล์อุปกรณ์พิเศษโดยใช้แฟล็ก -R การระบุ -R ทำให้ไฟล์พิเศษถูกสร้างขึ้นใหม่ภายใต้ชื่อพารามิเตอร์ การระบุแฟล็ก -r ทำให้คำสั่ง cp พยายามคัดลอกไฟล์พิเศษไปยังไฟล์ปกติ

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง cp:

- เมื่อต้องการจัดทำสำเนาของไฟล์ในไดเรกทอรีปัจจุบันให้พิมพ์ดังต่อไปนี้:

```
cp prog.c prog.bak
```

คำสั่งนี้ คัดลอก prog.c ไปยัง prog.bak หากไฟล์ prog.bak ยังไม่มีอยู่ คำสั่ง cp จะสร้างไฟล์ขึ้น หากมีอยู่แล้ว คำสั่ง cp จะแทนที่ไฟล์ ด้วยสำเนาของไฟล์ prog.c

- เมื่อต้องการคัดลอกไฟล์ในไดเรกทอรีปัจจุบันของคุณไปยังไดเรกทอรีอื่น ให้พิมพ์ ดังต่อไปนี้:

```
cp jones /home/nick/clients
```

คำสั่งนี้คัดลอกไฟล์ jones ไปยัง /home/nick/clients/jones

- เมื่อต้องการคัดลอกไฟล์ทั้งหมดในไดเรกทอรีไปยังไดเรกทอรีใหม่ ให้พิมพ์ดังต่อไปนี้:

```
cp /home/janet/clients/* /home/nick/customers
```

คำสั่งนี้คัดลอกเฉพาะไฟล์ในไดเรกทอรี clients ไปยังไดเรกทอรี customers

- เมื่อต้องการคัดลอกชุดของไฟล์เฉพาะไปยังไดเรกทอรีอื่น ให้พิมพ์ดังต่อไปนี้:

```
cp jones lewis smith /home/nick/clients
```

คำสั่งนี้คัดลอกไฟล์ jones, lewis และ smith ในไดเรกทอรีการทำงาน ปัจจุบันของคุณไปยังไดเรกทอรี /home/nick/clients

- เมื่อต้องการใช้อักขระการจับคู่รูปแบบเพื่อคัดลอกไฟล์ ให้พิมพ์ดังต่อไปนี้:

```
cp programs/*.c .
```

คำสั่งนี้ คัดลอกไฟล์ในไดเรกทอรี programs ที่ลงท้ายด้วย .c ไปยัง ไดเรกทอรีปัจจุบันซึ่งบ่งชี้โดยจุดเดี่ยว (.). คุณต้องพิมพ์ช่องว่างระหว่าง c และจุดตัวสุดท้าย

ให้ดูที่คำสั่ง cp ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

การค้นหาไฟล์ (คำสั่ง find):

ใช้คำสั่ง find เพื่อค้นหาซ้ำในแผนผังไดเรกทอรีสำหรับแต่ละ พาร ที่ระบุ เพื่อหาไฟล์ที่ตรงกับนิพจน์ บูลีนซึ่งเขียนโดยใช้เงื่อนไขที่กำหนดในข้อความต่อไปนี้

เอาต์พุตจากคำสั่ง find ขึ้นกับประโยคที่ระบุโดยพารามิเตอร์ *Expression*

ข้อมูลต่อไปนี้ เป็นตัวอย่างวิธีการใช้คำสั่ง find:

- เมื่อต้องการแสดงรายการไฟล์ทั้งหมดในระบบไฟล์ที่มีชื่อว่า .profile ให้พิมพ์ดังต่อไปนี้:

```
find / -name .profile
```

คำสั่งนี้ ค้นหาในระบบไฟล์ทั้งระบบและเขียนชื่อพารที่สมบูรณ์ของไฟล์ ที่ชื่อ .profile ทั้งหมด เครื่องหมายทับ (/) บอกให้คำสั่ง find ค้นหาไดเรกทอรี (/ราก) และไดเรกทอรีย่อยทั้งหมด

เพื่อประหยัดเวลา ควรจำกัดการค้นหาโดยระบุ ไดเรกทอรีซึ่งคุณคิดว่าอาจจะมีไฟล์อยู่

- เมื่อต้องการแสดงรายการไฟล์ที่มีไค้สิทธิ์อนุญาตเฉพาะเป็น 0600 ใน แผนผังไดเรกทอรีปัจจุบัน ให้พิมพ์ดังต่อไปนี้:

```
find . -perm 0600
```

คำสั่งนี้ แสดงรายชื่อไฟล์ที่มีสิทธิ์อนุญาตการอ่านระดับเจ้าของและการเขียนระดับ เจ้าของ เท่านั้น จุด (.) บอกให้คำสั่ง find ค้นหาไดเรกทอรีปัจจุบันและไดเรกทอรีย่อย หากต้องการคำอธิบายเกี่ยวกับ ไค้สิทธิ์อนุญาต ให้ดูที่คำสั่ง chmod

- เมื่อต้องการค้นหาหลายไดเรกทอรีเพื่อหาไฟล์ที่มีไค้สิทธิ์อนุญาต เฉพาะ ให้พิมพ์ดังต่อไปนี้:

```
find manual clients proposals -perm -0600
```

คำสั่งนี้ แสดงรายชื่อไฟล์ที่มีสิทธิอนุญาตการอ่านระดับเจ้าของและการเขียนระดับเจ้าของ และสิทธิอนุญาตอื่นที่เป็นไปได้ ไดรเร็กทอรี manual, clients และ proposals และไดเร็กทอรีย่อยจะถูกค้นหาในตัวอย่างก่อนหน้านี -perm 0600 เลือกเฉพาะไฟล์ที่มีโค้ดสิทธิอนุญาตที่ตรงกับ 0600 เท่านั้น ในตัวอย่างนี้ -perm -0600 เลือกไฟล์ที่มีโค้ดสิทธิอนุญาตที่อนุญาตการเข้าถึงที่ระบุโดย 0600 และการเข้าถึงอื่นเหนือระดับ 0600 และยังคงตรงกับโค้ดสิทธิอนุญาต 0622 และ 2744

- เมื่อต้องการแสดงไฟล์ทั้งหมดในไดเร็กทอรีปัจจุบันที่ถูกเปลี่ยนในระหว่างระยะเวลา 24-ชั่วโมงปัจจุบัน ให้พิมพ์ดังต่อไปนี้:

```
find . -ctime 1
```

- เมื่อต้องการค้นหาไฟล์ปกติที่มีหลายลิงก์ให้พิมพ์ดังต่อไปนี้:

```
find . -type f -links +1
```

คำสั่งนี้ แสดงรายชื่อของไฟล์ปกติ (-type f) ที่มี มากกว่าหนึ่งลิงก์ (-links +1)

หมายเหตุ: ทุกไดเร็กทอรี มีอย่างน้อยสองลิงก์: รายการในไดเร็กทอรีพาเรนต์และตัวไดเร็กทอรีเอง. (จุด) สำหรับข้อมูลเพิ่มเติมเกี่ยวกับลิงก์ไฟล์ โปรดดูคำสั่ง **ln**

- เมื่อต้องการค้นหาไฟล์ทั้งหมดที่มีความยาว 414 ไบต์พอดีให้พิมพ์ ดังต่อไปนี้:

```
find . -size 414c
```

ให้ดูที่คำสั่ง **find** ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงชนิดไฟล์ (คำสั่ง file):

ใช้คำสั่ง **file** เพื่ออ่านไฟล์ที่ระบุโดย พารามิเตอร์ *File* หรือ **-fFileList** ทำชุดของการทดสอบในแต่ละไฟล์ และพยายามจัดประเภทไฟล์ตามชนิด จากนั้นคำสั่งจะเขียนชนิดไฟล์ไปยังเอาต์พุต มาตรฐาน

หากปรากฏว่าไฟล์เป็น ASCII คำสั่ง **file** จะตรวจสอบ 512 ไบต์แรกและกำหนดภาษาของไฟล์ หากไฟล์ไม่ได้ อยู่ในรูปแบบ ASCII คำสั่ง **file** จะพยายามต่อไปเพื่อกำหนดว่าไฟล์ เป็นข้อมูลไบนารีหรือไฟล์ข้อความที่มี อักขระส่วนขยาย

หากพารามิเตอร์ *File* ระบุ โมดูลไฟล์ที่รันได้หรือที่เป็นอ็อบเจกต์ และหมายเลขเวอร์ชันมากกว่า 0 คำสั่ง **file** จะแสดงเวลาประทับของเวอร์ชัน

คำสั่ง **file** ใช้ไฟล์ `/etc/magic` เพื่อระบุไฟล์ที่มีหมายเลขวิเศษ นั่นคือไฟล์ใดๆ ที่มีค่าคงที่ตัวเลขหรือสตริงซึ่งบ่งชี้ชนิด

ข้อมูลต่อไปนี้ เป็นตัวอย่างวิธีการใช้คำสั่ง **file**:

- เมื่อต้องการแสดงชนิดของข้อมูลที่มีในไฟล์ชื่อ `myfile` ให้พิมพ์ดังต่อไปนี้:

```
file myfile
```

คำสั่งนี้ แสดงชนิดไฟล์ของ `myfile` (เช่น ไดรเร็กทอรี, ข้อมูล, ข้อความ ASCII, ซอร์สโปรแกรม C, หรือที่เก็บถาวร)

- เมื่อต้องการแสดงชนิดของแต่ละไฟล์ที่มีชื่ออยู่ในไฟล์ `filenames.lst` ซึ่งมีรายการของชื่อไฟล์ ให้พิมพ์ดังต่อไปนี้:

```
file -f filenames.lst
```

คำสั่งนี้ แสดงชนิดของแต่ละไฟล์ที่มีชื่ออยู่ในไฟล์ `filenames.lst` ชื่อไฟล์แต่ละชื่อต้องแสดงบนบรรทัดแยกต่างหากกัน

- เมื่อต้องการสร้างไฟล์ `filenames.lst` ที่มีชื่อไฟล์ทั้งหมด ในไดเรกทอรีปัจจุบัน ให้พิมพ์ดังต่อไปนี้:

```
ls > filenames.lst
```

แก้ไขไฟล์ `filenames.lst` ตามต้องการ

ให้ดูที่คำสั่ง `file` ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

คำสั่งสำหรับการแสดงเนื้อหาไฟล์ (คำสั่ง `pg`, `more`, `page`, และ `cat`):

คำสั่ง `pg`, `more`, และ `page` ช่วยให้คุณสามารถดูเนื้อหาของไฟล์และควบคุมความเร็วในการแสดงไฟล์ ของคุณได้

คุณยังสามารถใช้คำสั่ง `cat` เพื่อแสดงเนื้อหาของไฟล์ตั้งแต่หนึ่งไฟล์ขึ้นไปบนหน้าจอของคุณ การรวมคำสั่ง `cat` เข้ากับ คำสั่ง `pg` ช่วยให้คุณสามารถอ่านเนื้อหาของไฟล์บนหน้าจอเต็มทีละหน้าจอดี

คุณยังสามารถแสดงเนื้อหาของไฟล์โดยใช้การเปลี่ยนทิศทางอินพุตและเอาต์พุตได้ด้วย

หลักการที่เกี่ยวข้อง:

“การเปลี่ยนทิศทางอินพุตและเอาต์พุต” ในหน้า 375

ระบบปฏิบัติการ AIX อนุญาตให้คุณจัดวางอินพุตและเอาต์พุต (I/O) ของข้อมูลเข้าและ ออกจากระบบของคุณได้ โดยใช้คำสั่ง I/O และสัญลักษณ์เฉพาะ

การใช้คำสั่ง `pg`:

ใช้คำสั่ง `pg` เพื่ออ่านไฟล์ที่ระบุใน พารามิเตอร์ `File` และเขียนไฟล์นั้นไปยังเอาต์พุตมาตรฐาน ทีละหน้าจอดี

หากคุณระบุติ๊กด (-) เป็นพารามิเตอร์ `File` หรือรันคำสั่ง `pg` โดยไม่มีอ็อปชัน คำสั่ง `pg` จะอ่านอินพุตมาตรฐาน แต่ละหน้าจอดีพร้อมตามหลัง ถ้าคุณกดปุ่ม `Enter` หน้าจออื่นจะแสดงขึ้น คำสั่งย่อยที่ใช้ร่วมกับคำสั่ง `pg` อนุญาตให้คุณทบทวนเนื้อหาที่ผ่านไปแล้วได้

ตัวอย่างเช่น เมื่อต้องการดูเนื้อหาของไฟล์ `myfile` ทีละหน้า ให้พิมพ์ ดังต่อไปนี้:

```
pg myfile
```

ให้ดูที่คำสั่ง `pg` ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

การใช้คำสั่ง `more` หรือ `page`:

ใช้คำสั่ง `more` หรือ `page` เพื่อแสดงข้อความต่อเนื้อหนึ่งหน้าจอดีในแต่ละครั้ง

คำสั่งจะหยุดชั่วคราวหลังจากแต่ละหน้าจอดี และพิมพ์ `filename` และ เปอร์เซ็นต์ความสมบูรณ์ (ตัวอย่างเช่น `myfile (7%)`) ที่ด้านล่าง ของหน้าจอ จากนั้น ถ้าคุณกดปุ่ม `Enter` คำสั่ง `more` จะแสดงบรรทัดเพิ่มเติม ถ้าคุณกด `Spacebar` คำสั่ง `more` จะแสดงหน้าจออื่นของข้อความ

หมายเหตุ: บนเทอร์มินัลบางโมเดล คำสั่ง `more` จะล้าง หน้าจอแทนการเลื่อนหน้าจอ ก่อนการแสดงผลหน้าจอถัดไปของ ข้อความ

ตัวอย่างเช่น เมื่อต้องการดูไฟล์ที่ชื่อ `myfile` ให้พิมพ์ดังต่อไปนี้:

```
more myfile
```

กด Spacebar เพื่อดูหน้าจอดีไป

ให้ดูที่คำสั่ง `more` ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

คำสั่ง `cat`:

ใช้คำสั่ง `cat` เพื่ออ่านแต่ละพารามิเตอร์ *File* ตามลำดับ และเขียนไปยังเอาต์พุตมาตรฐาน

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการแสดงเนื้อหาของไฟล์ `notes` ให้พิมพ์ ดังต่อไปนี้:

```
cat notes
```

หากไฟล์ยาวมากกว่า 24 บรรทัด บางบรรทัดจะเลยหน้าจอ ในการแสดงไฟล์ ครั้งละหนึ่งหน้า ให้ใช้คำสั่ง `pg`

- เมื่อต้องการแสดงเนื้อหาของไฟล์ `notes`, `notes2`, และ `notes3` ให้พิมพ์ดังต่อไปนี้:

```
cat notes notes2 notes3
```

ให้ดูที่คำสั่ง `cat` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

การค้นหาสตริงข้อความภายในไฟล์ (คำสั่ง `grep`):

ใช้คำสั่ง `grep` เพื่อค้นหาไฟล์ที่ระบุ เพื่อหารูปแบบซึ่งระบุโดยพารามิเตอร์ *Pattern* และเขียนแต่ละ บรรทัดที่ตรงกันไปยังเอาต์พุตมาตรฐาน

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง `grep`:

- เมื่อต้องการค้นหาในไฟล์ที่ชื่อ `pgm.s` เพื่อหารูปแบบที่มีอักขระการจับคู่รูปแบบบางอย่าง `*`, `^`, `?`, `[,]`, `\(, \)`, `\{, \}`, และ `\|` ในกรณีนี้คือบรรทัดที่ขึ้นต้นด้วยตัวอักษรพิมพ์เล็กหรือพิมพ์ใหญ่ ให้พิมพ์ดังต่อไปนี้:

```
grep "[a-zA-Z]" pgm.s
```

คำสั่งนี้ แสดงบรรทัดทั้งหมดในไฟล์ `pgm.s` ที่ขึ้นต้นด้วย ตัวอักษร

- เมื่อต้องการแสดงบรรทัดทั้งหมดในไฟล์ที่ชื่อ `sort.c` ซึ่ง ไม่ตรงกับรูปแบบเฉพาะ ให้พิมพ์ดังต่อไปนี้:

```
grep -v bubble sort.c
```

คำสั่งนี้ แสดงบรรทัดทั้งหมดที่ไม่มีคำว่า `bubble` ใน ไฟล์ `sort.c`

- เมื่อต้องการแสดงบรรทัดในเอาต์พุตของคำสั่ง `ls` ที่ตรงกับ สตริง `staff` ให้พิมพ์ดังต่อไปนี้:

```
ls -l | grep staff
```

ให้ดูที่คำสั่ง `grep` ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

การเรียงลำดับไฟล์ข้อความ (คำสั่ง `sort`):

ใช้คำสั่ง `sort` เพื่อเรียงลำดับบรรทัดตามตัวอักษร ในไฟล์ที่ระบุโดยพารามิเตอร์ *File* และเขียนผลลัพธ์ไปยังเอาต์พุตมาตรฐาน

หากพารามิเตอร์ *File* ระบุมากกว่า หนึ่งไฟล์ คำสั่ง `sort` จะต่อไฟล์เข้าด้วยกัน และเรียงตามตัวอักษรเป็นไฟล์เดียว

หมายเหตุ: คำสั่ง `sort` เป็นแบบตรงตามตัวพิมพ์และจัดลำดับตัวอักษรพิมพ์ใหญ่ก่อนตัวพิมพ์เล็ก (ลักษณะการทำงานนี้ขึ้นอยู่กับโลแคล)

ในตัวอย่างต่อไปนี้ เนื้อหาของไฟล์ที่ชื่อ `names` คือ:

```
marta  
denise  
joyce  
endrica  
melanie
```

และเนื้อหาของไฟล์ที่ชื่อ `states` คือ:

```
texas  
colorado  
ohio
```

- เมื่อต้องการแสดงเนื้อหาที่เรียงลำดับของไฟล์ที่ชื่อ `names` ให้พิมพ์ดังต่อไปนี้:

```
sort names
```

ระบบ แสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
denise  
endrica  
joyce  
marta  
melanie
```

- เมื่อต้องการแสดงเนื้อหาที่เรียงลำดับของไฟล์ที่ชื่อ `names` และ `states` ให้พิมพ์ดังต่อไปนี้:

```
sort names states
```

ระบบ แสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
colorado  
denise  
endrica  
joyce  
marta  
melanie  
ohio  
texas
```

- เมื่อต้องการแทนที่เนื้อหาดั้งเดิมของไฟล์ที่ชื่อ `names` ด้วยเนื้อหาที่เรียงลำดับ ให้พิมพ์ดังต่อไปนี้:

```
sort -o names names
```

คำสั่งนี้ แทนที่เนื้อหาของไฟล์ `names` ด้วย ข้อมูลเดียวกันแต่มีการเรียงลำดับ

ให้ดูที่คำสั่ง `sort` ใน *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

การเปรียบเทียบไฟล์ (คำสั่ง `diff`):

ใช้คำสั่ง `diff` เพื่อเปรียบเทียบไฟล์ข้อความ สามารถเปรียบเทียบไฟล์เดี่ยวหรือเนื้อหาของไดเรกทอรี

เมื่อรันคำสั่ง diff บนไฟล์ปกติ และเมื่อคำสั่งเปรียบเทียบไฟล์ข้อความในไดเรกทอรีที่ต่างกัน คำสั่ง diff จะแจ้งถึงบรรทัดที่ต้องเปลี่ยนในไฟล์ เพื่อให้ไฟล์ตรงกัน

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง diff:

- เมื่อต้องการเปรียบเทียบสองไฟล์ ให้พิมพ์ดังต่อไปนี้:

```
diff chap1.bak chap1
```

คำสั่งนี้แสดงความแตกต่างระหว่างไฟล์ chap1.bak และ chap1

- เมื่อต้องการเปรียบเทียบสองไฟล์ในขณะที่ละเว้นความแตกต่างของจำนวนช่องว่าง ให้พิมพ์ดังต่อไปนี้:

```
diff -w prog.c.bak prog.c
```

ถ้าสองไฟล์แตกต่างกันเฉพาะในจำนวนของช่องว่างและแท็บระหว่างคำ คำสั่ง diff -w จะถือว่าทั้งสองไฟล์เหมือนกัน

ให้ดูที่คำสั่ง diff ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

การนับคำ บรรทัด และไบต์ในไฟล์ (คำสั่ง wc):

ใช้คำสั่ง wc เพื่อนับจำนวนของบรรทัด คำ และไบต์ในไฟล์ที่ระบุโดยพารามิเตอร์ *File*

หากไม่ได้ระบุไฟล์สำหรับพารามิเตอร์ *File* ระบบจะใช้ อินพุตมาตรฐาน คำสั่งเขียนผลลัพธ์ไปยังเอาต์พุตมาตรฐานและเก็บรักษาจำนวนทั้งหมดของไฟล์ที่ระบุทุกไฟล์ ถ้ามีการระบุแฟล็ก การจัดลำดับของ แฟล็กกำหนดการจัดลำดับของเอาต์พุต คำ มีการกำหนด เป็นสตริงของอักขระที่คั่นด้วยพื้นที่ว่าง แท็บ หรืออักขระบรรทัดใหม่

เมื่อมีการระบุไฟล์บนบรรทัดคำสั่ง ชื่อไฟล์จะถูกพิมพ์พร้อมกับจำนวน

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการแสดงจำนวนบรรทัด คำ และไบต์ของไฟล์ที่ชื่อ chap1 ให้พิมพ์ดังต่อไปนี้:

```
wc chap1
```

คำสั่งนี้แสดงจำนวนของบรรทัด คำ และไบต์ในไฟล์ chap1

- เมื่อต้องการแสดงเฉพาะจำนวนไบต์และคำเท่านั้น ให้พิมพ์ดังต่อไปนี้:

```
wc -cw chap*
```

คำสั่งนี้แสดงจำนวนไบต์และคำในแต่ละไฟล์ซึ่งชื่อขึ้นต้น ด้วย chap และแสดงผลรวม

ให้ดูที่คำสั่ง wc ใน *Commands Reference, Volume 6* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงผลบรรทัดแรกของไฟล์ (คำสั่ง head):

ใช้คำสั่ง head เพื่อเขียนสองสามบรรทัดแรกของแต่ละไฟล์ ที่ระบุ หรือของอินพุตมาตรฐานไปยังเอาต์พุตมาตรฐาน

หากไม่ระบุแฟล็กกับคำสั่ง head 10 บรรทัดแรกจะถูกแสดงโดยค่าดีฟอลต์

ตัวอย่างเช่น เมื่อต้องการแสดง ห้าบรรทัดแรกของไฟล์ Test ให้พิมพ์ดังต่อไปนี้:

```
head -5 Test
```

ให้ดูที่ คำสั่ง `head` ใน *Commands Reference, Volume 2* สำหรับไวยากรณ์ทั้งหมด

การแสดงบรรทัดสุดท้ายของไฟล์ (คำสั่ง `tail`):

ใช้คำสั่ง `tail` เพื่อเขียนไฟล์ที่ระบุโดยพารามิเตอร์ `File` ไปยังเอาต์พุตมาตรฐาน โดยเริ่มต้นที่จุดที่ระบุ

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการแสดง 10 บรรทัดสุดท้ายของไฟล์ `notes` ให้พิมพ์ดังต่อไปนี้:
`tail notes`
- เมื่อต้องการระบุจำนวนบรรทัดที่จะเริ่มต้นการอ่านนับจากตอนท้ายของไฟล์ `notes` ให้พิมพ์ดังต่อไปนี้:
`tail -20 notes`
- เมื่อต้องการแสดงไฟล์ `notes` ทีละหนึ่งหน้า โดยเริ่มต้นจาก ไบต์ลำดับที่ 200th ให้พิมพ์ดังต่อไปนี้:
`tail -c +200 notes | pg`
- เมื่อต้องการติดตามการเติบโตของไฟล์ที่ชื่อ `accounts` ให้พิมพ์ดังต่อไปนี้:
`tail -f accounts`

คำสั่งนี้แสดง 10 บรรทัดสุดท้ายของไฟล์ `accounts` คำสั่ง `tail` ยังคงแสดงบรรทัดต่อไปเมื่อมีการเพิ่มบรรทัดลงในไฟล์ `accounts` การแสดงยังคงดำเนินต่อไปจนกว่าคุณกดคีย์ (Ctrl-C) ตามลำดับเพื่อหยุด การแสดงผล

ให้ดูที่คำสั่ง `tail` ใน *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

การตัดส่วนของไฟล์ข้อความ (คำสั่ง `cut`):

ใช้คำสั่ง `cut` เพื่อเขียนไบต์ อักขระ หรือฟิลด์ที่เลือกจากแต่ละบรรทัดของไฟล์ไปยังเอาต์พุตมาตรฐาน

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการแสดงหลายฟิลด์ของแต่ละบรรทัดของไฟล์ ให้พิมพ์ดังต่อไปนี้:
`cut -f1,5 -d: /etc/passwd`

คำสั่งนี้แสดงฟิลด์ชื่อล็อกอิน และชื่อผู้ใช้แบบเต็มของไฟล์รหัสผ่านระบบ ซึ่งเป็นฟิลด์แรกและฟิลด์ที่ห้า (-f1,5) ที่คั่นด้วยเครื่องหมายจุดคู่ (-d:)

- ถ้าไฟล์ `/etc/passwd` มีลักษณะคล้ายเช่นนี้:
`su:*:0:0:User with special privileges:/:usr/bin/sh`
`daemon:*:1:1::/etc:`
`bin:*:2:2::usr/bin:`
`sys:*:3:3::usr/src:`
`adm:*:4:4:system administrator:/var/adm:/usr/bin/sh`
`pierre:*:200:200:Pierre Harper:/home/pierre:/usr/bin/sh`
`joan:*:202:200:Joan Brown:/home/joan:/usr/bin/sh`

คำสั่ง `cut` จัดทำ:

```
su:User with special privileges
daemon:
bin:
```

```
sys:
adm:system administrator
pierre: Pierre Harper
joan:Joan Brown
```

ให้ดูที่คำสั่ง `cut` ใน *Commands Reference, Volume 1* สำหรับไวยากรณ์ทั้งหมด

การวางส่วนของไฟล์ข้อความ (คำสั่ง `paste`):

ใช้คำสั่ง `paste` เพื่อผสานบรรทัดของไฟล์สูงสุดถึง 12 ไฟล์เข้าเป็นไฟล์เดียว

โปรดดูตัวอย่างต่อไปนี้:

- ถ้าคุณมีไฟล์ที่ชื่อ `names` ซึ่งมีข้อความต่อไปนี้:

```
rachel
jerry
mark
linda
scott
```

และอีกไฟล์หนึ่งที่ชื่อ `places` ซึ่งมีข้อความต่อไปนี้:

```
New York
Austin
Chicago
Boca Raton
Seattle
```

และอีกไฟล์หนึ่งที่ชื่อ `dates` ซึ่งมีข้อความต่อไปนี้:

```
February 5
March 13
June 21
July 16
November 4
```

เมื่อต้องการวางข้อความของไฟล์ `names`, `places` และ `dates` ไว้ด้วยกัน ให้พิมพ์ดังต่อไปนี้:

```
paste names places dates > npd
```

คำสั่งนี้สร้างไฟล์ที่ชื่อ `npd` ซึ่งประกอบด้วยข้อมูลจากไฟล์ `names` ในคอลัมน์หนึ่ง ข้อมูลจากไฟล์ `places` ในอีกคอลัมน์หนึ่ง และข้อมูลจากไฟล์ `dates` ในคอลัมน์ที่สาม ขณะนี้ไฟล์ `npd` มีข้อมูลต่อไปนี้:

```
rachel      New York      February 5
jerry       Austin        March 13
mark        Chicago       June 21
linda       Boca Raton    July 16
scott       Seattle       November 4
```

อักขระแท็บคั่น ชื่อ สถานที่ และวันที่ บนแต่ละบรรทัด คอลัมน์เหล่านี้ไม่ได้จัดแนว เนื่องจาก มีการตั้งค่าแท็บหยุดทุกคอลัมน์ที่แปด

- เมื่อต้องการคั่นคอลัมน์ด้วยอักขระอื่นที่ไม่ใช่แท็บ ให้พิมพ์ดังต่อไปนี้:

```
paste -d"!@" names places dates > npd
```

คำสั่งนี้ เปลี่ยน ! และ @ เป็นตัวค้นคอลัมน์ หากไฟล์ names, places, และ dates เหมือนกับในตัวอย่างที่ 1 ไฟล์ npd จะมีข้อมูล ต่อไปนี้:

```
rachel!New York@February 5
jerry!Austin@March 13
mark!Chicago@June 21
linda!Boca Raton@July 16
scott!Seattle@November 4
```

- เมื่อต้องการแสดงรายการโตเร็กทอรีปัจจุบันในสี่คอลัมน์ ให้พิมพ์ดังต่อไปนี้:

```
ls | paste - - - -
```

แต่ละยัติภังค์ (-) บอกคำสั่ง paste ให้ สร้างคอลัมน์ที่มีข้อมูลซึ่งอ่านมาจากอินพุตมาตรฐาน บรรทัดแรก ถูกวางไว้ในคอลัมน์แรก บรรทัดที่สองในคอลัมน์ที่สอง และต่อไป

ให้ดูที่คำสั่ง paste ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

การกำหนดหมายเลขบรรทัดในไฟล์ข้อความ (คำสั่ง nl):

ใช้คำสั่ง nl เพื่ออ่านไฟล์ที่ระบุ (อินพุตมาตรฐานโดยค่าดีฟอลต์) จำนวนบรรทัดในอินพุต และเขียน บรรทัดที่มีหมายเลขไปยังเอาต์พุตมาตรฐาน

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการกำหนดหมายเลขเฉพาะบรรทัดที่ไม่ว่างเปล่า ให้พิมพ์ดังต่อไปนี้:

```
nl chap1
```

คำสั่งนี้ แสดงรายการที่กำหนดหมายเลขของ chap1 โดยกำหนดหมายเลขเฉพาะ บรรทัดที่ไม่ว่างเปล่าในส่วนเนื้อหาเท่านั้น

- เมื่อต้องการกำหนดหมายเลขบรรทัดทั้งหมด ให้พิมพ์ดังต่อไปนี้:

```
nl -ba chap1
```

คำสั่งนี้ กำหนดหมายเลขบรรทัดทั้งหมดในไฟล์ที่ชื่อ chap1 รวมถึง บรรทัดที่ว่างเปล่า

ให้ดูที่คำสั่ง nl ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

การลบคอลัมน์ในไฟล์ข้อความ (คำสั่ง colrm):

ใช้คำสั่ง colrm เพื่อลบคอลัมน์ที่ระบุ ออกจากไฟล์ อินพุตถูกรับมาจากอินพุตมาตรฐาน เอาต์พุตถูกส่งไปยัง เอาต์พุตมาตรฐาน

หากคำสั่งถูกเรียกใช้ด้วยพารามิเตอร์หนึ่งตัว คอลัมน์ของแต่ละบรรทัด ตั้งแต่คอลัมน์ที่ระบุไปจนถึงคอลัมน์สุดท้ายจะถูกลบออก หากคำสั่ง ถูกเรียกใช้ด้วยพารามิเตอร์สองตัว คอลัมน์ตั้งแต่คอลัมน์แรกที่ระบุ ไปจนถึงคอลัมน์ที่สองที่ระบุจะถูกลบออก

หมายเหตุ: คอลัมน์ที่มีหมายเลขเริ่มต้นด้วยคอลัมน์ 1

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการลบคอลัมน์ออกจากไฟล์ text.fil ให้พิมพ์ ดังต่อไปนี้:

```
colrm 6 < text.fil
```

หาก text.fil ประกอบด้วย:

123456789

ดังนั้นคำสั่ง colrm จะแสดง:

12345

ให้ดูที่คำสั่ง colrm ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

ลิงก์ไฟล์และไดเรกทอรี

ลิงก์ คือการเชื่อมต่อระหว่างชื่อไฟล์และหมายเลขการอ้างอิง โหนดดัชนี (หมายเลขไอ-โหนด) ซึ่งใช้แสดงแทนไฟล์เป็นการภายใน เนื่องจากรายการไดเรกทอรีมีชื่อไฟล์ที่จับคู่กับหมายเลขไอ-โหนด ดังนั้นรายการไดเรกทอรีทุก รายการจึงเป็นลิงก์

หมายเลขไอ-โหนดระบุไฟล์โดยแท้จริงไม่ได้ ระบุชื่อไฟล์ โดยการใช้ลิงก์ หมายเลขไอ-โหนดหรือไฟล์ใดๆ จึงเป็นที่รู้จักแม้ว่า จะใช้ชื่อ ที่แตกต่างกันจำนวนมาก ตัวอย่างเช่น หมายเลขไอ-โหนด 798 มีบันทึกย่อ เกี่ยวกับยอดขายเดือนมิถุนายนในสำนักงาน Omaha รายการไดเรกทอรีที่แสดงสำหรับ บันทึกย่อนี้เป็นดังนี้:

หมายเลขไอ-โหนด	ชื่อไฟล์
798	บันทึกย่อ

เนื่องจากข้อมูลนี้เกี่ยวข้องกับข้อมูลที่จัดเก็บไว้ใน ไดเรกทอรี sales และ omaha จึงใช้ การลิงก์เพื่อแบ่งใช้ข้อมูลในตำแหน่งที่ จำเป็น โดยใช้คำสั่ง In ลิงก์ถูกสร้างขึ้นไปยังไดเรกทอรีเหล่านี้ ขณะนี้ ไฟล์มีชื่อไฟล์สามชื่อ ดังนี้:

หมายเลขไอ-โหนด	ชื่อไฟล์
798	memo
798	sales/june
798	omaha/junesales

เมื่อคุณใช้คำสั่ง pg หรือ cat เพื่อดูเนื้อหาของชื่อไฟล์ใดๆ ในสามชื่อนี้ ข้อมูลเดียวกัน จะแสดงขึ้น หากคุณแก้ไขเนื้อหาของ หมายเลขไอ-โหนดจากชื่อไฟล์ใดๆ ในสามชื่อนี้ เนื้อหาของข้อมูลที่แสดงโดยชื่อไฟล์ทั้งหมดจะ สะท้อนถึงการเปลี่ยนแปลง

ชนิดของลิงก์:

ลิงก์มีอยู่สองชนิดคือ: ฮาร์ดและสัญลักษณ์

ลิงก์ถูกสร้างขึ้นด้วยคำสั่ง In และเป็นชนิดต่อไปนี้:

ไอเท็ม	คำอธิบาย
ฮาร์ดลิงก์	อนุญาตให้เข้าถึงข้อมูลของไฟล์จากชื่อไฟล์ใหม่ ฮาร์ดลิงก์ ทำให้มั่นใจถึงการมีอยู่ของไฟล์ เมื่อฮาร์ดลิงก์ล่าสุดถูกลบออกไป หมายเลขไอ-โหนด และข้อมูลของฮาร์ดลิงก์นั้นจะถูกลบออก ฮาร์ดลิงก์สามารถสร้างขึ้นระหว่างไฟล์ที่อยู่ในระบบไฟล์เดียวกันเท่านั้น
ลิงก์สัญลักษณ์	อนุญาตให้เข้าถึงข้อมูลในระบบไฟล์อื่นจากชื่อไฟล์ใหม่ ลิงก์สัญลักษณ์ เป็นไฟล์ชนิดพิเศษที่มีชื่อพาร เมื่อโปรเซสพบลิงก์ สัญลักษณ์โปรเซสอาจค้นหาพารนั้น ลิงก์สัญลักษณ์ ไม่ได้ป้องกันไฟล์จากการลบออกจากระบบไฟล์

หมายเหตุ: ผู้ใช้ที่สร้างไฟล์ยังคงความเป็นเจ้าของในไฟล์นั้นไม่ว่าจะมี การสร้างลิงก์จำนวนเท่าใดก็ตาม เฉพาะเจ้าของไฟล์ หรือผู้ไ้รากเท่านั้นสามารถ ตั้งค่าโหมดเข้าถึงสำหรับไฟล์นั้นได้ อย่างไรก็ตาม สามารถทำการเปลี่ยนแปลงในไฟล์ จากชื่อไฟล์ ที่ลิงก์ด้วยโหมดการเข้าถึงที่เหมาะสม

ไฟล์หรือไดเรกทอรีมีอยู่ตรงเท่าที่มีหนึ่ง ฮาร์ดลิงก์ไปยังหมายเลขไอ-โหนดของไฟล์นั้น ในรายการฉบับสมบูรณ์ที่แสดงขึ้น โดยคำสั่ง `ls -l` มีการแสดงจำนวนของ ฮาร์ดลิงก์ไปยังแต่ละไฟล์และไดเรกทอรีย่อย ฮาร์ดลิงก์ทั้งหมดถูกจัดการ อย่างเท่าเทียมกันโดยระบบปฏิบัติการ โดยไม่คำนึงว่าลิงก์ใดถูกสร้างขึ้นก่อน

การลิงก์ไฟล์ (คำสั่ง `ln`):

การลิงก์ไฟล์กับคำสั่ง `ln` เป็นวิธีที่สะดวกในการทำงานกับข้อมูลเดียวกันราวกับว่าข้อมูลนั้นอยู่ในมากกว่าหนึ่งตำแหน่ง

ลิงก์มีการสร้างขึ้นโดยการให้ชื่ออื่น กับไฟล์ดั้งเดิม การใช้ลิงก์ช่วยให้ผู้ใช้หลายรายสามารถแบ่งใช้ไฟล์ขนาดใหญ่ เช่น ฐานข้อมูลหรือรายการส่งเมลได้ โดยไม่ต้องทำสำเนาของไฟล์ นั้น ลิงก์ไม่เพียงแต่ช่วยประหยัดพื้นที่ว่างดิสก์เท่านั้น แต่การเปลี่ยนแปลงในไฟล์หนึ่ง มีการสะท้อนในไฟล์ที่ลิงก์ทั้งหมดโดยอัตโนมัติ

คำสั่ง `ln` ลิงก์ไฟล์ที่กำหนดในพารามิเตอร์ `SourceFile` เข้ากับไฟล์ที่กำหนดโดยพารามิเตอร์ `TargetFile` หรือ เข้ากับชื่อไฟล์เดียวกันในอีกไดเรกทอรีหนึ่งที่ระบุโดยพารามิเตอร์ `TargetDirectory` โดยค่าดีฟอลต์ คำสั่ง `ln` จะสร้างฮาร์ดลิงก์ เมื่อต้องการใช้คำสั่ง `ln` เพื่อสร้างลิงก์สัญลักษณ์ให้เพิ่มแฟล็ก `-s`

หมายเหตุ: คุณไม่สามารถลิงก์ไฟล์ข้ามระบบไฟล์โดยไม่ใช้แฟล็ก `-s`

หาก คุณกำลังลิงก์ไฟล์ไปยังชื่อใหม่ คุณสามารถแสดงรายการได้เพียงไฟล์เดียวเท่านั้น หากคุณกำลัง ลิงก์ไปยังไดเรกทอรี คุณสามารถแสดงรายการได้มากกว่าหนึ่งไฟล์

พารามิเตอร์ `TargetFile` เป็นทางเลือก หากคุณไม่ได้กำหนดไฟล์เป้าหมาย คำสั่ง `ln` จะสร้างไฟล์ในไดเรกทอรีปัจจุบันของคุณ ไฟล์ใหม่ได้รับชื่อมาจาก ไฟล์ที่กำหนดในพารามิเตอร์ `SourceFile`

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการสร้างลิงก์ไปยังไฟล์ที่ชื่อ `chap1` ให้พิมพ์ดังต่อไปนี้:

```
ln -f chap1 intro
```

คำสั่งนี้ ลิงก์ `chap1` ไปยังชื่อใหม่ `intro` เมื่อใช้แฟล็ก `-f` ชื่อไฟล์ `intro` จะ ถูกสร้างขึ้นถ้ายังไม่มีอยู่ ถ้ามี `intro` ไฟล์จะถูกแทนที่ด้วยลิงก์ไปยัง `chap1` ทั้งชื่อไฟล์ `chap1` และ `intro` อ้างอิงไฟล์เดียวกัน

- เมื่อต้องการลิงก์ไฟล์ที่ชื่อ `index` ไปยังชื่อเดียวกันในไดเรกทอรีอื่นที่ชื่อ `manual` ให้พิมพ์ดังต่อไปนี้:

```
ln index manual
```

คำสั่งนี้ ลิงก์ `index` ไปยังชื่อใหม่ `manual/index`

- เมื่อต้องการลิงก์หลายไฟล์ไปยังชื่อในไดเรกทอรีอื่น ให้พิมพ์ดังต่อไปนี้:

```
ln chap2 jim/chap3 /home/manual
```

คำสั่งนี้ ลิงก์ `chap2` ไปยังชื่อใหม่ `/home/manual/chap2` และ `jim/chap3` ไปยัง `/home/manual/chap3`

- เมื่อต้องการใช้คำสั่ง `ln` กับอักขระการจับคู่รูปแบบ ให้พิมพ์ดังต่อไปนี้:

```
ln manual/* .
```

หมายเหตุ: คุณต้องพิมพ์ช่องว่างระหว่างเครื่องหมายดอกจันและเครื่องหมายมหัพภาค

สิ่งนี้จะลิงก์ไฟล์ทั้งหมดในไดเรกทอรี `manual` กับไดเรกทอรีปัจจุบัน นั่นคือ จุด (`.`) ซึ่งกำหนดให้ชื่อเดียวกันกับที่มีอยู่ในไดเรกทอรี `manual`

- เมื่อต้องการสร้างลิงก์สัญลักษณ์ ให้พิมพ์ดังต่อไปนี้:

```
ln -s /tmp/toc toc
```

คำสั่งนี้สร้างลิงก์สัญลักษณ์ `toc` ในไดเรกทอรีปัจจุบัน ไฟล์ `toc` ซึ่งไปยังไฟล์ `/tmp/toc` ถ้ามีไฟล์ `/tmp/toc` อยู่ คำสั่ง `cat toc` จะแสดงรายการเนื้อหาของไฟล์

- เพื่อให้ได้ผลลัพธ์เหมือนกันโดยไม่ต้องกำหนดพารามิเตอร์ `TargetFile` ให้พิมพ์ดังต่อไปนี้:

```
ln -s /tmp/toc
```

ให้ดูที่คำสั่ง `ln` ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

คำสั่งสำหรับการลบไฟล์ที่ลิงก์:

คำสั่ง `rm` ลบลิงก์ ออกจากชื่อไฟล์ซึ่งคุณระบุ

เมื่อชื่อไฟล์ที่ลิงก์แบบฮาร์ดชื่อใดชื่อหนึ่งถูกลบออกไป ไฟล์ไม่ได้ถูกลบออกโดยสมบูรณ์เนื่องจากไฟล์ยังคงอยู่ภายใต้ชื่ออื่น เมื่อลิงก์ล่าสุดไปยังหมายเลขไอ-โหนดถูกลบออก ข้อมูลจะถูกลบออก ด้วย จากนั้น หมายเลขไอ-โหนดมีอยู่เพื่อให้ระบบนำไปใช้อีกครั้งได้

ให้ดูที่คำสั่ง `rm` ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

ไฟล์ DOS

ระบบปฏิบัติการ AIX ช่วยให้คุณสามารถทำงานกับไฟล์ DOS บนระบบของคุณได้

คัดลอกไฟล์ DOS ซึ่งคุณต้องการ ทำงานไปยังดิสเก็ต ระบบของคุณสามารถอ่านไฟล์เหล่านี้เข้าในไดเรกทอรีระบบปฏิบัติการพื้นฐานในรูปแบบที่ถูกต้อง และกลับไปยังดิสเก็ตในรูปแบบ DOS

หมายเหตุ: อักขระ wildcard * และ ? (เครื่องหมายดอกจันและ เครื่องคำถาม) ทำงานได้ไม่ถูกต้องกับคำสั่งที่อธิบายในส่วนนี้ (แม้ว่าสามารถทำงานได้อย่างถูกต้องกับเซลล์ระบบปฏิบัติการพื้นฐาน) หากคุณไม่ได้ระบุ นามสกุลของชื่อไฟล์ ชื่อไฟล์จะถูกจับคู่ราวกับว่าคุณได้ระบุ นามสกุลว่างเปล่า

การคัดลอกไฟล์ DOS ไปยังไฟล์ระบบปฏิบัติการพื้นฐาน:

ใช้คำสั่ง `dosread` เพื่อคัดลอกไฟล์ DOS ที่ ระบุไปยังไฟล์ระบบปฏิบัติการพื้นฐานที่ระบุ

หมายเหตุ: แบบแผนการตั้งชื่อไฟล์ DOS มีการใช้พร้อมกับชื่อยกเว้นหนึ่งข้อ เนื่องจากอักขระ เครื่องหมายทับขา (\) สามารถมีความหมายพิเศษในระบบปฏิบัติการพื้นฐาน จึงควรใช้อักขระเครื่องหมายทับ (/) เป็นตัวคั่น เพื่อระบุชื่อไดเรกทอรีย่อยในชื่อพาร DOS

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการคัดลอกไฟล์ข้อความที่ชื่อ `chap1.doc` จากดิสเก็ต DOS ไปยังระบบไฟล์ปฏิบัติการพื้นฐาน ให้พิมพ์ดังต่อไปนี้:

```
dosread -a chap1.doc chap1
```

คำสั่งนี้คัดลอกไฟล์ข้อความ DOS \CHAP1.DOC บน อุปกรณ์ดีฟลอตต์ `/dev/fd0` ไปยังไฟล์ระบบปฏิบัติการพื้นฐาน `chap1` ในไดเรกทอรีปัจจุบัน

- เมื่อต้องการคัดลอกไฟล์ไบนารีจากดิสเก็ต DOS ไปยังระบบไฟล์ปฏิบัติการพื้นฐาน ให้พิมพ์ดังต่อไปนี้:

```
dosread -D/dev/fd0 /survey/test.dta /home/fran/testdata
```

คำสั่งนี้คัดลอกไฟล์ข้อมูล \SURVEY\TEST.DTA DOS บน /dev/fd0 ไปยังไฟล์ระบบปฏิบัติการพื้นฐาน /home/fran/testdata

ให้ดูที่คำสั่ง **dosread** ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

การคัดลอกไฟล์ระบบปฏิบัติการพื้นฐานไปยังไฟล์ DOS:

ใช้คำสั่ง **doswrite** เพื่อคัดลอกไฟล์ระบบปฏิบัติการพื้นฐานที่ระบุไปยังไฟล์ DOS ที่ระบุ

หมายเหตุ: แบบแผนการตั้งชื่อไฟล์ DOS มีการใช้พร้อมกันกับชื่อยกเว้นหนึ่งข้อ เนื่องจากอักขระ เครื่องหมายทับขา (\) สามารถมีความหมายพิเศษในระบบปฏิบัติการพื้นฐาน จึงควรใช้อักขระเครื่องหมายทับ (/) เป็นตัวคั่นเพื่อระบุชื่อไดเรกทอรีย่อยในชื่อพาธ DOS

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการคัดลอกไฟล์ข้อความที่ชื่อ chap1 จากระบบไฟล์ปฏิบัติการพื้นฐานไปยังดิสเก็ต DOS ให้พิมพ์ดังต่อไปนี้:

```
doswrite -a chap1 chap1.doc
```

คำสั่งนี้คัดลอกไฟล์ระบบปฏิบัติการพื้นฐาน chap1 ใน ไดเรกทอรีปัจจุบันไปยังไฟล์ข้อความ DOS \CHAP1.DOC บน /dev/fd0

- เมื่อต้องการคัดลอกไฟล์ไบนารีที่ชื่อ /survey/test.dta จาก ระบบไฟล์ปฏิบัติการพื้นฐานไปยังดิสเก็ต DOS ให้พิมพ์ดังต่อไปนี้:

```
doswrite -D/dev/fd0 /home/fran/testdata /survey/test.dta
```

คำสั่งนี้ คัดลอกไฟล์ข้อมูลระบบปฏิบัติการพื้นฐาน /home/fran/testdata ไปยังไฟล์ DOS \SURVEY\TEST.DTA บน /dev/fd0

ให้ดูที่คำสั่ง **doswrite** ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

การลบไฟล์ DOS:

ใช้คำสั่ง **dosdel** เพื่อลบไฟล์ DOS ที่ระบุ

หมายเหตุ: แบบแผนการตั้งชื่อไฟล์ DOS มีการใช้พร้อมกันกับชื่อยกเว้นหนึ่งข้อ เนื่องจากอักขระ เครื่องหมายทับขา (\) สามารถมีความหมายพิเศษในระบบปฏิบัติการพื้นฐาน จึงควรใช้อักขระเครื่องหมายทับ (/) เป็นตัวคั่นเพื่อระบุชื่อไดเรกทอรีย่อยในชื่อพาธ DOS

คำสั่ง **dosdel** แปลงอักขระตัวพิมพ์เล็กในชื่อไฟล์หรือไดเรกทอรี เป็นตัวพิมพ์ใหญ่ก่อนทำการตรวจสอบดิสก์ เนื่องจากชื่อไฟล์ทั้งหมดถือว่าเป็นชื่อพาธแบบเต็ม (ไม่ใช่แบบสัมพัทธ์) คุณจึงไม่จำเป็นต้องใส่เครื่องหมายทับนำหน้า (/)

ตัวอย่างเช่น เมื่อต้องการลบไฟล์ DOS ที่ชื่อ file.ext บน อุปกรณ์ดีฟอลต์ (/dev/fd0) ให้พิมพ์ดังต่อไปนี้:

```
dosdel file.ext
```

ให้ดูที่คำสั่ง **dosdel** ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงผลเนื้อหาของไดเรกทอรี DOS:

ใช้คำสั่ง `dosdir` เพื่อแสดงข้อมูล เกี่ยวกับไฟล์หรือไดเรกทอรี DOS ที่ระบุ

หมายเหตุ: แบบแผนการตั้งชื่อไฟล์ DOS มีการใช้พร้อมทั้งข้อยกเว้นหนึ่งข้อ เนื่องจากอักขระ เครื่องหมายทับขา (\) สามารถมีความหมายพิเศษในระบบปฏิบัติการ พื้นฐาน จึงควรใช้อักขระเครื่องหมายทับ (/) เป็นตัวคั่น เพื่อระบุชื่อไดเรกทอรีย่อยในชื่อพาธ DOS

คำสั่ง `dosdir` แปลงอักขระตัวพิมพ์เล็กในชื่อไฟล์หรือไดเรกทอรี เป็นตัวพิมพ์ใหญ่ก่อนทำการตรวจสอบดิสก์ เนื่องจากชื่อไฟล์ทั้งหมดถือว่าเป็นชื่อพาธแบบเต็ม (ไม่ใช่แบบสัมพัทธ์) คุณไม่จำเป็นต้องเพิ่ม / (เครื่องหมายทับ) เริ่มต้น

ตัวอย่าง เช่น เมื่อต้องการอ่านไดเรกทอรีของไฟล์ DOS บน `/dev/fd0` ให้พิมพ์ดังต่อไปนี้:

```
dosdir
```

คำสั่งส่งคืนข้อมูลชื่อของไฟล์ และพื้นที่ว่างดิสก์ คล้ายกับต่อไปนี้

```
PG3-25.TXT
PG4-25.TXT
PG5-25.TXT
PG6-25.TXT
Free space: 312320 bytes
```

ให้ดูที่คำสั่ง `dosdir` ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

ข้อมูลสรุปคำสั่งสำหรับไฟล์

ข้อมูลต่อไปนี้เป็นคำสั่งสำหรับไฟล์ โพรซีเดอร์การจัดการไฟล์ และไฟล์ DOS และยังมีรายการของคำสั่งสำหรับการลิงก์ไฟล์ และไดเรกทอรีด้วย

ตารางที่ 57. คำสั่งสำหรับไฟล์

ไอเท็ม	คำอธิบาย
*	Wildcard, ตรงกับอักขระใดๆ
?	Wildcard, ตรงกับอักขระเดี่ยวใดๆ
[]	เมตาอักขระ, ตรงกับอักขระที่ใส่ในเครื่องหมาย

ตารางที่ 58. คำสั่งสำหรับโพรซีเดอร์การจัดการไฟล์

ไอเท็ม	คำอธิบาย
cat	ต่อไฟล์ หรือแสดงไฟล์
cmp	เปรียบเทียบสองไฟล์
colrm	แยกคอลัมน์ออกจากไฟล์
cp	คัดลอกไฟล์
cut	เขียนไบต์ อักขระ หรือฟิลด์ที่เลือกจาก แต่ละบรรทัดของไฟล์
diff	เปรียบเทียบไฟล์ข้อความ
file	พิจารณาชนิดไฟล์
find	ค้นหาไฟล์ที่มีนิพจน์ตรงกัน
grep	ค้นหารูปแบบในไฟล์
head	แสดงสองสามบรรทัดแรกหรือไบต์ของไฟล์
more	แสดงข้อความต่อเนื่องทีละหนึ่งหน้าจอบนหน้าจอ แสดงผล
mv	ย้ายไฟล์
nl	จำนวนบรรทัดในไฟล์
pg	จัดรูปแบบไฟล์ที่จะแสดงผล
rm	ลบ (ยกเลิกลิงก์) ไฟล์หรือไดเรกทอรี
paste	ผสานบรรทัดของหลายไฟล์หรือบรรทัดที่ตามมา เป็นหนึ่งไฟล์

ตารางที่ 58. คำสั่งสำหรับโพรซีเดอร์ทอจัดการไฟล์ (ต่อ)

ไอเท็ม	คำอธิบาย
sort	เรียงลำดับไฟล์ ผสานไฟล์ที่เรียงลำดับแล้ว และ ตรวจสอบไฟล์เพื่อพิจารณาว่ามีการเรียงลำดับแล้วหรือไม่
tail	เขียนไฟล์ไปยังเอาต์พุตมาตรฐาน โดยเริ่มต้นที่จุด ที่ระบุ
wc	นับจำนวนของบรรทัด คำ และไบต์ในไฟล์

ตารางที่ 59. คำสั่งสำหรับการลิงก์ไฟล์และไดเรกทอรี

ไอเท็ม	คำอธิบาย
ln	ลิงก์ไฟล์และไดเรกทอรี

ตารางที่ 60. คำสั่งสำหรับไฟล์ DOS

ไอเท็ม	คำอธิบาย
dosdel	ลบไฟล์ DOS
dosdir	แสดงรายการไดเรกทอรีสำหรับไฟล์ DOS
dosread	คัดลอกไฟล์ DOS ไปยังไฟล์ระบบปฏิบัติการพื้นฐาน
doswrite	คัดลอกไฟล์ระบบปฏิบัติการพื้นฐานไปยังไฟล์ DOS

เซลล์ระบบปฏิบัติการ

อินเทอร์เน็ตของคุณสำหรับระบบปฏิบัติการเรียกว่า เซลล์

เซลล์เป็นเลเยอร์นอกสุดของระบบปฏิบัติการ เซลล์รวมภาษาโปรแกรมเข้าด้วยกันเพื่อควบคุมกระบวนการและไฟล์ รวมถึง เริ่มต้นและควบคุมโปรแกรมอื่นๆ เซลล์จัดการการโต้ตอบ ระหว่างคุณและระบบปฏิบัติการโดยการพ้อมต์คุณเพื่อรับอินพุต การแปล อินพุตนั้นสำหรับระบบปฏิบัติการ จากนั้นจัดการเอาต์พุตผลลัพธ์จากระบบปฏิบัติการ

เซลล์จัดให้มีวิธีสำหรับคุณเพื่อสื่อสารกับ ระบบปฏิบัติการ การสื่อสารนี้ถูกดำเนินการแบบโต้ตอบ (อินพุต จากคีย์บอร์ดถูก ดำเนินการในทันที) หรือเป็นเซลล์สคริปต์ โดย เซลล์ สคริปต์ คือลำดับของเซลล์และคำสั่งระบบปฏิบัติการที่เก็บ ในไฟล์

เมื่อคุณล็อกอินเข้าสู่ระบบ ระบบค้นหา ชื่อของโปรแกรมเซลล์เพื่อเรียกใช้งาน หลังจากเรียกใช้งาน เซลล์จะแสดง พ้อมต์คำสั่ง พ้อมต์นี้โดยทั่วไปคือ \$ (เครื่องหมายดอลลาร์) เมื่อคุณพิมพ์คำสั่งที่พ้อมต์ และกดปุ่ม Enter เซลล์หาค่า คำสั่งและ พ้อมต์เพื่อดำเนินการ ทั้งนี้ขึ้นอยู่กับคำสั่งของคุณ เซลล์เขียนเอาต์พุตคำสั่งไปยังหน้าจอ หรือเปลี่ยนทิศทางเอาต์พุต จากนั้น ส่งคืนพ้อมต์คำสั่ง และรอให้คุณพิมพ์คำสั่งอื่น

บรรทัดรับคำสั่ง คือบรรทัดที่คุณพิมพ์ ซึ่งมีเซลล์พ้อมต์ รูปแบบพื้นฐานสำหรับแต่ละบรรทัดเป็นดังนี้:

\$ Command Argument(s)

เซลล์พิจารณาคำแรกของบรรทัดรับคำสั่ง (จนถึงช่องว่างตัวแรก) เป็นคำสั่ง ส่วนคำที่ตามหลังทั้งหมดเป็นอาร์กิวเมนต์

หมายเหตุ: เมื่อ lib.c.a ถูกย้ายหรือเปลี่ยนชื่อ ข้อความแสดงข้อผิดพลาด Killed ถูกแสดงจากเซลล์เนื่องจากไม่มีไฟล์ lib.c.a สำหรับระบบเพื่อโหลดและรันยูทิลิตี้ คำสั่ง recsh เรียกใช้เซลล์การกู้คืน ซึ่งจัดให้มีความสามารถในการเปลี่ยนชื่อ lib.c.a ถ้าถูกย้ายโดยบังเอิญ

งานที่เกี่ยวข้อง:

“การแสดงรายการคำสั่งที่ป้อนก่อนหน้านี้ (คำสั่ง history)” ในหน้า 145

ใช้คำสั่ง history เพื่อแสดงรายการคำสั่งซึ่ง คุณป้อนก่อนหน้านี้

แนวคิดเซลล์

ก่อนคุณเริ่มต้นทำงานกับเซลล์ชนิดต่างๆ ที่มีอยู่ สำหรับ AIX คุณต้องเข้าใจ คำศัพท์และคุณลักษณะพื้นฐาน

เซลล์ที่มีอยู่:

ข้อมูลต่อไปนี้เป็นเซลล์ที่จัดเตรียมให้พร้อมกับ AIX

- Korn เซลล์ (เริ่มต้นด้วยคำสั่ง `ksh`)
- เซลล์เป้าหมาย (เริ่มต้นด้วยคำสั่ง `bsh`)
- เซลล์ที่จำกัด (เวอร์ชันที่จำกัดของเซลล์เป้าหมาย และเริ่มต้นด้วย คำสั่ง `Rsh`)
- POSIX เซลล์ (รู้จักกันอีกอย่างว่าเป็น Korn เซลล์ และเริ่มต้นด้วยคำสั่ง `psh`)
- เซลล์ที่จำกัดสำหรับเซลล์ Korn (`ksh` และ `ksh93`) เซลล์ `ksh` และ `ksh93` ถูก กำหนดด้วยเซลล์ที่จำกัดเทียบเท่ากับ `rksh` และ `rksh93`
- ดีฟอลต์เซลล์ (เริ่มต้นด้วยคำสั่ง `sh`)
- C เซลล์ (เริ่มต้นด้วยคำสั่ง `csh`)
- เซลล์ที่เชื่อถือได้ (เวอร์ชันที่จำกัดของ Korn เซลล์ และเริ่มต้นด้วย คำสั่ง `tsh`)
- รีโมตเซลล์ (เริ่มต้นด้วยคำสั่ง `rsh`)

เซลล์การล็อกอิน หมายถึงเซลล์ที่ถูกโหลดเมื่อคุณล็อกอิน เข้าสู่ระบบคอมพิวเตอร์ เซลล์การล็อกอินของคุณมีการตั้งค่าไว้ในไฟล์ `/etc/passwd` Korn เซลล์คือเซลล์การล็อกอินระบบปฏิบัติการมาตรฐาน และสามารถเข้ากันได้ย้อนหลังกับเซลล์เป้าหมาย

เซลล์ Korn (`/usr/bin/ksh`) ถูกตั้งค่าเป็น ดีฟอลต์เซลล์ ดีฟอลต์เซลล์ หรือเซลล์มาตรฐานอ้างอิงไปยังเซลล์ ที่ลิงก์กัน และเริ่มทำงานด้วยคำสั่ง `/usr/bin/sh` เซลล์ Bourne (`/usr/bin/sh`) สามารถแทน เป็นดีฟอลต์เซลล์ เซลล์ POSIX ซึ่งมีการเรียกใช้โดย คำสั่ง `/usr/bin/psh` มีอยู่เป็นลิงก์ไปยังคำสั่ง `/usr/bin/sh`

หลักการที่เกี่ยวข้อง:

“เซลล์เป้าหมาย” ในหน้า 283

เซลล์เป้าหมายคือตัวแปลคำสั่งแบบโต้ตอบและภาษา การเขียนโปรแกรมคำสั่ง

“คำสั่ง Korn เซลล์หรือ POSIX เซลล์” ในหน้า 275

Korn เซลล์คือตัวแปลคำสั่งแบบโต้ตอบและภาษา การเขียนโปรแกรมคำสั่ง Korn เซลล์สอดคล้องกับ Portable Operating System Interface for Computer Environments (POSIX) ซึ่งเป็นมาตรฐานสากลสำหรับ ระบบปฏิบัติการ

คำศัพท์เซลล์:

คำศัพท์และนิยามในตารางนี้มีประโยชน์ในการทำความเข้าใจกับ เซลล์

ไอเท็ม	คำอธิบาย
พื้นที่ว่างเปล่า	พื้นที่ว่างเปล่าเป็นอักขระหนึ่งในคลาสอักขระพื้นที่ว่างเปล่าที่กำหนดไว้ในหมวดหมู่ <code>LC_CTYPE</code> ใน POSIX เซลล์ พื้นที่ว่างเปล่าเป็นแท็บหรือช่องว่าง
คำสั่งในตัว	คำสั่งที่เซลล์ดำเนินการโดยไม่ต้องค้นหา และสร้างโปรเซสแยกต่างหาก
คำสั่ง	ลำดับของอักขระในไวยากรณ์ของภาษาเซลล์ เซลล์อ่านแต่ละคำสั่งและทำการดำเนินการที่ต้องการโดยตรง หรือโดยการเรียกใช้ยูทิลิตี้แยกต่างหาก
ข้อคิดเห็น	คำใดๆ ที่ขึ้นต้นด้วยสัญลักษณ์เงินปอนด์ (<code>#</code>) คำและอักขระทั้งหมดที่ตามหลังสัญลักษณ์นั้นจนถึงอักขระบรรทัดใหม่ ถัดไป ถูกละเว้นไป

ไอเท็ม	คำอธิบาย
ตัวระบุ	ลำดับของตัวอักษร ตัวเลข หรือขีดเส้นใต้จากชุดอักขระที่ใช้ได้ หลายระบบ ซึ่งขึ้นต้นด้วยตัวอักษรหรือขีดเส้นใต้ อักขระตัวแรกของตัวระบุต้องไม่ใช่ตัวเลข ตัวระบุมีการใช้เป็นชื่อของสมนาม ฟังก์ชัน และพารามิเตอร์ที่ระบุ
รายการ	<p>ลำดับของไฟฟ์ไลน์ตั้งแต่หนึ่งรายการขึ้นไปที่แบ่งโดยสัญลักษณ์ อย่างไรก็ตามหนึ่งต่อไปนี้: อัฒภาค (;), ampersand (&), ampersand คู่ (&&), หรือแถบคู่ () สามารถเลือกที่จะลงท้ายรายการด้วยสัญลักษณ์อย่างใดอย่างหนึ่งต่อไปนี้: อัฒภาค (;), ampersand (&), หรือแถบ ampersand (&)</p> <p>; ประมวลผลไฟฟ์ไลน์ก่อนหน้าตามลำดับ เซลล์ดำเนินการแต่ละคำสั่ง ตามรอบ และรอให้คำสั่งล่าสุดเสร็จสมบูรณ์</p> <p>& ประมวลผลไฟฟ์ไลน์ก่อนหน้าในแบบอะซิงโครนัส เซลล์ดำเนินการ แต่ละคำสั่งตามรอบ ประมวลผลไฟฟ์ไลน์ในพื้นที่หลังโดยไมรอให้เสร็จสมบูรณ์</p> <p> & ประมวลผลไฟฟ์ไลน์ก่อนหน้าในแบบอะซิงโครนัสและสร้างไฟฟ์สองทาง ไปยังเซลล์พาเรนต์ เซลล์ดำเนินการแต่ละคำสั่งตามรอบ ประมวลผลไฟฟ์ไลน์ ในพื้นที่หลังโดยไมรอให้เสร็จสมบูรณ์ เซลล์พาเรนต์สามารถอ่านจากอินพุตมาตรฐาน และเขียนลงเอาต์พุตมาตรฐานของคำสั่งที่ถูกสร้างโดยใช้คำสั่ง <code>read -p</code> และ <code>print -p</code> คำสั่งดังกล่าวสามารถใช้งานอยู่ได้เพียงคำสั่งเดียวเท่านั้นในเวลาที่กำหนด</p> <p>&& ประมวลผลรายการที่ตามหลังสัญลักษณ์นี้เฉพาะถ้าไฟฟ์ไลน์ก่อนหน้า ส่งคืนค่าการออกป็นศูนย์ (0)</p> <p> ประมวลผลรายการที่ตามหลังสัญลักษณ์นี้เฉพาะถ้าไฟฟ์ไลน์ก่อนหน้า ส่งคืนค่าการออกที่ไม่ใช่ศูนย์</p> <p>เครื่องหมายอัฒภาค (;), ampersand (&), และแถบ ampersand (&) มีลำดับความสำคัญต่ำกว่า ampersand คู่ (&&) และ แถบคู่ () สัญลักษณ์ ;, &, และ & มีลำดับความสำคัญเท่ากัน สัญลักษณ์ && และ มีลำดับความสำคัญเท่ากัน สามารถใช้อักขระบรรทัดใหม่ตั้งแต่หนึ่งตัวขึ้นไปแทนเครื่องหมายอัฒภาค เพื่อคั่นระหว่างสองคำสั่งในรายการ</p> <p>หมายเหตุ: สัญลักษณ์ & ถูกต้องใน Korn เซลล์เท่านั้น</p>
เมตาอักขระ	เมตาอักขระแต่ละตัวมีความหมายพิเศษในเซลล์ และส่งผลให้ยุติคำ ยกเว้นว่ามีการใส่ไว้ในเครื่องหมายอัญประกาศ เมตาอักขระคือ: ไฟฟ์ (), ampersand (&), อัฒภาค (;), เครื่องหมายน้อยกว่า (<), เครื่องหมายมากกว่า (>), วงเล็บซ้าย ((), วงเล็บขวา ()), สัญลักษณ์ดอลลาร์ (\$), backquote (`), ทับขวา (\), อัญประกาศขวา ('), เครื่องหมายอัญประกาศคู่ ("), อักขระบรรทัดใหม่, อักขระช่องว่าง, และอักขระแท็บ อักขระทั้งหมดที่ใส่อยู่ระหว่างเครื่องหมายอัญประกาศเดี่ยวถือว่าเป็นการใส่ไว้ในเครื่องหมายอัญประกาศ และถูกตีความตามตัวอักษรโดยเซลล์ ความหมายพิเศษของเมตาอักขระยังคงไว้ ถ้าไม่ได้ใส่ไว้ในเครื่องหมายอัญประกาศ (เมตาอักขระเรียกอีกอย่างว่า <i>เมตาอักขระตัวแฉงส่วน</i> ใน C เซลล์)
รายการกำหนดพารามิเตอร์	ประกอบด้วยคำสั่งตั้งแต่หนึ่งคำสั่งขึ้นไปในรูปแบบ <code>Identifier=Value</code> ซึ่งช่องว่างนำหน้าและตามหลังเครื่องหมายเท่ากับ (=) ต้องสมดุลกัน นั่นคือ ต้องใช้พื้นที่ว่างเปล่านำและตามหลัง หรือไม่มีพื้นที่ว่างเปล่า
ไฟฟ์ไลน์	<p>ลำดับของคำสั่งตั้งแต่หนึ่งรายการขึ้นไปที่แบ่งโดยไฟฟ์ () แต่ละคำสั่งในไฟฟ์ไลน์ ยกเว้นคำสั่งสุดท้าย มีการรันเป็นโปรเซสแยกต่างหาก อย่างไรก็ตาม เอาต์พุตมาตรฐานของแต่ละคำสั่งที่เชื่อมต่อกันโดยไฟฟ์ กลายเป็นอินพุตมาตรฐานของคำสั่งถัดไปในลำดับ หากรายการถูกใส่ไว้ในเครื่องหมายวงเล็บ จะมีการดำเนินการเป็นคำสั่งแบบง่าย ที่ดำเนินงานในเซลล์ย่อยแยกต่างหาก</p> <p>หากไม่มีคำสั่งส่งวนไว้! นำหน้าไฟฟ์ไลน์ สถานะการออกจะเป็น สถานะการออกของคำสั่งสุดท้ายที่ระบุในไฟฟ์ไลน์ มิฉะนั้น สถานะการออกเป็น NOT ทางตรรกะของสถานะการออกของ คำสั่งล่าสุด อธิบายอีกอย่างคือ ถ้าคำสั่งล่าสุดส่งคืนศูนย์ สถานะการออก จะเป็น 1 ถ้าคำสั่งล่าสุดส่งคืนค่าที่มากกว่าศูนย์ สถานะการออกจะเป็นศูนย์</p> <p>รูปแบบ สำหรับไฟฟ์ไลน์เป็นดังนี้:</p> <pre>[!] command1 [command2 ...]</pre> <p>หมายเหตุ: เวอร์ชันก่อนหน้าของเซลล์เป้าหมายใช้ caret (^) เพื่อบ่งชี้ไฟฟ์</p>

ไอเท็ม	คำอธิบาย
ตัวแปรเซลล์	ชื่อหรือพารามิเตอร์ซึ่งได้รับการกำหนดค่า กำหนด ตัวแปรโดยพิมพ์ชื่อตัวแปร เครื่องหมายเท่ากับ (=) ตามด้วยค่า สามารถทดแทนชื่อตัวแปรด้วยค่าที่กำหนดได้ โดยการใส่สัญลักษณ์ดอลลาร์ (\$) ไว้ข้างหน้าชื่อตัวแปร ตัวแปรมีประโยชน์มากเป็นพิเศษในการสร้างสัญลักษณ์สำหรับชื่อพาธที่ยาว เช่น \$HOME สำหรับไดเรกทอรีโฮม ตัวแปรที่กำหนดไว้ล่วงหน้าคือตัวแปรที่เซลล์กำหนดค่าไว้แล้ว ตัวแปรผู้ใช้กำหนด คือตัวแปรซึ่งผู้ใช้เป็นผู้กำหนดค่าให้
คำสั่งแบบง่าย	ลำดับของรายการกำหนดพารามิเตอร์และการเปลี่ยนทิศทางที่เป็น ทางเลือก ในลำดับใดๆ คำสั่งแบบง่ายอาจตามด้วย คำสั่ง คำ และการเปลี่ยนทิศทางที่เลือกกำหนดได้ และยุติด้วย ; , & , && & หรืออักขระบรรทัดใหม่ ชื่อคำสั่งมีการส่งผ่านเป็นพารามิเตอร์ 0 (ตามที่ กำหนดโดยรูทีนย่อย exec) ค่าของ คำสั่งแบบง่ายคือ สถานะการออกเป็นศูนย์ถ้ายุติลงตามปกติ หรือไม่ใช้ศูนย์ ถ้ายุติลงอย่างผิดปกติ รูทีนย่อย sigaction, sigvec, หรือ signal มีรายการ ของค่าสถานะการออกของสัญญาณ
เซลล์ย่อย	เซลล์ที่กำลังรันเป็นชายนของเซลล์การลือกอินหรือ เซลล์ปัจจุบัน
อักขระ wildcard	รู้จักในอีกชื่อหนึ่งว่า <i>อักขระการจับคู่รูปแบบ</i> เซลล์ เชื่อมโยงอักขระกับค่าที่กำหนดให้ Wildcards พื้นฐานคือ ?, *, [set], และ [!set] อักขระ wildcard มีประโยชน์มากเป็นพิเศษเมื่อทำ การทดแทนชื่อไฟล์
คำ	ลำดับของอักขระที่ไม่มีพื้นที่ว่างเปล่าใดๆ คำมีการแบ่งด้วยเมตาอักขระตั้งแต่หนึ่งตัวขึ้นไป

การระบุเซลล์สำหรับไฟล์สคริปต์:

เมื่อคุณรันเซลล์สคริปต์ปฏิบัติการใน Korn (POSIX เซลล์) หรือเซลล์เป้าหมาย คำสั่งในสคริปต์มีการดำเนินการภายใต้ การควบคุมของเซลล์ปัจจุบัน (เซลล์ซึ่งเริ่มต้นสคริปต์) ยกเว้นว่าคุณระบุเซลล์อื่น เมื่อคุณรันเซลล์สคริปต์ปฏิบัติการ ใน C เซลล์ คำสั่งในสคริปต์มีการดำเนินการภายใต้การควบคุม ของเซลล์เป้าหมาย (/usr/bin/bsh) ยกเว้นว่าคุณระบุ เซลล์อื่น

คุณสามารถรันเซลล์สคริปต์ในเซลล์เฉพาะได้โดยรวม เซลล์ไว้ภายในเซลล์สคริปต์

เมื่อต้องการรันเซลล์สคริปต์ปฏิบัติการภายใต้ เซลล์เฉพาะ ให้พิมพ์ `#! Path` บน บรรทัดแรกของเซลล์สคริปต์ และกด Enter อักขระ `#!` ระบุ ชนิดไฟล์ ตัวแปร `Path` ระบุ ชื่อพาธของเซลล์ซึ่งจะรันเซลล์สคริปต์

ตัวอย่าง เช่น เมื่อต้องการรันสคริปต์ `bsh` ในเซลล์เป้าหมาย ให้พิมพ์ ดังต่อไปนี้:

```
#!/usr/bin/bsh
```

เมื่อคุณ วางคำสั่งเซลล์ไว้ข้างหน้าชื่อไฟล์เซลล์สคริปต์ เซลล์ที่ระบุ บนบรรทัดคำสั่งจะยกเลิกเซลล์ใดๆ ที่ระบุภายในตัวไฟล์สคริปต์เอง ดังนั้น การพิมพ์ `ksh myfile` และกด Enter จะรัน ไฟล์ที่ชื่อ `myfile` ภายใต้การควบคุมของ Korn เซลล์ แม้ว่าบรรทัดแรกของ `myfile` คือ `#!/usr/bin/csh`

คุณลักษณะเซลล์:

มีข้อดีหลายอย่างในการใช้เซลล์เป็นอินเทอร์เฟซของ ระบบ

ข้อดีหลักของการอินเทอร์เฟซกับระบบผ่านทางเซลล์มี ดังนี้:

- การทดแทน wildcard ในชื่อไฟล์ (การจับคู่รูปแบบ)
 - ดำเนินการ คำสั่งในกลุ่มของไฟล์โดยการระบุรูปแบบที่จะจับคู่ แทน การระบุชื่อไฟล์จริง สำหรับข้อมูลเพิ่มเติม ให้ดูที่:
 - “การทดแทนชื่อไฟล์ใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 254
 - “การทดแทนชื่อไฟล์ในเซลล์เป้าหมาย” ในหน้า 286

– “การทดแทนชื่อไฟล์ใน C เซลล์” ในหน้า 303

• การประมวลผลพื้นหลัง

ตั้งค่าภารกิจที่ใช้เวลานานให้รันในพื้นหลัง เพื่อให้เทอร์มินัลว่างสำหรับการประมวลผลแบบโต้ตอบที่เกิดขึ้นพร้อมกัน สำหรับข้อมูลเพิ่มเติมให้ดูที่คำสั่ง **bg** ในส่วนต่อไปนี้:

- “การควบคุมงานใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 268
- “คำสั่งในตัว C เซลล์” ในหน้า 310

หมายเหตุ: เซลล์เป้าหมายไม่สนับสนุนการควบคุมงาน

• การตั้งสมนามให้กับคำสั่ง

ตั้งชื่อสมนามให้กับคำสั่งหรือวลี เมื่อเซลล์พบสมนามบนบรรทัดคำสั่ง หรือในเซลล์สคริปต์ เซลล์จะแทนที่ข้อความซึ่งสมนามอ้างอิง

สำหรับข้อมูลเพิ่มเติมให้ดูที่:

- “การกำหนดสมนามคำสั่งใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 281
- “การทดแทนสมนามใน C เซลล์” ในหน้า 300

หมายเหตุ: เซลล์เป้าหมายไม่สนับสนุนการตั้งสมนามให้กับคำสั่ง

• ประวัติคำสั่ง

บันทึกคำสั่งที่คุณป้อนไว้ในไฟล์ ประวัติ คุณสามารถใช้ไฟล์นี้เพื่อเข้าถึง แก้ไข และออกใช้คำสั่งที่แสดงรายการใดๆ ได้โดยง่าย

สำหรับข้อมูลเพิ่มเติมให้ดูที่คำสั่ง **history** ในส่วนต่อไปนี้:

- “ประวัติคำสั่ง Korn เซลล์หรือ POSIX เซลล์” ในหน้า 280
- “คำสั่งในตัว C เซลล์” ในหน้า 310
- “การทดแทนประวัติใน C เซลล์” ในหน้า 317

หมายเหตุ: เซลล์เป้าหมายไม่สนับสนุนประวัติคำสั่ง

• การทดแทนชื่อไฟล์

จัดทำรายชื่อไฟล์บนบรรทัดคำสั่ง โดยใช้อักขระการจับคู่รูปแบบโดยอัตโนมัติ

สำหรับข้อมูลเพิ่มเติมให้ดูที่:

- “การทดแทนชื่อไฟล์ใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 254
- “การทดแทนชื่อไฟล์ในเซลล์เป้าหมาย” ในหน้า 286
- “การทดแทนชื่อไฟล์ใน C เซลล์” ในหน้า 303

• การเปลี่ยนทิศทางอินพุตและเอาต์พุต

เปลี่ยนทิศทาง อินพุตจากคีย์บอร์ด และเปลี่ยนทิศทางเอาต์พุตไปยังไฟล์หรืออุปกรณ์ที่ไม่ใช่เทอร์มินัล ตัวอย่างเช่น อินพุตของโปรแกรมสามารถระบุจาก ไฟล์และเปลี่ยนทิศทางไปยังเครื่องพิมพ์หรือไฟล์อื่น

สำหรับข้อมูลเพิ่มเติมให้ดูที่:

- “การเปลี่ยนทิศทางอินพุตและเอาต์พุตใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 255
- “การเปลี่ยนทิศทางอินพุตและเอาต์พุตในเซลล์เป้าหมาย” ในหน้า 287
- “การเปลี่ยนทิศทางอินพุตและเอาต์พุตใน C เซลล์” ในหน้า 320

• การไฟฟ์

ลิงก์คำสั่งจำนวนหนึ่งเข้าด้วยกันเพื่อสร้างเป็นโปรแกรมที่ซับซ้อน เอาต์พุตมาตรฐานของโปรแกรมหนึ่งกลายเป็นอินพุตมาตรฐานของโปรแกรมถัดไป

สำหรับข้อมูลเพิ่มเติมให้ดูที่นิยาม *ไฟฟ์ไลน์* ใน “คำศัพท์เซลล์” ในหน้า 230

• การทดแทนตัวแปรเซลล์

จัดเก็บข้อมูลในตัวแปรผู้ใช้กำหนด และตัวแปรเซลล์ที่กำหนดไว้ล่วงหน้า

สำหรับข้อมูลเพิ่มเติมให้ดูที่:

- “การทดแทนพารามิเตอร์ใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 252
- “การทดแทนตัวแปรในเซลล์เป้าหมาย” ในหน้า 295
- “การทดแทนตัวแปรใน C เซลล์” ในหน้า 301

หลักการที่เกี่ยวข้อง:

“คำสั่ง” ในหน้า 141

บางคำสั่งสามารถป้อนได้โดยการพิมพ์เพียง คำเดียว และเป็นไปได้ที่จะรวมคำสั่งหลายคำสั่ง เพื่อให้เอาต์พุตจากคำสั่งหนึ่งเป็นอินพุตสำหรับอีกคำสั่งหนึ่ง

คลาสอักขระ:

คุณสามารถใช้คลาสอักขระเพื่อจับคู่ชื่อไฟล์

คุณสามารถใช้คลาสอักขระเพื่อจับคู่ชื่อไฟล์ได้ดังนี้:

```
[[ :charclass: ]]
```

รูปแบบนี้สั่งให้ระบบจับคู่อักขระเดี่ยวใดๆ ที่เป็น สมาชิกของคลาสที่ระบุ คลาสที่กำหนดสอดคล้องกับรูทिनย่อย `ctype` ดังนี้:

คลาสอักขระ	นิยาม
<code>alnum</code>	อักขระแบบตัวอักษรผสมตัวเลข
<code>alpha</code>	ตัวอักษรตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก
<code>blank</code>	ช่องว่างหรือแท็บแนวนอน
<code>cntrl</code>	อักขระควบคุม
<code>digit</code>	ตัวเลข
<code>graph</code>	อักขระกราฟิก
<code>lower</code>	ตัวอักษรตัวพิมพ์เล็ก
<code>print</code>	อักขระที่พิมพ์ได้
<code>punct</code>	อักขระวรรคตอน
<code>space</code>	ช่องว่าง แท็บแนวนอน ปิดแคร์ชั่นบรรทัดใหม่ แท็บแนวตั้ง หรืออักขระ ป้อนกระดาษ
<code>upper</code>	อักขระตัวพิมพ์ใหญ่
<code>xdigit</code>	ตัวเลขฐานสิบหก

เซลล์ที่จำกัด:

เซลล์ที่จำกัดใช้เพื่อตั้งค่าชื่อล็อกอินและสถานะแวดล้อมการดำเนินการ ซึ่งสามารถควบคุมความสามารถได้มากกว่าสถานะแวดล้อมของ เซลล์เป้าหมายปกติ

คำสั่ง `Rsh` หรือ `bsh -r` เปิดเซลล์ที่จำกัด ลักษณะการทำงานของคำสั่งเหล่านี้เหมือนกับ ของคำสั่ง `bsh` ยกเว้นว่าไม่อนุญาตให้ทำการดำเนินการ ต่อไปนี้:

- การเปลี่ยนไดเรกทอรี (โดยใช้คำสั่ง cd)
- การตั้งค่าของตัวแปร *PATH* หรือ *SHELL*
- การระบุพารามิเตอร์หรือชื่อคำสั่งที่มีเครื่องหมายทับ (/)
- การเปลี่ยนทิศทางเอาต์พุต

หากเซลล์ที่จำกัดกำหนดว่าคำสั่งที่จะรันเป็นเซลล์โพรซีเดอร์ เซลล์ที่จำกัดจะใช้เซลล์เป้าหมายเพื่อรันคำสั่ง ในวิธีนี้สามารถนำเสนอเซลล์โพรซีเดอร์ที่เข้าถึงศักยภาพ เต็มรูปแบบของเซลล์เป้าหมายในขณะที่ใช้เมนูคำสั่งที่จำกัดให้แก่ผู้ใช้ได้ สถานการณ์นี้สมมติว่า ผู้ใช้ไม่มีสิทธิ์การเขียนและการดำเนินการในไดเรกทอรีเดียวกัน

หากมีการระบุพารามิเตอร์ *ไฟล์* [*พารามิเตอร์*] เมื่อเซลล์เป้าหมายเริ่มต้นขึ้น เซลล์จะรันไฟล์สคริปต์ ที่ระบุโดยพารามิเตอร์ *ไฟล์* รวมถึง พารามิเตอร์ที่ระบุใดๆ ไฟล์สคริปต์ที่ระบุต้องมีสิทธิ์การอ่าน ค่าติดตั้ง *setuid* และ *setgid* ใดๆ สำหรับไฟล์สคริปต์ ถูกข้ามไป จากนั้น เซลล์อ่านคำสั่ง ถ้าใช้แฟล็ก *-c* หรือ *-s* ใดๆ อย่างใดอย่างหนึ่ง อย่ระบุไฟล์สคริปต์

เมื่อเริ่มต้นด้วยคำสั่ง *Rsh* เซลล์จะบังคับใช้ข้อจำกัดหลังจากการตีความไฟล์ *.profile* และ */etc/environment* ด้วยเหตุนี้ ผู้เขียนไฟล์ *.profile* จึงสามารถควบคุม การดำเนินการของผู้ใช้ได้อย่างเต็มที่โดยทำการดำเนินการเซ็ทอัพและปล่อย ผู้ใช้ไว้ในไดเรกทอรีที่เหมาะสม (อาจไม่ใช่ไดเรกทอรีล็อกอิน) ผู้ดูแลระบบ สามารถสร้างไดเรกทอรีของคำสั่งในไดเรกทอรี */usr/rbin* ที่คำสั่ง *Rsh* สามารถใช้ได้โดยการเปลี่ยนตัวแปร *PATH* ให้มีไดเรกทอรี ถ้าเริ่มต้นด้วยคำสั่ง *bsh -r* เซลล์จะใช้ข้อจำกัดเมื่อตีความไฟล์ *.profile*

เมื่อเรียกด้วยคำสั่ง *Rsh* เซลล์ที่จำกัดจะอ่านไฟล์ *.profile* ของผู้ใช้ (*\$HOME/.profile*) และทำหน้าที่เป็นเซลล์เป้าหมายปกติในขณะที่ทำเช่นนี้ ยกเว้นว่าการขัดจังหวะ ส่งผลให้ออกในทันทีแทนการกลับไปยังระดับคำสั่ง

เซลล์ Korn สามารถเริ่มทำงานเป็นเซลล์ที่จำกัดด้วยคำสั่ง *command ksh -r*

โดย inodes สำหรับ *ksh* และ *rksh* จะเหมือนกัน และ inodes สำหรับ *ksh93* และ *rksh93* เหมือนกัน

การสร้างและการรันเซลล์สคริปต์:

เซลล์สคริปต์ คือไฟล์ที่มีคำสั่งตั้งแต่หนึ่งคำสั่งขึ้นไป เซลล์สคริปต์นำเสนอวิธีที่ง่ายในการดำเนินการคำสั่งที่ยุ่งยาก ลำดับของคำสั่งจำนวนมากหรือซับซ้อน และภารกิจที่หนัก เมื่อคุณป้อนชื่อ ของไฟล์เซลล์สคริปต์ ระบบจะดำเนินการลำดับคำสั่งที่มีอยู่ในไฟล์

คุณสามารถสร้างเซลล์สคริปต์โดยใช้โปรแกรมแก้ไขข้อความ สคริปต์ของคุณ สามารถมีทั้งคำสั่งระบบปฏิบัติการและคำสั่งในตัวเซลล์

ขั้นตอนต่อไปนี้เป็นแนวทางทั่วไปสำหรับการเขียนเซลล์สคริปต์:

1. สร้างและบันทึกไฟล์โดยใช้โปรแกรมแก้ไขข้อความ คุณสามารถรวมการผสมผสานในอัตราส่วนใดๆ ระหว่างคำสั่งเซลล์ และระบบปฏิบัติการในไฟล์เซลล์สคริปต์ ตามระเบียบ เซลล์สคริปต์ที่ไม่ได้ตั้งค่าสำหรับการใช้โดยผู้ใช้งานจำนวนมากมีการจัดเก็บไว้ในไดเรกทอรี *\$HOME/bin*

หมายเหตุ: ระบบปฏิบัติการ ไม่สนับสนุนรูทีนย่อย *setuid* หรือ *setgid* ภายในสคริปต์เซลล์

2. ใช้คำสั่ง *chmod* เพื่ออนุญาตให้เจ้าของรัน (หรือเรียกทำงาน) ไฟล์เท่านั้น ตัวอย่างเช่น ถ้าไฟล์ของคุณ มีชื่อว่า *script1* ให้พิมพ์ดังต่อไปนี้:

```
chmod u=rwx script1
```

3. พิมพ์ชื่อสคริปต์บนบรรทัดคำสั่งเพื่อรันเชลล์สคริปต์ เมื่อต้องการรัน เชลล์สคริปต์ script1 ให้พิมพ์ดังต่อไปนี้:

```
script1
```

หมายเหตุ: คุณสามารถรันเชลล์สคริปต์โดยไม่ต้องทำให้เชลล์สคริปต์ดำเนินการได้ ถ้าคำสั่งเชลล์ (ksh, bsh, หรือ csh) นำหน้าชื่อไฟล์เชลล์สคริปต์บนบรรทัด คำสั่ง ตัวอย่างเช่น เมื่อต้องการรันไฟล์ที่ไม่ใช่ไฟล์ปฏิบัติการที่ชื่อ script1 ภายใต้การควบคุมของ Korn เชลล์ ให้พิมพ์ดังต่อไปนี้:

```
ksh script1
```

หลักการที่เกี่ยวข้อง:

“คำสั่ง” ในหน้า 141

บางคำสั่งสามารถป้อนได้โดยการพิมพ์เพียง คำเดียว และเป็นไปได้ที่จะรวมคำสั่งหลายคำสั่ง เพื่อให้เอาต์พุตจากคำสั่งหนึ่ง เป็นอินพุตสำหรับอีกคำสั่งหนึ่ง

Korn เชลล์

Korn เชลล์ (คำสั่ง ksh) เข้ากันได้ย้อนหลังกับ เชลล์เป้าหมาย (คำสั่ง bsh) และมีคุณลักษณะเชลล์เป้าหมายส่วนใหญ่ ตลอดจนคุณลักษณะที่ดีที่สุดของ C เชลล์หลายอย่าง

ตัวแปรที่ตั้งค่าโดย Korn เชลล์หรือ POSIX เชลล์:

ตัวแปรต่อไปนี้มีการตั้งค่าโดยเชลล์

ไอเท็ม	คำอธิบาย
<i>ขีดเส้นใต้</i> ()	บ่งชี้ชื่อพาสเวิร์ดโดยแรกเริ่มของเชลล์หรือ สคริปต์ที่กำลังดำเนินการเมื่อส่งผ่านในสภาวะแวดล้อม ในเวลาต่อมา มีการกำหนด อาร์กิวเมนต์ล่าสุดของคำสั่งก่อนหน้านี้ ไม่มีการตั้งค่าพารามิเตอร์นี้ สำหรับคำสั่งที่ชื่อซิกโครนัส พารามิเตอร์นี้ยังใช้เพื่อจัดเก็บ ชื่อของไฟล์ MAIL ที่ตรงกันเมื่อตรวจสอบ เมล
<i>ERRNO</i>	ระบุค่าที่ตั้งค่าโดยรูทีนย่อยที่ล้มเหลว ล่าสุด ค่านี้ขึ้นอยู่กับระบบและมีไว้สำหรับวัตถุประสงค์การดีบัก
<i>LINENO</i>	ระบุหมายเลขบรรทัดของบรรทัดปัจจุบันภายใน สคริปต์หรือฟังก์ชันที่กำลังดำเนินการ
<i>OLDPWD</i>	บ่งชี้ไดเรกทอรีการทำงานก่อนหน้านี้ที่ตั้งค่าโดยคำสั่ง <i>cd</i>
<i>OPTARG</i>	ระบุค่าของอาร์กิวเมนต์อ็อปชันล่าสุดที่ประมวลผลโดย คำสั่งในตัวปกติ <i>getopts</i>
<i>OPTIND</i>	ระบุดัชนีของอาร์กิวเมนต์อ็อปชันล่าสุดที่ประมวลผลโดย คำสั่งในตัวปกติ <i>getopts</i>
<i>PPID</i>	ระบุหมายเลขไพรเซสของพารেন্টของเชลล์
<i>PWD</i>	บ่งชี้ไดเรกทอรีการทำงานปัจจุบันที่ตั้งค่าโดยคำสั่ง <i>cd</i>
<i>RANDOM</i>	สร้างเลขจำนวนเต็มแบบสุ่ม ซึ่งกระจายอย่างเท่าเทียมกันระหว่าง 0 และ 32767 ลำดับของตัวเลขแบบสุ่มสามารถเริ่มต้นโดยการกำหนด ค่าตัวเลขให้กับตัวแปร <i>RANDOM</i>
<i>REPLY</i>	ตั้งค่าโดยคำสั่ง <i>select</i> และโดยคำสั่งในตัวปกติ <i>read</i> เมื่อไม่ได้ระบุอาร์กิวเมนต์
<i>SECONDS</i>	ระบุจำนวนวินาทีตั้งแต่สิ้นการเรียกใช้ เชลล์ ถ้าตัวแปรนี้ได้รับการกำหนดค่า ค่าที่ส่งคืนหลังจากการอ้างอิง จะเป็นค่าที่ถูกกำหนดบวกจำนวนวินาทีตั้งแต่ การกำหนดค่า

ตัวแปรที่ใช้โดย Korn เชลล์หรือ POSIX เชลล์:

เชลล์ใช้ตัวแปรต่อไปนี้

ไอเท็ม	คำอธิบาย
CDPATH	บ่งชี้พาทหารค้นหาสำหรับคำสั่ง cd (เปลี่ยนไดเรกทอรี)
COLUMNS	กำหนดความกว้างของหน้าต่างแก้ไขสำหรับโหมดแก้ไขของเซลล์ และสำหรับการพิมพ์รายการ select
EDITOR	หากค่าของพารามิเตอร์นี้สิ้นสุดใน emacs, gmacs, หรือ vi และไม่ได้ตั้งค่าตัวแปร VISUAL ด้วยคำสั่งในตัวพิเศษ set ผลคืออ็อปชันที่สอดคล้องกันจะเปิด
ENV	หากมีการตั้งค่าตัวแปรนี้จะทำการทดแทนพารามิเตอร์ ในค่าเพื่อสร้างชื่อพารของสคริปต์ที่จะถูกดำเนินการ เมื่อเรียกใช้เซลล์ โดยปกติ ไฟล์นี้ใช้สำหรับนิยามสมนาม และฟังก์ชัน ตัวแปรนี้จะถูกละเว้นสำหรับเซลล์แบบ ไม่โต้ตอบ
FCEDIT	ระบุชื่อโปรแกรมแก้ไขดีฟอลต์สำหรับคำสั่งในตัวปกติ fc
FPATH	ระบุพาทหารค้นหาสำหรับนิยามฟังก์ชัน พาทนี้จะถูกค้นหาเมื่อมีการอ้างอิงฟังก์ชันพร้อมกับแฟล็ก -u และเมื่อไม่พบคำสั่ง หากพบไฟล์ปฏิบัติการ ไฟล์จะถูกอ่าน และดำเนินการในสภาวะแวดล้อมปัจจุบัน
HISTFILE	หากมีการตั้งค่าตัวแปรนี้เมื่อเรียกใช้เซลล์ ค่าจะเป็น ชื่อพารของไฟล์ที่จะใช้เพื่อจัดเก็บประวัติ คำสั่ง โปรเซสการเริ่มต้นของไฟล์ history อาจขึ้นอยู่กับไฟล์สตาร์ทอัพระบบ เนื่องจากไฟล์สตาร์ทอัพบางไฟล์ สามารถมีคำสั่งที่เรียกใช้คำสั่งตั้งผู้ระบุ สำหรับ HISTFILE และ HISTSIZE ก่อนอย่างมีประสิทธิภาพ ตัวอย่างเช่น คำสั่งนิยามฟังก์ชันมีกรับบันทึกไว้ในไฟล์ history หากผู้ดูแลระบบ รวมนิยามฟังก์ชันไว้ในไฟล์สตาร์ทอัพระบบที่ถูกเรียกก่อนไฟล์ ENV หรือก่อนการตั้งค่าตัวแปร HISTFILE หรือ HISTSIZE ไฟล์ history จะมีการเริ่มต้นก่อนผู้ใช้ สามารถแก้ไขลักษณะได้
HISTSIZE	หากมีการตั้งค่าตัวแปรนี้เมื่อเรียกใช้เซลล์ จำนวนของคำสั่ง ที่ป้อนก่อนหน้าซึ่งเซลล์นี้สามารถเข้าถึงได้ จะมากกว่าหรือเท่ากับ ตัวเลขนี้ ค่าดีฟอลต์คือ 128 คำสั่ง สำหรับผู้ใช้ที่ไม่ใช่ผู้ไชราก และ 512 คำสั่งสำหรับผู้ไชราก
HOME	บ่งชี้ชื่อของไดเรกทอรีโฮมของคุณ ซึ่งกลายเป็น ไดเรกทอรีปัจจุบันในทันทีที่ล็อกอินเสร็จสมบูรณ์ โปรแกรม login จะเริ่มต้นตัวแปรนี้ คำสั่ง cd ใช้ค่าของ พารามิเตอร์ \$HOME เป็นค่าดีฟอลต์ การใช้ตัวแปรนี้แทนชื่อพารที่ชัดเจนในเซลล์โพธิ์เตอร์ช่วยให้โพธิ์เตอร์สามารถรันจากไดเรกทอรีอื่นได้โดยไม่ต้องทำการเปลี่ยนแปลง
IFS	ระบุ IFS (ตัวแบ่งฟิลด์ภายใน) ซึ่งโดยปกติแล้ว เป็น ช่องว่าง แท็บ และบรรทัดใหม่ ที่ใช้เพื่อแบ่งค่าของคำสั่งซึ่งเป็นผลมาจากการ ทดแทนคำสั่งหรือพารามิเตอร์ และสำหรับการแบ่งค่าด้วยคำสั่งในตัว ปกติ read อักขระตัวแรกของพารามิเตอร์ IFS ใช้เพื่อแบ่งอาร์กิวเมนต์สำหรับการทดแทน \$*
LANG	นำเสนอค่าดีฟอลต์สำหรับตัวแปร LC_*
LC_ALL	ยกเลิกค่าของตัวแปร LANG และ LC_*
LC_COLLATE	กำหนดลักษณะการทำงานของนิพจน์ช่วงภายในการจัดรูปแบบ
LC_CTYPE	กำหนดการจัดประเภทอักขระ การแปลงตัวพิมพ์ และแอตทริบิวต์ อักขระอื่น
LC_MESSAGES	กำหนดภาษาซึ่งใช้ในการเขียนข้อความ
LINES	กำหนดความยาวคอลัมน์สำหรับการพิมพ์การเลือก รายการเลือกพิมพ์ตามแนวตั้งจนถึงประมาณเศษสองส่วนสามของบรรทัดที่ระบุโดย การป้อนค่าตัวแปร LINES
MAIL	บ่งชี้ชื่อพารของไฟล์ที่ใช้โดยระบบเมล เพื่อตรวจสอบการมาถึงของเมลใหม่ หากมีการตั้งค่าตัวแปรนี้เป็นชื่อของไฟล์เมล และไม่ได้ตั้งค่าตัวแปร MAILPATH เซลล์จะแจ้งผู้ใช้ให้ทราบถึงเมลใหม่ในไฟล์ที่ระบุ
MAILCHECK	ระบุความถี่ (ในหน่วยวินาที) ที่เซลล์จะตรวจสอบหาการเปลี่ยนแปลง ในเวลาการแก้ไขของไฟล์ใดๆ ซึ่งระบุโดยตัวแปร MAILPATH หรือ MAIL ค่าดีฟอลต์คือ 600 วินาที เมื่อเวลาผ่านไป เซลล์จะตรวจสอบ ก่อนการออกใช้พร้อมตัดไป
MAILPATH	ระบุรายชื่อไฟล์ที่ค้นด้วยเครื่องหมายจุด หากมีการตั้งค่า ตัวแปรนี้ เซลล์จะแจ้งผู้ใช้ให้ทราบถึงการแก้ไขใดๆ ในไฟล์ที่ระบุ ซึ่งเกิดขึ้นในระหว่างรอบระยะเวลาในหน่วยวินาทีซึ่งระบุ โดยตัวแปร MAILCHECK ชื่อไฟล์แต่ละชื่อสามารถ ตามด้วย ? และข้อความที่จะถูกพิมพ์ ข้อความจะอยู่ภายใต้ การทดแทนตัวแปรด้วยตัวแปร &_ ที่กำหนดเป็นชื่อของไฟล์ที่เปลี่ยนแปลง ข้อความดีฟอลต์คือ you have mail in \$_ กำหนดที่ตั้งของแค็ตตาล็อกข้อความสำหรับการประมวลผล LC_MESSAGES
NLSPATH	
PATH	บ่งชี้พาทหารค้นหาของคำสั่ง ซึ่งเป็นรายการที่จัดลำดับของ ชื่อพารไดเรกทอรีที่ค้นด้วยเครื่องหมายจุด เซลล์ค้นหาไดเรกทอรีเหล่านี้ในลำดับที่ระบุเมื่อเซลล์ค้นหาคำสั่ง สตริง null ที่ใดๆ ในรายการ แสดงถึงไดเรกทอรีปัจจุบัน
PS1	ระบุสตริงที่จะใช้เป็นพร้อมต์ระบบหลัก ค่าของพารามิเตอร์นี้ถูกขยายสำหรับการทดแทนพารามิเตอร์เพื่อกำหนด สตริงพร้อมต์หลัก ซึ่งเป็น \$ โดยค่าดีฟอลต์ อักขระ ! ใน สตริงพร้อมต์หลักถูกแทนที่โดยหมายเลขคำสั่ง

ไอเท็ม	คำอธิบาย
PS2	ระบุค่าของสตริงพร้อมเครื่องหมาย > โดยค่าดีฟอลต์
PS3	ระบุค่าของสตริงพร้อมเครื่องหมาย #? โดยค่าดีฟอลต์
PS4	ค่าของตัวแปรนี้ถูกขยายสำหรับการทดแทนพารามิเตอร์ และนำหน้าแต่ละบรรทัดของการติดตามการดำเนินการ ถ้าข้ามไปพร้อมดีฟอลต์ตามการดำเนินการจะเป็น +
SHELL	ระบุชื่อพาธของเชลล์ซึ่งเก็บไว้ใน สภาวะแวดล้อม
SHELL PROMPT	เมื่อใช้แบบโต้ตอบ เชลล์จะพร้อมด้วยค่าของพารามิเตอร์ <i>PS1</i> ก่อนการอ่าน คำสั่ง หากมีการป้อนบรรทัดใหม่ในเวลาใดก็ตามและเชลล์ต้องการอินพุตเพิ่มเติม เพื่อทำคำสั่งให้สมบูรณ์ เชลล์จะออกใช้พร้อมเครื่องหมาย (ค่าของพารามิเตอร์ <i>PS2</i>)
TMOU	ระบุจำนวนวินาทีที่เชลล์รอโดยไม่มีการเรียกใช้ ก่อนจะออก หากมีการตั้งค่าตัวแปร <i>TMOU</i> เป็นค่าที่มากกว่าศูนย์ (0) เชลล์จะออกถ้าไม่ได้ป้อนคำสั่งภายในจำนวนวินาทีที่กำหนดไว้หลังการออกใช้พร้อมเครื่องหมาย <i>PS1</i> (โปรดทราบว่าสามารถคอมไพล์เชลล์ด้วยขอบเขตสูงสุดที่ไม่เกินกว่าค่านี้) หมายเหตุ: หลังจากช่วงเวลาไทม์เอาต์หมดไปแล้ว จะมีการหยุดชั่วคราว นาน 60-วินาทีก่อนเชลล์ออก
VISUAL	หากค่าของตัวแปรนี้สิ้นสุดใน emacs, gmacs, หรือ vi อีอ็อปชันที่สอดคล้องกันจะเปิด

เชลล์ให้ค่าดีฟอลต์กับพารามิเตอร์ *PATH*, *PS1*, *PS2*, *MAILCHECK*, *TMOU*, และ *IFS* แต่พารามิเตอร์ *HOME*, *SHELL*, *ENV*, และ *MAIL* ไม่มีการตั้งค่าโดย เชลล์ (แม้ว่าพารามิเตอร์ *HOME* มีการตั้งค่าโดยคำสั่ง *login*)

การทดแทนคำสั่งในเชลล์ Korn หรือเชลล์ POSIX:

Korn เชลล์หรือ POSIX เชลล์อนุญาตให้คุณทำการทดแทนคำสั่งได้ ในการทดแทนคำสั่ง เชลล์ดำเนินการคำสั่งที่ระบุในสภาวะแวดล้อมเชลล์ย่อย และแทนที่คำสั่งนั้นด้วยเอาต์พุต

เมื่อต้องการดำเนินการทดแทนคำสั่งใน Korn เชลล์หรือ POSIX เชลล์ให้พิมพ์ ดังต่อไปนี้:

```
$(command)
```

หรือสำหรับเวอร์ชัน backquoted ให้พิมพ์ดังต่อไปนี้:

```
`command`
```

หมายเหตุ: แม้ว่าไวยากรณ์ backquote เป็นที่ยอมรับโดย *ksh* แต่ถือว่ามีความล้าสมัยตามมาตรฐาน X/Open Portability Guide Issue 4 and POSIX มาตรฐานเหล่านี้แนะนำให้ แอ็พพลิเคชันที่ใช้ได้หลายระบบใช้ไวยากรณ์ $(command)$

เชลล์ขยายการทดแทนคำสั่งโดยการดำเนินการ *command* ใน สภาวะแวดล้อมเชลล์ย่อยและแทนที่การทดแทนคำสั่ง (ข้อความของ *command* บวกที่อยู่ในเครื่องหมาย $()$ หรือ backquotes) ด้วยเอาต์พุตมาตรฐาน ของคำสั่ง ลบลำดับของอักขระบรรทัดใหม่หนึ่งหรือหลายตัวที่ ตอนท้ายของการทดแทน

ในตัวอย่างต่อไปนี้ $()$ ที่ล้อมรอบคำสั่ง บ่งชี้ว่าเอาต์พุตของคำสั่ง *whoami* ถูกทดแทน:

```
echo My name is: $(whoami)
```

คุณสามารถทำการทดแทนคำสั่งเดียวกันด้วย:

```
echo My name is: `whoami`
```

เอาต์พุตจากทั้งสองตัวอย่างสำหรับผู้ใช้ *dee* คือ:

```
My name is: dee
```

คุณยังสามารถทดแทนนิพจน์ทางคณิตศาสตร์โดยใส่ไว้ใน () ตัวอย่างเช่น คำสั่ง:

```
echo Each hour contains  $((60 * 60))$  seconds
```

ให้ผลลัพธ์ต่อไปนี้:

```
Each hour contains 3600 seconds
```

Korn เชลล์หรือ POSIX เชลล์ลบอักขระบรรทัดใหม่ส่วนท้ายทั้งหมดเมื่อทำการทดแทนคำสั่ง ตัวอย่างเช่น ถ้าได้เรียกทอริปัจจุบันของคุณมีไฟล์ file1, file2, และ file3 คำสั่ง:

```
echo  $$(ls)$ 
```

ลบอักขระบรรทัดใหม่ และให้เอาต์พุตต่อไปนี้:

```
file1 file2 file3
```

เมื่อต้องการรักษอักขระบรรทัดใหม่ไว้ให้ใส่คำสั่งที่ทดแทนไว้ใน " ":

```
echo " $$(ls)$ "
```

การประเมินทางคณิตศาสตร์ใน Korn เชลล์หรือ POSIX เชลล์:

คำสั่งในตัว let ปกติของ Korn เชลล์หรือ POSIX เชลล์ช่วยให้คุณทำการคำนวณเลขจำนวนเต็มได้

ค่าคงที่อยู่ในรูปแบบ [Base]Number พารามิเตอร์ Base เป็นตัวเลขฐานสิบระหว่าง 2 และ 36 inclusive ที่แสดงถึงฐานทางคณิตศาสตร์ พารามิเตอร์ Number เป็นตัวเลขในฐานนั้น หากคุณละเว้นพารามิเตอร์ Base เชลล์จะใช้ฐาน 10

นิพจน์ทางคณิตศาสตร์ใช้ไวยากรณ์ การมาก่อน และความเชื่อมโยงของนิพจน์เดียวกันกับภาษาการเขียนโปรแกรม C ตัวดำเนินการจำนวนเต็มทั้งหมด ที่ไม่ใช่เครื่องหมายบวกคู่(++), เครื่องหมายยัติภังค์คู่(--), เครื่องหมายคำถาม-จุดคู่(?:), และเครื่องหมายจุลภาค(,) ได้รับการสนับสนุน ตารางต่อไปนี้แสดงรายการตัวดำเนินการ Korn เชลล์หรือ POSIX เชลล์ที่ต้องใช้ในลำดับการมาก่อนจากมากไปน้อย:

ตัวดำเนินการ	นิยาม
-	การลบแบบเอกภาค
!	การปฏิเสธทางตรรกะ
~	การปฏิเสธ bitwise
*	การคูณ
/	การหาร
%	การเตือน
+	การบวก
-	การลบ
<<, >>	เลื่อนไปทางซ้าย, เลื่อนไปทางขวา
<=, >=, <, >, ==, !=	การเปรียบเทียบ
&	Bitwise AND

ตัวดำเนินการ	นิยาม
^	Bitwise exclusive OR
	Bitwise OR
&&	AND ทางตรรกะ
	OR ทางตรรกะ
= *=, /=, & +=, -=, <<=, >>=, &=, ^=, =	การกำหนดค่า

ตัวดำเนินการทางคณิตศาสตร์จำนวนมาก เช่น *, &, <, และ > มีความหมายพิเศษใน Korn เซลล์หรือ POSIX เซลล์ อีกซึ่ขะเหล่านี้ต้องใส่ไว้ในเครื่องหมายอัญประกาศ ตัวอย่างเช่น เมื่อต้องการคูณค่าปัจจุบันของ y ด้วย 5 และกำหนดค่าใหม่ให้กับ y อีกครั้ง ให้ใช้นิพจน์:

```
let "y = y * 5"
```

การใส่ นิพจน์ไว้ในเครื่องหมายอัญประกาศลบ ความหมายพิเศษของอักขระ *

คุณสามารถจัดกลุ่มการดำเนินงานภายในนิพจน์คำสั่ง let เพื่อบังคับใช้การจัดกลุ่ม ตัวอย่างเช่น ในนิพจน์:

```
let "z = q * (z - 10)"
```

คำสั่งคูณ q ด้วย ค่าที่ลดลงของ z

Korn เซลล์หรือ POSIX เซลล์รวมรูปแบบอื่น ของคำสั่ง let ถ้าจะประเมินนิพจน์เดียว เท่านั้น เซลล์จัดการคำสั่งที่อยู่ใน (()) เหมือนกับ นิพจน์ที่อยู่ในเครื่องหมายอัญประกาศ ดังนั้น นิพจน์:

```
((x = x / 3))
```

เทียบเท่ากับ:

```
let "x = x / 3"
```

พารามิเตอร์ที่ระบุมีการอ้างอิงโดยชื่อภายในนิพจน์ ทางคณิตศาสตร์โดยไม่ต้องใช้ไวยากรณ์การทดแทนพารามิเตอร์ เมื่อมีการอ้างอิงพารามิเตอร์ ที่ระบุ ค่าของพารามิเตอร์จะถูกประเมินเป็นนิพจน์ทางคณิตศาสตร์

ระบุการสแดงแทนเลขจำนวนเต็มภายในของพารามิเตอร์ ที่ระบุด้วยแฟล็ก -i ของคำสั่งในตัวพิเศษ typeset โดยใช้แฟล็ก -i การประเมินทางคณิตศาสตร์มีการทำ บนค่าของแต่ละการกำหนดให้กับพารามิเตอร์ที่ระบุ หากคุณไม่ได้ระบุฐานทางคณิตศาสตร์ การกำหนดแรกให้กับพารามิเตอร์จะกำหนดฐาน ทางคณิตศาสตร์ ฐานนี้ใช้เมื่อการทดแทนพารามิเตอร์เกิดขึ้น

หลักการที่เกี่ยวข้อง:

“คำสั่ง Korn เซลล์หรือ POSIX เซลล์” ในหน้า 275

Korn เซลล์คือตัวแปลคำสั่งแบบโต้ตอบและภาษา การเขียนโปรแกรมคำสั่ง Korn เซลล์สอดคล้องกับ Portable Operating System Interface for Computer Environments (POSIX) ซึ่งเป็นมาตรฐานสากลสำหรับ ระบบปฏิบัติการ

“พารามิเตอร์ใน Korn เซลล์” ในหน้า 251

พารามิเตอร์ Korn เซลล์มีการอธิบายข้างล่าง

การแบ่งฟิลด์ใน Korn เซลล์หรือ POSIX เซลล์:

หลังจากทำการทดแทนคำสั่งแล้ว Korn เซลล์จะสแกนผลลัพธ์ของการทดแทนเพื่อหาอักขระตัวแบ่งฟิลด์ที่พบในตัวแปร IFS (Internal Field Separator) ในตำแหน่งที่พบอักขระดังกล่าว เซลล์จะแบ่ง การทดแทนออกเป็นอาร์กิวเมนต์ที่แยกกัน

เซลล์รักษาอาร์กิวเมนต์ explicit null ไว้ (" " หรือ ' ') และลบอาร์กิวเมนต์ implicit null (ที่เป็นผลมาจากพารามิเตอร์ซึ่งไม่มีค่า)

- หากค่าของ IFS เป็นช่องว่าง แท็บ หรืออักขระบรรทัดใหม่ หรือถ้าไม่มีการตั้งค่า ลำดับใดๆ ของช่องว่าง แท็บ หรืออักขระบรรทัดใหม่ที่ ตอนต้นหรือตอนท้ายของอินพุตจะถูกละเว้นไป และลำดับใดๆ ของอักขระ ดังกล่าวภายในอินพุตจะทำหน้าที่คั่นฟิลด์ ตัวอย่างเช่น อินพุตต่อไปนี้ ให้ผลเป็นสองฟิลด์คือ school และ days:

```
<newline><space><tab>school<tab><tab>days<space>
```

- มิฉะนั้น และถ้าค่าของ IFS ไม่ใช่ null กฎต่อไปนี้จะใช้ตามลำดับ IFS พื้นที่ว่าง ใช้เพื่อหมายถึงลำดับใดๆ (ศูนย์หรืออินสแตนท์เพิ่มเติม) ของอักขระพื้นที่ว่างที่อยู่ในค่า IFS (ตัวอย่างเช่น ถ้า IFS มีช่องว่าง/จุลภาค/แท็บ ลำดับใดๆ ของช่องว่างและอักขระแท็บจะถูกพิจารณาเป็น IFS พื้นที่ว่าง)

1. IFS พื้นที่ว่างถูกละเว้นที่ตอนต้นและตอนท้ายของอินพุต
2. การเกิดขึ้นแต่ละครั้งในอินพุตของอักขระ IFS ที่ไม่ใช่ IFS พื้นที่ว่าง ควบคู่ไปกับ IFS พื้นที่ว่างที่ต่อเนื่องใดๆ จะคั่นฟิลด์
3. IFS พื้นที่ว่างซึ่งมีความยาวไม่ใช่ศูนย์จะคั่นฟิลด์

รายการของคำสั่งในตัวพิเศษ Korn เซลล์หรือ POSIX เซลล์:

คำสั่งพิเศษมีการสร้างขึ้นใน Korn เซลล์และ POSIX เซลล์ และดำเนินการในโปรเซสเซลล์

ไอเท็ม	คำอธิบาย
:(จุดคู่)	ขยายอาร์กิวเมนต์อย่างเดี่ยว
.(จุด)	อ่านไฟล์ที่ระบุ แล้วดำเนินการคำสั่ง
break	ออกจากลูป for, while, until, หรือ select ที่ล้อมอยู่ ถ้ามี
continue	ทำการวนซ้ำถัดไปของลูป for, while, until, หรือ select ที่ล้อมอยู่ต่อไป
eval	อ่านอาร์กิวเมนต์เป็นอินพุตของเซลล์ และดำเนินการคำสั่ง ที่ได้
exec	ดำเนินการคำสั่งที่ระบุโดยพารามิเตอร์ อาร์กิวเมนต์ แทนเซลล์นี้ โดยไม่สร้างโปรเซสใหม่
exit	ออกจากเซลล์ที่เป็นเจ้าของสถานะการออกที่มีการระบุโดยพารามิเตอร์ n
export	ทำเครื่องหมายชื่อของเอ็กซ์พอร์ตอัตโนมัติในสภาวะแวดล้อมของ คำสั่งที่ดำเนินการในเวลาต่อมา
newgrp	เทียบเท่ากับคำสั่ง exec/usr/bin/newgrp [Group ...]
readonly	ทำเครื่องหมายชื่อแบบอ่านอย่างเดียวที่ระบุ
return	ส่งผลให้เซลล์กลับไปยังสคริปต์ที่เรียกใช้
set	เขียนชื่อและค่าของตัวแปรเซลล์ทั้งหมดในลำดับการจัดเรียงของโลแคล ปัจจุบัน ยกเว้นว่าระบุชื่อพจนานุกรมหรืออาร์กิวเมนต์
shift	เปลี่ยนชื่อพารามิเตอร์ตำแหน่ง
times	พิมพ์เวลาผู้ใช้และระบบสะสมสำหรับทั้งเซลล์และ โปรเซสที่รันจากเซลล์
trap	รันคำสั่งที่ระบุเมื่อเซลล์ได้รับสัญญาณ ที่ระบุ
typeset	ตั้งค่าแอตทริบิวต์และค่าสำหรับพารามิเตอร์เซลล์
unset	ยกเลิกการตั้งค่าและแอตทริบิวต์ของพารามิเตอร์ที่ระบุ

หลักการที่เกี่ยวข้อง:

“คำสั่งในตัว Korn เซลล์หรือ POSIX เซลล์” ในหน้า 258

คำสั่งพิเศษมีการสร้างขึ้นใน Korn เซลล์และ POSIX เซลล์ และดำเนินการในโปรเซสเซลล์

คำสั่งในตัวปกติ Korn เซลล์หรือ POSIX เซลล์:

ข้อมูลต่อไปนี้เป็นรายการของคำสั่งในตัวปกติของ Korn เซลล์หรือ POSIX เซลล์

ไอเท็ม	คำอธิบาย
alias	พิมพ์รายการของสมนามไปยังเอาต์พุตมาตรฐาน
bg	วางงานที่ระบุไว้ในพื้นหลัง
cd	เปลี่ยนไดเรกทอรีปัจจุบันเป็นไดเรกทอรีที่ระบุ หรือทดแทน สตริงปัจจุบันด้วยสตริงที่ระบุ
echo	เขียนสตริงอักขระไปที่เอาต์พุตมาตรฐาน
fc	เลือกช่วงของคำสั่งจากคำสั่งตัวแปร HISTSIZE ล่าสุด ที่พิมพ์ที่เทอร์มินัล ดำเนินการคำสั่งที่ระบุอีกครั้งหลังจากทำการทดแทนค่าเก่าด้วยค่าใหม่
fg	นำงานที่ระบุไปไว้ในพื้นหน้า
getopts	ตรวจสอบพารามิเตอร์ <i>Argument</i> สำหรับอ็อปชันที่ถูกต้อง
jobs	แสดงรายการข้อมูลสำหรับงานที่ระบุ
kill	ส่งสัญญาณ TERM (ยุติ) ไปยังงานหรือโปรเซสที่ระบุ
let	ประเมินนิพจน์ทางคณิตศาสตร์ที่ระบุ
print	พิมพ์เซลล์เอาต์พุต
pwd	เทียบเท่ากับคำสั่ง print -r - \$PWD
read	ใช้เซลล์อินพุต
ulimit	ตั้งค่าหรือแสดงขีดจำกัดรีซอร์สของโปรเซสผู้ใช้งานที่กำหนดไว้ในไฟล์ /etc/security/limits
umask	พิจารณาสิทธิ์ในไฟล์
unalias	ลบพารามิเตอร์ในรายชื่อออกจากรายการสมนาม
wait	รอกงานที่ระบุและยุติ
whence	บ่งชี้วิธีการตีความสำหรับแต่ละชื่อที่ระบุ ถ้าใช้เป็นชื่อคำสั่ง

สำหรับข้อมูลเพิ่มเติม ให้ดูที่ “คำสั่งในตัว Korn เซลล์หรือ POSIX เซลล์” ในหน้า 258

หลักการที่เกี่ยวข้อง:

“คำสั่งในตัว Korn เซลล์หรือ POSIX เซลล์” ในหน้า 258

คำสั่งพิเศษมีการสร้างขึ้นใน Korn เซลล์และ POSIX เซลล์ และดำเนินการในโปรเซสเซลล์

นิพจน์เงื่อนไขสำหรับ Korn เซลล์หรือ POSIX เซลล์:

นิพจน์เงื่อนไขใช้พร้อมกับคำสั่งผสม [[เพื่อทดสอบแอตทริบิวต์ของไฟล์และเพื่อเปรียบเทียบสตริง

การแบ่งค่าและการทดแทนชื่อไฟล์ไม่ทำบนค่าที่ปรากฏขึ้นระหว่าง [[และ]] แต่ละ นิพจน์สร้างขึ้นจากนิพจน์เอกภาคหรือไบนารีต่อไปนี้ตั้งแต่หนึ่งรายการ ขึ้นไป:

ไอเท็ม	คำอธิบาย
-a File	จริง ถ้าไฟล์ที่ระบุเป็นลิงก์สัญลักษณ์ที่ชี้ไปยังไฟล์อื่น ซึ่งมีอยู่
-b File	จริง ถ้ามีไฟล์ที่ระบุและเป็นไฟล์พิเศษบล็อก
-c File	จริง ถ้ามีไฟล์ที่ระบุและเป็นไฟล์พิเศษอักขระ
-d File	จริง ถ้ามีไฟล์ที่ระบุและเป็นไดเรกทอรี
-e File	จริง ถ้ามีไฟล์ที่ระบุ
-f File	จริง ถ้ามีไฟล์ที่ระบุและเป็นไฟล์ปกติ
-g File	จริง ถ้ามีไฟล์ที่ระบุและมีการตั้งค่าบิต setgid
-h File	จริง ถ้ามีไฟล์ที่ระบุและเป็นลิงก์สัญลักษณ์
-k File	จริง ถ้ามีไฟล์ที่ระบุและมีการตั้งค่าบิต sticky
-n String	จริง ถ้าความยาวของสตริงที่ระบุไม่ใช่ศูนย์
-o Option	จริง ถ้าอ็อปชันที่ระบุเป็น on
-p File	จริง ถ้ามีไฟล์ที่ระบุและเป็นไฟล์พิเศษ FIFO หรือ ไพพ์
-r File	จริง ถ้ามีไฟล์ที่ระบุและสามารถอ่านได้โดยโปรเซสปัจจุบัน
-s File	จริง ถ้ามีไฟล์ที่ระบุและมีขนาดมากกว่า 0
-t FileDescriptor	จริง ถ้าหมายเลขตัวอธิบายไฟล์ที่ระบุเปิดอยู่และเชื่อมโยงกับ อุปกรณ์เทอร์มินัล
-u File	จริง ถ้ามีไฟล์ที่ระบุและมีการตั้งค่าบิต setuid

ไอเท็ม	คำอธิบาย
<code>-w File</code>	จริง ถ้ามีไฟล์ที่ระบุและบิต write เป็น on อย่างไรก็ตาม จะไม่สามารถเขียนไฟล์บนระบบไฟล์แบบอ่านอย่างเดียวแม้ว่าการทดสอบนี้ บ่งชี้จริง
<code>-x File</code>	จริง ถ้ามีไฟล์ที่ระบุและแฟล็ก execute เป็น on ถ้ามีไฟล์ที่ระบุและเป็นไดเรกทอรี โพรเซสปัจจุบันจะมีสิทธิ์การค้นหาในไดเรกทอรี
<code>-z String</code>	จริง ถ้าความยาวของสตริงที่ระบุเป็น 0
<code>-L File</code>	จริง ถ้ามีไฟล์ที่ระบุและเป็นลิงก์สัญลักษณ์
<code>-O File</code>	จริง ถ้ามีไฟล์ที่ระบุและเป็นของ ID ผู้ใช้ที่มีประสิทธิภาพ ของโปรเซสนี้
<code>-G File</code>	จริง ถ้ามีไฟล์ที่ระบุและกลุ่มของไฟล์ตรงกับ ID กลุ่มที่มีประสิทธิภาพ ของโปรเซสนี้
<code>-S File</code>	จริง ถ้ามีไฟล์ที่ระบุและเป็นซ็อกเก็ต
<code>File1 -nt File2</code>	จริง ถ้ามี <i>File1</i> และใหม่กว่า <i>File2</i>
<code>File1 -ot File2</code>	จริง ถ้ามี <i>File1</i> และเก่ากว่า <i>File2</i>
<code>File1 -ef File2</code>	จริง ถ้ามี <i>File1</i> และ <i>File2</i> และอาจถึงไฟล์เดียวกัน
<code>String1 = String2</code>	จริง ถ้า <i>String1</i> เท่ากับ <i>String2</i>
<code>String1 != String2</code>	จริง ถ้า <i>String1</i> ไม่เท่ากับ <i>String2</i>
<code>String = Pattern</code>	จริง ถ้าสตริงที่ระบุตรงกับรูปแบบที่ระบุ
<code>String != Pattern</code>	จริง ถ้าสตริงที่ระบุไม่ตรงกับรูปแบบที่ระบุ
<code>String1 < String2</code>	จริง ถ้า <i>String1</i> มาก่อน <i>String2</i> ตาม ข้อมูลค่า ASCII ของอักขระ
<code>String1 > String2</code>	จริง ถ้า <i>String1</i> มาหลัง <i>String2</i> ตาม ข้อมูลค่า ASCII ของอักขระ
<code>Expression1 -eq Expression2</code>	จริง ถ้า <i>Expression1</i> เท่ากับ <i>Expression2</i>
<code>Expression1 -ne Expression2</code>	จริง ถ้า <i>Expression1</i> ไม่เท่ากับ <i>Expression2</i>
<code>Expression1 -lt Expression2</code>	จริง ถ้า <i>Expression1</i> น้อยกว่า <i>Expression2</i>
<code>Expression1 -gt Expression2</code>	จริง ถ้า <i>Expression1</i> มากกว่า <i>Expression2</i>
<code>Expression1 -le Expression2</code>	จริง ถ้า <i>Expression1</i> น้อยกว่าหรือเท่ากับ <i>Expression2</i>
<code>Expression1 -ge Expression2</code>	จริง ถ้า <i>Expression1</i> มากกว่าหรือเท่ากับ <i>Expression2</i>

หมายเหตุ: ในแต่ละนิพจน์ก่อนหน้า ถ้าตัวแปร *File* คล้ายกับ `/dev/fd/n` โดยที่ *n* คือ เลขจำนวนเต็ม การทดสอบจะใช้กับไฟล์เปิดที่มีหมายเลขตัวอธิบาย เป็น *n*

คุณสามารถสร้างนิพจน์ผสมจากค่าเบื้องต้นเหล่านี้ หรือส่วนที่เล็กกว่า โดยใช้นิพจน์ใดๆ ต่อไปนี้ที่แสดงรายการในลำดับการมาก่อน จากมากไปน้อย:

ไอเท็ม	คำอธิบาย
<code>(Expression)</code>	จริง ถ้านิพจน์ที่ระบุเป็นจริง ใช้เพื่อจัดกลุ่มนิพจน์
<code>!Expression</code>	จริง ถ้านิพจน์ที่ระบุเป็นเท็จ
<code>Expression1 &&Expression2</code>	จริง ถ้า <i>Expression1</i> และ <i>Expression2</i> เป็นจริงทั้งคู่
<code>Expression1 Expression2</code>	จริง ถ้า <i>Expression1</i> หรือ <i>Expression2</i> ใดๆ อย่างหนึ่งเป็นจริง

การใส่อักขระไว้ในเครื่องหมายอัญประกาศใน Korn เซลล์หรือ POSIX เซลล์:

เมื่อคุณต้องการให้ Korn เซลล์หรือ POSIX เซลล์อ่านอักขระ เป็นอักขระปกติ แทนความหมายที่เชื่อมโยงตามปกติใดๆ คุณต้องใส่อักขระไว้ใน *เครื่องหมายอัญประกาศ*

เมตาอักขระแต่ละตัวมีความหมายพิเศษในเซลล์ และส่งผลให้ยุติคำ ยกเว้นว่ามีการใส่ไว้ในเครื่องหมายอัญประกาศ อักขระต่อไปนี้ ถือว่าเป็นเมตาอักขระโดย Korn เซลล์หรือ POSIX เซลล์และต้องใส่ไว้ใน เครื่องหมายอัญประกาศถ้าต้องการแสดงถึงตัวอักขระเอง:

- ไพพ์ (|)
- ampersand (&)
- อัฒภาค (;)
- เครื่องหมายน้อยกว่า (<) และเครื่องหมายมากกว่า (>)

- วงเล็บซ้าย (() และวงเล็บขวา ())
- สัญลักษณ์ดอลลาร์ (\$))
- backquote (`) และเครื่องหมายอัฒประกาศเดี่ยว (')
- ทับขวา (\)
- เครื่องหมายอัฒประกาศคู่ (")
- อักขระบรรทัดใหม่
- อักขระช่องว่าง
- อักขระแท็บ

เมื่อต้องการปฏิเสธความหมายพิเศษของเมตาอักขระให้ใช้กลไกการใส่ไว้ในเครื่องหมายอัฒประกาศอย่างใดอย่างหนึ่ง ในรายการต่อไปนี้

ไอเท็ม	คำอธิบาย
ทับขวา	เครื่องหมายทับขวา (\) ที่ไม่ได้อยู่ในเครื่องหมายอัฒประกาศ รักษาตำแหน่งตามตัวอักษรของอักขระที่ตามมา โดยมีข้อยกเว้นของ อักขระบรรทัดใหม่ หากอักขระบรรทัดใหม่ตามหลังเครื่องหมายทับขวา เซลล์จะตีความว่าเป็นความต่อเนื่องของบรรทัด
เครื่องหมายอัฒประกาศเดี่ยว	การใส่อักขระไว้ในเครื่องหมายอัฒประกาศเดี่ยว (' ') รักษาตำแหน่งตามตัวอักษรของแต่ละอักขระภายในเครื่องหมาย อัฒประกาศเดี่ยว เครื่องหมายอัฒประกาศเดี่ยวไม่สามารถมีอยู่ภายในเครื่องหมาย อัฒประกาศ เดี่ยว แต่สามารถใช้เครื่องหมายทับขวาเพื่อ escape เครื่องหมาย อัฒประกาศเดี่ยวในสตริงที่ตั้งค่าในเครื่องหมายอัฒประกาศเดี่ยว เครื่องหมายอัฒประกาศที่ฝัง สามารถสร้างขึ้นได้โดยการเขียน ตัวอย่างเช่น: 'a\'\'b' ซึ่งให้ค่า a'b
เครื่องหมายอัฒประกาศคู่	การใส่อักขระไว้ในเครื่องหมายอัฒประกาศคู่ (" ") รักษาตำแหน่งตามตัวอักษรของอักขระทั้งหมดภายในเครื่องหมายอัฒประกาศ คู่ โดยมีข้อยกเว้นของสัญลักษณ์ดอลลาร์, backquote, และอักขระทับขวา ดังนี้: \$ สัญลักษณ์ดอลลาร์คงความหมายพิเศษในการแนะนำการขยายพารามิเตอร์ รูปแบบของการทดแทนคำสั่ง และการขยายทางคณิตศาสตร์ อักขระอินพุต ภายในสตริงที่ใส่ไว้ในเครื่องหมายอัฒประกาศและถูกใส่ไว้ระหว่าง \$ (และ) ที่ตรงกันด้วยจะไม่ได้รับผลกระทบจากเครื่องหมายอัฒประกาศคู่ แต่กำหนดคำสั่งซึ่งเอาต์พุตของคำสั่งนั้นจะแทนที่ \$ (...) เมื่อ ขยายค่า ภายในสตริงของอักขระ ตั้งแต่ \$ { ถึง } ที่ตรงกันซึ่งใส่ไว้ ต้องเป็นจำนวนคู่ของเครื่องหมาย อัฒประกาศคู่หรือเครื่องหมายอัฒประกาศเดี่ยว unescaped ถ้ามี ต้องใช้อักขระทับขวานำหน้าเพื่อ escape { หรือ } ตามตัวอักษร , เครื่องหมายทับขวาคงความหมายพิเศษในการแนะนำรูปแบบอื่นของ การทดแทนคำสั่งไว้ ส่วนของสตริงที่ใส่ไว้ในเครื่องหมาย ตั้งแต่ backquote แรกเริ่ม และอักขระไปจนถึง backquote ถัดไปซึ่งไม่ได้นำหน้าด้วยทับขวา กำหนดคำสั่งซึ่งเอาต์พุตของคำสั่งนั้นจะแทนที่ `... ` เมื่อขยายค่า \ ทับขวาคงความหมายพิเศษในการเป็นอักขระ escape เฉพาะถ้า ตามหลังด้วยอักขระอย่างใดอย่างหนึ่งต่อไปนี้เท่านั้น: \$, `, ", \, หรืออักขระบรรทัดใหม่

เครื่องหมายอัฒประกาศคู่ต้องมีทับขวานำหน้า เพื่อจะได้รวมไว้ในเครื่องหมายอัฒประกาศคู่ เมื่อคุณใช้เครื่องหมาย อัฒประกาศคู่ ถ้าทับขวามีการตามหลังต่อด้วยอักขระซึ่งจะถูกตีความว่ามี ความหมายพิเศษ ทับขวาจะถูกลบออก และอักขระ

ลำดับต่อมา จะถูกใช้ตามตัวอักษร หากทับขาไม่ได้ นำหน้าอักขระ ซึ่งจะถูกต้องความว่ามีความหมายพิเศษ ทับขาจะถูกปล่อยไว้ในตำแหน่งโดยไม่เปลี่ยน และอักขระที่ตามหลังต่อจากทับขานั้นยังไม่เปลี่ยนด้วย ตัวอย่างเช่น:

```
"\$" -> $
"a" -> \a
```

เงื่อนไขต่อไปนี้จะใช้กับเมตาอักขระและอักขระที่ใส่ไว้ในเครื่องหมายอัญประกาศใน Korn หรือ POSIX 셸ล์:

- ความหมายของสัญลักษณ์ดอลลาร์, ดอกจัน (\$*) และสัญลักษณ์ ดอลลาร์, สัญลักษณ์ at (\$@) เหมือนกับเมื่อไม่ได้ใส่ไว้ในเครื่องหมาย อัญประกาศ เมื่อใช้เป็นค่าการกำหนดพารามิเตอร์ หรือเมื่อใช้เป็นชื่อไฟล์
- เมื่อใช้เป็นอาร์กิวเมนต์คำสั่ง เครื่องหมายอัญประกาศคู่, สัญลักษณ์ดอลลาร์, ดอกจัน, เครื่องหมายอัญประกาศคู่ ("\$*") เทียบเท่ากับ "\$1\$2..." โดยที่ *d* คืออักขระแรกของพารามิเตอร์ IFS
- เครื่องหมายอัญประกาศคู่, สัญลักษณ์ at, ดอกจัน, เครื่องหมายอัญประกาศคู่ ("@\$") เทียบเท่ากับ "\$1" "\$2" ...
- ภายใน backquotes (` `) ทับขาจะอ้างอิงอักขระทับขา (\), เครื่องหมายอัญประกาศเดี่ยว ('), และสัญลักษณ์ดอลลาร์ (\$) หาก backquotes มีอยู่ภายในเครื่องหมาย อัญประกาศคู่ (" ") ทับขายังอ้างอิงอักขระเครื่องหมายอัญประกาศ คู่ด้วย
- การทดแทนพารามิเตอร์และคำสั่งเกิดขึ้นภายในเครื่องหมายอัญประกาศคู่ (" ")
- ความหมายพิเศษของค่าที่ส่งวนไว้หรือสมนามจะถูกลบออกโดยการใส่อักขระใดๆ ของค่าที่ส่งวนไว้ในเครื่องหมายอัญประกาศ คุณไม่สามารถใส่ชื่อฟังก์ชันหรือชื่อคำสั่งในตัว ไว้ในเครื่องหมายอัญประกาศ

Korn 셸ล์ที่จำกัด:

Korn 셸ล์ที่จำกัดใช้เพื่อตั้งค่าชื่อล็อกอินและสถานะแวดล้อมการดำเนินการ ซึ่งสามารถควบคุมความสามารถได้มากกว่าสถานะแวดล้อมของ Korn 셸ล์ปกติ

คำสั่ง `rksh` หรือ `ksh -r` เปิด Korn 셸ล์ที่จำกัด ลักษณะการทำงานของคำสั่งเหล่านี้เหมือนกับของคำสั่ง `ksh` ยกเว้นว่าไม่อนุญาตให้ทำการดำเนินการ ต่อไปนี้:

- เปลี่ยนไดเรกทอรีการทำงานปัจจุบัน
- ตั้งค่าของตัวแปร `SHELL`, `ENV`, หรือ `PATH`
- ระบุชื่อพารของคำสั่งที่มี / (เครื่องหมายทับ)
- เปลี่ยนทิศทางเอาต์พุตของคำสั่งด้วย `>` (caret ขวา), `>|` (caret ขวา, สัญลักษณ์ไพพ์), `<>` (caret ซ้าย, caret ขวา), หรือ `>>` (carets ขวาสองตัว)

หาก Korn 셸ล์ที่จำกัดกำหนดว่าคำสั่งที่จะรันเป็นเชลล์ไพธอน Korn 셸ล์ที่จำกัดจะใช้ Korn 셸ล์เพื่อรันคำสั่ง ในวิธีนี้สามารถนำเสนอเชลล์ไพธอนที่เข้าถึงศักยภาพเต็มรูปแบบของ Korn 셸ล์ในขณะที่ใช้เมนูคำสั่งที่จำกัดให้แก่ผู้ใช้ชั้นปลายได้ สถานการณ์นี้สมมติว่า ผู้ใช้ไม่มีสิทธิการเขียนและการดำเนินการในไดเรกทอรีเดียวกัน

หากมีการระบุพารามิเตอร์ `File [Parameter]` เมื่อ Korn 셸ล์เริ่มต้นขึ้น เชลล์จะรัน ไฟล์สคริปต์ที่ระบุโดยพารามิเตอร์ `File` รวมถึง พารามิเตอร์ที่ระบุใดๆ ไฟล์สคริปต์ที่ระบุต้องมีสิทธิการอ่าน ค่าติดตั้ง `setuid` และ `setgid` ใดๆ สำหรับไฟล์สคริปต์ถูกข้ามไปจากนั้น เชลล์อ่านคำสั่ง ถ้าใช้แฟล็ก `-c` หรือ `-s` อย่างไม่อย่างหนึ่ง อย่ระบุไฟล์สคริปต์

เมื่อเริ่มต้นด้วยคำสั่ง `rksh` เชลล์จะบังคับใช้ข้อจำกัดหลังจากการตีความไฟล์ `.profile` และ `/etc/environment` ด้วยเหตุนี้ ผู้เขียนไฟล์ `.profile` จึงสามารถควบคุม การดำเนินการของผู้ใช้ได้อย่างเต็มที่โดยทำการดำเนินการเซ็ทอัพและปล่อย ผู้ใช้ไว้

ในไอดีเรกทอรีที่เหมาะสม (อาจไม่ใช่ไอดีเรกทอรีล็อกอิน) ผู้ดูแลระบบ สามารถสร้างไอดีเรกทอรีของคำสั่งในไอดีเรกทอรี /usr/rbin ที่คำสั่ง `rksh` สามารถใช้ได้โดยการเปลี่ยนตัวแปร `PATH` ให้มีไอดีเรกทอรี ถ้าเริ่มต้นด้วยคำสั่ง `ksh -r` เซลล์จะใช้ข้อจำกัดเมื่อตีความไฟล์ `.profile`

เมื่อเรียกด้วยคำสั่ง `rksh` Korn เซลล์ที่จำกัด จะอ่านไฟล์ `.profile` ของผู้ใช้ (`$HOME/.profile`) และทำหน้าที่เป็น Korn เซลล์ปกติในขณะที่ทำเช่นนี้ ยกเว้นว่าการขัดจังหวะ ส่งผลให้ออกในทันทีแทนการกลับไปยังระดับคำสั่ง

คำที่สงวนไว้ใน Korn เซลล์หรือ POSIX เซลล์:

คำที่สงวนไว้ต่อไปนี้มีมีความหมายพิเศษใน Korn เซลล์ หรือ POSIX เซลล์

```
!           case    do
done        elif    else
esac       fi       for
function   if       in
select     then     time
until      while    {
}          [[      ]]
```

ระบบรับรู้อำนาจที่สงวนไว้เฉพาะถ้า คำที่สงวนไว้ปรากฏขึ้นโดยไม่มีเครื่องหมายอัฒภาค และเมื่อใช้คำในลักษณะต่อไปนี้เท่านั้น:

- คำแรกของคำสั่ง
- คำแรกที่ตามด้วยคำที่สงวนไว้ได้อย่างใดอย่างหนึ่งซึ่งไม่ใช่ `case`, `for`, หรือ `in`
- คำที่สามในคำสั่ง `case` หรือ `for` (เฉพาะ `in` ถูกต้องในกรณีนี้)

Korn เซลล์ที่เพิ่มประสิทธิภาพ (ksh93):

นอกเหนือจาก Korn เซลล์ระบบดีฟอลต์แล้ว (/usr/bin/ksh) AIX ยังนำเสนอเวอร์ชันที่เพิ่มประสิทธิภาพของ Korn เซลล์ /usr/bin/ksh93 ด้วย เวอร์ชันที่เพิ่มประสิทธิภาพนี้ โดยส่วนใหญ่แล้วเข้ากันได้กับเวอร์ชันดีฟอลต์ปัจจุบัน และมีคุณลักษณะเพิ่มเติมบางอย่างที่ไม่มีอยู่ใน Korn เซลล์ /usr/bin/ksh

บางสคริปต์อาจทำงานภายใต้ Korn เซลล์ ksh93 ในลักษณะที่แตกต่างจากที่ทำภายใต้เซลล์ดีฟอลต์ เนื่องจากการจัดการตัวแปรภายใต้สอง เซลล์แตกต่างกัน

หมายเหตุ: นอกจากนี้ ยังมีเวอร์ชันที่จำกัดของ Korn เซลล์ที่เพิ่มประสิทธิภาพ ซึ่งเรียกว่า `rksh93`

คุณลักษณะต่อไปนี้ไม่มีอยู่ใน Korn เซลล์ /usr/bin/ksh แต่มีอยู่ใน Korn เซลล์ /usr/bin/ksh93:

ไอเท็ม	คำอธิบาย
การเพิ่มประสิทธิภาพทางคณิตศาสตร์	คุณสามารถใช้ฟังก์ชัน <code>libm</code> (ฟังก์ชันคณิตศาสตร์ที่โดยปกติแล้ว พบในภาษาการเขียนโปรแกรม C) ภายใน นิพจน์คณิตศาสตร์ เช่น <code>\$ value=\$(sqrt(9))</code> มีตัวดำเนินการทางคณิตศาสตร์ เพิ่มเติม รวมถึงโครงสร้าง <code>+</code> , <code>++</code> , <code>--</code> , และ <code>?:</code> เอกภาค (ตัวอย่างเช่น <code>"x ? y : z"</code>) ตลอด จนตัวดำเนินการ , (จุลภาค) มีการสนับสนุนฐานทางคณิตศาสตร์สูงสุดถึง ฐาน 64 และยังสามารถลดยตัวทางคณิตศาสตร์ด้วย <code>"typeset -E"</code> (เลขชี้กำลัง) สามารถใช้เพื่อระบุจำนวนของตัวเลขที่สำคัญและ <code>"typeset -F"</code> (ลอยตัว) สามารถใช้เพื่อระบุจำนวนของ ตำแหน่งตัวเลขสำหรับตัวแปรทางคณิตศาสตร์ ขณะนี้ ตัวแปร <code>SECONDS</code> แสดงเศษหนึ่งส่วนร้อยของวินาทีที่ใกล้เคียงที่สุด แทนวินาทีที่ ใกล้เคียงที่สุด

ไอเท็ม	คำอธิบาย
ตัวแปรผสม	<p>สนับสนุนตัวแปรผสม ตัวแปรผสมช่วยให้ผู้ใช้สามารถระบุหลายค่าภายในชื่อตัวแปรเดียว ค่าแต่ละค่าถูกกำหนดด้วยตัวแปร subscript และแบ่งจากตัวแปรพารามิเตอร์ด้วยเครื่องหมายจุด (.). ตัวอย่างเช่น:</p> <pre>\$ myvar=(x=1 y=2) \$ print "\${myvar.x}" 1</pre>
การกำหนดผสม	<p>การกำหนดผสมได้รับการสนับสนุน เมื่อเริ่มต้นอาร์เรย์ ทั้งสำหรับอาร์เรย์ที่จัดทำดัชนีและอาร์เรย์เชื่อมโยง ค่าการกำหนดมีการใส่ไว้ในเครื่องหมายวงเล็บ ดังแสดงในตัวอย่าง ต่อไปนี้:</p> <pre>\$ numbers=(zero one two three) \$ print \${numbers[0]} \${numbers[3]} zero three</pre>
อาร์เรย์เชื่อมโยง	<p>อาร์เรย์เชื่อมโยงเป็นอาร์เรย์ที่มีสตริงเป็นดัชนี</p> <p>คำสั่ง <code>typeset</code> ที่ใช้พร้อมกับแฟล็ก <code>-A</code> ช่วยให้คุณสามารถระบุอาร์เรย์เชื่อมโยงภายใน <code>ksh93</code> ได้ ตัวอย่างเช่น:</p> <pre>\$ typeset -A teammates \$ teammates=([john]=smith [mary]=jones) \$ print \${teammates[mary]} jones</pre>
การอ้างอิงชื่อตัวแปร	<p>คำสั่ง <code>typeset</code> ที่ใช้พร้อมกับแฟล็ก <code>-n</code> ช่วยให้คุณสามารถกำหนดหนึ่งชื่อตัวแปรเป็นการอ้างอิงไปยังตัวแปรอื่นได้ ในวิธีนี้ การแก้ไขค่าของตัวแปรจะแก้ไข ค่าของตัวแปรที่อ้างอิง ตัวอย่างเช่น:</p> <pre>\$ greeting="hello" \$ typeset -n welcome=greeting # establishes the reference \$ welcome="hi there" # overrides previous value \$ print \$greeting hi there</pre>
การขยายพารามิเตอร์	<p>โครงสร้างการขยายพารามิเตอร์มีอยู่ต่อไปนี้:</p> <ul style="list-style-type: none"> <code>\${!varname}</code> คือชื่อของตัวแปรเอง <code>\${!varname[@]}</code> ตั้งชื่อดัชนีสำหรับ อาร์เรย์ <code>varname</code> <code>\${param:offset}</code> คือสตริงย่อยของ <code>param</code> โดยเริ่มต้นที่ <code>offset</code> <code>\${param:offset:num}</code> คือสตริงย่อยของ <code>param</code> โดยเริ่มต้นที่ <code>offset</code> สำหรับจำนวน <code>num</code> ของอักขระ <code>\${@:offset}</code> บ่งชี้พารามิเตอร์ตำแหน่งทั้งหมด โดยเริ่มต้นที่ <code>offset</code> <code>\${@:offset:num}</code> บ่งชี้พารามิเตอร์ตำแหน่ง <code>num</code> โดยเริ่มต้นที่ <code>offset</code> <code>\${param/pattern/repl}</code> ประเมิน เป็น <code>param</code> ด้วยการเกิดขึ้นครั้งแรกของ <code>pattern</code> ที่แทนที่โดย <code>repl</code> <code>\${param//pattern/repl}</code> ประเมิน เป็น <code>param</code> ด้วยการเกิดขึ้นทุกครั้งของ <code>pattern</code> ที่แทนที่โดย <code>repl</code> <code>\${param/#pattern/repl}</code> ถ้า <code>param</code> เริ่มต้น ด้วย <code>pattern</code> ผลคือ <code>param</code> จะถูกแทนที่โดย <code>repl</code> <code>\${param/%pattern/repl}</code> ถ้า <code>param</code> สิ้นสุด ด้วย <code>pattern</code> ผลคือ <code>param</code> จะถูกแทนที่โดย <code>repl</code>

ไอเท็ม	คำอธิบาย
ฟังก์ชัน ระเบียบวินัย	<p>ฟังก์ชันระเบียบวินัยเป็นฟังก์ชันที่เชื่อมโยงกับตัวแปรเฉพาะ ฟังก์ชันนี้ช่วยให้คุณสามารถกำหนดและเรียกฟังก์ชันได้ในทุกครั้งที่ตัวแปรถูกอ้างอิง ตั้งค่า หรือยกเลิกตั้งค่า ฟังก์ชัน เหล่านี้อยู่ในรูปแบบ <code>varname.function</code> โดยที่ <code>varname</code> คือ ชื่อของตัวแปรและ <code>function</code> คือฟังก์ชันระเบียบวินัย ฟังก์ชันระเบียบวินัยที่กำหนดไว้ล่วงหน้าคือ <code>get</code>, <code>set</code>, และ <code>unset</code></p> <ul style="list-style-type: none"> ฟังก์ชัน <code>varname.get</code> ถูกเรียกใช้ในทุกครั้งที่มีการอ้างอิง <code>varname</code> หากมีการตั้งค่าตัวแปรพิเศษ <code>.sh.value</code> ภายในฟังก์ชันนี้ ค่าของ <code>varname</code> จะเปลี่ยนเป็นค่านี้ ตัวอย่าง แบบง่ายคือเวลาของวัน: <pre>\$ function time.get > { > .sh.value=\$(date +%r) > } \$ print \$time 09:15:58 AM \$ print \$time # it will change in a few seconds 09:16:04 AM</pre> ฟังก์ชัน <code>varname.set</code> ถูกเรียกใช้ในทุกครั้งที่มีการตั้งค่า <code>varname</code> ตัวแปร <code>.sh.value</code> ได้รับค่าที่กำหนดให้ค่า ที่กำหนดให้กับ <code>varname</code> คือค่าของ <code>.sh.value</code> เมื่อ ฟังก์ชันเสร็จสมบูรณ์ ตัวอย่างเช่น: <pre>\$ function adder.set > { > let .sh.value=" \$ { .sh.value } + 1" > } \$ adder=0 \$ echo \$adder 1 \$ adder=\$adder \$ echo \$adder 2</pre> ฟังก์ชัน <code>varname.unset</code> มีการดำเนินการในทุกครั้งที่ยกเลิกการตั้งค่า <code>varname</code> ตัวแปรไม่ได้ถูกยกเลิกการตั้งค่าจริง ยกเว้นว่ามีการยกเลิกการตั้งค่าภายในตัวฟังก์ชันเอง มิฉะนั้นตัวแปรจะคงค่าของตัวเองไว้ <p>ภายในฟังก์ชันระเบียบวินัยทั้งหมด มีการตั้งค่าตัวแปรพิเศษ <code>.sh.name</code> เป็นชื่อของตัวแปร ในขณะที่ตั้งค่า <code>.sh.subscript</code> เป็นค่าของตัวแปร <code>subscript</code> ถ้าใช้ได้</p>
สภาวะแวดล้อมฟังก์ชัน	ฟังก์ชันที่ประกาศด้วยรูปแบบ <code>function myfunc</code> รันในสภาวะแวดล้อมฟังก์ชันแยกต่างหาก และสนับสนุนตัวแปรโลคัล ฟังก์ชันที่ประกาศเป็น <code>myfunc()</code> รันในสภาวะแวดล้อมเดียวกันกับเชลล์พาเรนต์
ตัวแปร	ตัวแปรที่ขึ้นต้นด้วย <code>.sh.</code> ถูกสำรองไว้โดยเชลล์และมีความหมายพิเศษ ให้ดูที่คำอธิบายของ ฟังก์ชันระเบียบวินัย ใน ตารางนี้ สำหรับคำอธิบายเกี่ยวกับ <code>.sh.name</code> , <code>.sh.value</code> , และ <code>.sh.subscript</code> นอกจากนี้ ยังมี <code>.sh.version</code> ซึ่งแสดงถึงเวอร์ชันของเชลล์
คำสั่งคืนของคำสั่ง	คำสั่งคืนของคำสั่ง มีดังนี้: <ul style="list-style-type: none"> หากไม่พบคำสั่งที่จะดำเนินการ จะมีการตั้งค่าสั่งคืนเป็น 127 หากพบคำสั่งที่จะดำเนินการแต่ไม่สามารถดำเนินการได้ จะมีการตั้งค่าสั่งคืนเป็น 126 หากดำเนินการคำสั่งแต่ถูกยุติโดยสัญญาณ คำสั่งคืน จะเป็น 256 บวกหมายเลขสัญญาณ
กฎการค้นหา PATH	คำสั่งในตัวพิเศษจะถูก ค้นหา ก่อน ตามด้วยฟังก์ชันทั้งหมด (รวมถึงฟังก์ชันในไต่เรียกทอริ <code>FPATH</code>) ตามด้วยส่วนในตัวอื่นๆ
ประวัติเชลล์	คำสั่ง <code>hist</code> ช่วยให้คุณสามารถแสดงและแก้ไขประวัติคำสั่งเชลล์ได้ใน <code>ksh</code> เชลล์ มีการใช้คำสั่ง <code>fc</code> คำสั่ง <code>fc</code> เป็น สมนามของ <code>hist</code> ตัวแปรคือ <code>HISTCMD</code> ซึ่งเพิ่มขึ้นหนึ่งครั้งสำหรับแต่ละคำสั่งที่ดำเนินการในประวัติเชลล์ปัจจุบัน และ <code>HISTEDIT</code> ซึ่งระบุโปรแกรมแก้ไขที่จะใช้เมื่อ ใช้คำสั่ง <code>hist</code>

คำสั่งในตัว

Korn shell ที่เพิ่มประสิทธิภาพมี คำสั่งในตัวต่อไปนี้:

- คำสั่ง **builtin** แสดงรายการคำสั่งในตัวที่มีอยู่ทั้งหมด
- คำสั่ง **printf** ทำงานในลักษณะ คล้ายกับรูทีน **printf()** C โลบรารี โปรดดูที่คำสั่ง **printf**
- **disown** บล็อกเซลล์ไม่ให้ส่ง SIGHUP ไปยัง คำสั่งที่ระบุ
- คำสั่ง **getconf** ทำงานในวิธีเดียวกับคำสั่ง **stand-alone /usr/bin/getconf** โปรดดูที่คำสั่ง **getconf**
- คำสั่งในตัว **read** มีแฟล็กต่อไปนี้:
 - **read -d {char}** อนุญาตให้คุณ ระบุตัวคั่นอักขระแทนอักขระบรรทัดใหม่ดีฟอลต์
 - **read -t {seconds}** อนุญาตให้คุณระบุขีดจำกัดเวลาในหน่วยวินาที ซึ่งเมื่อผ่านไปแล้วคำสั่ง **read** จะไทม์เอาต์ ถ้า **read** ไทม์เอาต์ จะส่งคืน FALSE
- คำสั่งในตัว **exec** มีแฟล็กต่อไปนี้:
 - **exec -a {name} {cmd}** ระบุว่า อาร์กิวเมนต์ 0 ของ **cmd** จะถูกแทนที่ด้วย **name**
 - **exec -c {cmd}** บอกให้ **exec** ล้างสถานะแวดล้อมก่อนดำเนินการ **cmd**
- คำสั่งในตัว **kill** มีแฟล็กต่อไปนี้:
 - **kill -n {signame}** ใช้สำหรับการระบุหมายเลขสัญญาณที่จะส่งไปยังโปรเซส ในขณะที่ **kill -s {signame}** ใช้เพื่อระบุชื่อสัญญาณ
 - **kill -l** ที่ไม่มีอาร์กิวเมนต์ แสดงรายการชื่อสัญญาณทั้งหมด แต่ไม่แสดงหมายเลขสัญญาณ
- คำสั่งในตัว **whence** มีแฟล็กต่อไปนี้:
 - แฟล็ก **-a** แสดงรายการที่ตรงกันทั้งหมด ไม่เฉพาะแต่รายการแรกที่พบเท่านั้น
 - แฟล็ก **-f** บอก **whence** ว่าไม่ต้องค้นหาฟังก์ชันใดๆ
- ลำดับอักขระ **escape** มีไว้สำหรับการใช้โดยคำสั่ง **print** และ **echo** ปุ่ม Esc (Escape) สามารถแสดงแทนโดยลำดับ \E
- คำสั่งในตัวปกติทั้งหมดรับแฟล็ก **-?** ซึ่งแสดง ไวยากรณ์ของคำสั่งที่ระบุ
- **getopts** ในตัว ต้องการ **optstring** เพื่อจะได้มี + นำหน้า เพื่อให้มีตัวเลือกต่างๆ ที่ขึ้นต้นด้วยสัญลักษณ์ +

ไอเท็ม	คำอธิบาย
ความแตกต่างเบ็ดเตล็ดอื่น ระหว่าง Korn shell ksh และ Korn shell ksh93	<p>ความแตกต่างอื่นมีดังนี้:</p> <ul style="list-style-type: none"> ด้วย Korn shell ksh93 คุณไม่สามารถเอ็กซ์พอร์ตฟังก์ชันโดยใช้คำสั่งในตัว <code>typeset -fx</code> ด้วย Korn shell ksh93 คุณไม่สามารถเอ็กซ์พอร์ตสมนามโดยใช้คำสั่งในตัว <code>alias -x</code> ด้วย Korn shell ksh93 เครื่องหมายดอลลาร์ตามด้วยเครื่องหมายอัฒภาคเดี่ยว (<code>'</code>) มีการตีความเป็นสตริง ANSIC คุณต้องใส่เครื่องหมายดอลลาร์ไว้ในเครื่องหมายอัฒภาค (<code>\\$\'</code>) เพื่อให้ได้ลักษณะการทำงานแบบเดิม (ksh) อาร์กิวเมนต์ที่แจ้งส่วนตรรกะสำหรับคำสั่งในตัว Korn shell ksh93 มีการเปลี่ยนแปลง ชุดของอาร์กิวเมนต์ที่แจ้งส่วนคำสั่งในตัว Korn shell ksh ซึ่งไม่มีการระบุใน เอกสารคู่มือไม่ทำงานใน Korn shell ksh93 ตัวอย่างเช่น <code>typeset -4i</code> ทำงานคล้ายกับ <code>typeset -i4</code> ใน Korn shell ksh แต่ไม่ทำงานใน Korn shell ksh93 ด้วย Korn shell ksh93 การทดแทนคำสั่งและการขยายทางคณิตศาสตร์มีการ ทำบนตัวแปรสภาวะแวดล้อมพิเศษ PS1, PS3, และ ENV ในขณะที่ขยาย ดังนั้น คุณต้อง escape สัญลักษณ์ grave (<code>`</code>) และเครื่องหมายดอลลาร์ และเครื่องหมายวงเล็บเปิด (<code>\$</code>) โดยใช้ เครื่องหมายทับขา (<code>\</code>) เพื่อคงลักษณะการทำงานเอาไว้ ตัวอย่างเช่น Korn shell ksh กำหนด <code>x=\$' name\ toperator'</code> ตามตัวอักษรเป็น <code>\$name\ toperator</code> ในขณะที่ Korn shell ksh93 ขยาย <code>\t</code> และกำหนดเป็น <code>name<\t expanded>operator</code> เพื่อรักษาลักษณะการทำงานของ Korn shell ksh คุณต้องใส่ <code>\$</code> ไว้ในเครื่องหมายอัฒภาค ตัวอย่างเช่น <code>x="\$' name\ toperator'</code> ตัวแปร <code>ERRNO</code> ถูกลบออกใน Korn shell ksh93 ใน Korn shell ksh93 ไม่มีการขยายชื่อไฟล์สำหรับเซลล์แบบไม่โต้ตอบ หลังจากสัญลักษณ์การเปลี่ยนทิศทาง ด้วย Korn shell ksh93 คุณต้องใช้อ็อปชัน <code>-t</code> ของคำสั่ง <code>alias</code> เพื่อแสดงสมนามที่ติดตาม คุณลักษณะนามแฝงที่ติดตามล้าสมัยแล้ว ดังนั้น นามแฝงที่แสดง อาจไม่ถูกติดตาม ด้วย Korn shell ksh93 ในโหมด emacs <code>Ctrl+T</code> จะสับเปลี่ยนอักขระปัจจุบัน กับอักขระก่อนหน้านี้ ด้วย ksh <code>Ctrl+T</code> จะสับเปลี่ยนอักขระปัจจุบันกับอักขระถัดไป Korn shell ksh93 ไม่อนุญาตให้ใช้วงเล็บที่ไม่สมดุลภายใน <code>\${name operator value}</code> ตัวอย่างเช่น <code>\${name- (}</code> ต้องการ escape เช่น <code>\${name- \ (}</code> เพื่อ ทำงานในทั้งสองเวอร์ชัน ด้วย Korn shell ksh93 คำสั่ง <code>kill -l</code> แสดงรายการเฉพาะ ชื่อสัญญาณ ไม่ได้แสดงค่าตัวเลขของสัญญาณ

สถานะการออกใน Korn shell หรือ POSIX shell:

ข้อผิดพลาดที่ตรวจพบโดยเซลล์ เช่น ข้อผิดพลาดไวยากรณ์ ส่งผลให้ เซลล์ส่งคืนสถานะการออกที่ไม่ใช่ศูนย์ มิฉะนั้น เซลล์จะส่งคืนสถานะการออก ของคำสั่งล่าสุดที่ดำเนินการ

เซลล์รายงานข้อผิดพลาดรันไทม์ที่ตรวจพบโดยการพิมพ์ คำสั่งหรือชื่อฟังก์ชันและสภาพข้อผิดพลาด หากจำนวนบรรทัด ซึ่งเกิดข้อผิดพลาดมีมากกว่า 1 จะมีการพิมพ์หมายเลขบรรทัด ใน `[]` (วงเล็บ) หลังจากคำสั่งหรือชื่อฟังก์ชันด้วย

สำหรับเซลล์แบบไม่โต้ตอบ ข้อผิดพลาดที่เกิดขึ้นเนื่องจาก คำสั่งในตัวพิเศษหรือคำสั่งชนิดอื่นจะส่งผลให้เซลล์เขียน ข้อความการวินิจฉัยดังแสดงในตารางต่อไปนี้:

ข้อผิดพลาด	ในตัวพิเศษ	ยูทิลิตี้อื่น
ข้อผิดพลาดไวยากรณ์ภาษาของเซลล์	จะออก	จะออก
ข้อผิดพลาดไวยากรณ์ยูทิลิตี้ (ข้อผิดพลาดอ็อปชันหรือตัวถูกดำเนินการ)	จะออก	จะไม่ออก
ข้อผิดพลาดการเปลี่ยนทิศทาง	จะออก	จะไม่ออก
ข้อผิดพลาดการกำหนดตัวแปร	จะออก	จะไม่ออก
ข้อผิดพลาดการขยาย	จะออก	จะออก
ไม่พบคำสั่ง	ใช้ไม่ได้	อาจออก
ไม่พบสคริปต์จุด	จะออก	ใช้ไม่ได้

หากข้อผิดพลาดใดๆ ที่แสดงเป็น "จะ (อาจ) ออก" เกิดขึ้น ในเซลล์ย่อย เซลล์ย่อยจะ (อาจ) ออกด้วยสถานะที่ไม่ใช่ศูนย์ แต่สคริปต์ที่มีเซลล์ย่อยจะไม่ออกเนื่องจากข้อผิดพลาด

ในทุกกรณีที่แสดงในตาราง เซลล์แบบโต้ตอบ จะเขียนข้อความการวินิจฉัยไปยังข้อผิดพลาดมาตรฐานโดยไม่ออก

พารามิเตอร์ใน Korn เซลล์:

พารามิเตอร์ Korn เซลล์มีการอธิบายข้างล่าง

พารามิเตอร์มีการกำหนดดังต่อไปนี้:

- ตัวระบุของอักขระเครื่องหมายดอกจัน (*), สัญลักษณ์ at (@), สัญลักษณ์เงินปอนด์ (#), เครื่องหมายคำถาม (?), ยัติภังค์ (-), สัญลักษณ์ดอลลาร์ (\$), และเครื่องหมายอัศเจรีย์ (!). อักขระเหล่านี้เรียกว่า *พารามิเตอร์พิเศษ*
- อาร์กิวเมนต์ที่แสดงแทนโดยตัวเลข (*พารามิเตอร์ตำแหน่ง*)
- พารามิเตอร์ที่แสดงแทนโดยตัวระบุ ด้วยค่าและแธตทริบิวต์ศูนย์หรือมากกว่า (*พารามิเตอร์/ตัวแปรที่ระบุ*)

คำสั่งในตัวพิเศษ `typeset` กำหนดค่าและแธตทริบิวต์ให้กับพารามิเตอร์ที่ระบุ แธตทริบิวต์ที่สนับสนุน โดย Korn เซลล์มีการอธิบายด้วยคำสั่งในตัวพิเศษ `typeset` พารามิเตอร์ที่เอ็กซ์พอร์ตส่งผ่านค่าและแธตทริบิวต์ไปยังสภาวะแวดล้อม

ค่าของพารามิเตอร์ที่ระบุมีการกำหนดโดย:

Name=Value [Name=Value] ...

หากแธตทริบิวต์เลขจำนวนเต็ม `-i` มีการตั้งค่าสำหรับพารามิเตอร์ Name พารามิเตอร์ Value ต้องถูกประเมินทางคณิตศาสตร์

เซลล์สนับสนุนสิ่งอำนวยความสะดวกอาร์เรย์มิติเดียว อิลิเมนต์ของพารามิเตอร์อาร์เรย์มีการอ้างอิงโดย `subscript Subscript` มีการแสดงแทนโดยนิพจน์ทางคณิตศาสตร์ที่ใส่ไว้ในเครื่องหมายวงเล็บ [] เมื่อต้องการกำหนดค่าให้กับอาร์เรย์ ให้ใช้ `set -A Name Value ...` ค่าของ `subscripts` ทั้งหมดต้องอยู่ในช่วงตั้งแต่ 0 ถึง 511 ไม่จำเป็น ต้องประกาศอาร์เรย์ การอ้างอิงใดๆ ถึงพารามิเตอร์ที่ระบุด้วย `subscript` ที่ถูกต้อง เป็นสิ่งที่ถูกต้องและอาร์เรย์จะถูกสร้างขึ้น ถ้าจำเป็น การอ้างอิงอาร์เรย์ โดยไม่มี `subscript` เทียบเท่ากับการอ้างอิงอิลิเมนต์ 0

พารามิเตอร์ตำแหน่งได้รับการกำหนดค่าโดยใช้คำสั่งพิเศษ `set` พารามิเตอร์ `$0` มีการตั้งค่าจากอาร์กิวเมนต์ 0 เมื่อ เรียกใช้เซลล์ อักขระ `$` ใช้เพื่อแนะนำ พารามิเตอร์ที่สามารถถูกทดแทนได้

หลักการที่เกี่ยวข้อง:

“เซลล์สตาร์ทอัพ” ในหน้า 278

คุณสามารถเริ่มต้น Korn เซลล์โดยใช้คำสั่ง `ksh`, คำสั่ง `psh` (POSIX เซลล์), หรือคำสั่ง `exec`

“ฟังก์ชัน Korn เซลล์” ในหน้า 279

คำที่สแกนไว้ ฟังก์ชัน กำหนดฟังก์ชันของเซลล์ เซลล์ อ่านและจัดเก็บฟังก์ชันไว้ภายใน ชื่อสมนามมีการแก้ไขเมื่อ อ่านฟังก์ชัน เซลล์ดำเนินการฟังก์ชันในลักษณะเดียวกันกับคำสั่ง โดยมีอาร์กิวเมนต์ที่ส่งผ่านเป็นพารามิเตอร์ตำแหน่ง

“การประเมินทางคณิตศาสตร์ใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 239

คำสั่งในตัว `let` ปกติของ Korn เซลล์หรือ POSIX เซลล์ ช่วยให้คุณสามารถทำการคำนวณเลขจำนวนเต็มได้

สิ่งอ้างอิงที่เกี่ยวข้อง:

“คำสั่งผสม Korn เซลล์” ในหน้า 277

คำสั่งผสมสามารถเป็นรายการของคำสั่งแบบง่ายหรือโฟลไลน์ หรือสามารถขึ้นต้นด้วยคำที่สำรองไว้ ในเวลาส่วนใหญ่ คุณจะใช้คำสั่งผสม เช่น `if`, `while`, และ `for` เมื่อ คุณเขียนเซลล์สคริปต์

การทดแทนพารามิเตอร์ใน Korn เซลล์หรือ POSIX เซลล์:

Korn เซลล์หรือ POSIX เซลล์อนุญาตให้คุณทำการทดแทนพารามิเตอร์ได้

พารามิเตอร์ที่ทดแทนได้มีดังต่อไปนี้:

ไอเท็ม	คำอธิบาย
<code>\$(Parameter)</code>	เซลล์อ่านอักขระทั้งหมดจาก <code>{</code> (ถึง) ที่ตรงกันเป็นส่วนหนึ่งของคำเดียวกัน แม้ว่าคำนั้น มีวงเล็บหรือเมตาอักขระ คำ ถ้ามี ชื่อพารามิเตอร์ที่ระบุ จะถูกทดแทน เครื่องหมายวงเล็บเป็นสิ่งที่จำเป็นเมื่อพารามิเตอร์ <code>Parameter</code> ตามด้วยตัวอักษร ตัวเลข หรือขีดเส้นใต้ซึ่งจะไม่ถูกตีความ เป็นส่วนหนึ่งของชื่อ หรือเมื่อพารามิเตอร์ที่ระบุเป็น subscripted หาก พารามิเตอร์ที่ระบุมีตัวเลขตั้งแต่หนึ่งตัวขึ้นไป พารามิเตอร์นั้นจะเป็น พารามิเตอร์ตำแหน่ง พารามิเตอร์ ตำแหน่งที่มีตัวเลขมากกว่าหนึ่งตัวต้องใส่ไว้ใน วงเล็บ ถ้าค่าของตัวแปรเป็น * หรือ@ พารามิเตอร์ตำแหน่งแต่ละตัวที่ขึ้นต้นด้วย \$1 จะถูกทดแทน (แบ่งด้วยอักขระตัวแบ่งฟิลด์) ถ้าใช้ตัวระบุอาร์เรย์ที่มี subscript * หรือ@ ค่าของแต่ละอิลิเมนต์ (ที่แบ่งด้วยอักขระตัวแบ่งฟิลด์) จะถูกทดแทน
<code>#{Parameter}</code>	หากค่าของพารามิเตอร์ <code>Parameter</code> เป็น * หรือ@ ตัวเลขของพารามิเตอร์ตำแหน่ง จะถูกทดแทน มิฉะนั้น ความยาวที่ระบุโดยพารามิเตอร์ <code>Parameter</code> จะถูกทดแทน
<code>#{Identifier[*]}</code>	จำนวนอิลิเมนต์ในอาร์เรย์ที่ระบุโดยพารามิเตอร์ <code>Identifier</code> จะถูกทดแทน
<code>{Parameter: - Word}</code>	หากมีการตั้งค่าพารามิเตอร์ <code>Parameter</code> และไม่ใช่ null ค่าของพารามิเตอร์จะถูกทดแทน มิฉะนั้น ค่าของ พารามิเตอร์ <code>Word</code> จะถูกทดแทน
<code>{Parameter:=Word}</code>	หากไม่มีการตั้งค่าพารามิเตอร์ <code>Parameter</code> หรือเป็น null พารามิเตอร์จะถูกตั้งค่าเป็นค่าของพารามิเตอร์ <code>Word</code> อย่างไรก็ตาม ไม่สามารถกำหนดพารามิเตอร์ตำแหน่งในวิธีนี้
<code>{Parameter: ? Word}</code>	หากมีการตั้งค่าพารามิเตอร์ <code>Parameter</code> และไม่ใช่ null ระบบจะทดแทนค่าของพารามิเตอร์นี้ มิฉะนั้น พิมพ์ค่าของ ตัวแปร <code>Word</code> และออกจากเซลล์ หากตัวแปร <code>Word</code> ถูกข้ามไป จะมีการพิมพ์ข้อความมาตรฐาน
<code>{Parameter:+Word}</code>	หากมีการตั้งค่าพารามิเตอร์ <code>Parameter</code> และไม่ใช่ null จะทดแทนค่าของตัวแปร <code>Word</code>
<code>{Parameter#Pattern} </code> <code>{Parameter##Pattern}</code>	หากพารามิเตอร์ <code>Pattern</code> เซลล์ที่ระบุ ตรงกับตอนต้นของค่าของพารามิเตอร์ <code>Parameter</code> ค่าของการทดแทนนี้จะเป็นค่าของพารามิเตอร์ <code>Parameter</code> ที่มีส่วนตรงกันถูกลบออกไป มิฉะนั้น ค่าของพารามิเตอร์ <code>Parameter</code> จะถูกทดแทน ในรูปแบบแรก รูปแบบการจับคู่ที่เล็กที่สุดจะถูกลบออก ในรูปแบบที่สอง รูปแบบการจับคู่ที่ใหญ่ที่สุดจะถูกลบออก

ไอเท็ม	คำอธิบาย
<code>\${Parameter%Pattern} \${Parameter%%Pattern}</code>	หาก <i>Pattern</i> เซลล์ที่ระบุตรงกับตอนท้ายของค่าของตัวแปร <i>Parameter</i> ค่าของการทดแทนนี้จะเป็นค่าของตัวแปร <i>Parameter</i> ที่มีส่วนตรงกันถูกลบออกไป มิฉะนั้น ทดแทนค่าของตัวแปร <i>Parameter</i> ในรูปแบบแรก รูปแบบการจับคู่ที่เล็กที่สุดจะถูกลบออก ในรูปแบบที่สอง รูปแบบการจับคู่ที่ใหญ่ที่สุดจะถูกลบออก ในนิพจน์ก่อนหน้า ตัวแปร <i>Word</i> ไม่ถูกประเมินยกเว้นว่าจะใช้ เป็นสตริงทดแทน ดังนั้น ในตัวอย่างต่อไปนี้ คำสั่ง <code>pwd</code> มีการดำเนินการ เฉพาะถ้าไม่ได้ตั้งค่าแฟล็ก <code>-d</code> หรือเป็น <code>null</code> เท่านั้น: <code>echo \${d:-\$(pwd)}</code>

หมายเหตุ: หาก : ถูกข้ามไปจากนิพจน์ก่อนหน้า เซลล์จะตรวจสอบว่ามีการตั้งค่าพารามิเตอร์ *Parameter* หรือไม่เพียงอย่างเดียว

หลักการที่เกี่ยวข้อง:

“เทอร์มินัลที่ไม่ได้ใส่ใจ” ในหน้า 322

ระบบทุกระบบอาจได้รับอันตรายได้ถ้าเทอร์มินัลมีการล็อกอินทิ้งไว้ และไม่ได้ใส่ใจ ปัญหาร้ายแรงที่สุดเกิดขึ้นเมื่อผู้จัดการระบบปล่อยเทอร์มินัลที่ไม่ได้ใส่ใจทิ้งไว้โดยที่เปิดใช้งานด้วยสิทธิการใช้งานรากแล้ว โดยทั่วไป ผู้ใช้ควรล็อกเอาต์ทุกครั้งที่ปล่อยเทอร์มินัลทิ้งไว้

พารามิเตอร์พิเศษที่กำหนดไว้ล่วงหน้าใน Korn เซลล์หรือ POSIX เซลล์:

บางพารามิเตอร์มีการตั้งค่าโดยอัตโนมัติโดย Korn เซลล์หรือ POSIX เซลล์

พารามิเตอร์ต่อไปนี้มีการตั้งค่าโดยอัตโนมัติ โดยเซลล์:

ไอเท็ม	คำอธิบาย
@	ขยายพารามิเตอร์ตำแหน่งโดยเริ่มต้นด้วย \$1 แต่ละพารามิเตอร์มีการค้นด้วยช่องว่าง หากคุณใส่ " ล้อมรอบ @\$ เซลล์จะพิจารณาพารามิเตอร์ตำแหน่งแต่ละตัวเป็นสตริงแยกต่างหาก ถ้าไม่มีพารามิเตอร์ตำแหน่ง เซลล์จะขยายคำสั่งเป็นสตริง null ที่ไม่ได้อยู่ในเครื่องหมายอัฒภาค
*	ขยายพารามิเตอร์ตำแหน่งโดยเริ่มต้นด้วย \$1 เซลล์ค้นแต่ละพารามิเตอร์ด้วยอักขระตัวแรกของค่าพารามิเตอร์ IFS หากคุณใส่ " ล้อมรอบ \$* เซลล์จะใส่ค่าพารามิเตอร์ตำแหน่งไว้ในเครื่องหมายอัฒภาคคู่ แต่ละค่ามีการค้นด้วยอักขระตัวแรกของพารามิเตอร์ IFS
#	ระบุจำนวน (ในหลักสิบ) ของพารามิเตอร์ตำแหน่ง ที่ส่งผ่านไปยังเซลล์ โดยไม่นับชื่อของตัวเซลล์ไพรซีเดอร์เอง ดังนั้นพารามิเตอร์ \$# ให้จำนวนของพารามิเตอร์ตำแหน่งสูงสุด ที่มีการตั้งค่า การใช้งานหลักอย่างหนึ่งของพารามิเตอร์นี้คือ การตรวจสอบการมีอยู่ของจำนวนอาร์กิวเมนต์ที่ต้องการ
-	จัดส่งแฟล็กให้กับเซลล์เมื่อเรียกใช้หรือด้วยคำสั่ง <code>set</code>
?	ระบุค่าออกของคำสั่งล่าสุดที่ดำเนินการ ค่า เป็นสตริงฐานสิบ คำสั่งส่วนใหญ่ส่งคืนค่าเป็น 0 เพื่อบ่งชี้ถึงความสำเร็จสมบูรณ์ ตัวเซลล์เองส่งคืนค่าปัจจุบันของพารามิเตอร์ <code>?</code> เป็น ค่าออก

ไอเท็ม	คำอธิบาย
\$	ระบุหมายเลขโปรเซสของเซลล์นี้ เนื่องจากหมายเลขโปรเซสไม่ซ้ำกันระหว่างโปรเซสทั้งหมดที่มีอยู่จึงมักใช้สตริงนี้สูงสุด 5 ตำแหน่ง เพื่อสร้างชื่อที่ไม่ซ้ำกันของไฟล์ชั่วคราว ตัวอย่างต่อไปนี้ แสดงวิธีปฏิบัติที่แนะนำของการสร้างไฟล์ชั่วคราวในไดเรกทอรีซึ่งใช้สำหรับวัตถุประสงค์นั้นเท่านั้น: temp=\$HOME/temp/\$\$ ls >\$temp . . . rm \$temp
!	ระบุหมายเลขโปรเซสของคำสั่งส่วนหลังล่าสุด ที่เรียกใช้
zero (0)	ขยายชื่อของเซลล์หรือเซลล์สคริปต์

การทดแทนชื่อไฟล์ใน Korn เซลล์หรือ POSIX เซลล์:

Korn เซลล์หรือ POSIX เซลล์ทำการทดแทนชื่อไฟล์ โดยการสแกนค่าของคำสั่งแต่ละคำซึ่งระบุโดยตัวแปร *Word* เพื่อหาอักขระบางตัว

ถ้าค่าของคำสั่งมีอักขระ `*`, `?` หรือ `[` และยังไม่ได้ตั้งค่าแฟล็ก `-f` เซลล์จะถือว่าค่าเป็นรูปแบบ เซลล์แทนที่ค่าด้วยชื่อไฟล์ซึ่งเรียงลำดับตามลำดับการจัดเรียงที่ใช้ในโลแคลปัจจุบัน ซึ่งตรงกับรูปแบบนั้น หาก เซลล์ไม่พบชื่อไฟล์ที่ตรงกับรูปแบบ เซลล์จะไม่เปลี่ยน คำ

เมื่อเซลล์ใช้รูปแบบสำหรับการทดแทนชื่อไฟล์ อักขระ `.` และ `/` ต้องตรงกันอย่างชัดเจน

หมายเหตุ: Korn เซลล์ ไม่ได้จัดการอักขระเหล่านี้โดยเฉพาะในอินสแตนซ์อื่นของการจับคู่รูปแบบ

อักขระการจับคู่รูปแบบเหล่านี้บ่งชี้การทดแทน ต่อไปนี้:

ไอเท็ม	คำอธิบาย
*	จับคู่สตริงใดๆ รวมถึงสตริง null
?	จับคู่อักขระเดี่ยวใดๆ
[...]	จับคู่อักขระอย่างใดอย่างหนึ่งใดๆ ที่อยู่ในเครื่องหมายวงเล็บ คู่ของอักขระที่คั่นด้วย เครื่องหมายยัติภังค์ (-) ตรงกับอักขระใดๆ ตามตัวอักษรภายใน ช่วงเฉพาะของคุณั้น ตามลำดับการจัดเรียงที่ใช้ใน โลแคลปัจจุบัน ถ้าอักขระตัวแรกที่ต่อจาก <code>[</code> เปิดเป็น <code>!</code> อักขระใดๆ ที่ไม่ได้อยู่ในเครื่องหมายวงเล็บจะถูกจับคู่ เครื่องหมายยัติภังค์ (-) สามารถรวมไว้ในชุดอักขระได้โดยใส่ไว้เป็นอักขระตัวแรกหรือตัวสุดท้าย

คุณยังสามารถใช้สัญกรณ์ `[:charclass:]` เพื่อ จับคู่อักขระที่อยู่ในช่วงที่ระบุได้ด้วย รูปแบบนี้สั่งให้ระบบ จับคู่อักขระเดี่ยวใดๆ ที่เป็นสมาชิกของ `class` นิยามของ อักขระที่ประกอบเป็นคลาสอักขระเฉพาะมีอยู่ผ่านทาง หมวดหมู่ `LC_CTYPE` ของรูทีนย่อย `setlocale` ระบบรับรู้ถึง คลาสอักขระทั้งหมดที่ระบุในโลแคลปัจจุบัน

ชื่อของบางคลาสอักขระมีดังนี้:

- **alnum**
- **alpha**
- **cntrl**
- **digit**
- **graph**

- lower
- print
- punct
- space
- upper
- xdigit

ตัวอย่างเช่น `[[:upper:]]` จับคู่ตัวอักษรพิมพ์ใหญ่ใดๆ

Korn เชลล์สนับสนุนการขยายชื่อไฟล์ตามข้อมูล อิลิเมนต์การจัดเรียง สัญลักษณ์ หรือคลาสการสมมูล

PatternList เป็นรายการของรูปแบบตั้งแต่หนึ่งรูปแบบขึ้นไป ซึ่งคั่นระหว่างกันด้วย | รูปแบบผสมประกอบขึ้นจาก ข้อมูลต่อไปนี้ตั้งแต่หนึ่งรายการขึ้นไป:

ไอเท็ม	คำอธิบาย
? (<i>PatternList</i>)	จับคู่รูปแบบอย่างใดอย่างหนึ่งที่กำหนดซึ่งเลือกกำหนดได้
* (<i>PatternList</i>)	จับคู่การเกิดขึ้นศูนย์หรือมากกว่านั้นของรูปแบบที่กำหนด
+ (<i>PatternList</i>)	จับคู่การเกิดขึ้นตั้งแต่หนึ่งครั้งขึ้นไปของรูปแบบที่กำหนด
@ (<i>PatternList</i>)	จับคู่รูปแบบที่กำหนดอย่างใดอย่างหนึ่งเท่านั้น
! (<i>PatternList</i>)	จับคู่ทุกอย่าง ยกเว้นรูปแบบที่กำหนดอย่างใดอย่างหนึ่ง

การจับคู่รูปแบบมีข้อจำกัดบางประการ ถ้าอักขระตัวแรก ของชื่อไฟล์คือจุด (.) แล้วระบบสามารถ จับคู่ตามรูปแบบที่ขึ้นต้นด้วย ตัวอย่างเช่น * ตรงกับชื่อไฟล์ myfile และ yourfile แต่ไม่ตรงกับชื่อไฟล์ .myfile และ .yourfile เมื่อต้องการจับคู่ไฟล์เหล่านี้ ให้ใช้รูปแบบดังต่อไปนี้:

```
.*file
```

หากรูปแบบไม่ตรงกับชื่อไฟล์ใดๆ ตัวรูปแบบเองจะถูก ส่งคืนเป็นผลลัพธ์ของการจับคู่ที่พยายาม

ชื่อไฟล์และไดเรกทอรีไม่ควรมีอักขระ *, ?, [, หรือ] เนื่องจากอักขระเหล่านี้อาจส่งผลให้เกิดซ้ำแบบไม่สิ้นสุด (นั่นคือ ลูปไม่สิ้นสุด) ในระหว่างความพยายามจับคู่รูปแบบ

การลบบัญประกาศ:

อักขระบางตัวจะถูกลบออกถ้าไม่ได้ใส่ไว้ในเครื่องหมายอัญประกาศ

อักขระอัญประกาศ, ทับขวา (\), อัญประกาศเดี่ยว ('), และอัญประกาศคู่ (") ที่มีอยู่ในคำสั่งเดิม จะถูกลบออก ยกเว้นว่าอักขระดังกล่าวถูกใส่ไว้ในเครื่องหมายอัญประกาศ

การเปลี่ยนทิศทางอินพุตและเอาต์พุตใน Korn เชลล์หรือ POSIX เชลล์:

ก่อน Korn เชลล์ดำเนินการคำสั่ง เชลล์จะสแกนบรรทัดคำสั่ง เพื่อหาอักขระการเปลี่ยนทิศทาง สัญกรณ์พิเศษเหล่านี้สั่งให้เชลล์เปลี่ยนทิศทางอินพุตและเอาต์พุต

อักขระการเปลี่ยนทิศทางสามารถปรากฏขึ้นที่ใดก็ได้ในคำสั่งแบบง่าย หรือสามารถนำหน้าหรือตามหลังคำสั่ง อักขระนี้ไม่ถูกส่งผ่านไปยังคำสั่งที่เรียกใช้

เซลล์ทำการทดแทนคำสั่งและพารามิเตอร์ก่อนการใช้พารามิเตอร์ Word หรือ Digit ยกเว้นมีการระบุเป็นอย่างอื่น การทดแทนชื่อไฟล์เกิดขึ้นเฉพาะถ้ารูปแบบตรงกับ ไฟล์เดียวและไม่ได้ทำการตีความพื้นที่ว่างเปล่า

ไอเท็ม	คำอธิบาย
<Word	ใช้ไฟล์ที่ระบุโดยพารามิเตอร์ Word เป็นอินพุตมาตรฐาน (ตัวอธิบายไฟล์ 0)
>Word	ใช้ไฟล์ที่ระบุโดยพารามิเตอร์ Word เป็นเอาต์พุตมาตรฐาน (ตัวอธิบายไฟล์ 1) ถ้าไฟล์ไม่มีอยู่ เซลล์จะ สร้างไฟล์ขึ้น ถ้ามีไฟล์และอ็อปชัน noclobber เป็น on จะส่งผลให้เกิด ข้อผิดพลาด มิฉะนั้น ไฟล์จะถูกตัดใหม่มีความยาวเป็นศูนย์ หมายเหตุ: เมื่อหลายเซลล์ มีการตั้งอ็อปชัน noclobber และเซลล์กลับทิศทางเอาต์พุตไปยัง ไฟล์เดียวกัน อาจทำให้เกิดเงื่อนไขในการแข่งขันซึ่งส่งผลให้มีโปรเซสเซลล์เขียน ไปยังไฟล์มากกว่าหนึ่งโปรเซส เซลล์ไม่ได้ตรวจพบ หรือป้องกันเงื่อนไขในการแข่งขันดังกล่าว
> Word	เหมือนกับคำสั่ง >Word ยกเว้นว่าคำสั่งการเปลี่ยนทิศทางนี้ยกเลิกอ็อปชัน noclobber
>>Word	ใช้ไฟล์ที่ระบุโดยพารามิเตอร์ Word เป็นเอาต์พุตมาตรฐาน หากมีไฟล์อยู่ในปัจจุบัน เซลล์จะผนวกเอาต์พุตเข้ากับไฟล์ (โดยค้นหาอักขระสิ้นสุดไฟล์ก่อน) ถ้าไฟล์ไม่มีอยู่ เซลล์จะ สร้างไฟล์ขึ้น
<>Word	เปิดไฟล์ที่ระบุโดยพารามิเตอร์ Word สำหรับการอ่านและการเขียนเป็นอินพุตมาตรฐาน
<<[-]Word	อ่านแต่ละบรรทัดของเซลล์อินพุตจนกว่าจะพบบรรทัดที่มี เฉพาะค่าของพารามิเตอร์ Word หรืออักขระสิ้นสุดไฟล์ เท่านั้น เซลล์ไม่ได้ทำการทดแทนพารามิเตอร์ การทดแทนคำสั่ง หรือการทดแทนชื่อไฟล์บนไฟล์ที่ระบุ เอกสารที่ได้ซึ่งเรียกว่า เอกสาร here กลายเป็นอินพุตมาตรฐาน หากอักขระใดๆ ของพารามิเตอร์ Word อยู่ในเครื่องหมายประกาศ จะไม่มีการตีความอักขระของเอกสาร

เอกสาร here มีการจัดการเป็นคำเดียวที่เริ่มต้นหลังจาก อักขระบรรทัดใหม่ถัดไป และทำต่อไปจนกว่ามีบรรทัดที่มี เฉพาะตัวค้นโดยไม่มีอักขระว่างเปล่าที่ส่วนท้าย จากนั้น เอกสาร here ถัดไป ถ้ามี จะเริ่มต้นขึ้น รูปแบบเป็นดังนี้:

```
[n]<<word
  เอกสาร here
delimiter
```

หากอักขระใดๆ ใน word อยู่ในเครื่องหมายอัญประกาศ ตัวค้นจะ เกิดขึ้นโดยการลบอัญประกาศบน word บรรทัดเอกสาร here จะไม่ถูกขยาย มิฉะนั้น ตัวค้นจะเป็นตัว word เอง หากไม่มีอักขระใน word อยู่ในเครื่องหมายอัญประกาศ บรรทัดทั้งหมด ของเอกสาร here จะถูกขยายสำหรับการขยายพารามิเตอร์ การทดแทนคำสั่ง และ การขยายทางคณิตศาสตร์

เซลล์ทำการทดแทนพารามิเตอร์สำหรับข้อมูลที่เปลี่ยนทิศทาง เพื่อป้องกันไม่ให้ เซลล์ตีความอักขระ \, \$, และ เครื่องหมายอัญประกาศเดี่ยว (') และอักขระตัวแรกของ พารามิเตอร์ Word ให้ใส่อักขระ \ ไว้ข้างหน้าอักขระ

ถ้ามีการผนวกยัติภังค์ (-) เข้ากับ << เซลล์จะ strips แท็บนำทั้งหมดจากพารามิเตอร์ Word และเอกสาร

ไอเท็ม	คำอธิบาย
<&Digit	ทำซ้ำอินพุตมาตรฐานจากตัวอธิบายไฟล์ที่ระบุโดยพารามิเตอร์ Digit
>&Digit	ทำซ้ำเอาต์พุตมาตรฐานในตัวอธิบายไฟล์ที่ระบุโดยพารามิเตอร์ Digit
<&-	ปิดอินพุตมาตรฐาน
>&-	ปิดเอาต์พุตมาตรฐาน
<&p	ย้ายอินพุตจากโปรเซสรวมไปยังอินพุตมาตรฐาน
>&p	ย้ายเอาต์พุตของโปรเซสรวมไปยังเอาต์พุตมาตรฐาน

หากอ็อปชันการเปลี่ยนทิศทางอย่างใดอย่างหนึ่งเหล่านี้มีตัวเลขนำหน้า หมายเลข ตัวอธิบายไฟล์ที่อ้างอิงจะถูกระบุเป็นตัวเลข (แทนค่าดีฟอลต์ 0 หรือ 1) ในตัวอย่างต่อไปนี้ เซลล์เปิดตัวอธิบายไฟล์ 2 เพื่อเขียน เป็นสำเนาของตัวอธิบายไฟล์ 1:

```
... 2>&1
```

ลำดับในการระบุการเปลี่ยนทิศทางเป็นสิ่งสำคัญ เซลล์ประเมิน การเปลี่ยนทิศทางแต่ละรายการในแง่ของ (FileDescriptor, File) ความเชื่อมโยง ณ เวลาของการประเมิน ตัวอย่างเช่น ในคำสั่ง:

... 1>File 2>&1

ตัวอธิบายไฟล์ 1 มีความเชื่อมโยงกับไฟล์ที่ระบุโดยพารามิเตอร์ File เชลล์เชื่อมโยงตัวอธิบายไฟล์ 2 เข้ากับไฟล์ที่เชื่อมโยงกับตัวอธิบายไฟล์ 1 (File) ถ้าลำดับของการเปลี่ยนทิศทางถูกย้อนกลับ ตัวอธิบายไฟล์ 2 จะถูกเชื่อมโยงกับเทอร์มินัล (สมมติว่า ตัวอธิบายไฟล์ 1 เคยเป็นก่อนหน้านี) และตัวอธิบายไฟล์ 1 จะถูกเชื่อมโยงเข้ากับไฟล์ซึ่งระบุโดยพารามิเตอร์ File

หากคำสั่งมีเครื่องหมาย ampersand (&) ตามหลังและการควบคุมงานไม่ได้ใช้งานอยู่ อินพุตมาตรฐานดีฟอลต์สำหรับคำสั่งจะเป็นไฟล์ว่าง /dev/null มิฉะนั้น สถานะแวดล้อมสำหรับการดำเนินการ ของคำสั่งจะมีตัวอธิบายไฟล์ของเชลล์ที่เรียกใช้ตามที่แก้ไข โดยข้อกำหนดคุณลักษณะอินพุตและเอาต์พุต

หลักการที่เกี่ยวข้อง:

“การเปลี่ยนทิศทางอินพุตและเอาต์พุต” ในหน้า 375

ระบบปฏิบัติการ AIX อนุญาตให้คุณจัดวางอินพุตและเอาต์พุต (I/O) ของข้อมูลเข้าและ ออกจากระบบของคุณได้ โดยใช้คำสั่ง I/O และสัญลักษณ์เฉพาะ

งานที่เกี่ยวข้อง:

“การเปลี่ยนทิศทางเอาต์พุตไปยังเอกสารอินไลน์อินพุต (here)” ในหน้า 378

คุณสามารถเปลี่ยนทิศทางเอาต์พุตไปยังเอกสารอินไลน์อินพุต (here)

สิ่งอำนวยความสะดวกโปรเซสรวม:

Korn เชลล์หรือ POSIX เชลล์อนุญาตให้คุณรันคำสั่งตั้งแต่หนึ่งคำสั่งขึ้นไป เป็นกระบวนการส่วนหลังได้ คำสั่งเหล่านี้ที่รันจากภายในเชลล์สคริปต์ เรียกว่า *โปรเซสรวม*

กำหนดโปรเซสรวมโดยการวางตัวดำเนินการ |& ไว้ข้างหลัง คำสั่ง ทั้งอินพุตและเอาต์พุตมาตรฐานของคำสั่งจะถูกโพรพเข้า กับสคริปต์ของคุณ

โปรเซสรวมต้องเป็นไปตามข้อจำกัดต่อไปนี้:

- มีอีกขระบรรทัดใหม่ที่ตอนท้ายของแต่ละข้อความ
- ส่งข้อความเอาต์พุตแต่ละรายการไปยังเอาต์พุตมาตรฐาน
- ล้างเอาต์พุตมาตรฐานหลังจากแต่ละข้อความ

ตัวอย่างต่อไปนี้สาธิตวิธีการส่งผ่านและส่งคืนอินพุต จากโปรเซสรวม:

```
echo "Initial process"
./FileB.sh |&
read -p a b c d
echo "Read from coprocess: $a $b $c $d"
print -p "Passed to the coprocess"
read -p a b c d
echo "Passed back from coprocess: $a $b $c $d"
```

```
FileB.sh
    echo "The coprocess is running"
    read a b c d
    echo $a $b $c $d
```

เอาต์พุตมาตรฐานที่ได้เป็นดังนี้:

Initial process
Read from coprocess: The coprocess is running
Passed back from coprocess: Passed to the coprocess

ใช้คำสั่ง `print -p` เพื่อเขียนลงในโปรเซสร่วม ใช้คำสั่ง `read -p` เพื่ออ่านจากโปรเซสร่วม

หลักการที่เกี่ยวข้อง:

“คำสั่ง Korn เชลล์หรือ POSIX เชลล์” ในหน้า 275

Korn เชลล์คือตัวแปลคำสั่งแบบโต้ตอบและภาษา การเขียนโปรแกรมคำสั่ง Korn เชลล์สอดคล้องกับ Portable Operating System Interface for Computer Environments (POSIX) ซึ่งเป็นมาตรฐานสากลสำหรับ ระบบปฏิบัติการ

การเปลี่ยนทิศทางอินพุตและเอาต์พุตโปรเซสร่วม:

อินพุตและเอาต์พุตมาตรฐานของโปรเซสร่วมถูกกำหนดใหม่ให้กับ ตัวอธิบายไฟล์ที่ระบุโดยใช้การเปลี่ยนทิศทาง I/O

ตัวอย่างเช่น คำสั่ง:

```
exec 5>&p
```

ย้าย อินพุตของโปรเซสร่วมไปยังตัวอธิบายไฟล์ 5

หลังจากโปรเซสร่วมนี้เสร็จสมบูรณ์แล้ว คุณสามารถใช้ไวยากรณ์การเปลี่ยนทิศทาง มาตรฐานเพื่อเปลี่ยนทิศทางเอาต์พุตคำสั่งไปยังโปรเซสร่วมได้ และยังสามารถ เริ่มต้นโปรเซสร่วมอื่นได้ด้วย เอาต์พุตจากทั้งสองโปรเซสร่วมถูกเชื่อมต่อเข้ากับ ไฟล์เดียวกันและอ่านด้วยคำสั่ง `read -p` เมื่อต้องการหยุด โปรเซสร่วม ให้พิมพ์ดังต่อไปนี้:

```
read -u5
```

คำสั่งในตัว Korn เชลล์หรือ POSIX เชลล์:

คำสั่งพิเศษมีการสร้างขึ้นใน Korn เชลล์และ POSIX เชลล์ และดำเนินการในโปรเซสเชลล์

ยกเว้นว่ามีการระบุเป็นอย่างอื่น เอาต์พุตจะถูกเขียนไปยังตัวอธิบายไฟล์ 1 และสถานะการออกคือ 0 (ศูนย์) ถ้าคำสั่งไม่มีข้อผิดพลาด ไวยากรณ์ใดๆ อนุญาตให้เปลี่ยนทิศทางอินพุตและเอาต์พุตได้ คำสั่งในตัวมีอยู่ สองชนิดคือ *คำสั่งในตัวพิเศษ* และ *คำสั่งในตัวปกติ*

คำสั่งในตัวพิเศษแตกต่างจากคำสั่งในตัวปกติใน ลักษณะต่อไปนี้:

- ข้อผิดพลาดไวยากรณ์ในคำสั่งในตัวพิเศษอาจส่งผลให้เชลล์ที่ดำเนินการ คำสั่งสิ้นสุดลง กรณีนี้ไม่เกิดขึ้นถ้าคุณมีข้อผิดพลาดไวยากรณ์ในคำสั่ง ในตัวปกติ ถ้าข้อผิดพลาดไวยากรณ์ในคำสั่งในตัวพิเศษไม่ได้สิ้นสุด โปรแกรมเชลล์ คำการออกจะไม่ใช่ศูนย์
- การกำหนดตัวแปรที่ระบุด้วยคำสั่งในตัวพิเศษยังคงมีผลบังคับใช้ หลังจากคำสั่งเสร็จสมบูรณ์
- การเปลี่ยนทิศทาง I/O มีการประมวลผลหลังจากการกำหนดพารามิเตอร์

นอกจากนี้ คำที่อยู่ในรูปแบบของการกำหนดพารามิเตอร์ตามหลังคำสั่งพิเศษ `export`, `readonly`, และ `typeset` ถูกขยายด้วยกฎเดียวกันกับการกำหนดพารามิเตอร์ การทดแทน เครื่องหมายการออกเสียงมีการทำหลังจาก = และไม่ทำการแบ่งคำ และการทดแทนชื่อไฟล์

หลักการที่เกี่ยวข้อง:

“คำสั่ง Korn เซลล์หรือ POSIX เซลล์” ในหน้า 275

Korn เซลล์คือตัวแปลคำสั่งแบบโต้ตอบและภาษา การเขียนโปรแกรมคำสั่ง Korn เซลล์สอดคล้องกับ Portable Operating System Interface for Computer Environments (POSIX) ซึ่งเป็นมาตรฐานสากลสำหรับ ระบบปฏิบัติการ

“ฟังก์ชัน Korn เซลล์” ในหน้า 279

คำสั่งสวอนไว้ ฟังก์ชัน กำหนดฟังก์ชันของเซลล์ เซลล์ อ่านและจัดเก็บฟังก์ชันไว้ใน ชื่อสมนามมีการแก้ไขเมื่อ อ่านฟังก์ชัน เซลล์ดำเนินการฟังก์ชันในลักษณะเดียวกันกับคำสั่ง โดยมีอาร์กิวเมนต์ที่ส่งผ่านเป็นพารามิเตอร์ตำแหน่ง

สิ่งอ้างอิงที่เกี่ยวข้อง:

“รายการของคำสั่งในตัวพิเศษ Korn เซลล์หรือ POSIX เซลล์” ในหน้า 241

คำสั่งพิเศษมีการสร้างขึ้นใน Korn เซลล์และ POSIX เซลล์ และดำเนินการในโปรเซสเซลล์

“คำสั่งในตัวปกติ Korn เซลล์หรือ POSIX เซลล์” ในหน้า 242

ข้อมูลต่อไปนี้เป็นรายการของคำสั่งในตัวปกติของ Korn เซลล์หรือ POSIX เซลล์

คำอธิบายคำสั่งในตัวพิเศษสำหรับ Korn เซลล์หรือ POSIX เซลล์:

คำสั่งพิเศษมีการสร้างขึ้นใน Korn เซลล์และ POSIX เซลล์ และดำเนินการในโปรเซสเซลล์

คำสั่งในตัวพิเศษของ Korn เซลล์มีการอธิบาย ข้างล่าง:

:	eval	newgrp	shift
.	exec	readonly	times
break	exit	return	trap
continue	export	set	typeset
			unset

ไอเท็ม	คำอธิบาย
: [Argument ...]	ขยายอาร์กิวเมนต์อย่างเดี่ยว ใช้เมื่อคำสั่งเป็นสิ่งจำเป็น เช่นในเงื่อนไข <i>then</i> ของคำสั่ง <i>if</i> แต่คำสั่งไม่ต้องดำเนินการใดๆ
.File [Argument ...]	อ่านไฟล์สมบุรณ์ที่ระบุ แล้วดำเนินการ คำสั่ง คำสั่งมีการดำเนินการในสภาวะแวดล้อมเซลล์ปัจจุบัน เซลล์ใช้พารามิเตอร์ค้นหาที่ระบุโดยตัวแปล <i>PATH</i> เพื่อค้นหาไดเรกทอรีซึ่งมี ไฟล์ที่ระบุ หากมีการระบุ อาร์กิวเมนต์ใดๆ อาร์กิวเมนต์จะกลายเป็นพารามิเตอร์ตำแหน่ง มิฉะนั้น พารามิเตอร์ตำแหน่งไม่เปลี่ยนแปลง สถานะการออกคือสถานะการออกของ คำสั่งล่าสุดที่ดำเนินการ โปรดดูที่ “การทดแทนพารามิเตอร์ใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 252 สำหรับ ข้อมูลเพิ่มเติมเกี่ยวกับพารามิเตอร์ตำแหน่ง หมายเหตุ: คำสั่ง <i>.File [Argument ...]</i> อ่านไฟล์ทั้งไฟล์ก่อนจะดำเนินการคำสั่งใดๆ ดังนั้น คำสั่ง <i>alias</i> และ <i>unalias</i> ในไฟล์จึงไม่ใช้กับฟังก์ชันใดๆ ที่กำหนดไว้ในไฟล์
break [n]	ออกจากลูป <i>for</i> , <i>while</i> , <i>until</i> , หรือ <i>select</i> ที่ล้อมอยู่ ถ้ามี หากคุณระบุพารามิเตอร์ <i>n</i> คำสั่งจะแบ่งตัวเลขของระดับที่ระบุ โดยพารามิเตอร์ <i>n</i> ค่า <i>n</i> เป็นเลขจำนวนเต็มใดๆ ที่เท่ากับหรือ มากกว่า 1
continue [n]	ทำการวนซ้ำถัดไปของลูป <i>for</i> , <i>while</i> , <i>until</i> , หรือ <i>select</i> ที่ล้อมอยู่ต่อไป หากคุณระบุพารามิเตอร์ <i>n</i> คำสั่งจะดำเนินการต่อที่ลูป <i>n</i> th ที่ล้อมอยู่ ค่า <i>n</i> เป็นเลขจำนวนเต็มใดๆ ที่เท่ากับหรือ มากกว่า 1
eval [Argument ...]	อ่านอาร์กิวเมนต์ที่ระบุเป็นอินพุตของเซลล์ และดำเนินการคำสั่งที่ได้
exec [Argument ...]	ดำเนินการคำสั่งที่ระบุโดยอาร์กิวเมนต์แทน เซลล์นี้ (โดยไม่สร้างโปรเซสใหม่) อินพุตและเอาต์พุตอาร์กิวเมนต์ สามารถปรากฏขึ้นและส่งผลกระทบต่อโปรเซสปัจจุบัน หากคุณไม่ได้ ระบุอาร์กิวเมนต์ คำสั่ง <i>exec</i> จะแก้ไข ตัวอธิบายไฟล์ตั้งที่อธิบายโดยรายการเปลี่ยนทิศทางอินพุตและ เอาต์พุต ในกรณีนี้ หมายเลขตัวอธิบายไฟล์ใดๆ ที่มากกว่า 2 ซึ่งเปิด ด้วยกลไกนี้จะถูกปิดเมื่อเรียกใช้โปรแกรมอื่น

ไอเท็ม	คำอธิบาย
<code>exit [n]</code>	ออกจากเซลล์ด้วยสถานะการออกที่ระบุโดย พารามิเตอร์ <i>n</i> พารามิเตอร์ <i>n</i> ต้องเป็นเลขจำนวนเต็มฐานสิบที่ไม่มีเครื่องหมายในช่วง 0-255 หากคุณข้าม พารามิเตอร์ <i>n</i> สถานะการออกคือสถานะการออกของคำสั่งล่าสุดที่ดำเนินการ อีกระยะสิ้นสุดไฟล์ยังออกจากเซลล์ ด้วย ยกเว้นว่าเปิดอ็อพชัน <code>ignoreeof</code> ของคำสั่งพิเศษ <code>set</code>
<code>export -p [Name= Value] ...</code>	ทำเครื่องหมายชื่อที่ระบุของเอ็กซ์พอร์ตอัตโนมัติในสถานะแวดล้อมของ คำสั่งที่ดำเนินการในเวลาต่อมา <p>-p เขียนชื่อและค่า ของตัวแปรที่เอ็กซ์พอร์ตทั้งหมดไปยังเอาต์พุตมาตรฐาน ในรูปแบบต่อไปนี้: "export %s= %s\n", <name> <value></p>
<code>newgrp [Group]</code>	เทียบเท่ากับคำสั่ง <code>exec/usr/bin/newgrp [Group]</code> หมายเหตุ: คำสั่งนี้ไม่ส่งคืน
<code>readonly -p [Name= Value] ...</code>	ทำเครื่องหมายชื่อที่ระบุโดยพารามิเตอร์ <i>Name</i> เป็นแบบอ่านอย่างเดียว ชื่อเหล่านี้ไม่สามารถเปลี่ยนโดยการกำหนดค่าในลำดับต่อมา <p>-p เขียนชื่อและค่าของตัวแปรที่เอ็กซ์พอร์ตทั้งหมดไปยังเอาต์พุตมาตรฐาน ในรูปแบบต่อไปนี้: "export %s= %s\n", <name> <value></p>
<code>return [n]</code>	ส่งผลให้ฟังก์ชันเซลล์กลับไปยังสคริปต์ที่ เรียกใช้ สถานะการส่งคืนมีการระบุโดยพารามิเตอร์ <i>n</i> หากคุณข้ามพารามิเตอร์ <i>n</i> สถานะการส่งคืนคือสถานะการส่งคืน ของคำสั่งล่าสุดที่ดำเนินการ หากคุณเรียกใช้ คำสั่ง <code>return</code> ภายนอกฟังก์ชันหรือสคริปต์ จะเหมือนกับคำสั่ง <code>exit</code>
<code>set [+ -abCefhkmnostuvx] [+ -o Option]... [+ -AName] [Argument ...]</code>	ถ้าไม่ได้ออ็อพชันหรืออาร์กิวเมนต์ คำสั่ง <code>set</code> จะเขียน ชื่อและค่าของตัวแปรเซลล์ทั้งหมดในลำดับการจัดเรียง ของโลแคลปัจจุบัน เมื่อระบุอ็อพชัน อ็อพชันจะตั้งค่าหรือ ยกเลิกการตั้งค่าแอตทริบิวต์ของเซลล์ ดังอธิบายต่อไปนี้: <p>-A การกำหนดอาร์เรย์ ยกเลิกการตั้งค่าพารามิเตอร์ <i>Name</i> และกำหนดค่าตามลำดับจากรายการพารามิเตอร์ <i>Argument</i> ที่ระบุ ถ้าใช้แฟล็ก +A พารามิเตอร์ <i>Name</i> จะไม่ถูกยกเลิกการตั้งค่าก่อน</p> <p>-a เอ็กซ์พอร์ตพารามิเตอร์ในลำดับต่อมาทั้งหมดซึ่งกำหนดไว้โดยอัตโนมัติ</p> <p>-b แจ้งผู้ใช้แบบอะซิงโครนัสเกี่ยวกับความสมบูรณ์ของงานพื้นหลัง</p> <p>-C เทียบเท่ากับ <code>set -o noclobber</code></p> <p>-e ดำเนินการดักจับ ERR ถ้าตั้งค่า และออกถ้าคำสั่งมีสถานะการออกที่ไม่ใช่ ศูนย์ยกเว้นว่าคำสั่งแบบง่ายเป็น: + ที่มีอยู่ใน <code>&&</code> หรือรายการ <code> </code> + คำสั่งที่ตามหลังต่อจาก <code>if</code>, <code>while</code> หรือ <code>until</code> + ที่มีอยู่ในไฟฟ์ไลน์หลังจาก !</p> <p>โหมดนี้ปิดใช้งานในขณะที่อ่านโปรไฟล์</p> <p>-f ปิดใช้การทดแทนชื่อไฟล์</p> <p>-h กำหนดแต่ละคำสั่งเป็นสมนามที่ติดตามเมื่อพบเป็นครั้งแรก</p> <p>-k วางอาร์กิวเมนต์การกำหนดพารามิเตอร์ทั้งหมดไว้ในสถานะแวดล้อมสำหรับคำสั่ง ไม่ใช่เฉพาะอาร์กิวเมนต์ที่นำหน้าชื่อคำสั่ง</p> <p>-m ใช้งานพื้นหลังในโปรเซสแยกต่างหากและพิมพ์บรรทัดในทันทีที่เสร็จสมบูรณ์ สถานะการออกของงานพื้นหลังมีการรายงานในข้อความ ความสมบูรณ์ บนระบบที่มีการควบคุมงาน แฟล็กนี้เปิดโดยอัตโนมัติ สำหรับเซลล์แบบโต้ตอบ สำหรับข้อมูลเพิ่มเติม โปรดดูที่ “การควบคุมงานใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 268</p> <p>-n อ่านคำสั่งและตรวจสอบหาข้อผิดพลาดไวยากรณ์ แต่ไม่ได้ ดำเนินการคำสั่ง แฟล็กนี้ถูกละเว้นสำหรับเซลล์แบบโต้ตอบ</p>

-o Option

พิมพ์ค่าที่ตั้งอ็อปชันปัจจุบันและข้อความแสดงข้อผิดพลาดถ้าคุณไม่ได้ระบุอาร์กิวเมนต์ คุณสามารถตั้งค่ามากกว่าหนึ่งอ็อปชันบนบรรทัดคำสั่ง ksh ได้ด้วย ถ้าใช้แฟล็ก +o อ็อปชันที่ระบุจะถูกยกเลิกการตั้งค่าเมื่อระบุอาร์กิวเมนต์ อาร์กิวเมนต์จะส่งผลให้ตั้งค่าหรือยกเลิกการตั้งค่า พารามิเตอร์ตำแหน่ง อาร์กิวเมนต์ตามที่ระบุ โดยตัวแปร *Option* สามารถทำอย่างใดอย่างหนึ่งต่อไปนี้:

allexport เหมือนกับแฟล็ก -a

bgnice รันงานพื้นหลังทั้งหมดในลำดับความสำคัญต่ำกว่า นี่เป็นโหมด ดีพอลด์

emacs ป้อนโปรแกรมแก้ไขแบบแทรกลักษณะ emacs สำหรับรายการคำสั่ง

erexit เหมือนกับแฟล็ก -e

gmacs ป้อนโปรแกรมแก้ไขแบบแทรกลักษณะ gmacs สำหรับรายการคำสั่ง

ignoreeof

ไม่ออกจากเซลล์เมื่อพบอักขระสิ้นสุดไฟล์ เมื่อต้องการออกจากเซลล์ คุณต้องใช้คำสั่ง exit หรือ กดลำดับคีย์ Ctrl-D มากกว่า 11 ครั้ง

keyword เหมือนกับแฟล็ก -k

หมายเหตุ: แฟล็กนี้ใช้เพื่อให้เข้ากันได้ย้อนหลังกับเซลล์เป้าหมายเท่านั้น ไม่ควรใช้แฟล็กนี้ถ้าไม่จำเป็น

markdirs

ผนวกเครื่องหมายทับขวา / เข้ากับชื่อไดเรกทอรีทั้งหมด ซึ่งเป็นผลลัพธ์ของการทดแทนชื่อไฟล์

monitor เหมือนกับแฟล็ก -m

noclobber

ป้องกันไม่ให้เกิดการเปลี่ยนทิศทางตัดไฟล์ที่มีอยู่ เมื่อคุณระบุ อ็อปชันนี้ แถบแนวตั้งต้องเป็นสัญลักษณ์การเปลี่ยนทิศทาง (>|) เพื่อตัดไฟล์

noexec เหมือนกับแฟล็ก -n

noglob เหมือนกับแฟล็ก -f

nolog ป้องกันไม่ให้บันทึกนิยามฟังก์ชันในไฟล์ .profile และ \$ENV ไว้ในไฟล์ประวัติ

nounset เหมือนกับแฟล็ก -u

privileged

เหมือนกับแฟล็ก -p

trackall	เหมือนกับแฟล็ก -h
verbose	เหมือนกับแฟล็ก -v
vi	ป้อนโหมดแทรกของโปรแกรมแก้ไขแบบแทรก vi-style สำหรับ รายการคำสั่ง การป้อนอักขระ escape 033 จะวางโปรแกรมแก้ไขไวโน โหมดย้าย การส่งคืนส่งบรรทัด
viraw	ประมวลผลอักขระแต่ละตัวเมื่อมีการพิมพ์ในโหมด vi
xtrace	เหมือนกับแฟล็ก -x
-p	ปิดใช้งานการประมวลผลของไฟล์ \$HOME/.profile และใช้ไฟล์ /etc/suid_profile แทน ไฟล์ ENV โหมดนี้มีการเปิดใช้งานในทุกเมื่อที่ ID ผู้ใช้ (UID) หรือ ID กลุ่ม (GID) ที่ใช้อยู่ไม่เหมือนกับ UID หรือ GID จริง การปิดอ็อพชันนี้จะตั้งค่า UID หรือ GID ที่ใช้อยู่เป็น UID และ GID จริง หมายเหตุ: ระบบไม่สนับสนุนอ็อพชัน -p เนื่องจากระบบปฏิบัติการไม่สนับสนุนเซลล์สคริปต์ setuid
-s	เรียงลำดับพารามิเตอร์ตำแหน่งตามลำดับอักษร
-t	ออกหลังการอ่านและเรียกใช้งานหนึ่งคำสั่ง หมายเหตุ: แฟล็กนี้ใช้เพื่อให้เข้ากันได้ย้อนหลังกับเซลล์เป้าหมายเท่านั้น ไม่ควรใช้แฟล็กนี้ถ้าไม่จำเป็น
-u	จัดการพารามิเตอร์ที่ยกเลิกการตั้งค่าเป็นข้อผิดพลาดเมื่อทดแทน
-v	พิมพ์บรรทัดเซลล์อินพุตขณะที่ถูกอ่าน
-x	พิมพ์คำสั่งและอาร์กิวเมนต์ของคำสั่งขณะที่ถูกดำเนินการ
-	ปิดแฟล็ก -x และ -v และหยุดการตรวจสอบ อาร์กิวเมนต์สำหรับแฟล็ก
-	ป้องกันไม่ให้เปลี่ยนแฟล็กใดๆ อ็อพชันนี้มีประโยชน์ในการตั้งค่าพารามิเตอร์ \$1 เป็นค่าที่ขึ้นต้นด้วย - หากไม่มีอาร์กิวเมนต์ตามหลังแฟล็กนี้ จะไม่มีการตั้งค่า พารามิเตอร์ตำแหน่ง

การนำหน้าแฟล็กคำสั่ง **set** ใดๆ ด้วย **+** แทน **-** จะปิดแฟล็ก คุณสามารถใช้แฟล็กเหล่านี้เมื่อคุณเรียกใช้เซลล์ เมื่อเรียกใช้ **'set +o'** โดยไม่มีอาร์กิวเมนต์ใดๆ คำสั่งจะแสดง ค่าที่ตั้งอ็อพชันปัจจุบันในรูปแบบที่เหมาะสมสำหรับการอินพุตอีกครั้ง ลงในเซลล์เป็นคำสั่งที่ให้ค่าที่ตั้งอ็อพชันเดียวกัน การตั้งค่าปัจจุบัน ของแฟล็กมีอยู่ในพารามิเตอร์ **\$-** ยกเว้นว่าคุณระบุแฟล็ก **-A** อาร์กิวเมนต์ที่เหลือเป็น พารามิเตอร์ตำแหน่งและมีการกำหนดตามลำดับ **\$1, \$2, ...**, และต่อไป หากไม่ได้กำหนดอาร์กิวเมนต์ จะมีการพิมพ์ชื่อและค่าของพารามิเตอร์ ที่ระบุทั้งหมดไปยังเอาต์พุตมาตรฐาน

shift [n]	เปลี่ยนชื่อพารามิเตอร์ตำแหน่งโดยเริ่มต้นด้วย \$n+1 ... ถึง \$1 ... ค่าดีฟอลต์ของพารามิเตอร์ m คือ 1 พารามิเตอร์ m เป็นนิพจน์ทางคณิตศาสตร์ใดๆ ที่ประเมินเป็นตัวเลขที่ไม่ใช่ค่าลบซึ่งน้อยกว่าหรือเท่ากับ พารามิเตอร์ \$#
times	พิมพ์เวลาผู้ใช้และระบบสะสมสำหรับเซลล์และ สำหรับโปรเซสที่รันจากเซลล์

ไอเท็ม**คำอธิบาย****trap** [*Command*] [*Signal*] ...

รับคำสั่งที่ระบุเมื่อเซลล์ได้รับสัญญาณ ที่ระบุ พารามิเตอร์ *Command* ถูกอ่าน หนึ่งครั้งเมื่อตั้งค่าการดักจับ และอีกครั้งเมื่อใช้การดักจับ พารามิเตอร์ *Signal* สามารถมีการกำหนดเป็นตัวเลขหรือ เป็นชื่อของสัญญาณ คำสั่ง **trap** มีการดำเนินการในลำดับของ หมายเลขสัญญาณ ความพยายามใดๆ ในการตั้งค่าการดักจับบนสัญญาณที่ถูกละเว้น บนรายการของ เซลล์ปัจจุบันจะไม่มีผล

หากคำสั่งเป็น - การดักจับทั้งหมดจะถูกรีเซ็ตเป็นค่าดั้งเดิม หากคุณข้ามคำสั่งไป และสัญญาณแรกเป็นหมายเลขสัญญาณตัวเลข คำสั่ง **ksh** จะรีเซ็ตค่าของพารามิเตอร์ *Signal* เป็นค่าดั้งเดิม

หมายเหตุ: หากคุณข้ามคำสั่งไปและสัญญาณแรก เป็นชื่อสัญลักษณ์ สัญญาณจะถูกตีความเป็นคำสั่ง

ถ้า คำของพารามิเตอร์ *Signal* เป็นสัญญาณ **ERR** จะมีการดำเนินการคำสั่งที่ระบุในทุกเมื่อที่คำสั่งมีสถานะการออก ที่ไม่ใช่ศูนย์ ถ้าสัญญาณเป็น **DEBUG** จะมีการดำเนินการคำสั่งที่ระบุ หลังจากแต่ละคำสั่ง ถ้าคำของพารามิเตอร์ *Signal* เป็น **0** หรือสัญญาณ **EXIT** และมีการดำเนินการคำสั่ง **trap** ภายในเนื้อหาของฟังก์ชัน จะมีการดำเนินการคำสั่งที่ระบุ หลังจากฟังก์ชันเสร็จสมบูรณ์แล้ว ถ้าพารามิเตอร์ *Signal* เป็น **0** หรือ **EXIT** สำหรับคำสั่ง **trap** ที่ตั้งค่า ภายนอกฟังก์ชันใดๆ จะมีการดำเนินการคำสั่งที่ระบุเมื่อออกจากเซลล์ คำสั่ง **trap** ที่ไม่มีอาร์กิวเมนต์ จะพิมพ์รายการของคำสั่งที่เชื่อมโยงกับแต่ละหมายเลขสัญญาณ หากคำสั่ง ที่ระบุเป็น **NULL** ซึ่งบ่งชี้เป็น "" (อัญประกาศว่าง) คำสั่ง **ksh** จะละเว้นสัญญาณ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีการที่ Korn เซลล์ หรือ POSIX เซลล์อ่านอักขระเป็นอักขระปกติ ให้ดูที่ "การใส่อักขระไว้ใน เครื่องหมายอัญประกาศใน Korn เซลล์หรือ POSIX เซลล์" ในหน้า 243

For a complete list of *Signal* parameter values used in the **trap** command without the **SIG** prefix, see the **sigaction**, **sigvec**, or **signal** subroutine in the *Technical Reference: Base Operating System and Extensions, Volume 2*.

ไอเท็ม

คำอธิบาย

typeset [+HLRZflirtux [n]]
[Name[=Value]] ...

ตั้งค่าแฉัตริบิวต์และค่าสำหรับพารามิเตอร์เซลล์ เมื่อเรียกใช้ภายในฟังก์ชัน จะมีการสร้างอินสแตนซ์ใหม่ของพารามิเตอร์ *Name* ค่าพารามิเตอร์และชนิดมีการเรียกคืนเมื่อฟังก์ชัน เสร็จสมบูรณ์ คุณสามารถระบุบุลเล็ตไปนี้พร้อมกับคำสั่ง **typeset**:

- H นำเสนอการแม็พ AIX กับไฟล์โฮสต์ บนเครื่องที่ไม่ใช่ AIX
- L ปรับขีดซ้ายและลบพื้นที่ว่างนำออกจากพารามิเตอร์ *Value* ถ้าพารามิเตอร์ *n* มีค่าที่ไม่ใช่ศูนย์ พารามิเตอร์นี้จะกำหนดความกว้างของฟิลด์ ถ้าเป็นศูนย์ จะถูกกำหนดโดย ความกว้างของค่าของการกำหนดค่าครั้งแรก เมื่อกำหนดพารามิเตอร์ จะมีการวางพารามิเตอร์ทางด้านขวาพร้อมกับพื้นที่ว่างหรือตัดให้สั้นลง ถ้าจำเป็น เพื่อให้พอดีกับฟิลด์ เลขศูนย์นำถูกลบออกถ้ามีการกำหนดแฟล็ก -Z ด้วย แฟล็ก -R ถูกปิด
- R ปรับขีดขวาและกรอกข้อมูลพร้อมกับพื้นที่ว่างนำ ถ้าพารามิเตอร์ *n* มีค่าที่ไม่ใช่ศูนย์ พารามิเตอร์นี้จะกำหนดความกว้างของฟิลด์ ถ้าเป็นศูนย์ จะถูกกำหนดโดย ความกว้างของค่าของการกำหนดค่าครั้งแรก ฟิลด์ยังคงมีการกรอกข้อมูลพร้อมกับพื้นที่ว่างหรือตัดให้สั้นลงจากตอนท้าย ถ้ากำหนดพารามิเตอร์อีกครั้ง แฟล็ก L ถูกปิด
- Z ปรับขีดขวาและกรอกข้อมูลพร้อมกับเลขศูนย์นำ ถ้าอักขระที่ไม่ว่างเปล่าตัวแรก เป็นตัวเลขและยังไม่ได้ตั้งค่าแฟล็ก -L ถ้าพารามิเตอร์ *n* มีค่าที่ไม่ใช่ศูนย์ พารามิเตอร์นี้จะกำหนดความกว้างของฟิลด์ ถ้าเป็นศูนย์ จะถูกกำหนดโดย ความกว้างของค่าของการกำหนดค่าครั้งแรก
- f บ่งชี้ว่าชื่อแสดงถึงชื่อฟังก์ชัน ไม่ใช่ชื่อ พารามิเตอร์ไม่สามารถกำหนดค่าและแฟล็กอื่นที่ถูกต้องคือ -t, -u, และ -x เท่านั้น แฟล็ก -t เปิดใช้การติดตามการดำเนินการ สำหรับฟังก์ชันนี้ แฟล็ก -u ส่งผลให้มีการทำเครื่องหมายฟังก์ชันนี้ เป็น ไม่ได้กำหนด มีการค้นหาตัวแปร *FPATH* เพื่อหานิยามฟังก์ชันเมื่อมีการอ้างอิงถึงฟังก์ชัน แฟล็ก -x อนุญาตให้นิยามฟังก์ชันยังคงมีผลบังคับใช้ บนเซลล์สคริปต์ต่างๆ ที่ไม่ใช่การเรียกใช้แยกต่างหากของคำสั่ง *ksh*
- i ระบุพารามิเตอร์เป็นเลขจำนวนเต็ม ซึ่งทำให้การคำนวณเร็วขึ้น ถ้าพารามิเตอร์ *n* มีค่าที่ไม่ใช่ศูนย์ พารามิเตอร์นี้จะกำหนดฐานทางคณิตศาสตร์ของเอาต์พุต ถ้าเป็นศูนย์ การกำหนดค่าครั้งแรกจะกำหนดฐานของเอาต์พุต
- l แปลงอักขระตัวพิมพ์ใหญ่ทั้งหมดเป็นอักขระตัวพิมพ์เล็ก แฟล็กการแปลงตัวพิมพ์ใหญ่ -u ถูกปิด
- r ทำเครื่องหมายชื่อที่ระบุโดยพารามิเตอร์ *Name* เป็นแบบอ่านอย่างเดียว ชื่อเหล่านี้ไม่สามารถเปลี่ยนโดยการกำหนดค่าในลำดับต่อมา
- t ติดแท็กพารามิเตอร์ที่ระบุ แท็กสามารถกำหนดได้โดยผู้ใช้ และ ไม่มีความหมายพิเศษสำหรับเซลล์
- u แปลงอักขระตัวพิมพ์เล็กทั้งหมดเป็นอักขระตัวพิมพ์ใหญ่ แฟล็กตัวพิมพ์เล็ก -l ถูกปิด
- x ทำเครื่องหมายชื่อที่ระบุโดยพารามิเตอร์ *Name* สำหรับเอ็กซ์พอร์ตอัตโนมัติไปยังสภาวะแวดล้อมของคำสั่งที่ดำเนินการในเวลาต่อมา

การใช้ + แทน - จะปิด แฟล็กคำสั่ง **typeset** หากคุณไม่ได้ระบุพารามิเตอร์ *Name* แต่ระบุบุลเล็ต จะมีการพิมพ์รายชื่อ (และค่าที่เลือกกำหนดได้) ของพารามิเตอร์ที่มีชุดตัวบ่งชี้เหล่านี้ (การใช้ + แทน - ป้องกันการพิมพ์ค่า) หาก คุณไม่ได้ระบุชื่อหรือแฟล็กใดๆ จะมีการพิมพ์ชื่อและแฉัตริบิวต์ของ พารามิเตอร์ทั้งหมด

unset [-fv] Name ...

ยกเลิกการตั้งค่าและแฉัตริบิวต์ของพารามิเตอร์ที่กำหนดโดยรายชื่อ ถ้ามีการระบุ -v *Name* จะหมายถึง ชื่อตัวแปรและเซลล์จะยกเลิกการตั้งค่าชื่อและลบชื่อนั้นออกจาก สภาวะแวดล้อม ตัวแปรแบบอ่านอย่างเดียวไม่สามารถยกเลิกการตั้งค่าได้ การยกเลิกการตั้งค่าตัวแปร *ERRNO*, *LINENO*, *MAILCHECK*, *OPTARG*, *OPTIND*, *RANDOM*, *SECONDS*, *TMOU*T, และขีดเส้นใต้ () จะลบความหมายพิเศษของตัวแปรนั้นแม้จะมีการกำหนดในเวลาต่อมา ก็ตาม ถ้าตั้งค่าแฟล็ก -f *Name* จะหมายถึง ชื่อฟังก์ชัน และเซลล์จะยกเลิกการตั้งค่านิยามฟังก์ชัน

คำอธิบายคำสั่งในตัวปกติสำหรับ Korn เซลล์หรือ POSIX เซลล์:

คำสั่งในตัวสำหรับ Korn หรือ POSIX เซลล์มีการอธิบาย ที่นี้

Korn เซลล์นำเสนอคำสั่งในตัวปกติต่อไปนี้:

alias	fg	print	ulimit	
bg	getopts	pwd	umask	
cd	jobs	read	unalias	
command	kill	setgroups	wait	
echo	let	setenv	test	whence
fc				

ไอเท็ม

คำอธิบาย

alias [-t] [-x] [AliasName=[String]] ...
สร้างหรือกำหนดนิยามสมนามอีกครั้ง หรือเขียน นิยามสมนามที่มีอยู่ไปยังเอาต์พุตมาตรฐาน สำหรับข้อมูลเพิ่มเติม ให้อูที่คำสั่ง **alias**

bg [JobID...]
วางแต่ละงานที่ระบุไว้ในพื้นหลัง งานปัจจุบัน ถูกวางไว้ในพื้นหลังถ้าไม่ได้ระบุพารามิเตอร์ *JobID* โปรดดูที่ “การควบคุมงานใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 268 สำหรับ ข้อมูลเพิ่มเติมเกี่ยวกับการควบคุมงาน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการรันงาน ในพื้นหลัง ให้อูที่คำสั่ง **bg**

cd [Argument]
cd *Old New*

คำสั่งนี้สามารถอยู่ในสองรูปแบบ ในรูปแบบแรก คำสั่งเปลี่ยนไดเรกทอรีปัจจุบันเป็นไดเรกทอรีที่ระบุโดยพารามิเตอร์ *Argument* หากค่าของพารามิเตอร์ *Argument* เป็นยัติภังค์ (-) ไดเรกทอรีจะถูกเปลี่ยนเป็นไดเรกทอรีก่อนหน้านี้ ตัวแปรเซลล์ **HOME** เป็นค่าดีฟอลต์ของ พารามิเตอร์ *Argument* ตัวแปร **PWD** มีการตั้งค่าเป็นไดเรกทอรีปัจจุบัน

ตัวแปรเซลล์ **CDPATH** กำหนดพารการค้นหาสำหรับไดเรกทอรีที่มีค่าของ พารามิเตอร์ *Argument* ชื่อไดเรกทอรีอื่น มีการแบ่งด้วยเครื่องหมายจุดคู่ (:) พารดีฟอลต์เป็น null ซึ่งระบุ ไดเรกทอรีปัจจุบัน ไดเรกทอรีปัจจุบันมีการระบุโดยชื่อพาร null ซึ่งปรากฏขึ้นทันทีหลังจากเครื่องหมายเท่ากับ หรือระหว่างตัวค้นจุดคู่ในใดๆ ในรายการพาร ถ้าอาร์กิวเมนต์ที่ระบุขึ้นต้นด้วยเครื่องหมายทับ (/) เซลล์จะไม่ใช้พารการค้นหา มิฉะนั้น จะมีการค้นหาแต่ละไดเรกทอรีในพาร เพื่อหาอาร์กิวเมนต์

รูปแบบที่สองของคำสั่ง **cd** ใช้สตริงที่ระบุโดยตัวแปร *New* ทดแทนสตริงที่ระบุโดยตัวแปร *Old* ในชื่อไดเรกทอรีปัจจุบัน **PWD** และพยายามเปลี่ยน เป็นไดเรกทอรีใหม่นี้

command [-p]
CommandName [Argument ...]

command [-v|-V]
CommandName
ทำให้เซลล์จัดการกับคำสั่งและอาร์กิวเมนต์ที่ระบุ เป็นคำสั่งแบบง่าย โดยยับยั้งการค้นหาเซลล์-ฟังก์ชัน สำหรับข้อมูลเพิ่มเติม ให้อูที่คำสั่ง **command**

echo [String ...]
เขียนสตริงอักขระไปที่เอาต์พุตมาตรฐาน โปรดดูที่คำสั่ง **echo** สำหรับการใช้งานและคำอธิบาย ไม่สนับสนุนแฟล็ก **-n**

fc [-r] [-eEditor] [First [Last]]
fc -l [-n] [-r] [First [Last]]
]

fc -s [Old=New] [First]
แสดงเนื้อหาของไฟล์ประวัติคำสั่งของคุณ หรือเรียกใช้ โปรแกรมแก้ไขเพื่อแก้ไขและดำเนินการคำสั่งที่ป้อนก่อนหน้านี้อีกครั้งในเซลล์
สำหรับข้อมูลเพิ่มเติม ให้อูที่คำสั่ง **fc**

ไอเท็ม	คำอธิบาย
<code>fg [JobID]</code>	นำแต่ละงานที่ระบุไปไว้ในพื้นหน้า ถ้าคุณไม่ได้ระบุงานใดๆ คำสั่งจะนำงานปัจจุบันไปไว้ในพื้นหน้า สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการรันงาน ในพื้นหน้า ให้ดูที่คำสั่ง <code>fg</code>
<code>getopts OptionString Name [Argument ...]</code>	ตรวจสอบพารามิเตอร์ <i>Argument</i> สำหรับอ็อปชันที่ถูกต้อง สำหรับข้อมูลเพิ่มเติม ให้ดูที่คำสั่ง <code>getopts</code>
<code>jobs [-l -n -p] [JobID ...]</code>	แสดงสถานะของงานที่เริ่มต้นในสภาวะแวดล้อมเซลล์ ปัจจุบัน หากไม่มีการระบุงานเฉพาะด้วยพารามิเตอร์ <i>JobID</i> ข้อมูลสถานะสำหรับงานที่ใช้งานอยู่ทั้งหมดจะแสดงขึ้น หากมีการรายงานการยุติงาน เซลล์จะลบ ID โปรเซสของงานนั้นออกจากรายการของงานที่ สภาวะแวดล้อมเซลล์ปัจจุบันรู้จัก สำหรับข้อมูลเพิ่มเติม ให้ดูที่คำสั่ง <code>jobs</code>
<code>kill [-s { SignalName SignalNumber } ProcessID...]</code> <code>kill [-SignalName -SignalNumber] ProcessID...</code> <code>kill -l [ExitStatus]</code>	ส่งสัญญาณ (โดยค่าดีฟอลต์คือสัญญาณ SIGTERM) ไปยังโปรเซสที่กำลังรัน โดยปกติ การดำเนินการดีฟอลต์นี้หยุดโปรเซส ถ้าคุณ ต้องการหยุดโปรเซส ให้ระบุ ID โปรเซส (PID) ในตัวแปร <i>ProcessID</i> เซลล์รายงาน PID ของแต่ละโปรเซสที่กำลังรันในพื้นที่หลัง (ยกเว้นว่าคุณเริ่มต้นมากกว่าหนึ่งโปรเซสในไฟฟ์ไลน์ ซึ่งในกรณีนี้ เซลล์ จะรายงานหมายเลขของโปรเซสล่าสุด) คุณยังสามารถใช้คำสั่ง <code>ps</code> เพื่อค้นหาหมายเลข ID โปรเซสของคำสั่งได้ด้วย แสดงรายชื่อสัญญาณ สำหรับข้อมูลเพิ่มเติม ให้ดูที่คำสั่ง <code>kill</code>
<code>let Expression ...</code>	ประเมินนิพจน์ทางคณิตศาสตร์ที่ระบุ สถานะการออก เป็น 0 ถ้าค่าของนิพจน์ล่าสุดไม่ใช่ศูนย์ และเป็น 1 ถ้าค่าของนิพจน์ล่าสุดเป็นศูนย์ โปรดดู “การประเมินทางคณิตศาสตร์ใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 239 สำหรับข้อมูลเพิ่มเติม
<code>print [-Rnrpsu [n] [Argument ...]</code>	พิมพ์เซลล์เอาต์พุต หากคุณไม่ได้ระบุแฟล็กใดๆ หรือ ถ้าคุณระบุแฟล็กยัติภังค์ (-) หรือยัติภังค์คู่ (--) อาร์กิวเมนต์จะถูกพิมพ์ไปยังเอาต์พุตมาตรฐานตั้งที่อธิบายโดยคำสั่ง <code>echo</code> แฟล็กทำดังต่อไปนี้: <ul style="list-style-type: none"> -R พิมพ์ในโหมดดิบ (ระเบียน escape ของคำสั่ง <code>echo</code> ถูกละเว้น) แฟล็ก -R พิมพ์อาร์กิวเมนต์และแฟล็กที่ตามมาทั้งหมด ยกเว้น -n -n ป้องกันไม่ให้เพิ่มอักขระบรรทัดใหม่ลงในเอาต์พุต -p เขียนอาร์กิวเมนต์ลงในไฟฟ์ของโปรเซสที่รันด้วย & แทน ของเอาต์พุตมาตรฐาน -r พิมพ์ในโหมดดิบ ระเบียน escape ของคำสั่ง <code>echo</code> ถูกละเว้น -s เขียนอาร์กิวเมนต์ลงในไฟฟ์ประวัติแทนลงในเอาต์พุตมาตรฐาน -u ระบุหมายเลขชนิดตัวอธิบายไฟล์หนึ่งตำแหน่ง <i>n</i> ซึ่งจะวางเอาต์พุต ดีฟอลต์คือ 1
<code>pwd</code>	เทียบเท่ากับ <code>print -r - \$PWD</code> หมายเหตุ: คำสั่งภายใน Korn เซลล์ <code>pwd</code> ไม่สนับสนุนลิงก์ สัญลักษณ์
<code>read [-prs [n]] [Name?] Prompt [Name...]</code>	ใช้เซลล์อินพุต มีการอ่านหนึ่งบรรทัดและแบ่งออกเป็นฟิลด์ต่างๆ โดยใช้อักขระในตัวแปร IFS เป็นตัวแบ่ง สำหรับข้อมูลเพิ่มเติม ให้ดูที่คำสั่ง <code>read</code>
<code>setgroups</code>	ดำเนินการคำสั่ง <code>/usr/bin/setgroups</code> ซึ่งรัน เป็นเซลล์แยกต่างหาก โปรดดูที่คำสั่ง <code>setgroups</code> สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งนี้ อย่างไรก็ตาม มีความแตกต่างหนึ่งข้อคือ คำสั่งในตัว <code>setgroups</code> เรียกใช้เซลล์ย่อย แต่คำสั่ง <code>setgroups</code> แทนที่ เซลล์ที่กำลังดำเนินการในปัจจุบัน เนื่องจากคำสั่งในตัวมีการสนับสนุนสำหรับความเข้ากันได้ เท่านั้น จึงแนะนำให้สคริปต์ใช้ชื่อพาสลับ <code>/usr/bin/setgroups</code> แทนคำสั่งในตัวเซลล์
<code>setseiv</code>	ดำเนินการคำสั่ง <code>/usr/bin/setseiv</code> ซึ่งแทนที่เซลล์ที่กำลังดำเนินการในปัจจุบัน โปรดดูที่คำสั่ง <code>setseiv</code> สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งนี้
<code>test</code>	เหมือนกับ <code>[expression]</code> โปรดดูที่ “นิพจน์เงื่อนไขสำหรับ Korn เซลล์หรือ POSIX เซลล์” ในหน้า 242 สำหรับการใช้งานและคำอธิบาย

ไอเท็ม	คำอธิบาย
ulimit [-HSacdfmst] [Limit]	<p>ตั้งค่าหรือแสดงขีดจำกัดรีซอร์สของโปรเซสผู้ใช้ที่กำหนดไว้ในไฟล์ /etc/security/limits ไฟล์นี้มีขีดจำกัดดีฟอลต์ต่อไปนี้:</p> <pre> fsize = 2097151 core = 2048 cpu = 3600 data = 131072 rss = 65536 stack = 8192 threads = -1 </pre> <p>ค่าเหล่านี้ใช้เป็นค่าติดตั้งดีฟอลต์ เมื่อเพิ่มผู้ใช้งานในระบบ ค่ามีการตั้งค่าด้วยคำสั่ง mkuser เมื่อเพิ่มผู้ใช้งานในระบบ หรือเปลี่ยนด้วยคำสั่ง chuser</p> <p>ขีดจำกัดมีการจัดหมวดหมู่เป็นแบบซอฟต์แวร์หรือฮาร์ด อย่างใดอย่างหนึ่ง ผู้ใช้อาจเปลี่ยนขีดจำกัดซอฟต์แวร์ของตนได้ไม่เกินค่าสูงสุดที่กำหนดโดยขีดจำกัดฮาร์ด โดยใช้คำสั่ง ulimit คุณต้องมีสิทธิ์ผู้ใช้รากเพื่อเปลี่ยนขีดจำกัดฮาร์ดของรีซอร์ส</p> <p>ระบบจำนวนมาก ไม่มีขีดจำกัดเหล่านี้ ขีดจำกัดสำหรับรีซอร์สที่ระบบมีการตั้งค่าเมื่อระบบบูต <i>Limit</i> ค่าของพารามิเตอร์ <i>Limit</i> สามารถเป็นตัวเลข ในหน่วยที่ระบุพร้อมกับแต่ละรีซอร์ส หรือเป็นค่า unlimited คุณสามารถระบุแฟล็กคำสั่ง ulimit ต่อไปนี้:</p> <ul style="list-style-type: none"> -H ระบุว่ามีการตั้งค่าขีดจำกัดฮาร์ดสำหรับรีซอร์สที่กำหนด หากคุณมี สิทธิ์ผู้ใช้ราก คุณสามารถเพิ่มขีดจำกัดฮาร์ดได้ ผู้ใช้ทุกรายสามารถลดขีดจำกัดนี้ได้ -S ระบุว่ามีการตั้งค่าขีดจำกัดซอฟต์แวร์สำหรับรีซอร์สที่กำหนด ขีดจำกัดซอฟต์แวร์ สามารถเพิ่มได้สูงสุดถึงค่าของขีดจำกัดฮาร์ด หากไม่มีการระบุทั้งอ็อปชัน -H และ -S ขีดจำกัดจะใช้กับทั้งสองอย่าง -a แสดงรายการขีดจำกัดรีซอร์สปัจจุบันทั้งหมด -c ระบุจำนวนของบล็อกขนาด 512-ไบต์ในขนาดของดัมพ์หลัก -d ระบุขนาดในหน่วย KB ของพื้นที่ข้อมูล -f ระบุจำนวนของบล็อกขนาด 512-ไบต์สำหรับไฟล์ที่เขียนโดยโปรเซสชายน (สามารถอ่านไฟล์ได้ทุกขนาด) -m ระบุจำนวน KB สำหรับขนาดของหน่วยความจำเสมือน -n ระบุขีดจำกัดจำนวนของตัวอธิบายไฟล์ซึ่งโปรเซสอาจเปิดไว้ -r ระบุขีดจำกัดจำนวนของเรดต่อโปรเซส -s ระบุจำนวน KB สำหรับขนาดของพื้นที่สแต็ก -t ระบุจำนวนวินาทีที่จะใช้โดยแต่ละโปรเซส <p>มีการพิมพ์ขีดจำกัดรีซอร์สปัจจุบันเมื่อคุณข้าม ตัวแปร <i>Limit</i> มีการพิมพ์ขีดจำกัดซอฟต์แวร์ยกเว้นว่า คุณระบุแฟล็ก -H เมื่อคุณระบุรีซอร์สมากกว่าหนึ่งรายการ จะมีการพิมพ์ ชื่อขีดจำกัดและยูนิตก่อนหน้าค่า หากไม่ได้กำหนดอ็อปชัน จะสมมติเป็นแฟล็ก -f เมื่อคุณเปลี่ยนค่า จะตั้งค่าทั้งขีดจำกัดฮาร์ดและซอฟต์แวร์เป็น <i>Limit</i> ยกเว้นว่า คุณระบุ -H หรือ -S</p> <p>สำหรับข้อมูลเพิ่มเติมเกี่ยวกับขีดจำกัดรีซอร์ส ผู้ใช้และระบบ ให้อูที่ getrlimit, setrlimit, หรือ vlimit subroutine</p>
umask [-S] [Mask]	<p>พิจารณาสิทธิในไฟล์ ค่านี้ พร้อมกับสิทธิของการสร้างกระบวนการ จะใช้พิจารณาสิทธิของไฟล์เมื่อไฟล์ ถูกสร้างขึ้น ค่าดีฟอลต์คือ 022 ถ้าไม่ได้อะไรพารามิเตอร์ <i>Mask</i> คำสั่ง umask แสดงมาส์กการสร้างโหมดไฟล์ ของสภาวะแวดล้อมเชลล์ปัจจุบันไปยังเอาต์พุตมาตรฐาน</p> <p>สำหรับ ข้อมูลเพิ่มเติมเกี่ยวกับสิทธิอนุญาตในไฟล์ ให้อูที่คำสั่ง umask</p>

ไอเท็ม	คำอธิบาย
<code>unalias { -a AliasName... }</code>	ลบนิยามสำหรับแต่ละชื่อสมนามที่ระบุ หรือลบ นิยามสมนามทั้งหมดถ้าใช้แฟล็ก <code>-a</code> นิยามสมนาม ถูกลบออกจากสภาวะแวดล้อมเซลล์ปัจจุบัน สำหรับข้อมูลเพิ่มเติม ให้ดูที่คำสั่ง <code>unalias</code>
<code>wait [ProcessID...]</code>	รอกงานที่ระบุและยุติ หากคุณไม่ได้ ระบุงาน คำสั่งจะรอโปรเซสชาชนที่ใช้งานอยู่ในปัจจุบันทั้งหมด สถานะการออกจากคำสั่งนี้คือสถานะการออกของโปรเซสซึ่งคำสั่งรอ สำหรับข้อมูลเพิ่มเติม ให้ดูที่คำสั่ง <code>wait</code>
<code>whence [-pv] Name ...</code>	บ่งชี้วิธีการตีความสำหรับแต่ละชื่อที่ระบุ ถ้าใช้เป็นชื่อคำสั่ง เมื่อใช้โดยไม่มีแฟล็ก <code>whence</code> จะ แสดงชื่อพาสสัมบูรณ์ ถ้ามีที่สอดคล้องกับแต่ละชื่อ <code>-p</code> ทำการค้นหาพาสเพื่อหาชื่อที่ระบุแม้ว่าเป็น สมนาม ฟังก์ชัน หรือคำที่สงวนไว้ <code>-v</code> จัดทำรายงานซึ่งรายละเอียดมากขึ้นที่ระบุชนิดของแต่ละชื่อ

การควบคุมงานใน Korn เซลล์หรือ POSIX เซลล์:

Korn เซลล์หรือ POSIX เซลล์นำเสนอสิ่งอำนวยความสะดวกในการควบคุม ลำดับคำสั่ง หรือ งาน

เมื่อคุณดำเนินการคำสั่งพิเศษ `set -m` Korn เซลล์เชื่อมโยงงานเข้ากับแต่ละไฟฟ์ไลน์ เซลล์เก็บตารางของงานปัจจุบันที่พิมพ์โดยคำสั่ง `jobs` และกำหนดเลขจำนวนเต็มค่าต่ำให้กับตาราง

เมื่องานเริ่มต้นในพื้นที่ด้วย ampersand (&) เซลล์จะพิมพ์บรรทัดที่คล้ายตัวอย่างต่อไปนี้:

```
[1] 1234
```

เอาต์พุตนี้บ่งชี้ว่างานซึ่งเริ่มต้นในพื้นที่ มีหมายเลขงานเป็น 1 และยังแสดงว่างานมีหนึ่งโปรเซส (ระดับบนสุด) ที่มี ID โปรเซสเป็น 1234

หากคุณกำลังรันงานและต้องการทำบางอย่างให้ใช้ลำดับคีย์ `Ctrl-Z` ลำดับคีย์นี้ส่งสัญญาณ `STOP` ไปยัง งานปัจจุบัน โดยปกติ เซลล์บ่งชี้ว่างานถูกหยุดแล้ว จากนั้นแสดงเซลล์พร้อมต์ จากนั้น คุณสามารถจัดการกับสถานะของงานนี้ (โดยการวางงานไว้ในพื้นที่หลังโดยใช้คำสั่ง `bg`) รันคำสั่งอื่น ในท้ายที่สุด ส่งงานกลับไปยังพื้นที่หน้าโดยใช้ คำสั่ง `fg` ลำดับคีย์ `Ctrl-Z` มีผลบังคับใช้ในทันที และคล้ายกับการขัดจังหวะในลักษณะที่เซลล์ทิ้งเอาต์พุตที่ค้างอยู่และอินพุตที่ไม่ได้ อ่านเมื่อคุณพิมพ์ลำดับ

งานที่กำลังรันในพื้นที่หลังจะหยุดถ้างานนั้นพยายาม อ่านจากเทอร์มินัล โดยปกติแล้วงานพื้นที่หลังได้รับอนุญาตให้จัดทำ เอาต์พุตได้ คุณสามารถปิดใช้งานอ็อพชันนี้โดยออกใช้คำสั่ง `stty tostop` หากคุณตั้งค่า อ็อพชันเทอร์มินัลนี้ งานพื้นที่หลังจะหยุดเมื่องานนั้นพยายามจัดทำ เอาต์พุตหรืออ่านอินพุต

คุณสามารถอ้างอิงงานใน Korn เซลล์ได้หลาย วิธีงานมีการอ้างอิงโดย ID โปรเซสของโปรเซส หรือใน วิธีอย่างใดอย่างหนึ่งต่อไปนี้:

ไอเท็ม	คำอธิบาย
%Number	ระบุงานที่มีหมายเลขที่กำหนด
%String	ระบุงานใดๆ ที่มีบรรทัดคำสั่งเริ่มต้นด้วยตัวแปร String
%?String	ระบุงานใดๆ ที่มีบรรทัดคำสั่งซึ่งมีตัวแปร String
%%	ระบุงานปัจจุบัน
%+	เทียบกับ %%
%-	ระบุงานก่อนหน้านี้

เซลล์นี้รับรู้การเปลี่ยนแปลงในสถานะโปรเซสในทันทีโดยปกติ เซลล์แจ้งให้คุณทราบในทุกเมื่อที่งานถูกบล็อกและไม่สามารถ คืบหน้าต่อไปได้ เซลล์ทำเช่นนั้นก่อนหน้าจะพิมพ์พร้อมท์เพียงเล็กน้อยเท่านั้น เพื่อไม่ให้รบกวนงานของคุณ

เมื่อโหมดมอนิเตอร์เปิดอยู่ งานพื้นหลังที่เสร็จสมบูรณ์แต่ละงาน จะทริกเกอร์การดักจับที่ตั้งค่าสำหรับสัญญาณ CHLD

หากคุณพยายามออกจากเซลล์ (โดยพิมพ์ exit หรือ ใช้ลัดคีย์ Ctrl-D) ในขณะที่งานถูกหยุดหรือกำลังรัน ระบบจะเตือนคุณด้วยข้อความ There are stopped (running) jobs. ใช้คำสั่ง jobs เพื่อดูว่างานใดได้รับผลกระทบ หากคุณ พยายามออกอีกครั้งในทันที เซลล์จะยุติงานที่หยุดและที่กำลังรัน โดยไม่มีค่าเตือน

การจัดการสัญญาณ:

สัญญาณ SIGINT และ SIGQUIT ของคำสั่งที่เรียกใช้ถูกละเว้นถ้าคำสั่งมี ampersand (&) ตามหลังและอ็อพชัน มอนิเตอร์ งานไม่ได้ใช้งานอยู่ มิฉะนั้น สัญญาณมีค่าที่เซลล์ได้รับมาจากพาเรนต์

เมื่อได้รับสัญญาณซึ่งตั้งการดักจับไว้ ในขณะที่เซลล์กำลังรอให้คำสั่งพื้นหลังเสร็จสมบูรณ์ การดักจับ ที่เชื่อมโยงกับสัญญาณนั้นจะไม่ถูกดำเนินการจนกว่าคำสั่งพื้นหลัง เสร็จสมบูรณ์แล้ว ดังนั้น จึงไม่ทำการดักจับบนสัญญาณ CHLD จนกว่างานพื้นหลังจะยุติลง

การแก้ไขแบบแทรกใน Korn เซลล์หรือ POSIX เซลล์:

โดยปกติ คุณพิมพ์บรรทัดคำสั่งแต่ละบรรทัดจากอูปรแกรมเทอร์มินัลและ ตามด้วยอักขระบรรทัดใหม่ (RETURN หรือ LINE FEED) เมื่อคุณเรียกใช้อ็อพชันการแก้ไขแบบแทรก emacs, gmacs, หรือ vi คุณสามารถแก้ไข บรรทัดคำสั่งได้

คำสั่งต่อไปนี้ป้อนโหมดแก้ไข:

ไอเท็ม	คำอธิบาย
set -o emacs	ป้อนโหมดการแก้ไข emacs และเริ่มต้นโปรแกรมแก้ไขแบบแทรกลักษณะ emacs
set -o gmacs	ป้อนโหมดการแก้ไข gmacs และเริ่มต้นโปรแกรมแก้ไขแบบแทรกลักษณะ gmacs
set -o vi	ป้อนโหมดการแก้ไข vi และเริ่มต้นโปรแกรมแก้ไขแบบแทรกลักษณะ vi

อ็อพชันการแก้ไขมีการเลือกโดยอัตโนมัติในแต่ละครั้งที่ตัวแปร VISUAL หรือ EDITOR ได้รับการกำหนดค่าซึ่งลงท้ายด้วยชื่ออ็อพชันใดๆ เหล่านี้

หมายเหตุ: เพื่อใช้คุณลักษณะการแก้ไข เทอร์มินัลของคุณต้องยอมรับ RETURN เป็นแป้นปัดแคร์โดยไม่ป้อนบรรทัด ช่องว่างต้องเขียนทับอักขระปัจจุบัน บนหน้าจอ

แต่ละโหมดการแก้ไขเปิดหน้าต่างที่บรรทัดปัจจุบัน ความกว้างของหน้าต่าง คือค่าของตัวแปร COLUMNS ถ้ามีการกำหนดไว้ มิฉะนั้น ความกว้างคือ 80 ช่องว่างอักขระ หากบรรทัดยาวกว่า ความกว้างหน้าต่างลบด้วยสอง ระบบจะแจ้งให้คุณทราบโดยการแสดงเครื่องหมาย ที่ตอนท้ายของหน้าต่าง เมื่อเคอร์เซอร์เลื่อนไปถึงขอบหน้าต่าง หน้าต่างจะถูกปรับตำแหน่งโดยมีศูนย์

กลางเป็นตำแหน่งของเคอร์เซอร์ เครื่องหมายที่แสดงขึ้นมีดังนี้:

ไอเท็ม	คำอธิบาย
>	บ่งชี้ว่าบรรทัดขยายไปทางด้านขวาของหน้าต่าง
<	บ่งชี้ว่าบรรทัดขยายไปทางด้านซ้ายของหน้าต่าง
*	บ่งชี้ว่าบรรทัดขยายไปทั้งด้านซ้ายและด้านขวาของหน้าต่าง

คำสั่ง search ในแต่ละโหมดแก้ไขนำเสนอการเข้าถึงไฟล์ประวัติของ Korn shell มีการจับคู่สตริงเท่านั้น หากอักขระนำในสตริงเป็น carat (^) การจับคู่ต้องเริ่มต้นที่อักขระตัวแรกในบรรทัด

หลักการที่เกี่ยวข้อง:

“คำสั่ง Korn shell หรือ POSIX shell” ในหน้า 275

Korn shell คือตัวแปลคำสั่งแบบโต้ตอบและภาษา การเขียนโปรแกรมคำสั่ง Korn shell สอดคล้องกับ Portable Operating System Interface for Computer Environments (POSIX) ซึ่งเป็นมาตรฐานสากลสำหรับ ระบบปฏิบัติการ

โหมดการแก้ไข emacs:

มีการป้อนโหมดการแก้ไข emacs เมื่อคุณเปิดใช้งานอ็อปชัน emacs หรือ gmacs ความแตกต่างเพียงอย่างเดียวระหว่างสองโหมดเหล่านี้คือ วิธีการจัดการกับคำสั่ง แก้ไข Ctrl-T

เมื่อต้องการแก้ไข ให้ย้ายเคอร์เซอร์เพื่อชี้การแก้ไขที่ต้องการและแทรก หรือลบอักขระหรือคำ ตามต้องการ คำสั่งการแก้ไขทั้งหมดเป็น อักขระควบคุมหรือลำดับ escape

คำสั่งแก้ไขดำเนินงานจากที่ใดๆ บนบรรทัด (ไม่เฉพาะที่ ตอนต้นเท่านั้น) อย่ากดปุ่ม Enter หรือคีย์ป้อนบรรทัด (ลูกศรลง) หลังจาก คำสั่งแก้ไข ยกเว้นตามที่ระบุ

ไอเท็ม	คำอธิบาย
Ctrl-F	ย้ายเคอร์เซอร์ไปข้างหน้า (ขวา) หนึ่งอักขระ
Esc-F	ย้ายเคอร์เซอร์ไปข้างหน้าหนึ่งคำ (สตริงของอักขระ ที่มีเฉพาะตัวอักษร ตัวเลข และขีดเส้นใต้)
Ctrl-B	ย้ายเคอร์เซอร์ย้อนกลับ (ซ้าย) หนึ่งอักขระ
Esc-B	ย้ายเคอร์เซอร์ย้อนกลับหนึ่งคำ
Ctrl-A	ย้ายเคอร์เซอร์ไปที่ตอนต้นของบรรทัด
Ctrl-E	ย้ายเคอร์เซอร์ไปที่ตอนท้ายของบรรทัด
Ctrl-]c	ย้ายเคอร์เซอร์ไปข้างหน้าบนบรรทัดปัจจุบันไปยังอักขระ ที่บ่งชี้
Esc-Ctrl-]c	ย้ายเคอร์เซอร์ย้อนกลับบนบรรทัดปัจจุบันไปยังอักขระ ที่บ่งชี้
Ctrl-X Ctrl-X	แลกเปลี่ยนเคอร์เซอร์และเครื่องหมาย
ERASE	ลบอักขระก่อนหน้า (อักขระลบที่ผู้ใช้กำหนด ตามที่กำหนดโดยคำสั่ง stty โดยปกติคือลำดับคีย์ Ctrl-H)
Ctrl-D	ลบอักขระปัจจุบัน
Esc-D	ลบคำปัจจุบัน
Esc-Backspace	ลบคำก่อนหน้า
Esc-H	ลบคำก่อนหน้า
Esc-Delete	ลบคำก่อนหน้า หากอักขระขีดจagged ของคุณเป็น ปุ่ม Delete คำสั่งนี้จะไม่ทำงาน
Ctrl-T	สลับที่อักขระปัจจุบันกับอักขระถัดไป ในโหมด emacs สลับที่อักขระก่อนหน้าทั้งสองตัวในโหมด gmacs

ไอเท็ม	คำอธิบาย
Ctrl-C	ทำให้อักขระปัจจุบันเป็นตัวพิมพ์ใหญ่
Esc-C	ทำให้คำปัจจุบันเป็นตัวพิมพ์ใหญ่
Esc-L	เปลี่ยนคำปัจจุบันเป็นตัวพิมพ์เล็ก
Ctrl-K	ลบจากเคอร์เซอร์ไปที่ตอนท้ายของบรรทัด หากมีพารามิเตอร์ตัวเลข นำหน้าที่มีค่าน้อยกว่าตำแหน่งเคอร์เซอร์ปัจจุบัน คำสั่งการแก้ไขนี้จะลบจากตำแหน่งที่กำหนดขึ้นไปยังเคอร์เซอร์ หาก มีพารามิเตอร์ตัวเลขนำหน้าที่มีค่ามากกว่าตำแหน่งเคอร์เซอร์ ปัจจุบัน คำสั่งการแก้ไขนี้จะลบจากเคอร์เซอร์ขึ้นไปยังตำแหน่งเคอร์เซอร์ ที่กำหนด
Ctrl-W	ลบจากเคอร์เซอร์ไปที่เครื่องหมาย
Esc-P	กดพื้นที่จากเคอร์เซอร์ไปยังเครื่องหมายบนสแต็ค
KILL	อักขระทำลายที่ผู้ใช้กำหนดตามที่กำหนดโดยคำสั่ง stty โดยปกติคือลำดับคีย์ Ctrl-G หรือ @ ทำลายบรรทัดปัจจุบัน ทั้งบรรทัด หากป้อนอักขระทำลายสองตัวต่อกัน อักขระทำลายในลำดับต่อมาทั้งหมด ส่งผลให้เกิดการป้อนบรรทัด (มีประโยชน์เมื่อใช้เทอร์มินัลกระดาษ)
Ctrl-Y	เรียกคืนไอเท็มล่าสุดที่ลบออกจากบรรทัด (นำไอเท็ม กลับไปยังบรรทัด)
Ctrl-L	ป้อนบรรทัดและพิมพ์บรรทัดปัจจุบัน
Ctrl-@	(อักขระ null) ตั้งค่าเครื่องหมาย
Esc-space	ตั้งค่าเครื่องหมาย
Ctrl-J	(บรรทัดใหม่) ดำเนินการบรรทัดปัจจุบัน
Ctrl-M	(ส่งคืน) ดำเนินการบรรทัดปัจจุบัน
EOF	ประมวลผลอักขระสิ้นสุดไฟล์ โดยปกติคือลำดับคีย์ Ctrl-D เป็นสิ้นสุดไฟล์เฉพาะถ้าบรรทัดปัจจุบันเป็น null เท่านั้น
Ctrl-P	นำคำสั่งก่อนหน้านี้ไปใช้ ในแต่ละครั้งที่ป้อนลำดับคีย์ Ctrl-P จะมีการเข้าถึงคำสั่งก่อนหน้านี้ ย้ายกลับไปหนึ่งบรรทัด เมื่อไม่ใช้บรรทัดแรกของคำสั่งหลายบรรทัด
Esc-<	นำบรรทัดประวัติก่อนสุด (เก่าสุด) ไปใช้
Esc->	นำบรรทัดประวัติล่าสุด (ใหม่สุด) ไปใช้
Ctrl-N	นำบรรทัดคำสั่งถัดไปไปใช้ ในแต่ละครั้งที่ป้อนลำดับคีย์ Ctrl-N จะมีการเข้าถึงบรรทัดคำสั่งถัดไป
Ctrl-R String	ย้อนกลับประวัติการค้นหาเพื่อหาบรรทัดคำสั่งก่อนหน้านี้ที่มี สตริงซึ่งระบุโดยพารามิเตอร์ String หากกำหนดค่า เป็น 0 การค้นหาค่าจะดำเนินไปข้างหน้า สตริงที่ระบุถูกยุติโดย Enter หรืออักขระบรรทัดใหม่ ถ้าสตริงมีเครื่องหมาย carat (^) นำหน้า บรรทัดที่ตรงกันต้องขึ้นต้นด้วยพารามิเตอร์ String ถ้าพารามิเตอร์ String ถูกข้ามไป จะมีการเข้าถึงบรรทัดคำสั่งถัดไป ที่มีพารามิเตอร์ String ล่าสุด ในกรณีนี้ ค่า 0 จะย้อนกลับทิศทางของการค้นหา
Ctrl-O	(ดำเนินการ) ดำเนินการบรรทัดปัจจุบันและนำบรรทัดถัดไปที่ สัมพันธ์กับบรรทัดปัจจุบันจากไฟล์ประวัติไปใช้
Esc Digits	(Escape) กำหนดพารามิเตอร์ตัวเลข ตัวเลขถูกใช้ เป็นพารามิเตอร์ของคำสั่งถัดไป คำสั่งที่ยอมรับพารามิเตอร์คือ Ctrl-F, Ctrl-B, ERASE, Ctrl-C, Ctrl-D, Ctrl-K, Ctrl-R, Ctrl-P, Ctrl-N, Ctrl-I, Esc-., Esc-Ctrl-I, Esc-_, Esc-B, Esc-C, Esc-D, Esc-F, Esc-H, Esc-L, และ Esc-Ctrl-H
Esc Letter	(Soft-key) ค้นหารายการสมนามสำหรับสมนามที่ชื่อ Letter. หากมีการกำหนดสมนามของชื่อนี้ ค่าจะถูกวางเข้าในคิวอินพุต พารามิเตอร์ Letter ต้องไม่ระบุฟังก์ชัน escape อย่างไม่อย่างหนึ่ง
Esc-[Letter	(Soft-key) ค้นหารายการสมนามสำหรับสมนามที่ชื่อ double underscore Letter (__Letter) หากมีการกำหนดสมนามของชื่อนี้ ค่าจะถูกวางเข้าในคิวอินพุต คำสั่งนี้สามารถใช้เพื่อเขียนโปรแกรม ฟังก์ชันคีย์บนเทอร์มินัลจำนวนมาก
Esc-.	แทรกคำสุดท้ายของคำสั่งก่อนหน้านี้บนบรรทัด ถ้ามีพารามิเตอร์ตัวเลขนำหน้า คำของพารามิเตอร์นี้กำหนด คำที่จะแทรกแทนคำสุดท้าย
Esc- _	เหมือนกับลำดับคีย์ Esc-
Esc-*	พยายามการทดแทนชื่อไฟล์บนคำปัจจุบัน มีการผนวก เครื่องหมายดอกจัน (*) ถ้าคำไม่ตรงกับไฟล์ใดๆ หรือมีอักขระรูปแบบพิเศษใดๆ

ไอเท็ม	คำอธิบาย
Esc-Esc	ทำชื่อไฟล์ให้เสร็จสมบูรณ์ แทนที่คำปัจจุบันด้วยคำเดิมนำทั่วไปที่ยาวที่สุด ของชื่อไฟล์ทั้งหมดซึ่งตรงกับคำปัจจุบันพร้อมกับผนวก เครื่องหมายดอกจัน หากรายการตรงกันไม่ซ้ำกัน จะผนวกเครื่องหมายทับ (/) ถ้าไฟล์เป็นไดเรกทอรี และจะผนวกช่องว่างถ้าไฟล์ไม่ใช่ ไดเรกทอรี
Esc==	แสดงรายการไฟล์ที่ตรงกับรูปแบบคำปัจจุบันราวกับว่า ผนวกเครื่องหมายดอกจัน (*)
Ctrl-U	คุณพารามิเตอร์ของคำสั่งถัดไปด้วย 4
\	Escapes อักขระถัดไป สามารถป้อนอักขระการแก้ไขและอักขระ ERASE, KILL และ INTERRUPT (โดยปกติคือ ปุ่ม Delete) ในบรรทัดคำสั่งหรือในสตริงการค้นหา ถ้านำหน้าด้วยเครื่องหมายทับขาว (\) เครื่องหมายทับขาวลบ คุณลักษณะการแก้ไขของอักขระถัดไป ถ้ามี
Ctrl-V	แสดงเวอร์ชันของเซลล์
Esc-#	แทรกสัญลักษณ์เงินปอนด์ (#) ที่ตอนต้น ของบรรทัด จากนั้นดำเนินการบรรทัด ซึ่งส่งผลให้ข้อคิดเห็นถูกแทรก ลงในไฟล์ประวัติ

โหมดการแก้ไข vi:

โหมดการแก้ไข vi มีโหมดการพิมพ์สองโหมด

โหมดคือ:

- โหมดอินพุต เมื่อคุณป้อนคำสั่ง โปรแกรมแก้ไข vi อยู่ในโหมด อินพุต
- โหมดควบคุม กดปุ่ม Esc เพื่อเข้าสู่โหมดควบคุม

คำสั่งควบคุมส่วนใหญ่ยอมรับพารามิเตอร์ Count ทำซ้ำที่เป็นทางเลือก ก่อนหน้าคำสั่ง เมื่ออยู่ในโหมด vi บนระบบส่วนใหญ่ การประมวลผลที่เป็นที่ยอมรับ มีการเปิดใช้งานโดยแรกเริ่ม คำสั่งถูกสะท้อนอีกครั้งถ้าเงื่อนไขหนึ่งหรือหลายข้อต่อไปนี้เป็นจริง:

- ความเร็วคือ 1200 baud หรือมากกว่า
- คำสั่งมีอักขระควบคุมใดๆ
- ผ่านไปน้อยกว่าหนึ่งวินาทีนับตั้งแต่พิมพ์พร้อมต์

อักขระ Esc ยุติการประมวลผลที่เป็นที่ยอมรับสำหรับส่วนที่เหลือของ คำสั่ง จากนั้น คุณสามารถแก้ไขบรรทัดคำสั่งได้แบบ แขนงนี้มีข้อดีของการประมวลผลที่เป็นที่ยอมรับซึ่งมีการสะท้อนแบบไปข้างหน้าของโหมดดิบ หากมีการตั้งค่าอ็อพชัน viraw ด้วยการประมวลผลที่เป็นที่ยอมรับจะถูกปิดใช้งานเสมอ โหมดนี้เป็นโหมดโดยปริยายสำหรับระบบที่ไม่สนับสนุนตัวค้นการ ลื่นสุดบรรทัดอื่น สองตัว และอาจมีประโยชน์สำหรับบางเทอร์มินัล

คำสั่ง vi edit ที่มีอยู่มีการแบ่งกลุ่มเป็นหมวดหมู่ หมวดหมู่ มีดังนี้:

คำสั่ง input edit:

คำสั่ง input edit สำหรับ Korn เซลล์มีการอธิบายข้างล่าง

หมายเหตุ: โดยค่าดีฟอลต์ โปรแกรมแก้ไขอยู่ในโหมดอินพุต

ไอเท็ม	คำอธิบาย
ERASE	ลบอักขระก่อนหน้า (อักขระลบที่ผู้ใช้กำหนด ตามที่กำหนดโดยคำสั่ง <code>stty</code> โดยปกติคือ <code>Ctrl-H</code> หรือ <code>#</code>)
Ctrl-W	ลบคำที่คั่นด้วยพื้นที่ว่างก่อนหน้า
Ctrl-D	ยุติเชลล์
Ctrl-V	Escapes อักขระถัดไป สามารถป้อนอักขระการแก้ไข เช่น อักขระ ERASE หรือ KILL ในบรรทัดคำสั่ง หรือในสตริงการค้นหาลำหน้าด้วยลำดับคีย์ Ctrl-V ลำดับคีย์ Ctrl-V ลบคุณลักษณะการแก้ไขของอักขระถัดไป (ถ้ามี)
\	Escapes อักขระ ERASE หรือ KILL ถัดไป

คำสั่ง *motion edit*:

คำสั่ง *motion edit* สำหรับ Korn เชลล์มีการอธิบายข้างล่าง

คำสั่ง *motion edit* ย้ายเคอร์เซอร์ดังนี้:

ไอเท็ม	คำอธิบาย
[Count]l	ย้ายเคอร์เซอร์ไปข้างหน้า (ขวา) หนึ่งอักขระ
[Count]w	ย้ายเคอร์เซอร์ไปข้างหน้าหนึ่งคำที่เป็นตัวอักษรและตัวเลข
[Count]W	ย้ายเคอร์เซอร์ไปที่ตอนต้นของคำถัดไปซึ่ง ตามหลังพื้นที่ว่าง
[Count]e	ย้ายเคอร์เซอร์ไปที่ตอนท้ายของคำปัจจุบัน
[Count]E	ย้ายเคอร์เซอร์ไปที่ตอนท้ายของคำปัจจุบันซึ่งแบ่งด้วย พื้นที่ว่าง
[Count]h	ย้ายเคอร์เซอร์ย้อนกลับ (ซ้าย) หนึ่งอักขระ
[Count]b	ย้ายเคอร์เซอร์ย้อนกลับหนึ่งคำ
[Count]B	ย้ายเคอร์เซอร์ไปที่คำก่อนหน้าซึ่งแบ่งด้วยพื้นที่ว่าง
[Count]l	ย้ายเคอร์เซอร์ไปยังคอลัมน์ที่ระบุโดยพารามิเตอร์ <i>Count</i>
[Count]fc	ค้นหาอักขระ <i>c</i> ถัดไปในบรรทัด ปัจจุบัน
[Count]Fc	ค้นหาอักขระ <i>c</i> ก่อนหน้าในบรรทัด ปัจจุบัน
[Count]tc	เทียบเท่ากับ <i>f</i> ตามด้วย <i>h</i>
[Count]Tc	เทียบเท่ากับ <i>F</i> ตามด้วย <i>l</i>
[Count];	ทำซ้ำตามจำนวนครั้งที่ระบุโดยพารามิเตอร์ <i>Count</i> ในคำสั่งค้นหาอักขระเดี่ยวล่าสุด: <i>f</i> , <i>F</i> , <i>t</i> , หรือ <i>T</i>
[Count],	ย้อนกลับคำสั่งค้นหาอักขระเดี่ยวล่าสุดตามจำนวนครั้งที่ระบุโดยพารามิเตอร์ <i>Count</i>
o	ย้ายเคอร์เซอร์ไปที่ตอนต้นของบรรทัด
^	ย้ายเคอร์เซอร์ไปที่อักขระซึ่งไม่ว่างเปล่าตัวแรกในบรรทัด
\$	ย้ายเคอร์เซอร์ไปที่ตอนท้ายของบรรทัด

คำสั่ง *search edit*:

คำสั่ง *search edit* เข้าถึงประวัติคำสั่งของคุณดังนี้:

ไอเท็ม	คำอธิบาย
[Count]k	นำคำสั่งก่อนหน้าไปใช้
[Count]-	เทียบเท่ากับคำสั่ง k
[Count]j	นำคำสั่งถัดไปไปใช้ ในแต่ละครั้งที่ป้อนคำสั่ง j จะมีการเข้าถึงคำสั่งถัดไป
[Count]+	เทียบเท่ากับคำสั่ง j
[Count]G	นำคำสั่งที่มีหมายเลขที่ระบุโดยพารามิเตอร์ Count ไปใช้ ค่าดีฟอลต์คือคำสั่งประวัติล่าสุด
/String	ค้นหาประวัติในแบบย้อนกลับเพื่อหาคำสั่งก่อนหน้า ซึ่งมีสตริงที่ระบุ สตริงถูกยัดโดย RETURN หรือ อักขระบรรทัดใหม่ ถ้าสตริงที่ระบุมีเครื่องหมาย carat (^) นำหน้า บรรทัดที่ตรงกันต้องขึ้นต้นด้วยพารามิเตอร์ String ถ้าค่าของพารามิเตอร์ String เป็น null จะใช้สตริงก่อนหน้า
?String	เหมือนกับ /String ยกเว้น ว่าการค้นหาอยู่ในทิศทางไปข้างหน้า
n	ค้นหารายการตรงกันถัดไปของรูปแบบล่าสุดในคำสั่ง /String หรือ ?
N	ค้นหารายการตรงกันถัดไปของรูปแบบล่าสุดในคำสั่ง /String หรือ ? แต่ในทิศทางตรงกันข้าม ค้นหาประวัติของสตริงที่ป้อนด้วย คำสั่ง /String ก่อนหน้า

คำสั่ง text modification edit:

คำสั่ง text-modification edit แก้ไขบรรทัดดังนี้:

ไอเท็ม	คำอธิบาย
a	ป้อนโหมดอินพุตและป้อนข้อความหลังจากอักขระ ปัจจุบัน
A	ผนวกข้อความที่ตอนท้ายของบรรทัด เทียบเท่ากับคำสั่ง \$a
[Count]cMotion c[Count]Motion	ลบอักขระปัจจุบันผ่านทางอักขระซึ่งพารามิเตอร์ Motion ระบุเพื่อย้ายเคอร์เซอร์ และ ป้อนโหมดอินพุต หากค่าของพารามิเตอร์ Motion เป็น c ทั้งบรรทัดจะถูกลบออกและป้อนโหมดอินพุต
C	ลบอักขระปัจจุบันผ่านทางตอนท้ายของบรรทัด และป้อนโหมดอินพุต เทียบเท่ากับคำสั่ง c\$
S	เทียบเท่ากับคำสั่ง cc
D	ลบอักขระปัจจุบันผ่านทางตอนท้ายของบรรทัด เทียบเท่ากับ คำสั่ง d\$

ไอเท็ม	คำอธิบาย
[Count]dMotion d[Count]Motion	ลบอักขระปัจจุบันจนถึงและรวมอักขระที่ระบุโดยพารามิเตอร์ Motion หาก Motion เป็น d ทั้งบรรทัดจะถูกลบออก
i	ป้อนโหมดอินพุตและแทรกข้อความก่อนหน้าอักขระ ปัจจุบัน
I	ป้อนข้อความก่อนหน้าตอนต้นของบรรทัด เทียบเท่ากับ คำสั่ง Oi
[Count]P	วางการแก้ไขข้อความก่อนหน้าไว้ก่อนหน้าเคอร์เซอร์
[Count]p	วางการแก้ไขข้อความก่อนหน้าไว้หลังจากเคอร์เซอร์
R	ป้อนโหมดอินพุตและพิมพ์อักขระบน หน้าจอ
[Count]rc	แทนที่จำนวนอักขระที่ระบุโดยพารามิเตอร์ Count โดยเริ่มต้นที่ตำแหน่งเคอร์เซอร์ปัจจุบัน ด้วยอักขระที่ระบุโดยพารามิเตอร์ c คำสั่งนี้ยังเลื่อนเคอร์เซอร์ไปข้างหน้า หลังจากแทนที่อักขระแล้วด้วย
[Count]x	ลบอักขระปัจจุบัน
[Count]X	ลบอักขระที่นำหน้า

ไอเท็ม	คำอธิบาย
[Count].	ทำซ้ำคำสั่งการแก้ไขข้อความก่อนหน้านี้
[Count]~	แปลงตัวพิมพ์ของจำนวนอักขระที่ระบุโดย พารามิเตอร์ Count โดยเริ่มต้นที่ตำแหน่งเคอร์เซอร์ปัจจุบัน และเลื่อนเคอร์เซอร์ไปยังหน้า
[Count]_	ผนวกคำที่ระบุโดยพารามิเตอร์ Count ของคำสั่งก่อนหน้านี้และป้อนโหมดอินพุต จะใช้ค่าล่าสุดถ้าพารามิเตอร์ Count ถูกข้ามไป
*	ผนวกเครื่องหมายดอกจัน (*) เข้ากับค่าปัจจุบัน และพยายามทดแทนชื่อไฟล์ ถ้าไม่พบรายการที่ตรงกัน เสียงระฆังจะดังขึ้น มิฉะนั้น คำถูกแทนที่โดยรูปแบบการจับคู่และป้อน โหมดอินพุต
\	ทำชื่อไฟล์ให้เสร็จสมบูรณ์ แทนที่ค่าปัจจุบันด้วยค่าเดิมนำตัวไปที่ยาวที่สุด ของชื่อไฟล์ทั้งหมดซึ่งตรงกับค่าปัจจุบันพร้อมกับผนวก เครื่องหมายดอกจัน (*) หากรายการตรงกันไม่ซ้ำกัน จะผนวกเครื่องหมายทับ / ถ้าไฟล์เป็นไดเรกทอรี จะผนวกช่องว่างถ้าไฟล์ไม่ใช่ ไดเรกทอรี

คำสั่งแก้ไขเบ็ดเตล็ด:

คำสั่งการแก้ไขต่อไปนี้มักใช้บ่อยๆ

ไอเท็ม	คำอธิบาย
[Count]yMotion	
y[Count]Motion	เลื่อนอักขระปัจจุบันขึ้นและรวมอักขระ ที่ทำเครื่องหมายโดยตำแหน่งเคอร์เซอร์ซึ่งระบุโดยพารามิเตอร์ Motion และวางพารามิเตอร์ทั้งหมดเหล่านี้ไว้ในบัฟเฟอร์ลบ ข้อความและเคอร์เซอร์ไม่เปลี่ยนแปลง
Y	เลื่อนจากตำแหน่งปัจจุบันไปที่ตอนท้ายของบรรทัด เทียบเท่ากับคำสั่ง y\$
u	เลิกทำคำสั่งที่แก้ไขข้อความล่าสุด
U	เลิกทำคำสั่งที่แก้ไขข้อความทั้งหมดซึ่งทำบน บรรทัด
[Count]v	ส่งคืนคำสั่ง fc -e \$ {VISUAL: - \$ {EDITOR: - vi}} Count ใน บัฟเฟอร์อินพุต หากพารามิเตอร์ Count ถูกข้ามไป จะใช้บรรทัดปัจจุบัน
Ctrl-L	ป้อนบรรทัดและพิมพ์บรรทัดปัจจุบัน คำสั่งนี้มีผล ในโหมดควบคุมเท่านั้น
Ctrl-J	(บรรทัดใหม่) ดำเนินการบรรทัดปัจจุบันโดยไม่คำนึงถึงโหมด
Ctrl-M	(ส่งคืน) ดำเนินการบรรทัดปัจจุบันโดยไม่คำนึงถึงโหมด
#	ส่งบรรทัดหลังจากการแทรกสัญลักษณ์เงินปอนด์ (#) ที่ตอนต้นของบรรทัด มีประโยชน์ถ้าคุณต้องการแทรกบรรทัดปัจจุบันในประวัติ โดยไม่ดำเนินการบรรทัด ถ้าบรรทัดคำสั่งมีไพพ์หรือเครื่องหมายอัฒภาค หรืออักขระบรรทัดใหม่ สัญลักษณ์เงินปอนด์เพิ่มเติม (#) จะ ถูกแทรกที่ด้านหน้าของแต่ละสัญลักษณ์เหล่านี้ เมื่อต้องการลบสัญลักษณ์เงินปอนด์ทั้งหมด ให้ตั้งข้อมูลบรรทัดคำสั่งจากประวัติ และป้อนสัญลักษณ์เงินปอนด์อีกสัญลักษณ์หนึ่ง (#)
=	แสดงรายการชื่อไฟล์ที่ตรงกับค่าปัจจุบันราวกับว่า ผนวกเครื่องหมายดอกจัน
@Letter	ค้นหารายการสมนามสำหรับสมนามที่ชื่อ _Letter หากมีการกำหนดสมนามของชื่อนี้ ค่าจะถูกวางไว้ในคิว อินพุต สำหรับการประมวลผล

คำสั่ง Korn เชลล์หรือ POSIX เชลล์:

Korn เชลล์คือตัวแปลคำสั่งแบบโต้ตอบและภาษา การเขียนโปรแกรมคำสั่ง Korn เชลล์สอดคล้องกับ Portable Operating System Interface for Computer Environments (POSIX) ซึ่งเป็นมาตรฐานสากลสำหรับ ระบบปฏิบัติการ

POSIX ไม่ใช่ระบบปฏิบัติการ แต่เป็น *มาตรฐาน* ที่มุ่งเน้น portability ของแอปพลิเคชันที่ระดับซอร์สในหลายระบบ คุณลักษณะ POSIX มีการสร้างขึ้นที่ด้านบนสุดของ Korn shell Korn shell (หรือรู้จักในอีกชื่อหนึ่งว่า POSIX shell) นำเสนอคุณลักษณะหลายอย่างซึ่งเหมือนกันกับ shell เป้าหมายหรือ C shell เช่น ความสามารถในการเปลี่ยนทิศทาง I/O, การทดแทนที่เปลี่ยนแปลงได้ และการทดแทนชื่อไฟล์ นอกจากนี้ยังมีคำสั่งเพิ่มเติมหลายอย่างและคุณลักษณะ ภาษาการเขียนโปรแกรม:

หมายเหตุ: มีเวอร์ชันที่จำกัดของ Korn shell ที่มีอยู่ซึ่งเรียกว่า `rksh` สำหรับรายละเอียดเพิ่มเติมให้อ้างอิงคำสั่ง `rksh`

ไอเท็ม

คำอธิบาย

การประเมินทางคณิตศาสตร์

Korn shell หรือ POSIX shell สามารถทำการคำนวณเลขจำนวนเต็มโดยใช้ คำสั่ง `let` ในตัว ซึ่งใช้ฐานใดๆ ตั้งแต่ 2 ถึง 36

เพื่อให้สามารถรับรู้ถึงตัวเลขที่ขึ้นต้นด้วย 0 (ฐานแปด) และ 0x (ฐานสิบหก) ใน Korn shell ให้รับคำสั่งต่อไปนี้:

```
export XPG_SUS_ENV=ON
```

การเอ็กซ์พอร์ตตัวแปร `XPG_SUS_ENV` ส่งผลให้คำสั่งที่รันและ โลบรารีที่คำสั่งใช้สอดคล้องกับ POSIX โดยสมบูรณ์

หมายเหตุ: เนื่องจาก ระบบโลบรารีทั้งระบบสอดคล้องกับ POSIX ลักษณะการทำงานดีฟอลต์ที่คาดไว้ของคำสั่งที่กำหนดจึงอาจเปลี่ยนได้

```
export OCTAL_CONST=ON
```

การเอ็กซ์พอร์ตตัวแปรนี้ส่งผลให้การตีความของค่าคงที่ซึ่งประกาศใน Korn shell สอดคล้องกับ POSIX ตราบเท่าที่มีการรับรู้ถึงค่าคงที่ฐานแปด และฐานสิบหก

ประวัติคำสั่ง

Korn shell หรือ POSIX shell จัดเก็บไฟล์ที่บันทึกคำสั่งทั้งหมด ซึ่งคุณป้อน คุณสามารถใช้โปรแกรมแก้ไขข้อความเพื่อเปลี่ยนคำสั่งในไฟล์ประวัตินี้ จากนั้น ออกใช้คำสั่งอีกครั้ง

สิ่งอำนวยความสะดวกโปร

ช่วยให้คุณสามารถรันโปรแกรมในพื้นหลัง และส่งและได้รับ ข้อมูลจากกระบวนการส่วนหลังเหล่านี้

เซสรวม

การแก้ไข

Korn shell หรือ POSIX shell นำเสนออ็อปชันการแก้ไขแบบแทรกซึ่งช่วยให้คุณแก้ไขบรรทัดคำสั่งได้ มีโปรแกรมแก้ไขที่คล้ายกับ `emacs`, `gmacs`, และ `vi`

คำสั่ง Korn shell เป็นอย่างไรต่อไปนี้:

- คำสั่งแบบง่าย
- ไฟล์ไลน์
- รายการ
- คำสั่งผสม
- ฟังก์ชัน

เมื่อคุณออกใช้คำสั่งใน Korn shell หรือ POSIX shell shell จะประเมินคำสั่งและทำต่อไปนี้:

- ทำการทดแทนที่บ่งชี้ทั้งหมด
- พิจารณาว่าคำสั่งมีเครื่องหมายทับ (/) หรือไม่ ถ้ามี shell จะรันโปรแกรมที่ระบุโดยชื่อพาที่ระบุ

ถ้าคำสั่งไม่มีเครื่องหมายทับ (/) Korn shell หรือ POSIX shell จะทำการดำเนินการต่อไปนี้ต่อไป:

- พิจารณาว่าคำสั่งเป็นคำสั่งในตัวพิเศษหรือไม่ ถ้าเป็น shell จะรันคำสั่งภายในโปรเซส shell ปัจจุบัน
- เปรียบเทียบคำสั่งกับฟังก์ชันที่ผู้ใช้กำหนด หากคำสั่งตรงกับ ฟังก์ชันที่ผู้ใช้กำหนด พารามิเตอร์ตำแหน่งจะถูกบันทึกไว้ จากนั้น รีเซตเป็นอาร์กิวเมนต์ของการเรียก *function* เมื่อ ฟังก์ชันเสร็จสมบูรณ์หรือออกการส่งคืน รายการพารามิเตอร์ตำแหน่งจะถูกเรียกคืน และดำเนินการ `trap` ใดๆ ที่ตั้งค่าไว้บน EXIT ภายใน ฟังก์ชัน ค่าของฟังก์ชันเป็นค่าของคำสั่งล่าสุดที่ดำเนินการ ฟังก์ชันมีการดำเนินการในโปรเซส shell ปัจจุบัน
- หากชื่อคำสั่งตรงกับชื่อของคำสั่งในตัวปกติ คำสั่งในตัวปกติ นั้น จะถูกเรียกใช้

- สร้างโปรเซสและพยายามดำเนินการคำสั่งโดยใช้คำสั่ง exec (ถ้าคำสั่งไม่ใช่คำสั่งในตัวและไม่ใช้ฟังก์ชันที่ผู้กำหนด)

Korn shell หรือ POSIX shell ค้นหาแต่ละไดเรกทอรีในพาธที่ระบุ เพื่อหาไฟล์ปฏิบัติการ ตัวแปรเซลล์ PATH กำหนด พาธการค้นหาสำหรับไดเรกทอรีที่มีคำสั่ง ชื่อไดเรกทอรีอื่นมีการแบ่งด้วยเครื่องหมายจุดคู่ (:) ดีโฟลต์พาธคือ /usr/bin: (การระบุไดเรกทอรี /usr/bin และไดเรกทอรีปัจจุบันในลำดับนั้น) ไดเรกทอรีปัจจุบันมีการระบุโดยเครื่องหมายจุดคู่ที่ต่อกันตั้งแต่สองเครื่องหมายขึ้นไป หรือโดยเครื่องหมายจุดคู่ที่ตอนต้นหรือตอนท้ายของรายการพาธ

หากไฟล์มีสิทธิดำเนินการแต่ไม่ใช่ไดเรกทอรี หรือไฟล์ a.out เซลล์จะสมมติว่าไฟล์มี คำสั่งเซลล์ โปรเซสเซลล์ปัจจุบันสร้างเซลล์ย่อยเพื่ออ่านไฟล์ สมมติที่ไม่ได้เอ็กซ์พอร์ตทั้งหมด ฟังก์ชัน และพารามิเตอร์ที่ระบุถูกลบออกจาก ไฟล์ ถ้าไฟล์คำสั่งเซลล์มีสิทธิอนุญาต อ่าน หรือถ้าเปิด setuid หรือ setgid มีการตั้งค่าบนไฟล์ เซลล์จะรันเอเจนต์ที่ตั้งค่าสิทธิอนุญาต และดำเนินการเซลล์ด้วยไฟล์คำสั่งเซลล์ซึ่งส่งผ่านเป็นไฟล์ เปิด คำสั่งในวงเล็บรันในเซลล์ย่อยโดยไม่ได้อับปรมาณ ที่ไม่ได้เอ็กซ์พอร์ต

หลักการที่เกี่ยวข้อง:

“เซลล์ที่มีอยู่” ในหน้า 230

ข้อมูลต่อไปนี้เป็นเซลล์ที่จัดเตรียมให้พร้อมกับ AIX

“สิ่งอำนวยความสะดวกโปรเซสรวม” ในหน้า 257

Korn shell หรือ POSIX shell อนุญาตให้คุณรันคำสั่งตั้งแต่หนึ่งคำสั่งขึ้นไป เป็นกระบวนการส่วนหลังได้ คำสั่งเหล่านี้ที่รันจากภายในเซลล์สคริปต์ เรียกว่า *โปรเซสรวม*

“การแก้ไขแบบแทรกใน Korn shell หรือ POSIX shell” ในหน้า 269

โดยปกติ คุณพิมพ์บรรทัดคำสั่งแต่ละบรรทัดจากอุปกรณ์เทอร์มินัลและ ตามด้วยอักขระบรรทัดใหม่ (RETURN หรือ LINE FEED) เมื่อคุณเรียกใช้ออปชันการแก้ไขแบบแทรก emacs, gmacs, หรือ vi คุณสามารถแก้ไข บรรทัดคำสั่งได้

“การประเมินทางคณิตศาสตร์ใน Korn shell หรือ POSIX shell” ในหน้า 239

คำสั่งในตัว let ปกติของ Korn shell หรือ POSIX shell ช่วยให้คุณสามารถทำการคำนวณเลขจำนวนเต็มได้

“คำสั่งในตัว Korn shell หรือ POSIX shell” ในหน้า 258

คำสั่งพิเศษมีการสร้างขึ้นใน Korn shell และ POSIX shell และดำเนินการในโปรเซสเซลล์

คำสั่งผสม Korn shell:

คำสั่งผสมสามารถเป็นรายการของคำสั่งแบบง่ายหรือโฟฟไลน์ หรือสามารถขึ้นต้นด้วยคำที่สำรองไว้ ในเวลาส่วนใหญ่ คุณจะใช้คำสั่งผสม เช่น if, while, และ for เมื่อ คุณเขียนเซลล์สคริปต์

ข้อมูลต่อไปนี้เป็นรายการของคำสั่งผสม Korn shell หรือ POSIX shell:

ไวยากรณ์คำสั่ง	คำอธิบาย
<code>for Identifier [in Word ...];do List;done</code>	ในแต่ละครั้งที่ดำเนินการคำสั่ง for พารามิเตอร์ Identifier ถูกตั้งค่าเป็นคำถัดไป ที่นำมาจากรายการ in Word ... หากคำสั่ง in Word ... ถูกข้ามไป ผลคือคำสั่ง for จะดำเนินการคำสั่ง do List หนึ่งครั้งสำหรับแต่ละพารามิเตอร์ตำแหน่งซึ่งตั้งค่าไว้ การดำเนินการสิ้นสุดเมื่อไม่มี คำอยู่ในรายการอีกแล้ว
<code>select Identifier [in Word ...];do List;done</code>	คำสั่ง select พิมพ์ชุดของคำที่ระบุไว้บน ข้อผิดพลาดมาตรฐาน (ตัวอธิบายไฟล์ 2) โดยแต่ละคำ มีหมายเลขหน้าหน้า หากคำสั่ง in Word ... ถูกข้ามไป ผลคือจะใช้พารามิเตอร์ตำแหน่งแทน มีการพิมพ์พร้อมด PS3 และอ่าน บรรทัดจากอินพุตมาตรฐาน ถ้าบรรทัดนี้ประกอบด้วย หมายเลขของคำที่แสดงรายการอย่างใดอย่างหนึ่ง ผลคือค่าของพารามิเตอร์ Identifier จะถูกตั้งค่าเป็นคำซึ่งสอดคล้องกับหมายเลขนี้ หากบรรทัดที่อ่านจาก อินพุตมาตรฐานว่าง รายการเลือกจะถูกพิมพ์อีกครั้ง มิฉะนั้น ค่าของ พารามิเตอร์ Identifier มีการตั้งค่าเป็น null เนื้อหาของ บรรทัดที่อ่านจากอินพุตมาตรฐานมีการบันทึกไว้ในพารามิเตอร์ REPLY พารามิเตอร์ List มีการดำเนินการสำหรับแต่ละการเลือกจนกว่าจะพบ ตัวแบ่งหรืออักขระสิ้นสุดไฟล์

ไวยากรณ์คำสั่ง	คำอธิบาย
<code>case Word in [(Pattern [Pattern] ...) List ;] ... esac</code>	คำสั่ง <code>case</code> ดำเนินการ พารามิเตอร์ <code>List</code> ที่เชื่อมโยงกับพารามิเตอร์ <code>Pattern</code> แรก ซึ่งตรงกับพารามิเตอร์ <code>Word</code> รูปแบบเหมือนกับที่ใช้สำหรับการทดแทนชื่อไฟล์
<code>if List ;then List [elif List ;then List] ... [;else List] ;fi</code>	พารามิเตอร์ <code>List</code> ระบุรายการของคำสั่งที่จะรัน เซลล์ดำเนินการคำสั่ง <code>if List</code> ก่อน หากมีการส่งคืนสถานะการออก เป็นศูนย์ เซลล์จะดำเนินการคำสั่ง <code>then List</code> มิฉะนั้น จะดำเนินการคำสั่งที่ระบุโดยพารามิเตอร์ <code>List</code> ตามหลังคำสั่ง <code>elif</code> หากค่า ที่ส่งคืนโดยคำสั่งล่าสุดในคำสั่ง <code>elif List</code> เป็นศูนย์ จะดำเนินการคำสั่ง <code>then List</code> หากค่า ที่ส่งคืนโดยคำสั่งล่าสุดในคำสั่ง <code>then List</code> เป็นศูนย์ จะดำเนินการคำสั่ง <code>else List</code> ถ้าไม่มีการดำเนินการ คำสั่งที่ระบุโดยพารามิเตอร์ <code>List</code> สำหรับคำสั่ง <code>else</code> หรือ <code>then</code> คำสั่ง <code>if</code> จะส่งคืน สถานะการออกเป็นศูนย์
<code>while List ;do List ;done until List ;do List ;done</code>	พารามิเตอร์ <code>List</code> ระบุ รายการของคำสั่งที่จะรัน คำสั่ง <code>while</code> ดำเนินการคำสั่งที่ระบุโดย พารามิเตอร์ <code>List</code> ซ้ำ ถ้า สถานะการออกของคำสั่งล่าสุดในคำสั่ง <code>while List</code> เป็นศูนย์ จะมีการดำเนินการคำสั่ง <code>do List</code> ถ้าสถานะการออกของคำสั่งล่าสุดในคำสั่ง <code>while List</code> ไม่ใช่ศูนย์ ลูปจะยุติลง หากไม่มีการดำเนินการคำสั่งในคำสั่ง <code>do List</code> ผลคือคำสั่ง <code>while</code> จะส่งคืนสถานะการออกเป็นศูนย์ อาจใช้คำสั่ง <code>until</code> แทนคำสั่ง <code>while</code> เพื่อลบวงจรทดสอบการยุติ ลูป
<code>(List)</code>	พารามิเตอร์ <code>List</code> ระบุ รายการของคำสั่งที่จะรัน เซลล์ดำเนินการพารามิเตอร์ <code>List</code> ในสภาวะแวดล้อมแยกต่างหาก หมายเหตุ: หากจำเป็นต้องใช้วงเล็บเปิดที่ต่อกันสองเครื่องหมายสำหรับ การจัดกลุ่ม คุณต้องแทรกช่องว่าง ระหว่างวงเล็บทั้งสองเพื่อแยกความแตกต่าง ระหว่างคำสั่งและการประเมินทางคณิตศาสตร์
<code>{ List ; }</code>	พารามิเตอร์ <code>List</code> ระบุ รายการของคำสั่งที่จะรัน พารามิเตอร์ <code>List</code> มีการดำเนินการ แบบง่าย หมายเหตุ: สิ่งที่แตกต่างกันจากเมตาอักขระคือ <code>()</code> , <code>{ }</code> แสดงถึงค่าที่สำรองไว้ (ใช้สำหรับวัตถุประสงค์พิเศษ ไม่ใช่ตัวระบุ ที่ผู้ใช้กำหนด) เพื่อให้รับรู้ ค่าที่สำรองไว้เหล่านี้ต้องปรากฏขึ้นที่ตอนต้น ของบรรทัดหรือหลังจากเครื่องหมายอัฒภาค <code>(;)</code>
<code>[[Expression]]</code>	ประเมินพารามิเตอร์ <code>Expression</code> หาก นิพจน์เป็นจริง คำสั่งจะส่งคืนสถานะการออกเป็นศูนย์
<code>function Identifier { List ; } or function Identifier () { List ; }</code>	กำหนดฟังก์ชันที่อ้างถึงโดย พารามิเตอร์ <code>Identifier</code> เนื้อหาของฟังก์ชันคือรายการ ของคำสั่งที่ระบุไว้ในเครื่องหมาย <code>{ } ()</code> ประกอบด้วยตัวดำเนินการสองตัว ดังนั้นจึงสามารถผสมอักขระว่างเปล่ากับ <code>identifier</code> , <code>(and)</code> ได้ แต่ไม่ใช่สิ่งจำเป็น
<code>time Pipeline</code>	ดำเนินการพารามิเตอร์ <code>Pipeline</code> เวลาที่ผ่านไป เวลาผู้ใช้ และเวลาระบบมีการพิมพ์ไปยังข้อผิดพลาดมาตรฐาน

หลักการที่เกี่ยวข้อง:

“พารามิเตอร์ใน Korn เซลล์” ในหน้า 251
พารามิเตอร์ Korn เซลล์มีการอธิบายข้างล่าง

เซลล์สตาร์ทอัพ:

คุณสามารถเริ่มต้น Korn เซลล์โดยใช้คำสั่ง `ksh`, คำสั่ง `psh` (POSIX เซลล์), หรือคำสั่ง `exec`

หากเซลล์มีการเริ่มต้นโดยคำสั่ง `exec` และอักขระตัวแรกของอาร์กิวเมนต์ศูนย์ (`$0`) เป็นยัติภังค์ (`-`) เซลล์จะมีการสมมติว่าเป็นเซลล์การล็อกอิน อันดับแรก เซลล์อ่านคำสั่ง จากไฟล์ `/etc/profile` จากนั้น อ่านจากไฟล์ `.profile` ในไดเรกทอรีปัจจุบันหรือจากไฟล์ `$HOME/.profile` ถ้ามี ไฟล์ใดไฟล์หนึ่งอยู่ ถัดไป เซลล์อ่านคำสั่งจากไฟล์ที่ระบุ โดยทำการทดแทนพารามิเตอร์ในค่าของตัวแปรสภาวะแวดล้อม `ENV` ถ้ามีไฟล์อยู่

หากคุณระบุพารามิเตอร์ `File [Parameter]` เมื่อเรียกใช้ Korn เซลล์หรือ POSIX เซลล์ เซลล์จะรันไฟล์สคริปต์ ที่ระบุโดยพารามิเตอร์ `File` รวมถึง พารามิเตอร์ใดๆ ที่ระบุไว้ ไฟล์สคริปต์ที่ระบุต้องมีสิทธิการอ่าน ค่าติดตั้ง `setuid` และ `setgid` ถูกละเว้นไปจากนั้น เซลล์อ่าน คำสั่ง

หมายเหตุ: อย่าระบุไฟล์สคริปต์พร้อมกับแฟล็ก `-c` หรือ `-s` เมื่อเรียกใช้ Korn เซลล์หรือ POSIX เซลล์

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับพารามิเตอร์ตำแหน่ง ให้ดูที่ “พารามิเตอร์ใน Korn shell” ในหน้า 251

หลักการที่เกี่ยวข้อง:

“พารามิเตอร์ใน Korn shell” ในหน้า 251

พารามิเตอร์ Korn shell มีการอธิบายข้างล่าง

สภาวะแวดล้อม Korn shell:

ตัวแปรทั้งหมด (พร้อมด้วยค่าที่เชื่อมโยง) ซึ่งคำสั่งรู้จัก เมื่อเริ่มต้นการดำเนินการของคำสั่งประกอบกันเป็น *สภาวะแวดล้อม*

สภาวะแวดล้อมนี้ รวมถึงตัวแปรที่คำสั่งได้รับสืบทอดมาจากโปรเซสพารেন্ট และตัวแปรที่ระบุเป็นพารามิเตอร์คีย์เวิร์ดบนบรรทัดคำสั่งซึ่งเรียกคำสั่ง shell โต้ตอบกับสภาวะแวดล้อม ในหลายวิธี เมื่อเริ่มต้นขึ้น shell จะสแกนสภาวะแวดล้อมและสร้างพารามิเตอร์สำหรับแต่ละชื่อที่พบ ให้ค่าที่สอดคล้องกันกับพารามิเตอร์ และทำเครื่องหมายเพื่อเอ็กซ์พอร์ต คำสั่งที่ดำเนินการได้รับสืบทอดสภาวะแวดล้อม

หากคุณแก้ไขค่าของพารามิเตอร์ shell หรือ สร้างพารามิเตอร์ใหม่โดยใช้คำสั่ง `export` หรือ `typeset -x` พารามิเตอร์นั้นจะกลายเป็นส่วนหนึ่งของสภาวะแวดล้อม ดังนั้น สภาวะแวดล้อมซึ่งคำสั่งที่ดำเนินการ มองเห็นจึงประกอบด้วยคู่ของชื่อ-ค่าใดๆ ซึ่ง shell ได้รับสืบทอดมา ตั้งแต่ดั้งเดิมและอาจมีการแก้ไขค่าโดย shell ปัจจุบัน บวกส่วนเพิ่มใดๆ ที่เป็นผลมาจากการใช้คำสั่ง `export` หรือ `typeset -x` คำสั่งที่ดำเนินการ (shell ย่อย) จะเห็นการแก้ไขใดๆ ซึ่งตัวเอง ทำในตัวแปรสภาวะแวดล้อมที่ได้รับสืบทอดมา แต่เพื่อให้ shell ย่อยหรือโปรเซส เห็นค่าที่แก้ไข shell ย่อยต้องเอ็กซ์พอร์ตตัวแปรเหล่านี้

สภาวะแวดล้อมสำหรับคำสั่งหรือฟังก์ชันแบบง่ายสามารถ เปลี่ยนได้โดยการเติมหน้าด้วยการกำหนดพารามิเตอร์ตั้งแต่หนึ่งรายการขึ้นไป อาร์กิวเมนต์การกำหนด พารามิเตอร์เป็นค่าในรูปแบบ `Identifier=Value` ดังนั้น นิพจน์สองรายการต่อไปนี้จึงเทียบเท่ากัน (ตราที่ยังเกี่ยวข้องกับ การดำเนินการของคำสั่ง):

```
TERM=450 Command arguments
```

```
(export TERM; TERM=450; Command arguments)
```

ฟังก์ชัน Korn shell:

ค่าที่ส่งวนไว้ ฟังก์ชัน กำหนดฟังก์ชันของ shell shell อ่านและจัดเก็บฟังก์ชันไว้ใน ชื่อสมนามมีการแก้ไขเมื่อ อ่านฟังก์ชัน shell ดำเนินการฟังก์ชันในลักษณะเดียวกันกับคำสั่ง โดยมีอาร์กิวเมนต์ที่ส่งผ่านเป็นพารามิเตอร์ตำแหน่ง

Korn shell หรือ POSIX shell ดำเนินการฟังก์ชันใน สภาวะแวดล้อมซึ่งเรียกใช้ฟังก์ชัน ข้อมูลทั้งหมดต่อไปนี้มีการแบ่งใช้โดยฟังก์ชันและสคริปต์ที่เรียกใช้ และอาจเกิดผลข้างเคียงได้:

- ค่าตัวแปรและแอตทริบิวต์ (ยกเว้นว่าคุณใช้คำสั่ง `typeset` ภายในฟังก์ชันเพื่อประกาศตัวแปรโลคัล)
- ไตเร็กทอรีการทำงาน
- สมนาม นิยามฟังก์ชัน และแอตทริบิวต์
- พารามิเตอร์พิเศษ \$
- ไฟล์เปิด

ข้อมูลต่อไปนี้ไม่มีการแบ่งใช้ ระหว่างฟังก์ชันและสคริปต์ที่เรียกใช้ และไม่มีผลข้างเคียง:

- พารามิเตอร์ตำแหน่ง
- พารามิเตอร์พิเศษ #

- ตัวแปรในรายการกำหนดตัวแปรเมื่อเรียกใช้ฟังก์ชัน
- ตัวแปรที่ประกาศโดยใช้คำสั่ง `typeset` ภายใน ฟังก์ชัน
- อี้อพชัน
- การดักจับ อย่างไรก็ตาม สัญญาณที่ละเว้นโดยสคริปต์ที่เรียกใช้จะถูกละเว้น โดยฟังก์ชันด้วย

หมายเหตุ: ในเวอร์ชันก่อนหน้าของ Korn เซลล์ การดักจับอื่นที่ไม่ใช่ `EXIT` และ `ERR` มีการ แบ่งใช้โดยฟังก์ชันและสคริปต์ที่เรียกใช้

หากการดักจับ `O` หรือ `EXIT` มีการดำเนินการ *ภายใน* ตัวของฟังก์ชัน การดำเนินการจะทำหลังจากฟังก์ชันเสร็จสมบูรณ์ในสถานะแวดล้อมที่เรียกฟังก์ชัน หากมีการดำเนินการดักจับ *ภายนอก* ตัวของฟังก์ชัน การดำเนินการจะทำในทันทีที่ออกจาก Korn เซลล์ ในเวอร์ชันก่อนหน้าของ Korn เซลล์ ไม่มีการดำเนินการดักจับ `O` หรือ `EXIT` ภายนอกตัวของฟังก์ชันในทันทีที่ออกจากฟังก์ชัน

เมื่อดำเนินการฟังก์ชัน ฟังก์ชันมีไวยากรณ์-ข้อผิดพลาดและคุณสมบัติ การกำหนดตัวแปรเหมือนกับที่อธิบายไว้ในคำสั่งในตัว Korn เซลล์หรือ POSIX เซลล์

คำสั่งผสมมีการดำเนินการในทุกครั้งที่ระบุชื่อฟังก์ชัน เป็นชื่อของคำสั่งแบบง่าย ตัวถูกดำเนินการของคำสั่งจะกลายเป็น พารามิเตอร์ตำแหน่งเป็นการชั่วคราวในระหว่างการดำเนินการของ คำสั่งผสม พารามิเตอร์พิเศษ# จะเปลี่ยนเพื่อสะท้อน ถึงจำนวนของตัวถูกดำเนินการด้วย อย่างไรก็ตาม พารามิเตอร์พิเศษ 0 จะ ไม่เปลี่ยน

คำสั่งพิเศษ `return` ใช้เพื่อ กลับจากการเรียกฟังก์ชัน ข้อผิดพลาดภายในฟังก์ชันส่งคืนการควบคุมไปยัง ผู้เรียก

ตัวระบุฟังก์ชันมีการแสดงรายการพร้อมด้วยอี้อพชัน `-f` หรือ `+f` ของคำสั่งพิเศษ `typeset` อี้อพชัน `-f` ยังแสดงรายการ ข้อความของฟังก์ชันด้วย ฟังก์ชันถูกยกเลิกกำหนดด้วยอี้อพชัน `-f` ของ คำสั่งพิเศษ `unset`

โดยปกติ ฟังก์ชันถูกยกเลิกการตั้งค่าเมื่อเซลล์ดำเนินการ เซลล์สคริปต์ อี้อพชัน `-xf` ของคำสั่งพิเศษ `typeset` อนุญาตให้เอ็กซ์พอร์ต ฟังก์ชันเป็นสคริปต์ที่จะดำเนินการโดยเซลล์ไม่ต้องเรียกใช้ แยกต่างหาก ฟังก์ชันที่ต้องกำหนดในการเรียกใช้แยกต่างหากของเซลล์ ควรมีการระบุไว้ในไฟล์ `ENV` ด้วย อี้อพชัน `-xf` ของคำสั่งพิเศษ `typeset`

สถานะการออกของนิยามฟังก์ชันเป็นศูนย์ถ้า ประกาศฟังก์ชันไม่สำเร็จ มิฉะนั้น สถานะการออกจะมากกว่า ศูนย์ สถานะการออกของการเรียกใช้ฟังก์ชันคือสถานะการออกของ คำสั่งล่าสุดที่ดำเนินการโดยฟังก์ชัน

หลักการที่เกี่ยวข้อง:

“พารามิเตอร์ใน Korn เซลล์” ในหน้า 251

พารามิเตอร์ Korn เซลล์มีการอธิบายข้างล่าง

“คำสั่งในตัว Korn เซลล์หรือ POSIX เซลล์” ในหน้า 258

คำสั่งพิเศษมีการสร้างขึ้นใน Korn เซลล์และ POSIX เซลล์ และดำเนินการในโปรเซสเซลล์

ประวัติคำสั่ง Korn เซลล์หรือ POSIX เซลล์:

Korn เซลล์หรือ POSIX เซลล์บันทึกคำสั่งที่บ่อนจาก อุปกรณ์เทอร์มินัลไว้ในไฟล์ประวัติ

ถ้าตั้งค่า ค่าตัวแปร *HISTFILE* เป็นชื่อของไฟล์ประวัติ ถ้าไม่ได้ตั้งค่าตัวแปร *HISTFILE* หรือไม่สามารถเขียนได้ ไฟล์ประวัติที่ใช้คือ `$HOME/.sh_history` หากไม่มีไฟล์ประวัติและ Korn shell ไม่สามารถสร้างไฟล์ประวัติได้ หรือ ถ้ามีอยู่แต่ Korn shell ไม่มีสิทธิผนวกข้อมูลเข้ากับไฟล์ประวัติ Korn shell จะใช้ไฟล์ชั่วคราวเป็นไฟล์ประวัติ เซลล์เข้าถึง คำสั่งของเซลล์ได้ตอบทั้งหมด โดยใช้ไฟล์ประวัติที่ระบุเดียวกัน ซึ่งมีสิทธิอนุญาตที่เหมาะสม

โดยค่าดีฟอลต์ Korn shell หรือ POSIX shell จะบันทึกข้อความ ของ 128 คำสั่งล่าสุดของผู้ใช้ที่ไม่ใช่ผู้ใช้ราก และ 512 คำสั่งของผู้ใช้ ราก ขนาดของไฟล์ประวัติ (ที่ระบุโดยตัวแปร *HISTSZ*) ไม่มีขีดจำกัด แม้ว่าไฟล์ประวัติขนาดใหญ่อาจส่งผลให้ Korn shell เริ่มต้นได้ช้าก็ตาม

การทดแทนประวัติคำสั่ง:

ใช้คำสั่งในตัว `fc` เพื่อแสดงรายการหรือแก้ไข ส่วนต่างๆ ของไฟล์ประวัติ เมื่อต้องการเลือกส่วนของไฟล์ที่จะแก้ไขหรือแสดงรายการ ให้ระบุ หมายเลขหรืออักขระตัวแรกของคำสั่ง

คุณสามารถระบุคำสั่งเดียวหรือช่วงของคำสั่ง

หากคุณไม่ได้รับโปรแกรมแก้ไขเป็นอาร์กิวเมนต์ ของคำสั่งในตัวปกติ `fc` จะมีการใช้โปรแกรมแก้ไขที่ระบุโดยตัวแปร *FCEDIT* ถ้าไม่ได้กำหนดตัวแปร *FCEDIT* จะใช้ไฟล์ `/usr/bin/ed` คำสั่งที่แก้ไขมีการพิมพ์ และรันเมื่อคุณออกจากโปรแกรมแก้ไข

เครื่องหมายยัติภังค์ (-) ในชื่อโปรแกรมแก้ไข ใช้เพื่อข้ามระยะการแก้ไขและรันคำสั่งอีกครั้ง ในกรณีนี้ สามารถใช้พารามิเตอร์การทดแทนในรูปแบบ `01d=New` เพื่อแก้ไขคำสั่ง ก่อนจะรัน ตัวอย่างเช่น ถ้า `r` เป็นสมนามของ `fc -e` การพิมพ์ `r bad=good c` จะรันคำสั่งล่าสุด ที่ขึ้นต้นด้วยตัวอักษร `c` และแทนที่ การเกิดขึ้นครั้งแรกของสตริง `bad` ด้วยสตริง `good`

งานที่เกี่ยวข้อง:

“การแสดงรายการคำสั่งที่ป้อนก่อนหน้านี้ (คำสั่ง history)” ในหน้า 145
ใช้คำสั่ง `history` เพื่อแสดงรายการคำสั่งซึ่ง คุณป้อนก่อนหน้านี้

การกำหนดสมนามคำสั่งใน Korn shell หรือ POSIX shell:

Korn shell หรือ POSIX shell อนุญาตให้คุณสร้างสมนามให้กับ คำสั่งแบบกำหนดเองได้

คำสั่ง `alias` กำหนดค่าใน รูปแบบ `Name=String` เป็นสมนาม เมื่อคุณใช้สมนามเป็นคำแรก ของบรรทัดคำสั่ง Korn shell จะตรวจสอบดูว่ามีการประมวลผลสมนาม ที่มีชื่อเดียวกันแล้วหรือไม่ ถ้ามี Korn shell จะไม่ แทนที่ชื่อสมนาม หากสมนามที่มีชื่อเดียวกันยังไม่ได้ประมวลผล Korn shell จะแทนที่ชื่อสมนามด้วยค่าของสมนาม

อักขระตัวแรกของชื่อสมนามสามารถเป็นอักขระที่พิมพ์ได้ใดๆ ยกเว้น เมตาอักขระ อักขระที่เหลือต้องเหมือนกับตัวระบุที่ถูกต้อง สตริงที่นำมาแทนที่สามารถประกอบด้วยข้อความเซลล์ที่ถูกต้องใดๆ รวมถึง เมตาอักขระ

หากอักขระตัวสุดท้ายของค่าสมนามเป็นวงเล็บ เซลล์จะตรวจสอบ คำที่ต่อจากสมนามสำหรับการทดแทนสมนามด้วย คุณสามารถใช้สมนามเพื่อกำหนดคำสั่งในตัว พิเศษอีกครั้ง แต่ไม่ใช้กำหนดค่าที่ส่งวนไว้อีกครั้ง นิยาม สมนามไม่มีการสืบทอดเมื่อเรียกใช้ `ksh` อย่างไรก็ตาม ถ้าคุณระบุ `alias -x` สมนามยังคงมีผลบังคับใช้ สำหรับสคริปต์ที่เรียกใช้โดยชื่อซึ่งไม่ได้เรียกใช้เซลล์แยกต่างหาก เมื่อต้องการเอ็กซ์พอร์ต นิยามสมนามและทำให้โปรเซสชายนี้อีกมีสิทธิเข้าถึงนิยามได้ คุณ ต้องระบุ `alias -x` และนิยามสมนามในไฟล์ สภาวะแวดล้อมของคุณ

ใช้คำสั่ง `alias` เพื่อสร้าง แสดงรายการ และเอ็กซ์พอร์ตสมนาม

ใช้คำสั่ง `unalias` เพื่อลบสมนาม

รูปแบบสำหรับการสร้างสมนามเป็นดังนี้:

```
alias Name=String
```

โดยที่พารามิเตอร์ `Name` ระบุชื่อของสมนาม และพารามิเตอร์ `String` ระบุค่าของสมนาม

สมนามที่เอ็กซ์พอร์ตต่อไปนี้มีกำหนดไว้ล่วงหน้าโดย Korn เชลล์แต่สามารถยกเลิกการตั้งค่าหรือกำหนดใหม่ได้ไม่แนะนำให้คุณเปลี่ยนสมนามเหล่านี้ เนื่องจาก อาจสร้างความสับสนให้แก่ผู้ที่ต้องการใช้สมนามดังที่กำหนดไว้ล่วงหน้าโดย Korn เชลล์ในภายหลังได้

```
autoload='typeset -fu'  
false='let 0'  
functions='typeset -f'  
hash='alias -t'  
history='fc -l'  
integer='typeset -i'  
nohup='nohup '  
r='fc -e -'  
true=':'  
type='whence -v'
```

ไม่สนับสนุนสมนามในการเรียกใช้แบบไม่โต้ตอบของ Korn เชลล์ (`ksh`) ตัวอย่างเช่น ในเชลล์สคริปต์ หรือด้วยอ็อปชัน `-c` ใน `ksh` ดังเช่นในต่อไปนี้:

```
ksh -c alias
```

งานที่เกี่ยวข้อง:

“การสร้างสมนามคำสั่ง (คำสั่ง `alias shell`)” ในหน้า 148

สมนาม อนุญาตให้คุณสร้างชื่อช็อตคัตสำหรับคำสั่ง ชื่อไฟล์ หรือข้อความเชลล์ใดๆ โดยใช้สมนาม คุณสามารถประหยัดเวลาได้มากเมื่อทำการกิจซึ่งคุณทำบ่อยครั้ง คุณสามารถสร้างสมนามคำสั่งได้

สมนามที่ติดตาม:

บ่อยครั้งที่มีการใช้สมนามเป็นการเขียนย่อสำหรับชื่อพารแบบเต็ม อ็อปชันสั่งอำนวยความสะดวกกำหนดสมนามหนึ่งช่วยให้คุณสามารถตั้งค่าของ สมนามให้กับชื่อพารแบบเต็มของคำสั่งที่สอดคล้องกัน สมนามชนิดพิเศษนี้ คือสมนาม *ที่ติดตาม*

สมนามที่ติดตามเพิ่มความเร็วในการดำเนินการโดยการลดความจำเป็นที่ เชลล์ต้องค้นหาตัวแปร `PATH` สำหรับชื่อพารแบบเต็ม

คำสั่ง `set -h` จะเปิด *การติดตาม* คำสั่งเพื่อให้ ในแต่ละครั้งที่มีการอ้างอิงคำสั่ง เชลล์จะกำหนดค่าของสมนามที่ ติดตาม คำนี้ถูกยกเลิกการกำหนดในแต่ละครั้งที่คุณรีเซตตัวแปร `PATH`

สมนามเหล่านี้ยังคงถูกติดตามเพื่อให้การอ้างอิงในลำดับถัดไปจะ กำหนดค่าอีกครั้ง สมนามที่ติดตามหลายรายการมีการคอมไพล์เข้าในเชลล์

การทดแทนเครื่องหมายการออกเสียง:

หลังจากเซลล์ทำการทดแทนสมนามแล้ว เซลล์จะตรวจสอบแต่ละคำเพื่อดูว่าคำขึ้นต้นด้วยเครื่องหมายการออกเสียง (~) ที่ไม่ได้อยู่ในอัญประกาศหรือไม่ ถ้าเป็นเช่นนั้น เซลล์จะตรวจสอบคำไปจนถึงเครื่องหมายทับแรก (/) เพื่อดูว่าตรงกับชื่อผู้ใช้ในไฟล์ /etc/passwd หรือไม่ หากเซลล์พบรายการที่ตรงกัน เซลล์จะแทนที่อักขระ ~ และ ชื่อด้วยไอดีเรกทอรีล็อกอินของผู้ใช้ที่ตรงกัน โพรเซสนี้เรียกว่า การทดแทน เครื่องหมายการออกเสียง

เซลล์ไม่เปลี่ยนข้อความดั้งเดิมถ้าเซลล์ไม่พบรายการที่ตรงกัน Korn เซลล์ยังจัดทำการแทนที่พิเศษด้วยถ้าอักขระ ~ เป็นเพียงอักขระเดียวในคำหรือตามด้วยเครื่องหมายบวก (+) หรือยัติภังค์ (-):

ไอเท็ม	คำอธิบาย
~	แทนที่ด้วยค่าของตัวแปร HOME
~+	แทนที่ด้วยตัวแปร \$PWD (ชื่อพารามิเตอร์ของไอดีเรกทอรีปัจจุบัน)
~-	แทนที่ด้วยตัวแปร \$OLDPWD (ชื่อพารามิเตอร์ของไอดีเรกทอรีก่อนหน้านี้)

นอกจากนี้ เซลล์พยายามทดแทนเครื่องหมายการออกเสียงเมื่อค่าของ พารามิเตอร์กำหนดตัวแปรขึ้นต้นด้วยอักขระเครื่องหมายการออกเสียง ~

เซลล์เป้าหมาย

เซลล์เป้าหมายคือตัวแปลคำสั่งแบบโต้ตอบและภาษา การเขียนโปรแกรมคำสั่ง

คำสั่ง bsh รันเซลล์เป้าหมาย

เซลล์เป้าหมายสามารถรันเป็นเซลล์การล็อกอิน หรือเซลล์ย่อยภายใต้เซลล์การล็อกอิน อย่างใดอย่างหนึ่ง เฉพาะคำสั่ง login เท่านั้นสามารถเรียกเซลล์เป้าหมาย เป็นเซลล์การล็อกอินได้ โดยวิธีการใช้รูปแบบพิเศษของชื่อคำสั่ง bsh : -bsh เมื่อเรียกโดยมีเครื่องหมายยัติภังค์นำหน้า (-) เซลล์จะอ่านและรันคำสั่งที่พบในไฟล์ระบบ /etc/profile ก่อน และ \$HOME/.profile ของคุณ ถ้ามีอยู่ไฟล์ /etc/profile กำหนดตัวแปรที่ต้องการโดยผู้ใช้งานทั้งหมด สุดท้าย เซลล์พร้อมที่จะอ่านคำสั่ง จากอินพุตมาตรฐานของคุณ

หากมีการระบุพารามิเตอร์ ไฟล์ [พารามิเตอร์] เมื่อเซลล์เป้าหมายเริ่มต้นขึ้น เซลล์จะรันไฟล์สคริปต์ ที่ระบุโดยพารามิเตอร์ ไฟล์ รวมถึง พารามิเตอร์ที่ระบุใดๆ ไฟล์สคริปต์ที่ระบุต้องมีสิทธิ์การอ่าน ค่าติดตั้ง setuid และ setgid ถูกละเว้นไป จากนั้น เซลล์อ่านคำสั่ง ถ้าใช้แฟล็ก -c หรือ -s อยู่ระบุสคริปต์

หลักการที่เกี่ยวข้อง:

“เซลล์ที่มีอยู่” ในหน้า 230

ข้อมูลต่อไปนี้เป็นเซลล์ที่จัดเตรียมให้พร้อมไปกับ AIX

สถานะแวดล้อมเซลล์เป้าหมาย:

ตัวแปรทั้งหมด (พร้อมด้วยค่าที่เชื่อมโยง) ซึ่งคำสั่งรู้จัก เมื่อเริ่มต้นการดำเนินการของคำสั่งประกอบกันเป็น สถานะแวดล้อม สถานะแวดล้อมนี้ รวมถึงตัวแปรที่คำสั่งได้รับสืบทอดมาจากโปรเซสพารেন্ট และตัวแปรที่ระบุเป็นพารามิเตอร์คือไอดีเรกทอน บรรทัดคำสั่งซึ่งเรียก คำสั่ง

เซลล์ส่งผ่านตัวแปรที่ระบุชื่อเป็นอาร์กิวเมนต์ของคำสั่ง export ในตัวไปยังโปรเซสชายน คำสั่งนี้วางตัวแปรที่ระบุชื่อไว้ใน สถานะแวดล้อมของทั้งเซลล์และโปรเซสชายนในขนาดคตทั้งหมดของเซลล์นั้น

พารามิเตอร์คีเวิร์ดคือค่าตัวแปรที่ปรากฏขึ้นในรูปแบบของการกำหนด โดยปกติแล้ว อยู่ก่อนหน้าชื่อโพรซีเจอร์บนบรรทัดคำสั่ง (แต่ยังสามารถเห็นได้ในแฟล็กของคำสั่ง set ด้วย) ตัวแปรเหล่านี้มีการวางไว้ในสถานะแวดล้อมของโพรซีเจอร์ที่ถูกเรียกอยู่

โปรดดูตัวอย่างต่อไปนี้:

- พิจารณาโพรซีเจอร์ต่อไปนี้ ซึ่งแสดงค่าของตัวแปรสองตัว (ซึ่งบันทึกไว้ในไฟล์คำสั่งชื่อ key_command):

```
# key_command  
echo $a $b
```

บรรทัดคำสั่งต่อไปนี้จัดทำเอาต์พุตที่แสดง:

Input	Output
a=key1 b=key2 key_command	key1 key2
a=tom b=john key_command	tom john

พารามิเตอร์คีเวิร์ดของโพรซีเจอร์ไม่ได้รวมอยู่ในจำนวนพารามิเตอร์ ที่จัดเก็บไว้ใน \$#

โพรซีเจอร์สามารถเข้าถึงค่าของตัวแปรใดๆ ในสถานะแวดล้อมของตน อย่างไรก็ตาม ถ้าโพรซีเจอร์เปลี่ยนค่าใดๆ เหล่านี้ การเปลี่ยนแปลงไม่มีการสะท้อนให้เห็นในสถานะแวดล้อมเซลล์ การเปลี่ยนแปลงเป็นแบบโลคัลสำหรับโพรซีเจอร์ที่ดำเนินการ หากต้องการวางการเปลี่ยนแปลงในสถานะแวดล้อมที่โพรซีเจอร์ส่งผ่านไปยัง โพรเซสชายน คุณต้องเอ็กซ์พอร์ตค่าใหม่ภายในโพรซีเจอรันนั้น

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการรับรายการของตัวแปรที่เอ็กซ์พอร์ตจากเซลล์ปัจจุบัน ให้พิมพ์ดังต่อไปนี้:

```
export
```

- เมื่อต้องการรับรายการของตัวแปรแบบอ่านอย่างเดียวจากเซลล์ปัจจุบัน ให้พิมพ์ดังต่อไปนี้:

```
readonly
```

- เมื่อต้องการรับรายการของค่าตัวแปรในสถานะแวดล้อมปัจจุบัน ให้พิมพ์ดังต่อไปนี้:

```
env
```

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับสถานะแวดล้อมผู้ใช้ ให้ดูที่ “ไฟล์ /etc/environment” ในหน้า 344

การทดแทนแบบมีเงื่อนไขในเซลล์เป้าหมาย:

โดยปกติ เซลล์แทนที่นิพจน์ $\$Variable$ ด้วย ค่าสตริงที่กำหนดให้กับตัวแปร $Variable$ ถ้ามีอยู่ อย่างไรก็ตาม มีสัญลักษณ์พิเศษที่อนุญาต การทดแทนแบบมีเงื่อนไข ขึ้นอยู่กับว่าตัวแปรมีการตั้งค่า หรือไม่ใช้ null หรือ ทั้งสองอย่าง

โดยนิยาม ตัวแปรมีการตั้งค่าถ้าตัวแปรไม่เคยได้รับการกำหนด ค่า ค่าของตัวแปรสามารถเป็นสตริง null ซึ่งคุณสามารถกำหนดให้กับตัวแปรในวิธีอย่างใดอย่างหนึ่งต่อไปนี้:

ไอเท็ม A= bcd="" Efg='' set '' ''	คำอธิบาย กำหนดสตริง null ให้กับ A, bcd, และ Efg ตั้งค่าพารามิเตอร์ตำแหน่งที่หนึ่งและที่สองเป็นสตริง null และยกเลิกการตั้งค่าพารามิเตอร์ตำแหน่งอื่นทั้งหมด
---	---

ข้อมูลต่อไปนี้เป็นรายการของนิพจน์ที่มีอยู่ซึ่งคุณสามารถใช้เพื่อทำการทดแทนแบบมีเงื่อนไข:

ไอเท็ม	คำอธิบาย
<code>#{Variable-String}</code>	ถ้าตัวแปรมีการตั้งค่าให้ทดแทนค่า <i>Variable</i> ในตำแหน่งของนิพจน์นี้ มิฉะนั้น แทนที่นิพจน์นี้ด้วยค่า <i>String</i>
<code>#{Variable:-String}</code>	ถ้าตัวแปรมีการตั้งค่าและไม่ใช่ null ให้ทดแทนค่า <i>Variable</i> ในตำแหน่งของนิพจน์นี้ มิฉะนั้น แทนที่นิพจน์นี้ด้วยค่า <i>String</i>
<code>#{Variable=String}</code>	ถ้าตัวแปรมีการตั้งค่าให้ทดแทนค่า <i>Variable</i> ในตำแหน่งของนิพจน์นี้ มิฉะนั้น ให้ตั้งค่า <i>Variable</i> เป็นค่า <i>String</i> จากนั้นทดแทนค่า <i>Variable</i> ในตำแหน่งของนิพจน์นี้ คุณไม่สามารถกำหนดค่าให้กับพารามิเตอร์ตำแหน่ง ในลักษณะนี้
<code>#{Variable:=String}</code>	ถ้าตัวแปรมีการตั้งค่าและไม่ใช่ null ให้ทดแทนค่า <i>Variable</i> ในตำแหน่งของนิพจน์นี้ มิฉะนั้น ให้ตั้งค่า <i>Variable</i> เป็นค่า <i>String</i> จากนั้นทดแทนค่า <i>Variable</i> ในตำแหน่งของ นิพจน์นี้ คุณไม่สามารถกำหนดค่าให้กับพารามิเตอร์ตำแหน่งในลักษณะนี้
<code>#{Variable?String}</code>	ถ้าตัวแปรมีการตั้งค่าให้ทดแทนค่า <i>Variable</i> ในตำแหน่งของนิพจน์นี้ มิฉะนั้น แสดงข้อความในรูปแบบต่อไปนี้: Variable: String และออกจากเซลล์ปัจจุบัน (ยกเว้นว่าเซลล์เป็นเซลล์การลือกอื่น) หากคุณไม่ได้ระบุค่าสำหรับตัวแปร <i>String</i> เซลล์จะแสดงข้อความต่อไปนี้: Variable: parameter null or not set
<code>#{Variable?String}</code>	ถ้าตัวแปรมีการตั้งค่าและไม่ใช่ null ให้ทดแทนค่า <i>Variable</i> ในตำแหน่งของนิพจน์นี้ มิฉะนั้น แสดงข้อความในรูปแบบต่อไปนี้: Variable: String และออกจากเซลล์ปัจจุบัน (ยกเว้นว่าเซลล์เป็นเซลล์การลือกอื่น) หากคุณ ไม่ได้ระบุค่า <i>String</i> เซลล์จะแสดงข้อความต่อไปนี้: Variable: parameter null or not set
<code>#{Variable+String}</code>	ถ้าตัวแปรมีการตั้งค่าให้ทดแทนค่า <i>String</i> ในตำแหน่งของนิพจน์นี้ มิฉะนั้น ทดแทนสตริง null
<code>#{Variable:+String}</code>	ถ้าตัวแปรมีการตั้งค่าและไม่ใช่ null ให้ทดแทนค่า <i>String</i> ในตำแหน่งของนิพจน์นี้ มิฉะนั้น ทดแทนสตริง null

ในการทดแทนแบบมีเงื่อนไข เซลล์ไม่ได้ประเมินตัวแปร *String* จนกว่าเซลล์ใช้ตัวแปรนี้เป็นสตริงทดแทน ดังนั้น ในตัวอย่างต่อไปนี่ เซลล์ดำเนินการคำสั่ง `pwd` เฉพาะถ้า `d` ไม่มี การตั้งค่าหรือเป็น null:

```
echo ${d:-`pwd`}
```

หลักการที่เกี่ยวข้อง:

“ตัวแปรผู้ใช้กำหนดในเซลล์เป้าหมาย” ในหน้า 295

เซลล์เป้าหมายรู้จักตัวแปรตัวอักษรและตัวเลขซึ่งสามารถ กำหนดค่าสตริงได้

พารามิเตอร์ตำแหน่งในเชลล์เป้าหมาย:

เมื่อคุณรันเชลล์โพธิ์เตอร์ เชลล์จะสร้างพารามิเตอร์ตำแหน่งที่อ้างอิง แต่ละค่าบนบรรทัดคำสั่งโดยใช้ตำแหน่งของค่านับบนบรรทัดคำสั่งขึ้นโดยปริยาย

ค่าในตำแหน่ง 0 (ชื่อโพธิ์เตอร์) เรียกว่า \$0, ค่าถัดไป (พารามิเตอร์แรก) เรียกว่า \$1, และเป็นเช่นนี้ต่อไปจนถึง \$9 เมื่อต้องการอ้างอิงพารามิเตอร์บรรทัดคำสั่งที่มีหมายเลขสูงกว่า 9 ให้ใช้คำสั่งในตัว shift

คุณสามารถรีเซ็ตค่าของพารามิเตอร์ตำแหน่งได้อย่างชัดเจน โดยใช้คำสั่งในตัว set

หมายเหตุ: ถ้าไม่ได้รับอาร์กิวเมนต์สำหรับตำแหน่ง พารามิเตอร์ตำแหน่ง จะมีการตั้งค่าเป็น null พารามิเตอร์ตำแหน่งเป็นแบบสากลและสามารถส่งผ่าน ไปยังเชลล์โพธิ์เตอร์ที่จัดกลุ่มได้

หลักการที่เกี่ยวข้อง:

“ตัวแปรผู้กำหนดในเชลล์เป้าหมาย” ในหน้า 295

เชลล์เป้าหมายรู้จักตัวแปรตัวอักษรและตัวเลขซึ่งสามารถ กำหนดค่าสตริงได้

สิ่งอ้างอิงที่เกี่ยวข้อง:

“คำพิเศษที่กำหนดไว้ล่วงหน้าในเชลล์เป้าหมาย” ในหน้า 298

ตัวแปรหลายตัวมีความหมายพิเศษ ตัวแปรต่อไปนี้ มีการตั้งค่าโดยเชลล์เป้าหมายเท่านั้น:

การทดแทนชื่อไฟล์ในเชลล์เป้าหมาย:

เชลล์เป้าหมายอนุญาตให้คุณทำการทดแทนชื่อไฟล์ได้

พารามิเตอร์คำสั่งมักจะเป็นชื่อไฟล์ คุณสามารถจัดทำรายการของชื่อไฟล์ เป็นพารามิเตอร์บนบรรทัดคำสั่งได้โดยอัตโนมัติ เมื่อต้องการทำเช่นนี้ ให้ระบุ อักขระที่เชลล์รู้จักว่าเป็นอักขระการจับคู่รูปแบบ เมื่อ คำสั่งมีอักขระดังกล่าว เชลล์จะแทนที่อักขระนั้นด้วยชื่อไฟล์ในไดเรกทอรี

หมายเหตุ: เชลล์เป้าหมายไม่สนับสนุนการขยายชื่อไฟล์ ตามข้อมูลการจัดประเภทเทียบเท่าของอักขระ

อักขระส่วนใหญ่ในรูปแบบดังกล่าวตรงกับตัวเอง แต่คุณยังสามารถใช้ อักขระการจับคู่รูปแบบพิเศษบางตัวในรูปแบบของคุณได้ด้วย อักขระพิเศษเหล่านี้ มีดังนี้:

ไอเท็ม	คำอธิบาย
*	จับคู่สตริงใดๆ รวมถึงสตริง null
?	จับคู่หนึ่งอักขระใดๆ
[. . .]	จับคู่อักขระอย่างใดอย่างหนึ่งใดๆ ที่อยู่ในเครื่องหมายวงเล็บสี่เหลี่ยม
[! . . .]	จับคู่อักขระใดๆ ภายในเครื่องหมายวงเล็บสี่เหลี่ยม <i>ที่ไม่ใช่</i> อักขระ ที่ตามหลังเครื่องหมายอัศเจรีย์

ภายในวงเล็บสี่เหลี่ยม คู่ของอักขระที่คั่นด้วยเครื่องหมายยัติภังค์ (-) ระบุชุดของอักขระทั้งหมดตามลำดับอักษรภายในช่วงเฉพาะ ของคุณั้น ตามการจัดลำดับไบนารีของค่าอักขระ

การจับคู่รูปแบบมีข้อจำกัดบางประการ ถ้าอักขระตัวแรกของชื่อไฟล์คือ จุด (.) แล้ว ระบบสามารถจับคู่ตามรูปแบบที่ขึ้นต้นด้วย จุดเท่านั้น ตัวอย่างเช่น * ตรงกับชื่อไฟล์ *myfile* และ *yourfile* แต่ ไม่ตรงกับชื่อไฟล์ *.myfile* และ *.yourfile* เมื่อต้องการจับคู่ไฟล์เหล่านี้ ให้ใช้รูปแบบดังต่อไปนี้:

```
.*file
```

หากรูปแบบไม่ตรงกับชื่อไฟล์ใดๆ ตัวรูปแบบเองจะถูกส่งคืนเป็นผลลัพธ์ของการจับคู่ที่พยายาม

ชื่อไฟล์และไดเรกทอรีไม่ควรใช้อักขระ *, ?, [, หรือ] เนื่องจากอักขระเหล่านี้อาจส่งผลให้เกิดซ้ำแบบไม่สิ้นสุด (นั่นคือ ลูปไม่สิ้นสุด) ในระหว่างความพยายามจับคู่รูปแบบ

การเปลี่ยนทิศทางอินพุตและเอาต์พุตในเซลล์เป้าหมาย:

มีอ็อปชันการเปลี่ยนทิศทางซึ่งสามารถใช้ใน คำสั่งได้

โดยทั่วไป คำสั่งส่วนใหญ่ไม่ทราบว่าอินพุตหรือ เอาต์พุตของตนเชื่อมโยงกับคีย์บอร์ด หน้าจอแสดงผล หรือไฟล์ ดังนั้น จึงสามารถใช้คำสั่งได้อย่างสะดวกที่คีย์บอร์ด หรือในไฟล์ไลน์

อ็อปชันการเปลี่ยนทิศทางต่อไปนี้สามารถปรากฏขึ้น ที่ใดก็ได้ในคำสั่งแบบง่าย และยังสามารถอยู่ก่อนหน้าหรือตามหลังคำสั่ง แต่ไม่มีการ ส่งผ่านไปยังคำสั่ง

ไอเท็ม	คำอธิบาย
<File	ใช้ไฟล์ที่ระบุเป็นอินพุตมาตรฐาน
>File	ใช้ไฟล์ที่ระบุเป็นเอาต์พุตมาตรฐาน สร้างไฟล์ ถ้าไฟล์ไม่มีอยู่ มิฉะนั้น ตัดไฟล์ให้มีความยาวเป็นศูนย์
>>File	ใช้ไฟล์ที่ระบุเป็นเอาต์พุตมาตรฐาน สร้างไฟล์ ถ้าไฟล์ไม่มีอยู่ มิฉะนั้น เพิ่มเอาต์พุตลงในตอนท้ายของไฟล์
<<[-]eofstr	อ่านบรรทัดทั้งหมดจากตัวแปร eofstr จนถึงบรรทัดที่มี เฉพาะ eofstr หรือจนถึงอักขระตัวสุดท้ายของไฟล์ เป็นอินพุตมาตรฐาน ถ้ามีการอ้างอิงถึงอักขระใดๆ ในตัวแปร eofstr เซลล์จะไม่ขยายหรือตีความอักขระใดๆ ในเส้นอินพุต มิฉะนั้น เซลล์จะทำการทดแทนตัวแปรและคำสั่ง และละเว้นอักขระบรรทัดใหม่ ที่อ้างอิงถึง (\newline) ใช้เครื่องหมายทับขวา (\) เพื่ออ้างอิงถึงอักขระภายในตัวแปร eofstr หรือภายใน เส้นอินพุต ถ้าคุณเพิ่มยัติภังค์ (-) ลงในอ็อปชันการเปลี่ยนทิศทาง << แท้บนำทั้งหมด จะถูกกำหนดแยกจากตัวแปร eofstr และจาก เส้นอินพุต
<&Digit	เชื่อมโยงอินพุตมาตรฐานเข้ากับตัวอธิบายไฟล์ที่ระบุ โดยตัวแปร Digit
>&Digit	เชื่อมโยงเอาต์พุตมาตรฐานเข้ากับตัวอธิบายไฟล์ที่ระบุ โดยตัวแปร Digit
<&-	ปิดอินพุตมาตรฐาน
>&-	ปิดเอาต์พุตมาตรฐาน

หมายเหตุ: เซลล์ที่จำกัดไม่อนุญาตให้เปลี่ยนทิศทางเอาต์พุต

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการเปลี่ยนทิศทาง ให้ดูที่ “การเปลี่ยนทิศทางอินพุตและเอาต์พุต” ในหน้า 375

รายการของคำสั่งในตัวเซลล์เป้าหมาย:

ข้อมูลต่อไปนี้เป็นรายการของคำสั่งในตัวเซลล์เป้าหมาย

ไอเอ็ม	คำอธิบาย
:	ส่งคืนค่าออกเลขศูนย์
.	อ่านและดำเนินการคำสั่งจากพารามิเตอร์ไฟล์ จากนั้นส่งคืน
break	ออกจากลูปคำสั่ง for, while, หรือ until ที่ล้อมอยู่ ถ้ามี
cd	เปลี่ยนไดเรกทอรีปัจจุบันเป็นไดเรกทอรีที่ระบุ
continue	ทำการวนซ้ำถัดไปของลูปคำสั่ง for, while, หรือ until ที่ล้อมอยู่ต่อไป
echo	เขียนสตริงอักขระไปที่เอาต์พุตมาตรฐาน
eval	อ่านอาร์กิวเมนต์เป็นอินพุตของเชลล์ และดำเนินการคำสั่งที่ได้
exec	ดำเนินการคำสั่งที่ระบุโดยพารามิเตอร์ อาร์กิวเมนต์ แทนเชลล์นี้ โดยไม่สร้างโปรเซสใหม่
exit	ออกจากเชลล์ที่เป็นเจ้าของสถานะการออกที่มีการระบุโดยพารามิเตอร์
export	ทำเครื่องหมายชื่อของเอ็กซ์พอร์ตอโต้โนมัติในสภาวะแวดล้อมของ คำสั่งที่ดำเนินการในเวลาต่อมา
hash	ค้นหาและจัดจำที่ตั้งในพาดการค้นหาของคำสั่งที่ระบุ
pwd	แสดงไดเรกทอรีปัจจุบัน
read	อ่านหนึ่งบรรทัดจากอินพุตมาตรฐาน
readonly	ทำเครื่องหมายชื่อที่ระบุโดยพารามิเตอร์ชื่อ เป็นแบบอ่านอย่างเดียว
return	ส่งผลให้ฟังก์ชันออกด้วยค่าส่งคืนที่ระบุ
set	ควบคุมการแสดงของพารามิเตอร์ต่างๆ ในเอาต์พุตมาตรฐาน
shift	เลื่อนอาร์กิวเมนต์บรรทัดคำสั่งไปทางซ้าย
test	ประเมินนิพจน์แบบมีเงื่อนไข
times	แสดงเวลาผู้ใช้และระบบสะสมสำหรับโปรเซสที่รันจาก เชลล์
trap	รันคำสั่งที่ระบุเมื่อเชลล์ได้รับสัญญาณ ที่ระบุ
type	ตีความว่าเชลล์จะตีความชื่อที่ระบุเป็นชื่อคำสั่ง อย่างไร
ulimit	แสดงหรือปรับรีซอร์สเชลล์ที่จัดสรร
umask	พิจารณาสิทธิ์ในไฟล์
unset	ลบตัวแปรหรือฟังก์ชันที่สอดคล้องกับชื่อที่ระบุ
wait	รอให้โปรเซสชาวนที่ระบุสิ้นสุดลง และรายงานสถานะการสิ้นสุดของ โปรเซสนั้น

สิ่งอ้างอิงที่เกี่ยวข้อง:

“คำสั่งในตัวเชลล์เป้าหมาย” ในหน้า 291

คำสั่งพิเศษมีการสร้างขึ้นภายในเชลล์เป้าหมายและรันใน โปรเซสเชลล์

คำสั่งเชลล์เป้าหมาย:

คุณสามารถออกใช้คำสั่งในเชลล์เป้าหมายได้

เมื่อคุณออกใช้คำสั่งในเชลล์เป้าหมาย เชลล์จะประเมินคำสั่งก่อน และทำการทดแทนที่ระบุทั้งหมด จากนั้นรันคำสั่งถ้า:

- ชื่อคำสั่งคือคำสั่งในตัวพิเศษเชลล์เป้าหมาย
- ชื่อคำสั่งตรงกับชื่อของฟังก์ชันที่กำหนดไว้ หากเป็นกรณีนี้ เชลล์จะกำหนดพารามิเตอร์ตำแหน่งให้กับพารามิเตอร์ของฟังก์ชัน

ถ้าชื่อคำสั่งไม่ตรงกับคำสั่งภายในหรือ ชื่อของฟังก์ชันที่นิยามและคำสั่งที่ตั้งชื่อไฟล์เรียกใช้งาน ที่ถูกคอมไพล์ (ไบนารี) โปรแกรม เชลล์ (ในฐานะ พาเรนต) สร้างโปรเซส (ชาวน) ใหม่ที่รันโปรแกรมทันที ถ้าไฟล์มีการทำเครื่องหมายเป็นไฟล์ปฏิบัติการ แต่ไม่ใช่โปรแกรมคอมไพล์ เชลล์จะสมมติว่าเป็นเชลล์โพธิ์เตอร์ ในกรณีนี้ เชลล์ จะสร้างอีกอินสแตนซ์ของตัวเอง (เชลล์ย่อย) เพื่ออ่าน ไฟล์และเรียกใช้คำสั่งที่รวมอยู่ในนั้น เชลล์ยังรัน คำสั่งในวงเล็บในเชลล์ย่อยด้วย สำหรับผู้ใช้ โปรแกรมคอมไพล์ รันในวิธีเดียวกันกับเชลล์โพธิ์เตอร์ทุกประการ โดยปกติแล้ว เชลล์ค้นหา คำสั่งในไดเรกทอรีระบบไฟล์ในลำดับนี้:

1. /usr/bin
2. /etc
3. /usr/sbin

4. /usr/ucb
5. \$HOME/bin
6. /usr/bin/X11
7. /sbin
8. Current directory

เชลล์ค้นหาแต่ละไดเรกทอรีในลักษณะเป็นรอบต่อเนื่องไปยังไดเรกทอรีถัดไป ถ้าไม่พบคำสั่งในไดเรกทอรีปัจจุบัน

หมายเหตุ: ตัวแปร *PATH* กำหนด ลำดับซึ่งเชลล์ค้นหาไดเรกทอรี คุณสามารถเปลี่ยน ลำดับเฉพาะของไดเรกทอรีที่ค้นหาได้ โดยการรีเซ็ตตัวแปร *PATH*

หากคุณกำหนดชื่อพาธเฉพาะเมื่อรันคำสั่ง (ตัวอย่างเช่น /usr/bin/sort) เชลล์จะไม่ค้นหาไดเรกทอรีใดๆ นอกเหนือจากไดเรกทอรีซึ่งคุณระบุ ถ้าชื่อคำสั่งมีเครื่องหมายทับ (/) เชลล์จะไม่ใช้ พาธการค้นหา

คุณสามารถกำหนดชื่อพาธแบบเต็มที่เริ่มต้นด้วยไดเรกทอรีราก (เช่น /usr/bin/sort) คุณยังสามารถระบุชื่อพาธที่สัมพันธ์กับไดเรกทอรีปัจจุบันได้ด้วย หากคุณระบุ ตัวอย่างเช่น:

```
bin/myfile
```

เชลล์ จะค้นหาในไดเรกทอรีปัจจุบันเพื่อหาไดเรกทอรีที่ชื่อ bin และ ค้นหาไฟล์ myfile ในไดเรกทอรี bin

หมายเหตุ: เชลล์ที่จำกัด ไม่รันคำสั่งที่มีเครื่องหมายทับ (/)

เชลล์จดจำที่ตั้งในพาธการค้นหาของแต่ละคำสั่งที่ดำเนินการ (เพื่อหลีกเลี่ยงคำสั่ง exec ที่ไม่จำเป็นในภายหลัง) หากเชลล์พบคำสั่งในไดเรกทอรีสัมพันธ์ (คำสั่งที่ชื่อไม่ได้ขึ้นต้นด้วย /) เชลล์ต้องพิจารณาที่ตั้งของคำสั่งอีกครั้งในทุกครั้งที่ไดเรกทอรีปัจจุบัน เปลี่ยน เชลล์จะลืมที่ตั้งซึ่งจดจำไว้ทั้งหมดในแต่ละครั้งที่คุณเปลี่ยนตัวแปร *PATH* หรือรันคำสั่ง `hash -r`

การอ้างอิงอักขระ:

อักขระจำนวนมากมีความหมายพิเศษในเชลล์ บางครั้ง คุณต้องการปิดบังความหมายนั้น เครื่องอัญประกาศเดี่ยว (') และคู่ (") ที่ล้อมรอบสตริง หรือเครื่องหมายทับขวา (\) ก่อนหน้าอักขระเดี่ยวช่วยให้คุณสามารถปิดบังความหมายของอักขระได้

อักขระทั้งหมด (ยกเว้นที่อยู่ในเครื่องหมายอัญประกาศเดี่ยว) จะถูกใช้ ตามตัวอักษร โดยมีความหมายพิเศษที่ถูกลบออก ดังนั้นคำสั่ง:

```
stuff='echo $? $*; ls * | wc'
```

กำหนด สตริงตามตัวอักษร echo \$? \$*; ls * | wc ให้กับตัวแปร stuff เชลล์ไม่ได้ดำเนินการคำสั่ง echo, ls, และ wc หรือขยายตัวแปร \$? และ \$* และอักขระพิเศษเครื่องหมายดอกจัน (*)

ภายในเครื่องหมายอัญประกาศคู่ ความหมายพิเศษของอักขระเครื่องหมายดอลลาร์ (\$), backquote (`), และอัญประกาศคู่ (") ยังคงมีผลบังคับใช้ ในขณะที่อักขระอื่นทั้งหมดมีการใช้ตามตัวอักษร ดังนั้น ภายในเครื่องหมายอัญประกาศคู่ การทดแทนคำสั่งและตัวแปร จึงเกิดขึ้น นอกจากนี้ เครื่องหมายอัญประกาศไม่ส่งผลกระทบต่อคำสั่งภายใน การทดแทนคำสั่งซึ่งเป็นส่วนหนึ่งของสตริงที่อยู่ในอัญประกาศ ดังนั้นอักขระที่นั้น จึงรักษาความหมายพิเศษของตนไว้

พิจารณาลำดับต่อไปนี้:

```
ls *
file1 file2 file3
message="This directory contains `ls * ` "
echo $message
This directory contains file1 file2 file3
```

คำสั่งนี้แสดงว่าอักขระพิเศษเครื่องหมายดอกจัน (*) ภายในการทดแทนคำสั่งถูก ขยาย

เมื่อต้องการซ่อนความหมายพิเศษของอักขระเครื่องหมายดอลลาร์ (\$), backquote (`), และอัญประกาศคู่ (") ภายใต้อัญประกาศคู่ ให้ใส่เครื่องหมายทับขวา (\) ไว้ข้างหน้าอักขระเหล่านี้ เมื่อคุณไม่ได้ใช้อัญประกาศคู่ การใส่เครื่องหมายทับขวาไว้ข้างหน้าอักขระมีผลเทียบเท่ากับการวางอักขระไว้ภายใน เครื่องหมายอัญประกาศเดี่ยว ดังนั้น เครื่องหมายทับขวาที่อยู่ข้างหน้าติดกับอักขระบรรทัดใหม่ (นั่นคือ เครื่องหมายทับขวาที่ตอนท้ายของบรรทัด) จะซ่อนอักขระบรรทัดใหม่ และคุณสามารถทำบรรทัดคำสั่งบนบรรทัดฟิลิคัลถัดไปได้

การจัดการสัญญาณ:

เซลล์เว้นสัญญาณ INTERRUPT และ QUIT สำหรับ คำสั่งที่เรียกใช้ถ้าคำสั่งสิ้นสุดลงด้วยเครื่องหมาย ampersand (&) นั่นคือถ้ากำลังรันในพื้นที่หลัง มิฉะนั้น สัญญาณมีค่าที่เซลล์ได้รับสืบทอด มาจากพาเรนต์ โดยมีข้อยกเว้นของสัญญาณ

SEGMENTATION VIOLATION

สำหรับข้อมูลเพิ่มเติม ให้ดูที่คำสั่ง trap ในตัวเซลล์เป้าหมาย

คำสั่งผสมเซลล์เป้าหมาย:

คำสั่งผสมเป็นอย่างใดอย่างหนึ่งต่อไปนี้

- ไฟล์ไลน์ (คำสั่งแบบง่ายตั้งแต่หนึ่งคำสั่งขึ้นไปที่ค้นด้วยสัญลักษณ์ไพพ์ (|))
- รายการของคำสั่งแบบง่าย
- คำสั่งที่ขึ้นต้นด้วยคำที่สงวนไว้
- คำสั่งที่ขึ้นต้นด้วยตัวดำเนินการควบคุมเครื่องหมายวงเล็บซ้าย (())

ยกเว้นว่ามีกระบวนการเป็นอย่างอื่น คำที่ส่งคืนโดยคำสั่งผสม คือค่าของคำสั่งแบบง่ายล่าสุดที่ดำเนินการ

คำที่สงวนไว้:

คำที่สงวนไว้ต่อไปนี้สำหรับเซลล์เป้าหมายมีการรับรู้เฉพาะถ้า คำดังกล่าวปรากฏขึ้นโดยไม่มีเครื่องหมายอัญประกาศเป็นคำแรกของคำสั่ง

```
for          do          done
case        esac
if          then        fi
elif       else
while      until
{          }
(          )
```

ไวยากรณ์	คำอธิบาย
<code>for Identifier [in Word. . .] do List done</code>	ตั้งค่าพารามิเตอร์ <i>Identifier</i> เป็นค่าที่ระบุโดยพารามิเตอร์ <i>Word</i> (ทีละหนึ่งคำ) และรันคำสั่งที่ระบุในพารามิเตอร์ <i>List</i> ถ้าคุณละเว้น <code>in Word. . .</code> ดังนั้น คำสั่ง <code>for</code> จะรันพารามิเตอร์ <i>List</i> สำหรับแต่ละพารามิเตอร์ตำแหน่งที่ตั้งค่าไว้ และการประมวลผลสิ้นสุดลง เมื่อใช้พารามิเตอร์ตำแหน่งแล้ว
<code>case Word in Pattern [Pattern] . . .) List;; [Pattern [Pattern] . . .) List;;]... esac</code>	รันคำสั่งที่ระบุในพารามิเตอร์ <i>List</i> ซึ่งเชื่อมโยงกับพารามิเตอร์ <i>Pattern</i> แรกที่ ตรงกับค่าของพารามิเตอร์ <i>Word</i> ใช้สัญลักษณ์การจับคู่อักขระ เดียวกันในรูปแบบที่ใช้สำหรับการทดแทนชื่อไฟล์ ยกเว้นว่าเครื่องหมายสแลช (/), จุดนำ (.) หรือจุดที่ตามหลังเครื่องหมายสแลช (/.) ไม่จำเป็นต้องตรงกัน
<code>if List then List [elif List then List] . . . [else List] fi</code>	รันคำสั่งที่ระบุในพารามิเตอร์ <i>List</i> ตามหลังคำสั่ง <code>if</code> ถ้าคำสั่งส่งคืนค่าออกเป็นศูนย์ เซลล์จะรันพารามิเตอร์ <i>List</i> ตามหลัง คำสั่ง <code>then</code> แรก มิฉะนั้น จะรันพารามิเตอร์ <i>List</i> ตามหลังคำสั่ง <code>elif</code> (ถ้ามีอยู่) ถ้าออกนี้เป็นศูนย์ เซลล์จะรันพารามิเตอร์ <i>List</i> ตามหลังคำสั่ง <code>then</code> ถัดไป ถ้าคำสั่งส่งคืนค่าออกที่ไม่ใช่ศูนย์ เซลล์จะรันพารามิเตอร์ <i>List</i> ตามหลัง คำสั่ง <code>else</code> (ถ้ามีอยู่) ถ้าไม่มีการทำ <code>else List</code> หรือ <code>then List</code> คำสั่ง <code>if</code> จะส่งคืน ค่าออกเป็นศูนย์
<code>while List do List done</code>	รันคำสั่งที่ระบุในพารามิเตอร์ <i>List</i> ตามหลังคำสั่ง <code>while</code> ถ้าค่าออกของคำสั่งหลังสุด ใน <code>while List</code> เป็นศูนย์ เซลล์จะรันพารามิเตอร์ <i>List</i> ตามหลังคำสั่ง <code>do</code> และวนซ้ำในรายการต่อไปจนกว่าค่าออกของคำสั่งหลังสุด ใน <code>while List</code> ไม่ใช่ศูนย์ หากไม่มีการทำ คำสั่งใน <code>do List</code> คำสั่ง <code>while</code> จะส่งคืนค่าออกเป็นศูนย์
<code>until List do List done</code>	รันคำสั่งที่ระบุในพารามิเตอร์ <i>List</i> ตามหลังคำสั่ง <code>until</code> ถ้าค่าออกของคำสั่งหลังสุด ใน <code>until List</code> ไม่ใช่ศูนย์ เซลล์จะรันพารามิเตอร์ <i>List</i> ตามหลังคำสั่ง <code>do</code> และวนซ้ำในรายการต่อไปจนกว่าค่าออกของคำสั่งหลังสุด ใน <code>until List</code> เป็นศูนย์ หากไม่มีการทำ คำสั่งใน <code>do List</code> คำสั่ง <code>until</code> จะส่งคืนค่าออกเป็นศูนย์
<code>(List)</code>	รันคำสั่งในพารามิเตอร์ <i>List</i> ในเซลล์ย่อย
<code>{ List; }</code>	รันคำสั่งในพารามิเตอร์ <i>List</i> ในโปรเซสเซลล์ปัจจุบันและไม่ได้เริ่มต้นเซลล์ย่อย
<code>Name () { List }</code>	กำหนดฟังก์ชันที่มีการอ้างอิงโดยพารามิเตอร์ <i>Name</i> เนื้อหาของฟังก์ชันคือรายการของคำสั่งระหว่างวงเล็บปีกกาที่ระบุโดยพารามิเตอร์ <i>List</i>

คำสั่งในตัวเซลล์เป้าหมาย:

คำสั่งพิเศษมีการสร้างขึ้นภายในเซลล์เป้าหมายและรันใน โปรเซสเซลล์

ยกเว้นว่ามีการระบุเป็นอย่างอื่น เอาต์พุตจะถูกเขียนไปยังตัวอธิบายไฟล์ 1 (เอาต์พุตมาตรฐาน) และสถานะการออกคือ 0 (ศูนย์) ถ้าคำสั่งไม่มี ข้อผิดพลาดไวยากรณ์ใดๆ อนุญาตให้เปลี่ยนทิศทางการอินพุตและเอาต์พุตได้

คำสั่งพิเศษต่อไปนี้มีการจัดการแตกต่างจาก คำสั่งในตัวพิเศษอื่น:

```
: (colon)      exec          shift
. (dot)       exit          times
break         export       trap
continue     readonly     wait
eval         return
```

เซลล์เป้าหมายประมวลผลคำสั่งเหล่านี้ดังนี้:

- รายการกำหนดพารามิเตอร์คือเวิร์ดที่อยู่ข้างหน้าคำสั่งยังคงมีผลบังคับใช้ เมื่อคำสั่งเสร็จสมบูรณ์
- การเปลี่ยนทิศทาง I/O มีการประมวลผลหลังจากการกำหนดพารามิเตอร์
- ข้อผิดพลาดในเซลล์สคริปต์ส่งผลให้สคริปต์หยุดการประมวลผล

สิ่งอ้างอิงที่เกี่ยวข้อง:

“รายการของคำสั่งในตัวเซลล์เป้าหมาย” ในหน้า 287

ข้อมูลต่อไปนี้เป็นรายการของคำสั่งในตัวเซลล์เป้าหมาย

คำอธิบายคำสั่งพิเศษ:

เชลล์เป้าหมายแสดงคำสั่งในตัวพิเศษต่อไปนี้

ไอเท็ม	คำอธิบาย
:	ส่งคืนค่าออกเลขศูนย์
.File	อ่านและรันคำสั่งจากพารามิเตอร์ <i>File</i> และส่งคืน ไม่ได้เริ่มต้นเชลล์ย่อย เชลล์ใช้พาทการณ์หาที่ระบุโดย ตัวแปร <i>PATH</i> เพื่อค้นหาไดเรกทอรีซึ่งมีไฟล์ที่ระบุ
break [<i>n</i>]	ออกจากลูปคำสั่ง for , while , หรือ until ที่ล้อมอยู่ ถ้ามี หากคุณระบุตัวแปร <i>n</i> คำสั่ง break จะแบ่งตัวเลขของระดับที่ระบุโดยตัวแปร <i>n</i>
continue [<i>n</i>]	ทำการวนซ้ำถัดไปของลูปคำสั่ง for , while , หรือ until ที่ล้อมอยู่ต่อไป หากคุณระบุตัวแปร <i>n</i> คำสั่งจะดำเนินการต่อที่ลูป <i>n</i> ^h ที่ล้อมอยู่
cd <i>Directory</i>]	เปลี่ยนไดเรกทอรีปัจจุบันเป็น <i>Directory</i> ถ้าคุณไม่ได้ระบุ <i>Directory</i> จะใช้ค่าของตัวแปรเชลล์ <i>HOME</i> ตัวแปรเชลล์ <i>CDPATH</i> กำหนด พาทการณ์หาสำหรับ <i>Directory</i> <i>CDPATH</i> คือ รายการที่ค้นด้วยเครื่องหมายจุดของชื่อไดเรกทอรีอื่น ชื่อพารามิเตอร์ <i>cd</i> ระบุ ไดเรกทอรีปัจจุบัน (ซึ่งเป็นพาทดีฟอลต์) ชื่อพารามิเตอร์ <i>cd</i> นี้ปรากฏขึ้น ทันทีหลังจากเครื่องหมายเท่ากับในการกำหนด หรือระหว่างตัวค้นจุดจุดที่อื่นใดในรายการพาร ถ้า <i>Directory</i> ขึ้นต้นด้วย เครื่องหมายทับ (/) เชลล์จะไม่ใช้พาทการณ์หา มิฉะนั้น เชลล์จะค้นหาแต่ละไดเรกทอรีในตัวแปรเชลล์ <i>CDPATH</i> หมายเหตุ: เชลล์ที่จำกัด ไม่สามารถรันคำสั่งเชลล์ cd
echo <i>String</i> . . .]	เขียนสตริงอักขระไปที่เอาต์พุตมาตรฐาน โปรดดูที่คำสั่ง echo สำหรับข้อมูลการใช้งานและพารามิเตอร์ ไม่สนับสนุนแฟล็ก -n
eval [<i>Argument</i> . . .]	อ่านอาร์กิวเมนต์เป็นอินพุตของเชลล์ และ รันคำสั่งที่ได้
exec [<i>Argument</i> . . .]	รันคำสั่งที่ระบุโดยพารามิเตอร์ <i>exec</i> แทนเชลล์นี้ โดยไม่สร้างโปรเซสใหม่ อินพุตและเอาต์พุตอาร์กิวเมนต์ สามารถปรากฏขึ้น และถ้าไม่มีอาร์กิวเมนต์อื่นปรากฏขึ้น ส่งผลให้เชลล์อินพุต หรือเอาต์พุตจะถูกแก้ไข ซึ่งไม่แนะนำสำหรับเชลล์การล็อกอินของคุณ
exit [<i>n</i>]	ส่งผลให้เชลล์ออกด้วยค่าออกที่ระบุโดย พารามิเตอร์ <i>n</i> หากคุณข้ามพารามิเตอร์นี้ ค่าออกคือ ค่าของคำสั่งล่าสุดที่ดำเนินการ (ปุ่ม Ctrl-D ตามลำดับส่งผลให้ เชลล์ออกเช่นกัน) ค่าของพารามิเตอร์ <i>n</i> สามารถเป็นตั้งแต่ 0 ถึง 255 แบบ inclusive
export [<i>Name</i> . . .]	ทำเครื่องหมายชื่อที่ระบุของเอ็กซ์พอร์ตอัตโนมัติในสภาวะแวดล้อมของ คำสั่งที่ดำเนินการในเวลาต่อมา ถ้าคุณไม่ได้ระบุพารามิเตอร์ <i>Name</i> คำสั่ง export จะแสดงรายการของชื่อทั้งหมดที่ถูกเอ็กซ์พอร์ตในเชลล์นี้ คุณไม่สามารถเอ็กซ์พอร์ต ชื่อฟังก์ชัน
hash [-r][<i>Command</i> . . .]	ค้นหาและจดจำที่ตั้งในพาทการณ์หา ของแต่ละ <i>Command</i> ที่ระบุ แฟล็ก -r ส่งผลให้ เชลล์ลิมิตที่ตั้งทั้งหมด หากคุณไม่ได้ระบุแฟล็กหรือคำสั่งใดๆ เชลล์จะแสดงข้อมูลเกี่ยวกับคำสั่งที่จดจำไว้ในรูปแบบ ต่อไปนี้: Hits Cost <i>Command</i> Hits บ่งชี้จำนวนครั้งที่รันคำสั่ง โดยโปรเซสเชลล์ Cost คือการประเมินงานที่ต้องใช้ เพื่อระบุตำแหน่งคำสั่งในพาทการณ์หา <i>Command</i> แสดง ชื่อพารามิเตอร์ของแต่ละคำสั่งที่ระบุ บางสถานการณ์จำเป็นต้องคำนวณที่ตั้งซึ่งจัดเก็บไว้ของคำสั่งอีกครั้ง ตัวอย่างเช่น ที่ตั้งของชื่อพารามิเตอร์ที่สัมพันธ์เมื่อไดเรกทอรีปัจจุบันเปลี่ยน คำสั่งซึ่งอาจจะทำ มีการบ่งชี้โดยเครื่องหมายดอกจัน (*) ถัดจาก ข้อมูล Hits Cost เพิ่มขึ้น เมื่อทำการคำนวณใหม่
pwd	แสดงไดเรกทอรีปัจจุบัน โปรดดูที่คำสั่ง pwd สำหรับคำอธิบายอ็อปชันคำสั่ง
read [<i>Name</i> . . .]	อ่านหนึ่งบรรทัดจากอินพุตมาตรฐาน กำหนดค่าแรก ในบรรทัดเป็นพารามิเตอร์ <i>Name</i> แรก คำที่สองเป็น พารามิเตอร์ <i>Name</i> ที่สอง และต่อไปโดยมีค่าที่เหลือ ซึ่งกำหนดให้กับพารามิเตอร์ <i>Name</i> สุดท้าย คำสั่งนี้ ส่งคืนค่าเป็น 0 ยกเว้นว่าพบอักขระสิ้นสุดไฟล์
readonly [<i>Name</i> . . .]	ทำเครื่องหมายชื่อที่ระบุโดยพารามิเตอร์ <i>Name</i> เป็นแบบอ่านอย่างเดียว ค่าของชื่อไม่สามารถรีเซ็ตได้ ถ้าคุณไม่ได้ระบุ <i>Name</i> คำสั่ง readonly จะแสดง รายการของชื่อแบบอ่านอย่างเดียวทั้งหมด
return [<i>n</i>]	ส่งผลให้ฟังก์ชันออกด้วยค่าส่งคืนเป็น <i>n</i> ถ้าคุณไม่ได้ระบุตัวแปร <i>n</i> ฟังก์ชันจะส่งคืน สถานะของคำสั่งล่าสุดที่ทำในฟังก์ชันนั้น คำสั่งนี้ ถูกต้องเฉพาะถ้ารันภายในฟังก์ชันเชลล์เท่านั้น

ไอเท็ม	คำอธิบาย
<pre>set [Flag [Argument] . . .]</pre>	<p>ตั้งค่าแฟล็กต่อไปนี้ตั้งแต่หนึ่งรายการขึ้นไป:</p> <ul style="list-style-type: none"> -a ทำเครื่องหมายเอ็กซ์พอร์ตตัวแปรทั้งหมดที่ถูกดำเนินการกำหนดค่า หากการกำหนดค่านำหน้าชื่อคำสั่งแอ็ททริบิวต์เอ็กซ์พอร์ตจะใช้ได้ เฉพาะสำหรับสภาวะแวดล้อมการดำเนินการของคำสั่งนั้น ยกเว้นเมื่อการกำหนดค่านำหน้าค่าหนึ่งในคำสั่งในตัวพิเศษ ในกรณีนี้ แอ็ททริบิวต์เอ็กซ์พอร์ตมีอยู่หลังจากคำสั่งในตัวเสร็จสมบูรณ์แล้ว หากการกำหนดค่าไม่ได้มาก่อนหน้าชื่อคำสั่ง หรือหากการกำหนดค่าเป็นผลลัพธ์ของการดำเนินงาน ของคำสั่ง <code>getopts</code> หรือ <code>read</code> แอ็ททริบิวต์เอ็กซ์พอร์ตจะยังคงอยู่จนกระทั่งตัวแปรไม่ถูกตั้งค่า -e ออกจากการทำงานโดยทันทีหากมีเงื่อนไขทั้งหมดต่อไปนี้เกิดขึ้นกับ คำสั่ง: <ul style="list-style-type: none"> • ออกด้วยค่าส่งคืนมากกว่า 0 (ศูนย์) • ไม่ได้เป็นส่วนหนึ่งของรายการผสมของคำสั่ง <code>while</code>, <code>until</code> หรือ <code>if</code> • ไม่ได้ถูกทดสอบการใช้รายการ <code>AND</code> หรือ <code>OR</code> • ไม่ได้เป็นโฟฟไลน์ที่นำหน้าด้วย! (เครื่องหมาย อัศเจรีย์) คำสวางน -f ปิดใช้การแทนชื่อไฟล์ -h ค้นหาและจดจำคำสั่งที่เรียกใช้ภายในฟังก์ชันตามที่ ฟังก์ชันถูกกำหนด (โดยปกติ คำสั่งเหล่านี้จะมีการระบุตำแหน่งเมื่อทำฟังก์ชัน ให้ดูที่คำสั่ง <code>hash</code>) -k วางคีย์เวิร์ดพารามิเตอร์ทั้งหมดในสภาวะแวดล้อมสำหรับคำสั่ง ไม่ใช่เฉพาะ คีย์เวิร์ดที่นำหน้าชื่อคำสั่ง -n อ่านคำสั่งแต่ไม่ได้อ่านคำสั่ง เมื่อต้องการตรวจสอบข้อผิดพลาดไวยากรณ์ เซลล์สคริปต์ให้ใช้แฟล็ก <code>-n</code> -t ออกจากการอ่านและเรียกใช้งานหนึ่งคำสั่ง -u ปฏิบัติต่อตัวแปรที่ไม่ได้ตั้งค่าเป็นข้อผิดพลาด และออกโดยทันทีเมื่อดำเนินการ แทนค่าตัวแปร เซลล์แบบโต้ตอบจะไม่ออกจากการทำงาน -v แสดงบรรทัดอินพุตของเซลล์ขณะที่ถูกอ่าน -x แสดงคำสั่งและอาร์กิวเมนต์ของคำสั่งก่อนจะรัน - ไม่เปลี่ยนแปลงใดๆ มีประโยชน์ในการตั้งค่าพารามิเตอร์ตำแหน่ง \$1 เป็นสตริงที่ขึ้นต้นด้วยยัติภังค์ (-) <p>การใช้เครื่องหมายบวก (+) แทน ยัติภังค์ (-) จะยกเลิกการตั้งค่าแฟล็ก คุณยังสามารถระบุแฟล็กเหล่านี้ บนบรรทัดคำสั่ง เซลล์ได้ด้วย ตัวแปรพิเศษ <code>\$-</code> มีการตั้งค่าปัจจุบันของแฟล็ก</p> <p><i>อาร์กิวเมนต์</i> ใดๆ ของคำสั่ง <code>set</code> กลายเป็นพารามิเตอร์ตำแหน่งและมีการกำหนดตามลำดับเป็น \$1, \$2, . . . , และต่อไป หากคุณไม่ได้รับ <code>แฟล็ก</code> หรือ <i>อาร์กิวเมนต์</i> คำสั่ง <code>set</code> จะแสดงชื่อทั้งหมดและค่าของตัวแปรเซลล์ปัจจุบัน</p>
<pre>shift [n]</pre>	<p>เลื่อนอาร์กิวเมนต์บรรทัดคำสั่งไปทางซ้าย นั่นคือกำหนดค่าของ พารามิเตอร์ตำแหน่งอีกครั้งโดยที่ค่าปัจจุบันของ \$1 และกำหนดค่าของ \$2 เป็น \$1, ค่าของ \$3 เป็น \$2, และต่อไป หากมีอาร์กิวเมนต์บรรทัดคำสั่งมากกว่า 9 รายการ ค่า 10th จะถูกกำหนดเป็น \$9 และค่าส่วนที่เหลือจะยังคงไม่มีการกำหนด (จนกว่าจะมี <code>shift</code> อีกครั้ง) หากมี 9 อาร์กิวเมนต์หรือน้อยกว่านั้น คำสั่ง <code>shift</code> จะยกเลิก การตั้งค่าพารามิเตอร์ตำแหน่งที่มีหมายเลขสูงสุดซึ่งมีค่า</p> <p>พารามิเตอร์ตำแหน่ง \$0 ไม่เคยมีการเลื่อน คำสั่ง <code>shift n</code> คือสัญกรณ์ย่อที่ระบุหมายเลข <i>n</i> ของการเลื่อน ที่ต่อเนื่อง ค่าตีฟอลต์ของพารามิเตอร์ <i>n</i> คือ 1</p>
<pre>test Expression [Expression]</pre>	<p>ประเมินนิพจน์แบบมีเงื่อนไข โปรดดูที่คำสั่ง <code>test</code> สำหรับคำอธิบายแฟล็กคำสั่งและพารามิเตอร์ แฟล็ก <code>-h</code> ไม่ได้รับการสนับสนุน โดยคำสั่ง <code>test</code> ในตัวใน <code>bsh</code></p>
<pre>times</pre>	<p>แสดงเวลาผู้ใช้และระบบสะสมสำหรับโปรเซสที่รันจาก เซลล์</p>

ไอเท็ม	คำอธิบาย
<code>trap [Command] [n]</code>]. . .	รันคำสั่งที่ระบุโดยพารามิเตอร์ <i>Command</i> เมื่อเซลล์ได้รับสัญญาณที่ระบุโดยพารามิเตอร์ <i>n</i> คำสั่ง <code>trap</code> รันในลำดับของหมายเลขสัญญาณ ความพยายามใดๆ ในการตั้งค่า <code>trap</code> บนสัญญาณที่ถูกละเว้นบนรายการของเซลล์ ปัจจุบันจะไม่เป็นผล หมายเหตุ: เซลล์สแกนพารามิเตอร์ <i>Command</i> หนึ่งครั้ง เมื่อตั้งค่า <code>trap</code> และอีกครั้งเมื่อใช้ <code>trap</code> ถ้าคุณไม่ได้ ระบุคำสั่ง ผลคือ <code>traps</code> ทั้งหมดที่ระบุโดยพารามิเตอร์ <i>n</i> จะถูกรีเซ็ตเป็นค่าปัจจุบัน ถ้าคุณระบุสตริง <code>null</code> สัญญาณนี้ จะถูกละเว้นโดยเซลล์และโดยคำสั่งต่างๆ ที่เรียกใช้ ถ้าพารามิเตอร์ <i>n</i> เป็นศูนย์ (0) คำสั่งที่ระบุจะรันเมื่อคุณออกจากเซลล์ ถ้า คุณไม่ได้ระบุคำสั่งหรือสัญญาณ คำสั่ง <code>trap</code> จะแสดงรายการของคำสั่งที่เชื่อมโยงกับแต่ละหมายเลขสัญญาณ
<code>type [Name . . .]</code>	บ่งชี้ว่าเซลล์จะตีความ เป็นชื่อคำสั่งสำหรับแต่ละ <i>Name</i> ที่ระบุอย่างไร
<code>ulimit [-HS] [-c -d -f -m -r -s -t -u] [limit]</code>	แสดงหรือปรับรีเซ็ตเซลล์ที่จัดสรร ค่าติดตั้งรีเซ็ตเซลล์สามารถแสดงทีละค่าหรือเป็นกลุ่ม โหมดตีพอลต์คือแสดงรีเซ็ตที่ตั้งค่าเป็นค่าติดตั้ง <code>soft</code> , หรือ <code>lower bound</code> เป็นกลุ่ม ค่าติดตั้งของรีเซ็ตเซลล์ขึ้นอยู่กับ ID ผู้ใช้ ที่ใช้ของเซลล์ปัจจุบัน ระดับ <code>hard</code> ของรีเซ็ตสามารถตั้งค่าได้เฉพาะถ้า ID ผู้ใช้ ที่ใช้ของเซลล์ปัจจุบันไม่ใช่ราก คุณจะได้รับข้อผิดพลาด ถ้าคุณไม่ใช่ผู้ใช้รากและคุณกำลังพยายามตั้งค่าระดับ <code>hard</code> ของรีเซ็ต โดยค่าตีพอลต์ ผู้ใช้รากตั้งค่าทั้งขีดจำกัด <code>hard</code> และ <code>soft</code> ของ รีเซ็ตเฉพาะ ดังนั้น ผู้ใช้รากจึงควรระมัดระวังในการใช้ <code>-S</code> , <code>-H</code> , หรือการใช้แฟล็กตีพอลต์ของค่าติดตั้งขีดจำกัด ยกเว้นว่า คุณเป็นผู้ใช้ราก คุณสามารถตั้งค่าได้เฉพาะขีดจำกัด <code>soft</code> ของรีเซ็ตเท่านั้น หลังจาก ขีดจำกัดมีจำนวนลดลงเนื่องจากผู้ใช้ที่ไม่ใช่ผู้ใช้ราก ไม่สามารถเพิ่มขีดจำกัดได้ แม้แต่กลับไปยังขีดจำกัดระบบดั้งเดิม เมื่อต้องการตั้งค่าขีดจำกัดรีเซ็ต ให้เลือก แฟล็กที่เหมาะสมและค่าจำกัดของรีเซ็ตใหม่ ซึ่งควรจะเป็น เลขจำนวนเต็ม คุณสามารถตั้งค่าขีดจำกัดรีเซ็ตได้ครั้งละหนึ่งขีดจำกัดเท่านั้น หากมีการระบุ แฟล็กรีเซ็ตมากกว่าหนึ่งแฟล็ก คุณจะได้รับผลลัพธ์ที่ไม่ได้กำหนด โดยค่าตีพอลต์ <code>ulimit</code> ที่มี เฉพาะค่าใหม่บนบรรทัดคำสั่งจะตั้งค่าขนาดไฟล์ของเซลล์ การใช้แฟล็ก <code>-f</code> เป็นทางเลือก คุณสามารถระบุแฟล็กคำสั่ง <code>ulimit</code> ต่อไปนี้:
<code>-c</code>	ตั้งค่าหรือแสดงเช็คเมนต์หลักของเซลล์
<code>-d</code>	ตั้งค่าหรือแสดงเช็คเมนต์ข้อมูลของเซลล์
<code>-f</code>	ตั้งค่าหรือแสดงขนาดไฟล์ของเซลล์
<code>-H</code>	ตั้งค่าหรือแสดงขีดจำกัด <code>hard</code> รีเซ็ต (ผู้ใช้รากเท่านั้น)
<code>-m</code>	ตั้งค่าหรือแสดงหน่วยความจำของเซลล์
<code>-r</code>	ตั้งค่าหรือแสดงจำนวนเรตสูงสุดต่อโปรเซส
<code>-s</code>	ตั้งค่าหรือแสดงสแต็กเช็คเมนต์ของเซลล์
<code>-S</code>	ตั้งค่าหรือแสดงขีดจำกัด <code>soft</code> รีเซ็ต
<code>-t</code>	ตั้งค่าหรือแสดงเวลา CPU สูงสุดของเซลล์
<code>-u</code>	ตั้งค่าหรือแสดงจำนวนโปรเซสสูงสุดต่อผู้ใช้
<code>umask [nnn]</code>	พิจารณาลิทธิในไฟล์ คำนี้ พร้อมกับสิทธิของการสร้างกระบวนการ จะใช้พิจารณาลิทธิของไฟล์เมื่อไฟล์ ถูกสร้างขึ้น ค่าตีพอลต์คือ 022 ถ้าไม่ได้ป้อนค่า <code>umask</code> จะแสดง ค่าปัจจุบัน
<code>unset [Name . . .]</code>	ลบตัวแปรหรือฟังก์ชันที่สอดคล้องกันสำหรับแต่ละชื่อ ที่ระบุโดยพารามิเตอร์ <i>Name</i> ตัวแปรเซลล์ <i>PATH</i> , <i>PS1</i> , <i>PS2</i> , <i>MAILCHECK</i> , และ <i>IFS</i> ไม่สามารถยกเลิกการตั้งค่าได้
<code>wait [n]</code>	รอโปรเซสชาวยิ่งซึ่งเป็นเจ้าของหมายเลขโปรเซสที่ระบุโดย พารามิเตอร์ <i>n</i> เพื่อออก จากนั้นส่งคืนสถานะการออก ของโปรเซสนั้น ถ้าคุณไม่ได้ระบุพารามิเตอร์ <i>n</i> เซลล์จะรอโปรเซสชาวยิ่งทั้งหมดที่ใช้งานอยู่ในปัจจุบัน และค่าส่งคืน คือ 0

การทดแทนคำสั่งในเซลล์เป้าหมาย:

การทดแทนคำสั่งช่วยให้คุณสามารถดักจับเอาต์พุตของคำสั่งใดๆ เป็นอาร์กิวเมนต์ไปยังคำสั่งอื่น

เมื่อคุณใส่บรรทัดคำสั่งไว้ในเครื่องหมาย backquotes (``) เซลล์จะรันคำสั่งก่อน จากนั้นแทนที่ทั้งนิพจน์ รวมถึง backquotes ด้วยเอาต์พุต คุณลักษณะนี้มักใช้เพื่อให้ค่าแก้ตัวแปรเซลล์ ตัวอย่างเช่น คำสั่ง:

```
today=`date`
```

กำหนด สตริงที่แสดงถึงวันที่ปัจจุบันให้กับตัวแปร *today* การกำหนดต่อไปนี้บันทึกจำนวนไฟล์ในไดเรกทอรีปัจจุบันไว้ในตัวแปร *files*:

```
files=`ls | wc -l`
```

คุณสามารถทำการทดแทนคำสั่งบนคำสั่งใดๆ ที่จะเขียน ไปยังเอาต์พุตมาตรฐาน

เมื่อต้องการจัดกลุ่มการทดแทนคำสั่ง ให้ใส่เครื่องหมายทับขวา (\) นำหน้าแต่ละ backquotes ที่จัดกลุ่ม ดังเช่นใน:

```
logmsg=`echo Your login directory is \`pwd\``
```

คุณยังสามารถกำหนดค่าให้กับตัวแปรเซลล์ทางอ้อมได้โดยใช้คำสั่งพิเศษ *read* คำสั่งนี้ใช้บรรทัดจากอินพุตมาตรฐาน (โดยปกติ คีย์บอร์ดของคุณ) และกำหนดค่าที่ต่อเนื่องบนบรรทัดนั้นให้กับตัวแปรที่ระบุใดๆ ตัวอย่างเช่น:

```
read first init last
```

ใช้ เส้นอินพุตของรูปแบบ:

```
J. Q. Public
```

และมีผล เหมือนกับคุณพิมพ์:

```
first=J. init=Q. last=Public
```

คำสั่งพิเศษ *read* กำหนดค่าเพิ่มเติมใดๆ ให้กับ ตัวแปรล่าสุด

การทดแทนตัวแปรในเซลล์เป้าหมาย:

เซลล์เป้าหมายอนุญาตให้คุณทำการทดแทนตัวแปรได้

เซลล์เป้าหมายมีหลายกลไกสำหรับการสร้างตัวแปร (การกำหนดค่าสตริงให้กับชื่อ) บางตัวแปร พารามิเตอร์ตำแหน่ง และ พารามิเตอร์คีย์เวิร์ด โดยปกติแล้ว มีการตั้งค่าบนบรรทัดคำสั่งเท่านั้น ตัวแปรอื่น เป็นชื่อแบบง่ายซึ่งคุณหรือเซลล์สามารถ กำหนดค่าสตริงได้

หลักการที่เกี่ยวข้อง:

“เทอร์มินัลที่ไม่ได้ใส่ใจ” ในหน้า 322

ระบบทุกระบบอาจได้รับอันตรายได้ถ้าเทอร์มินัลมีการล็อกอินทิ้งไว้ และไม่ได้ใส่ใจ ปัญหาร้ายแรงที่สุดเกิดขึ้นเมื่อผู้จัดการระบบปล่อยเทอร์มินัลที่ไม่ได้ใส่ใจ ทิ้งไว้โดยที่เปิดใช้งานด้วยสิทธิการใช้งานรากแล้ว โดยทั่วไป ผู้ใช้ควรล็อกเอาต์ทุกครั้ง ที่ปล่อยเทอร์มินัลทิ้งไว้

ตัวแปรผู้ใช้กำหนดในเซลล์เป้าหมาย:

เซลล์เป้าหมายรู้จักตัวแปรตัวอักษรและตัวเลขซึ่งสามารถ กำหนดค่าสตริงได้

เมื่อต้องการกำหนดค่าสตริงให้กับชื่อ ให้พิมพ์ดังต่อไปนี้:

```
Name=String
```

ชื่อคือลำดับของตัวอักษร ตัวเลข และขีดเส้นใต้ที่ขึ้นต้นด้วยขีดเส้นใต้หรือตัวอักษร เมื่อต้องการใช้ค่าซึ่งคุณได้กำหนดให้กับตัวแปร ให้เพิ่มเครื่องหมายดอลลาร์ (\$) ที่ตอนต้นของชื่อ ดังนั้น ตัวแปร \$Name จึงให้ค่าที่ระบุโดยตัวแปร String โปรดทราบว่าไม่มีช่องว่างอยู่บนทั้งสองด้านของเครื่องหมายเท่ากับ (=) ในคำสั่งการกำหนดค่า (พารามิเตอร์ตำแหน่งไม่สามารถปรากฏขึ้นในคำสั่ง การกำหนดค่า) คุณสามารถวางได้มากกว่าหนึ่งการกำหนดค่าบนบรรทัดคำสั่ง แต่จำไว้ว่าเชลล์ทำการกำหนดค่าจากขวาไปซ้าย

ถ้าคุณใส่ตัวแปร String ไว้ในเครื่องหมายัญประกาศคู่หรือเดี่ยว (" or ') เชลล์จะไม่ถือว่า พื้นที่ว่างแท้ับ เครื่องหมายัญภาค และอักขระบรรทัดใหม่ภายในสตริง เป็นตัวค้นค่า แต่จะฝังอักขระเหล่านั้นตามตัวอักษรในสตริง

ถ้าคุณใส่ตัวแปร String ไว้ในเครื่องหมายัญประกาศคู่ (") เชลล์ยังคงรู้จักชื่อตัวแปรในสตริงและทำการทดแทนตัวแปรนั้นคือเชลล์แทนที่การอ้างอิง ไปยังพารามิเตอร์ตำแหน่งและชื่อตัวแปรอื่นที่มีเครื่องหมายดอลลาร์ (\$) นำหน้าด้วยค่าที่สอดคล้องกัน ถ้ามี นอกจากนี้ เชลล์ ยังทำการทดแทนคำสั่งภายในสตริงที่อยู่ในเครื่องหมาย ัญประกาศคู่ด้วย

ถ้าคุณใส่ตัวแปร String ไว้ในเครื่องหมายัญประกาศเดี่ยว (') เชลล์จะไม่ทดแทนตัวแปรหรือคำสั่ง ภายในสตริง ลำดับต่อไปนี้แสดงความแตกต่างนี้:

```
You:          num=875
              number1="Add $num"
              number2='Add $num'
              echo $number1
System:       Add 875
You:         echo $number2
System:      Add $num
```

เชลล์ไม่ได้ตีความพื้นที่ว่างเปล่าในการกำหนดค่าอีกครั้งหลังการทดแทนตัวแปร ดังนั้น การกำหนดค่าต่อไปนี้ส่งผลให้ \$first และ \$second มีค่าเหมือนกัน:

```
first='a string with embedded blanks'
second=$first
```

เมื่อคุณอ้างอิงตัวแปร คุณสามารถใส่ชื่อตัวแปร (หรือ ตัวเลขที่กำหนดพารามิเตอร์ตำแหน่ง) ไว้ในวงเล็บปีกกา {} เพื่อค้นชื่อตัวแปรจากสตริงใดๆ ที่ตามมา โดยเฉพาะ ถ้าอักขระ ที่ตามหลังต่อจากชื่อเป็นตัวอักษร ตัวเลข หรือขีดเส้นใต้ และตัวแปรไม่ใช่พารามิเตอร์ตำแหน่ง จำเป็นต้องใช้วงเล็บปีกกา:

```
You:          a='This is a'
              echo "${a}n example"
System:       This is an example
You:         echo "$a test"
System:      This is a test
```

หลักการที่เกี่ยวข้อง:

“พารามิเตอร์ตำแหน่งในเชลล์เป้าหมาย” ในหน้า 286

เมื่อคุณรันเชลล์ไพธอน เชลล์จะสร้างพารามิเตอร์ตำแหน่งที่อ้างอิง แต่ละค่าบนบรรทัดคำสั่งโดยใช้ตำแหน่งของค่านั้นบนบรรทัดคำสั่งขึ้นโดยปริยาย

สิ่งอ้างอิงที่เกี่ยวข้อง:

“การทดแทนแบบมีเงื่อนไขในเชลล์เป้าหมาย” ในหน้า 284

โดยปกติ เชลล์แทนที่นิพจน์ \$ Variable ด้วย ค่าสตริงที่กำหนดให้กับตัวแปร Variable ถ้ามีอยู่ อย่างไรก็ตาม มีสัญญาณพิเศษที่อนุญาต การทดแทนแบบมีเงื่อนไข ขึ้นอยู่กับว่าตัวแปรมีการตั้งค่า หรือไม่ใช่ null หรือ ทั้งสองอย่าง

ตัวแปรที่ใช้โดยเชลล์เป้าหมาย:

เชลล์ใช้ตัวแปรต่อไปนี้ แม้ว่าเชลล์ตั้งค่า บางตัวแปร แต่คุณสามารถตั้งค่าหรือรีเซ็ตตัวแปรทั้งหมดได้

ไอเท็ม	คำอธิบาย
<i>CDPATH</i>	ระบุพาธการค้นหาสำหรับคำสั่ง <i>cd</i> (เปลี่ยน ไดเรกทอรี)
<i>HOME</i>	บ่งชี้ชื่อของ <i>ไดเรกทอรีโฮม</i> ของคุณซึ่ง เป็นไดเรกทอรีที่กลายเป็นไดเรกทอรีปัจจุบันในทันทีที่ล็อกอินเสร็จสมบูรณ์ โปรแกรม ล็อกอิน จะเริ่มต้นตัวแปรนี้ คำสั่ง <i>cd</i> ใช้ค่าของตัวแปร <i>\$HOME</i> เป็นค่าดีฟอลต์ การใช้ตัวแปรนี้แทนชื่อพาธที่ชัดเจนในเชลล์ไพรซีเตอร์ ช่วยให้ไพรซีเตอร์สามารถรันจากไดเรกทอรีอื่นได้โดยไม่ต้องทำการเปลี่ยนแปลง
<i>IFS</i>	อักขระที่เป็น IFS (internal field separators หรือตัวแบ่งฟิลด์ภายใน) ซึ่งเป็นอักขระที่เชลล์ใช้ในระหว่างการตีความพื้นที่ว่าง โดยแรกเริ่ม เชลล์ตั้งค่าตัวแปร <i>IFS</i> เพื่อรวม พื้นที่ว่าง แท็บ และอักขระบรรทัดใหม่
<i>LANG</i>	กำหนดโลแคลที่จะใช้สำหรับหมวดหมู่โลแคลเมื่อ ทั้งตัวแปร <i>LC_ALL</i> และตัวแปรสภาวะแวดล้อมที่สอดคล้องกัน (ขึ้นต้นด้วย <i>LC_</i>) ไม่ได้ระบุโลแคล
<i>LC_ALL</i>	กำหนดโลแคลที่จะใช้เพื่อเขียนทับค่าสำหรับ หมวดหมู่โลแคลที่ระบุโดยค่าติดตั้งของตัวแปรสภาวะแวดล้อม <i>LANG</i> หรือตัวแปรสภาวะแวดล้อมใดๆ ที่ขึ้นต้นด้วย <i>LC_</i>
<i>LC_COLLATE</i>	กำหนดลำดับการจัดเรียงที่จะใช้เมื่อเรียงลำดับชื่อ และเมื่อช่วงอักขระเกิดขึ้นในรูปแบบ
<i>LC_CTYPE</i>	กำหนดโลแคลสำหรับการตีความลำดับไบนารี ของข้อมูลข้อความเป็นอักขระ (นั่นคือ อักขระไบต์เดียวและอักขระหลายไบต์ในอาร์กิวเมนต์และไฟล์อินพุต) ซึ่งอักขระมีการกำหนดเป็นตัวอักษร (คลาสอักขระ <i>alpha</i>) และลักษณะการทำงานของคลาสอักขระภายในการจัดรูปแบบ
<i>LC_MESSAGES</i>	กำหนดภาษาซึ่งข้อความควรถูกเขียนขึ้น
<i>LIBPATH</i>	ระบุพาธการค้นหาสำหรับไลบรารีแบบแบ่งใช้
<i>LOGNAME</i>	ระบุชื่อล็อกอินของคุณ ที่มีการทำเครื่องหมาย <i>readonly</i> ในไฟล์ <i>/etc/profile</i>
<i>MAIL</i>	บ่งชี้ชื่อพาธของไฟล์ที่ใช้โดยระบบเมล เพื่อตรวจสอบการมาถึงของเมลใหม่ หากมีการตั้งค่าตัวแปรนี้ เชลล์จะตรวจสอบเวลาแก้ไขของไฟล์นี้เป็นครั้งคราว และแสดงค่าของ <i>\$MAILMSG</i> ถ้า เวลาเปลี่ยนและความยาวของไฟล์มากกว่า 0 ให้ตั้งค่าตัวแปร <i>MAIL</i> ในไฟล์ <i>.profile</i> โดยปกติ ค่าที่กำหนดโดยผู้ใช้ของคำสั่ง <i>mail</i> คือ <i>/usr/spool/mail/\$LOGNAME</i>
<i>MAILCHECK</i>	จำนวนวินาทีที่เชลล์อนุญาตให้ผ่านไปก่อนทำการตรวจสอบ การมาถึงของเมลอีกครั้งในไฟล์ซึ่งระบุโดยตัวแปร <i>MAILPATH</i> หรือ <i>MAIL</i> ค่าดีฟอลต์คือ 600 วินาที (10 นาที) หากคุณตั้งค่าตัวแปร <i>MAILCHECK</i> เป็น 0 เชลล์จะตรวจสอบก่อนแต่ละพร้อมต์
<i>MAILMSG</i>	ข้อความการแจ้งเตือนทางเมล หากคุณตั้งค่าตัวแปร <i>MAILMSG</i> เป็น สตริง <i>null</i> อย่างชัดเจน (<i>MAILMSG=""</i>) จะไม่มีข้อความแสดงขึ้น
<i>MAILPATH</i>	รายการชื่อไฟล์ที่ค้นด้วยเครื่องหมายจุด ถ้าตั้งค่าตัวแปรนี้ เชลล์จะแจ้งการมาถึงของเมลให้คุณทราบในไฟล์ใดๆ ที่ระบุในรายการ คุณสามารถใส่เครื่องหมาย % ต่อท้ายชื่อไฟล์แต่ละชื่อเพื่อให้ข้อความแสดงขึ้นเมื่อมีเมลมาถึง มิฉะนั้น เชลล์จะใช้ค่าของตัวแปร <i>MAILMSG</i> หรือข้อความ [YOU HAVE NEW MAIL] โดยค่าดีฟอลต์ หมายเหตุ: เมื่อมีการตั้งค่าตัวแปร <i>MAILPATH</i> จะมีการตรวจสอบไฟล์เหล่านี้แทนชุดไฟล์ที่ตั้งค่าโดยตัวแปร <i>MAIL</i> เมื่อต้องการตรวจสอบชุดไฟล์โดยตัวแปร <i>MAILPATH</i> และชุดไฟล์ โดยตัวแปร <i>MAIL</i> ให้ระบุไฟล์ <i>MAIL</i> ในรายการไฟล์ <i>MAILPATH</i> ของคุณ

ไอเท็ม	คำอธิบาย
<i>PATH</i>	<p>พาดการค้นหาของคำสั่ง ซึ่งเป็นรายการที่จัดลำดับของ ชื่อพาราดิเรกทอรีที่ค้นด้วยเครื่องหมายจุดคู่ เซลล์ค้นหาไดเรกทอรีเหล่านี้ในลำดับที่ระบุเมื่อเซลล์ค้นหาคำสั่ง สตริง null ที่ใดๆ ในรายการ แสดงถึงไดเรกทอรีปัจจุบัน</p> <p>โดยปกติ ตัวแปร <i>PATH</i> มีการเริ่มต้นขึ้นในไฟล์ <code>/etc/environment</code> ซึ่งโดยปกติเป็น <code>/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin</code> คุณสามารถใช้ตัวแปรนี้เพื่อให้เหมาะกับความต้องการของคุณเอง ตัวแปร <i>PATH</i> ที่จัดเตรียมไว้ในไฟล์ <code>.profile</code> ของคุณยังมี <code>\$HOME/bin</code> และ ไดเรกทอรีปัจจุบันของคุณด้วย</p> <p>หากคุณมีไดเรกทอรีเฉพาะโครงการของคำสั่ง ตัวอย่างเช่น <code>/project/bin</code> ซึ่งคุณต้องการค้นหา ก่อนไดเรกทอรีระบบมาตรฐาน ให้ตั้งค่าตัวแปร <i>PATH</i> ของคุณ ดังนี้:</p> <p><code>PATH=/project/bin:\$PATH</code></p> <p>สถานที่ที่ดีที่สุด ในการตั้งค่าตัวแปร <i>PATH</i> เป็นค่าอื่นซึ่งไม่ใช่ ค่าดีฟอลต์คือในไฟล์ <code>\$HOME/.profile</code> ของคุณ คุณไม่สามารถใช้ตัวแปร <i>PATH</i> ถ้าคุณกำลังดำเนินการคำสั่งภายใต้ เซลล์ที่จำกัด</p>
<i>PS1</i>	สตริงที่จะใช้เป็นพร้อมระบบหลัก เซลล์เชิงโต้ตอบ แสดงสตริงพร้อมต้นเมื่อคาดหวังอินพุต ค่าดีฟอลต์ ของตัวแปร <i>PS1</i> คือ \$ ตามด้วย พื้นที่ว่างเปล่าสำหรับผู้ที่ไม่ใช่ผู้ใช้ราก
<i>PS2</i>	ค่าของสตริงพร้อมตรง หากเซลล์คาดหวัง อินพุตเพิ่มเติมเมื่อพบอักขระบรรทัดใหม่ในอินพุต เซลล์จะพร้อมด้วยค่าของตัวแปร <i>PS2</i> ค่าดีฟอลต์ของ ตัวแปร <i>PS2</i> คือ > ตามด้วยพื้นที่ว่างเปล่า
<i>SHACCT</i>	ชื่อของไฟล์ที่คุณเป็นเจ้าของ ถ้าตั้งค่าตัวแปรนี้ เซลล์จะเขียนเรกคอร์ดการบัญชีลงในไฟล์สำหรับแต่ละเซลล์สคริปต์ที่ดำเนินการ คุณสามารถใช้โปรแกรมการบัญชี เช่น <code>acctcom</code> และ <code>acctcms</code> เพื่อวิเคราะห์ข้อมูลที่รวบรวม
<i>SHELL</i>	ชื่อพาราดิเรกทอรีของเซลล์ซึ่งเก็บไว้ในสภาวะแวดล้อม ตัวแปรนี้ควรมีการตั้งค่าและเอ็กซ์พอร์ตโดยไฟล์ <code>\$HOME/.profile</code> ของแต่ละลิ้อกอินที่จำกัด
<i>TIMEOUT</i>	จำนวนนาทีที่เซลล์ยังคงไม่ได้ใช้งานก่อนจะ ออก หากตั้งค่าตัวแปรนี้เป็นค่าที่มากกว่าศูนย์ (0) เซลล์จะออก ถ้าไม่ได้ป้อนคำสั่งภายในจำนวนวินาทีที่ระบุไว้ นับจากออกใช้พร้อม <i>PS1</i> (โปรดทราบว่าสามารถคอมไพล์ เซลล์ด้วยขอบเขตสูงสุดที่ไม่เกินกว่าค่านี้) ค่าศูนย์บ่งชี้การไม่มีขีดจำกัดเวลา

หลักการที่เกี่ยวข้อง:

“การตีความพื้นที่ว่างเปล่า” ในหน้า 299

หลังจากเซลล์ทำการทดแทนตัวแปรและคำสั่งแล้ว เซลล์จะ สแกนผลลัพธ์เพื่อดูตัวแบ่งฟิลด์ภายใน (ตัวแบ่งที่กำหนดไว้ในตัวแปรเซลล์ *IFS*)

คำพิเศษที่กำหนดไว้ล่วงหน้าในเซลล์เป้าหมาย:

ตัวแปรหลายตัวมีความหมายพิเศษ ตัวแปรต่อไปนี้ มีการตั้งค่าโดยเซลล์เป้าหมายเท่านั้น:

ไอเท็ม	คำอธิบาย
<i>\$@</i>	<p>ขยายพารามิเตอร์ตำแหน่งโดยเริ่มต้นด้วย \$1 แต่ละพารามิเตอร์มีการค้นด้วยช่องว่าง</p> <p>หากคุณใส่เครื่องหมายอัฒภาคคู่ (" ") รอบ <i>\$@</i> เซลล์จะพิจารณาว่า พารามิเตอร์ตำแหน่งแต่ละตัวเป็นสตริงแยกต่างหาก ถ้าไม่มีพารามิเตอร์ตำแหน่ง เซลล์เป้าหมายจะขยายคำสั่งเป็นสตริง null ที่ไม่อ้างอิง</p>
<i>\$*</i>	<p>ขยายพารามิเตอร์ตำแหน่งโดยเริ่มต้นด้วย \$1 เซลล์ค้นหาแต่ละพารามิเตอร์ด้วยอักขระตัวแรกของค่าตัวแปร <i>IFS</i></p> <p>หากคุณใส่เครื่องหมายอัฒภาคคู่ (" ") รอบ <i>\$*</i> เซลล์จะใส่ค่า พารามิเตอร์ตำแหน่งไว้ในเครื่องหมายอัฒภาคคู่ แต่ละค่ามีการค้นด้วยอักขระตัวแรกของตัวแปร <i>IFS</i></p>
<i>\$#</i>	ระบุจำนวนของพารามิเตอร์ตำแหน่งที่ส่งผ่านไป ยัง เซลล์ โดยไม่นับชื่อของตัวเซลล์โปรแกรมเอง ดังนั้นตัวแปร <i>\$#</i> ให้จำนวนของพารามิเตอร์ตำแหน่งสูงสุดที่มีการ ตั้งค่า การใช้งานหลักอย่างหนึ่งของตัวแปรนี้คือ การตรวจสอบการมีอยู่ ของจำนวนอาร์กิวเมนต์ที่ต้องการ เฉพาะพารามิเตอร์ตำแหน่ง \$0 ถึง \$9 เท่านั้นสามารถเข้าถึงได้ผ่านทางเซลล์

ไอเท็ม คำอธิบาย

\$?	ระบุค่าออกของคำสั่งล่าสุดที่ดำเนินการ ค่า เป็นสตริงฐานสิบ คำสั่งส่วนใหญ่ส่งคืนค่าเป็น 0 เพื่อบ่งชี้ถึงความสำเร็จ สมบูรณ์ ตัวเซลล์เองส่งคืนค่าปัจจุบันของตัวแปร \$? เป็น คาคอก
\$\$	ระบุจำนวนโปรเซสของโปรเซสปัจจุบัน เนื่องจาก หมายเลขโปรเซสไม่ซ้ำกันระหว่างโปรเซสทั้งหมดที่มีอยู่ จึงมักใช้สตริงนี้ เพื่อสร้างชื่อที่ไม่ซ้ำกันของไฟล์ชั่วคราว ตัวอย่างต่อไปนี้ แสดงวิธีปฏิบัติที่แนะนำของการสร้างไฟล์ชั่วคราวในไดเรกทอรี ซึ่งใช้สำหรับวัตถุประสงค์นั้นเท่านั้น: <pre>temp=/tmp/\$\$ ls >\$temp . . rm \$temp</pre>
#!	ระบุหมายเลขโปรเซสของโปรเซสล่าสุดที่รันใน พื้นหลังโดยใช้ตัวหยุด &
\$-	สตริงประกอบด้วยชื่อของแฟล็กการดำเนินการ ที่ตั้งค่าในปัจจุบันในเซลล์

หลักการที่เกี่ยวข้อง:

“พารามิเตอร์ตำแหน่งในเซลล์เป้าหมาย” ในหน้า 286

เมื่อคุณรันเซลล์โพรซีเดอร์ เซลล์จะสร้างพารามิเตอร์ตำแหน่งที่อ้างอิง แต่ละค่าบนบรรทัดคำสั่งโดยใช้ตำแหน่งของค่านับบนบรรทัดคำสั่งขึ้นโดยปริยาย

การตีความพื้นที่ว่างเปล่า:

หลังจากเซลล์ทำการทดแทนตัวแปรและคำสั่งแล้ว เซลล์จะ สแกนผลลัพธ์เพื่อดูตัวแบ่งฟิลด์ภายใน (ตัวแบ่งที่กำหนดไว้ในตัวแปรเซลล์ *IFS*)

เซลล์แบ่งบรรทัดออกเป็นคำที่แยกกันในแต่ละตำแหน่งที่พบอักขระเหล่านี้ ตั้งแต่หนึ่งตัวขึ้นไปซึ่งแบ่งแต่ละคำที่แยกกันด้วยช่องว่างหนึ่งช่อง จากนั้น รักษาอาร์กิวเมนต์ explicit null ไว้ (" หรือ ') และทิ้งอาร์กิวเมนต์ implicit null (ที่เป็นผลมาจากพารามิเตอร์ซึ่ง ไม่มีค่า)

สิ่งอ้างอิงที่เกี่ยวข้อง:

“ตัวแปรที่ใช้โดยเซลล์เป้าหมาย” ในหน้า 297

เซลล์ใช้ตัวแปรต่อไปนี้ แม้ว่าเซลล์ตั้งค่า บางตัวแปร แต่คุณสามารถตั้งค่าหรือรีเซ็ตตัวแปรทั้งหมดได้

C เซลล์

C เซลล์คือตัวแปลคำสั่งแบบโต้ตอบและภาษา การเขียนโปรแกรมคำสั่ง C เซลล์ใช้ไวยากรณ์ที่คล้ายกับภาษาการเขียนโปรแกรม C

คำสั่ง csh เริ่มต้น C เซลล์

เมื่อคุณล็อกอิน คำสั่ง *csh* จะค้นหาไฟล์เช็ตอัพ ทั้งระบบ */etc/csh.cshrc* ก่อน ถ้ามีไฟล์เช็ตอัพ C เซลล์จะดำเนินการคำสั่งที่จัดเก็บไว้ในไฟล์นั้น ถัดไป C เซลล์จะดำเนินการไฟล์เช็ตอัพทั้งระบบ */etc/csh.login* ถ้า มีอยู่ จากนั้น ค้นหาไดเรกทอรีโฮมของคุณเพื่อหาไฟล์ *.cshrc* และ *.login* ถ้ามีไฟล์ดังกล่าว ไฟล์เหล่านั้นมีข้อมูลผู้ใช้ที่กำหนดเองซึ่งเหมาะสมสำหรับการรัน C เซลล์ ตัวแปรทั้งหมดที่ตั้งค่าในไฟล์ */etc/csh.cshrc* และ */etc/csh.login* อาจถูกยกเลิกโดยไฟล์ *.cshrc* และ *.login* ในไดเรกทอรี *\$HOME* ของคุณ เฉพาะผู้ใช้รากเท่านั้นสามารถแก้ไขไฟล์ */etc/csh.cshrc* และ */etc/csh.login* ได้

ไฟล์ `/etc/csh.login` และ `$HOME/.login` มีการดำเนินการเพียงครั้งเดียวเท่านั้นในเวลาล็อกอิน โดยทั่วไปแล้ว ไฟล์เหล่านี้ใช้เพื่อเก็บนิยาม ตัวแปรสถานะแวดล้อม คำสั่งซึ่งคุณต้องดำเนินการเพียงครั้งเดียวที่ ล็อกอิน หรือคำสั่งที่ตั้งค่าลักษณะเทอร์มินัล

ไฟล์ `/etc/csh.cshrc` และ `$HOME/.cshrc` มีการดำเนินการในเวลาล็อกอิน และทุกครั้งที่ใช้คำสั่ง `csh` หรือ C เซลล์สคริปต์ โดยทั่วไป ไฟล์ดังกล่าวใช้เพื่อกำหนดลักษณะ C เซลล์ เช่น สมนามและตัวแปร C เซลล์ (ตัวอย่างเช่น `history`, `noclobber`, หรือ `ignoreeof`) ขอแนะนำให้คุณใช้เฉพาะคำสั่งในตัว C เซลล์ในไฟล์ `/etc/csh.cshrc` และ `$HOME/.cshrc` เท่านั้น เนื่องจากการใช้คำสั่งอื่นเพิ่มเวลาสตาร์ทของเซลล์สคริปต์

สิ่งอ้างอิงที่เกี่ยวข้อง:

“รายการคำสั่งในตัว C เซลล์” ในหน้า 307

ข้อมูลต่อไปนี้เป็นคำสั่งในตัว C เซลล์

ขีดจำกัด C เซลล์:

ขีดจำกัดของ C เซลล์มีดังต่อไปนี้

- ค่าไม่สามารถยาวเกินกว่า 1024 ไบต์
- รายการอาร์กิวเมนต์ถูกจำกัดไม่เกิน `ARG_MAX` ไบต์ ค่าของตัวแปร `ARG_MAX` มีอยู่ในไฟล์ `/usr/include/sys/limits.h`
- จำนวนอาร์กิวเมนต์ของคำสั่งที่เกี่ยวข้องกับการขยายชื่อไฟล์ ถูกจำกัดเป็น $1/6^{\text{th}}$ ของจำนวนไบต์ที่ใช้ได้ในรายการ อาร์กิวเมนต์
- การทดแทนคำสั่งสามารถทดแทนไบต์ได้ไม่เกินจำนวนที่ได้รับอนุญาตใน รายการอาร์กิวเมนต์
- เพื่อตรวจสอบการวนซ้ำ เซลล์จะจำกัดจำนวนของการทดแทนสมนาม บนบรรทัดเดียวเป็น 20
- คำสั่ง `csh` ไม่สนับสนุนการขยายชื่อไฟล์ ตามข้อมูลการจัดประเภทเทียบเท่าของอักขระ
- ตัวอธิบายไฟล์ (นอกเหนือจาก `standard in`, `standard out`, และข้อผิดพลาดมาตรฐาน) ที่เปิดก่อน `csh` ดำเนินการแอฟพลิเคชันใดๆ ไม่มีอยู่สำหรับแอฟพลิเคชันนั้น

การทดแทนสมนามใน C เซลล์:

สมนาม คือชื่อที่กำหนดให้กับคำสั่งหรือสตริงคำสั่ง C เซลล์อนุญาตให้คุณกำหนดสมนามและใช้สมนามเมื่อคุณจะทำคำสั่ง เซลล์เก็บรักษารายการของสมนามซึ่งคุณกำหนด

หลังจากเซลล์สแกนบรรทัดคำสั่ง เซลล์จะแบ่ง คำสั่งออกเป็นคำที่แยกกันและตรวจสอบคำแรกของแต่ละคำสั่ง จากซ้ายไปขวา เพื่อดูว่ามีสมนามอยู่หรือไม่ หากพบสมนาม เซลล์จะใช้ กลไกประวัติเพื่อแทนที่ข้อความของสมนามด้วยข้อความ ของคำสั่งที่แสดงแทนโดยสมนาม คำที่ได้แทนที่รายการคำสั่ง และอาร์กิวเมนต์ หากไม่มีการทำการอ้างอิงไปยังรายการประวัติ รายการ อาร์กิวเมนต์ จะยังคงไม่มีการเปลี่ยนแปลง

คำสั่งในตัว `alias` และ `unalias` สร้าง แสดง และแก้ไขรายการสมนาม ใช้คำสั่งสมนาม ในรูปแบบต่อไปนี้:

```
alias [Name [WordList]]
```

ตัวแปร `Name` ที่เป็นทางเลือกระบุ สมนามสำหรับชื่อที่ระบุ ถ้าคุณระบุรายการคำด้วยตัวแปร `WordList` คำสั่งจะกำหนดรายการ คำนั้นเป็นสมนามของตัวแปร `Name` หากคุณรันคำสั่ง `alias` โดยไม่มีตัวแปรทางเลือกอย่างใดอย่างหนึ่ง ระบบจะแสดงสมนาม C เซลล์ทั้งหมด

ถ้าสมนามของคำสั่ง ls เป็น ls -l คำสั่งต่อไปนี้:

```
ls /usr
```

จะถูก แทนที่โดยคำสั่ง:

```
ls -l /usr
```

รายการอาร์กิวเมนต์ไม่มีการเปลี่ยนแปลงเนื่องจากไม่มีการอ้างอิง ไปยังรายการประวัติในคำสั่งที่มีสมนาม ในลักษณะคล้ายกัน ถ้าสมนามของคำสั่ง lookup เป็นดังนี้:

```
grep \!^ /etc/passwd
```

เซลล์จะแทนที่ lookup bill ด้วยข้อมูลต่อไปนี้:

```
grep bill /etc/passwd
```

ใน ตัวอย่างนี้ !^ อ้างถึงรายการประวัติ และเซลล์ แทนที่ข้อมูลนั้นด้วยอาร์กิวเมนต์แรกในเส้นอินพุต ซึ่งในกรณีนี้คือ bill

คุณสามารถใช้อักขระการจับคู่รูปแบบพิเศษใน สมนามได้ คำสั่งต่อไปนี้:

```
alias lprint 'pr &bslash2.!*' >
```

```
> print'
```

สร้างคำสั่งที่จัดรูปแบบอาร์กิวเมนต์ไปยังเครื่องพิมพ์ บรรทัด อักขระ ! ถูกป้องกันจากเซลล์ในสมนาม โดยการใช้เครื่องหมายอัฒประกาศเดี่ยว เพื่อให้สมนามไม่ถูกขยายจนกว่า คำสั่ง pr รัน

ถ้าเซลล์ระบุตำแหน่งสมนาม เซลล์จะทำการแปลงค่า ของข้อความอินพุต และเริ่มต้นโปรเซสสมนามอีกครั้งบนเส้นอินพุตที่แปลง ถ้าค่าแรกของข้อความถัดไปเหมือนกับข้อความ ก่อนหน้านั้น จะมีการป้องกันการวนซ้ำโดยการแพลิกสมนามเพื่อยุติโปรเซสสมนาม การวนซ้ำลำดับต่อมาอื่นถูกตรวจพบและส่งผลให้เกิดข้อผิดพลาด

หลักการที่เกี่ยวข้อง:

“การทดแทนประวัติใน C เซลล์” ในหน้า 317

การทดแทนประวัติอนุญาตให้คุณแก้ไขคำแต่ละคำจากคำสั่งก่อนหน้านี้ เพื่อสร้างคำสั่งใหม่ การทดแทนประวัติช่วยให้ง่ายในการทำซ้ำ คำสั่ง ทำซ้ำอาร์กิวเมนต์ของคำสั่งก่อนหน้านี้ในคำสั่งปัจจุบัน หรือแก้ไขข้อผิดพลาดการสะกดในคำสั่งก่อนหน้านี้ ด้วยการพิมพ์เพียงเล็กน้อยเท่านั้น

การทดแทนตัวแปรใน C เซลล์:

C เซลล์รักษาชุดของตัวแปร โดยแต่ละชุดมีค่ารายการค่า เป็นศูนย์หรือมากกว่านั้น บางตัวแปรเหล่านี้มีการตั้งค่าโดย เซลล์ หรืออ้างอิงโดยเซลล์ ตัวอย่างเช่น ตัวแปร *argv* เป็นรูปภาพของรายการตัวแปรเซลล์ และค่าที่ประกอบเป็นค่า ของตัวแปรนี้มีการอ้างอิงในวิธีพิเศษ

เมื่อต้องการเปลี่ยนและแสดงค่าของตัวแปร ให้ใช้คำสั่ง *set* และ *unset* ในบรรดาตัวแปร ที่อ้างอิงโดยเซลล์ มีการสลับตัวเลข (ตัวแปรที่เปิดและปิด บางสิ่ง) เซลล์ไม่ได้ตรวจสอบการสลับค่า แต่ตรวจสอบว่ามีการตั้งค่า หรือไม่เท่านั้น ตัวอย่างเช่น ตัวแปรเซลล์ *verbose* เป็นการสลับที่ส่งผลให้อินพุตคำสั่งถูกสะท้อน การตั้งค่าของตัวแปรนี้ เป็นผลมาจากการออกใช้แฟล็ก *-v* บนบรรทัดคำสั่ง

การดำเนินงานอื่นจัดการตัวแปรเป็นตัวเลข คำสั่ง @ ทำการคำนวณตัวเลข และผลลัพธ์ถูกกำหนดให้กับตัวแปร อย่างไรก็ตาม คำสั่งตัวแปรมีการแสดงแทนเป็นสตริง (ศูนย์หรือมากกว่า) เสมอ สำหรับการดำเนินงาน ตัวเลข สตริง null จะถูกพิจารณาเป็น ศูนย์ และค่าที่ส่งและที่ตามมา ของค่าหลายค่าจะถูกละเว้นไป

เมื่อคุณออกใช้คำสั่ง เซลล์จะแจกส่วนเส้นอินพุต และทำการทดแทนสมนาม ถัดไป ก่อนการรันคำสั่ง เซลล์ จะทำการทดแทน ตัวแปร อักขระ \$ ระบุ การทดแทน อย่างไรก็ตาม มีการส่งผ่านโดยไม่เปลี่ยนแปลงถ้าตามหลังด้วยพื้นที่ว่าง แท็บ หรืออักขระ บรรทัดใหม่ อักขระ \$ ที่มี \ นำหน้าป้องกัน การขยายนี้ ยกเว้นในสองกรณีคือ:

- คำสั่งที่อยู่ใน " " ในกรณีนี้ เซลล์ ทำการทดแทนเสมอ
- คำสั่งที่อยู่ใน ' ' ในกรณีนี้ เซลล์ ไม่เคยทำการทดแทน สตริงที่ใส่อยู่ใน ' ' จะถูกตีความ สำหรับการทดแทนคำสั่ง

เซลล์รับการเปลี่ยนทิศทางอินพุตและเอาต์พุตก่อน การขยายตัวแปร และขยายแต่ละตัวแยกต่างหากกัน มิฉะนั้น ชื่อคำสั่ง และรายการอาร์กิวเมนต์ทั้งหมดจะขยายด้วยกัน ดังนั้นจึงอาจเป็นไปได้ที่ค่า (คำสั่ง) แรก จะสร้างมากกว่าหนึ่งค่า ซึ่งค่าแรก กลายเป็นชื่อคำสั่ง และค่าที่เหลือกลายเป็นพารามิเตอร์

ยกเว้นว่าใส่อยู่ใน " " หรือกำหนดตัวแก้ไข :q ผลลัพธ์ของการทดแทนตัวแปรอาจต้องถูกทดแทนคำสั่งและ ชื่อไฟล์ต่อไป เมื่อใส่อยู่ในเครื่องหมายอัฒภาคคู่ ตัวแปร ที่มีค่าซึ่งประกอบด้วยหลายค่าขยายเป็นค่าเดียวหรือ ส่วนของค่าเดียว โดย คำของค่าตัวแปรถูกแบ่งด้วย พื้นที่ว่าง เมื่อคุณใช้ตัวแก้ไข :q กับการทดแทน ตัวแปรจะขยายเป็นหลายค่า แต่ละค่าถูกแบ่ง ด้วยพื้นที่ว่าง และใส่ไว้ในเครื่องหมายอัฒภาคคู่เพื่อป้องกันการทดแทนคำสั่งหรือ ชื่อไฟล์ในภายหลัง

สัญญาณต่อไปนี้อนุญาตให้คุณแนะนำค่าตัวแปร ในเซลล์อินพุต ยกเว้นตามที่ระบุ ถือเป็นข้อผิดพลาดในการอ้างอิง ตัวแปรซึ่ง ไม่ได้ตั้งค่าด้วยคำสั่ง set

คุณสามารถใช้ตัวแก้ไข :gh, :gt, :gr, :h, :r, :q, และ :x กับการทดแทน ต่อไปนี้ ถ้า {} ปรากฏขึ้นในรูปแบบคำสั่ง แสดงว่า ต้องใส่ ตัวแก้ไขไว้ภายในวงเล็บปีกกา อนุญาตให้ใช้ได้เพียงหนึ่งตัวแก้ไข : ในแต่ละการขยายตัวแปร

ไอเท็ม	คำอธิบาย
\$Name	
#{Name}	แทนที่โดยค่าที่กำหนดให้กับตัวแปร Name แต่ละค่า ถูกแบ่งด้วยพื้นที่ว่าง วงเล็บปีกกาแยกตัวแปร Name ออกจากอักขระที่ ตามมา ซึ่งจะเป็นส่วนหนึ่งของตัวแปรถ้าไม่ใช้วงเล็บปีกกา ชื่อตัวแปรเซลล์ เริ่มต้นด้วยตัวอักษรและประกอบด้วยตัวอักษรและ ตัวเลขได้สูงสุด 20 ตัว รวมถึง อักขระขีดเส้นใต้ (_) ถ้าตัวแปร Name ไม่ได้ ระบุตัวแปรเซลล์แต่มีการตั้งค่าในสภาวะแวดล้อม ค่าของตัวแปรนั้น จะถูกส่งคืน ตัวแก้ไขที่มีเครื่องหมายจุดนำหน้า และรูปแบบอื่นที่อธิบายที่นี่ ไม่มีอยู่ในกรณีนี้
\$Name[number]	
#{Name[number]}	เลือกเพียงบางค่าเท่านั้นจากค่า ของตัวแปร Name ตัวเลขอยู่ภายใต้เงื่อนไขการทดแทนตัวแปร และอาจประกอบด้วยตัวเลข เดียวหรือสองตัวเลขที่คั่น ด้วยยัติภังค์ (-) ค่าแรกของค่าสตริงตัวแปร มีหมายเลขเป็น 1 ถ้าตัวเลขแรกของช่วงถูกข้ามไป จะมี ค่าดีฟอลต์เป็น 1 ถ้าตัวเลขสุดท้ายของช่วงถูกข้ามไป จะมีค่าดีฟอลต์เป็น \$#Name สัญลักษณ์เครื่องหมายดอกจัน (*) เลือก ทุกค่า ไม่เป็นข้อผิดพลาด สำหรับช่วงที่จะวางถ้าอาร์กิวเมนต์ที่สองถูกข้ามไปหรืออยู่ในช่วง
##Name	
#{#Name}	ให้จำนวนค่าในตัวแปร Name ซึ่งสามารถใช้ใน [number] ดัง แสดงข้างบน ตัวอย่างเช่น \$Name[\$#Name]
\$0	ทดแทนชื่อไฟล์ซึ่งอินพุตคำสั่ง กำลังถูกอ่าน ข้อผิดพลาดเกิดขึ้นถ้าไม่รู้จักชื่อ
\$number	
#{number}	เทียบเท่ากับ \$argv[number]
\$	เทียบเท่ากับ \$argv[]

การทดแทนต่อไปนี้อาจไม่ถูกเปลี่ยนแปลงด้วยตัวแก้ไข ::

ไอเท็ม	คำอธิบาย
<code>\$(?name)</code>	ทดแทนสตริง 1 ถ้ามีการตั้งค่าตัวแปร <code>name</code> และศูนย์ (0) ถ้าไม่ได้ตั้งค่าตัวแปรนี้
<code>\$(?0)</code>	ทดแทน 1 ถ้ารู้จักชื่อไฟล์อินพุตปัจจุบัน และศูนย์ (0) ถ้าไม่รู้จักชื่อไฟล์
<code>\$\$</code>	ทดแทนหมายเลขโปรเซส (ฐานสิบ) ของพาเรนต์เชลล์
<code>\$(<</code>	ทดแทนบรรทัดจากอินพุตมาตรฐาน โดยไม่มีการตีความเพิ่มเติม ใช้การทดแทนนี้เพื่ออ่านจากคีย์บอร์ดในเชลล์ไพธอน

หลักการที่เกี่ยวข้อง:

“การทดแทนคำสั่งในเชลล์ C” ในหน้า 316

ใน *การทดแทนคำสั่ง* เชลล์ดำเนินการคำสั่งที่ระบุ และแทนที่คำสั่งนั้นด้วยเอาต์พุต

การทดแทนชื่อไฟล์ใน C เชลล์:

C เชลล์อนุญาตให้คุณทำการทดแทนชื่อไฟล์ได้

C เชลล์นำเสนอข้อคัดหลายรายการ เพื่อช่วยประหยัดเวลาและการกดแป้นพิมพ์ หากคำมีอักขระ `*`, `?`, `[]`, หรือ `{ }` หรือขึ้นต้นด้วยเครื่องหมายการออกเสียง (`~`) คำจะเป็นตัวเลือกสำหรับการทดแทนชื่อไฟล์ C เชลล์ถือว่าคำ เป็นรูปแบบ และแทนที่คำ ด้วยรายการที่เรียงตามตัวอักษรของชื่อไฟล์ ซึ่งตรงกับรูปแบบ

มีการใช้ลำดับการจัดเรียงปัจจุบันตามที่ระบุโดย ตัวแปรสถานะแวดล้อม `LC_COLLATE` หรือ `LANG` ในรายการของคำที่ระบุ การทดแทนชื่อไฟล์ จะส่งผลให้เกิดข้อผิดพลาด ถ้าไม่มีรูปแบบตรงกับชื่อไฟล์ที่มีอยู่ อย่างไรก็ตาม ไม่จำเป็นว่า ทุกรูปแบบ ต้องตรงกัน เฉพาะสัญลักษณ์การจับคู่อักขระ `*`, `?`, และ `[]` เท่านั้นบ่งชี้การจับคู่รูปแบบหรือการขยายชื่อไฟล์ อักขระเครื่องหมายการออกเสียง (`~`) และ `{ }` บ่งชี้ คำย่อชื่อไฟล์

การขยายชื่อไฟล์ใน C เชลล์:

อักขระเครื่องหมายดอกจัน (`*`) จับคู่สตริงต่างๆ ของ อักขระ รวมถึงสตริง `null`

ตัวอย่างเช่น ในไดเรกทอรีที่มีไฟล์:

```
a aa aax alice b bb c cc
```

คำสั่ง `echo a*` จะพิมพ์ชื่อไฟล์ทั้งหมดที่ขึ้นต้นด้วย อักขระ `a`:

```
a aa aax alice
```

หมายเหตุ: เมื่อชื่อไฟล์ ตรงกัน อักขระจุด (`.`) และเครื่องหมายสแลช (`/`) ต้องตรงกัน

อักขระเครื่องหมายคำถาม (?) ตรงกับอักขระเดี่ยวใดๆ คำสั่งต่อไปนี้:

```
ls a?x
```

แสดงรายการ ชื่อไฟล์ทุกชื่อที่ขึ้นต้นด้วยตัวอักษร `a` ตามด้วย อักขระเดี่ยว และสิ้นสุดด้วยตัวอักษร `x`:

```
aa x
```

เมื่อต้องการจับคู่อักขระเดี่ยวหรือช่วงของอักขระ ให้ใส่อักขระไว้ภายในเครื่องหมาย `[]` คำสั่งต่อไปนี้:

```
ls [abc]
```

แสดง รายการชื่อไฟล์ทั้งหมดที่ตรงกับอักขระในเครื่องหมายวงเล็บตัวใดตัวหนึ่งอย่างชัดเจน:

a b c

ภายในวงเล็บ ช่วงศัพท์ของอักขระมีการบ่งชี้โดย [a-z] อักขระที่ตรงกับรูปแบบนี้มีการกำหนด โดยลำดับการจัดเรียงปัจจุบัน

คำย่อชื่อไฟล์ใน C เซลล์:

เครื่องหมายการออกเสียง (~) และอักขระ { บ่งชี้คำย่อชื่อไฟล์ ~ ที่ตอนต้นของ ชื่อไฟล์ใช้เพื่อแสดงถึงไดเรกทอรีโฮม อักขระ ~ ซึ่งปรากฏขึ้นโดยลำพัง จะขยายไดเรกทอรีโฮมของคุณตั้งสะท้อนในค่าของตัวแปรเซลล์ *home*

ตัวอย่างเช่น คำสั่งต่อไปนี้:

```
ls ~
```

แสดงรายการ ไฟล์และไดเรกทอรีทั้งหมดที่ตั้งอยู่ในไดเรกทอรี \$HOME ของคุณ

เมื่อคำสั่งมีการตามหลังด้วยชื่อที่ประกอบด้วยอักขระ ตัวอักษร ตัวเลข และยัติภังค์ (-) เซลล์จะค้นหา ผู้ใช้ที่มีชื่อนั้นและทดแทนไดเรกทอรี \$HOME ของผู้ใช้นั้น

หมายเหตุ: ถ้า อักขระ ~ มีการตามด้วยอักขระอื่นที่ไม่ใช่ตัวอักษร หรือเครื่องหมายทับ (/) หรือปรากฏขึ้นในที่ใดๆ ยกเว้นที่ตอนต้น ของคำ อักขระจะไม่ขยาย

เมื่อต้องการจับคู่อักขระในชื่อไฟล์โดยไม่ต้องพิมพ์ชื่อไฟล์ ทั้งชื่อ ใช้ใส่ { } รอบชื่อไฟล์ รูปแบบ a{b,c,d}e เป็น อีกวิธีหนึ่งในการเขียน abe ace ade เซลล์รักษาลำดับ จากซ้ายไปขวาและจัดเก็บผลลัพธ์ของรายการที่ตรงกันไว้แยกต่างหากที่ระดับ ต่ำเพื่อรักษาลำดับนี้ การสร้างนี้อาจมีการจัดกลุ่มได้ ดังนั้น คำสั่งต่อไปนี้:

```
~source/s1/{oldls,ls}.c
```

จะขยายเป็น:

```
/usr/source/s1/oldls.c /usr/source/s1/ls.c
```

ถ้า ไดเรกทอรีโฮมสำหรับ **source** เป็น /usr/source ในลักษณะคล้ายกัน คำสั่งต่อไปนี้:

```
../{memo,*box}
```

อาจขยายเป็น:

```
../memo ../box ../mbox
```

หมายเหตุ: memo ไม่มี การเรียงลำดับด้วยผลลัพธ์ของ *box ที่ตรงกัน เนื่องจากเป็นตัวพิมพ์ พิเศษ อักขระ {, }, และ {} จึงมีการส่งผ่านอย่างราบรื่น

คลาสอักขระใน C เซลล์:

คุณสามารถใช้คลาสอักขระเพื่อจับคู่อักขระชื่อไฟล์ภายในช่วง ที่ระบุ

รูปแบบต่อไปนี้แนะนำให้ระบบ จับคู่อักขระเดี่ยวใดๆ ที่เป็นสมาชิกของคลาสที่ระบุ:

```
[:charclass:]
```

คลาสต่อไปนี้สอดคล้องกับรูทีนย่อย ctype:

คลาสอักขระ	นิยาม
alnum	อักขระแบบตัวอักษรผสมตัวเลข
alpha	ตัวอักษรตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก
cntrl	อักขระควบคุม
digit	ตัวเลข
graph	อักขระกราฟิก
lower	ตัวอักษรตัวพิมพ์เล็ก
print	อักขระที่พิมพ์ได้
punct	อักขระวรรคตอน
space	ช่องว่าง แท็บ แนวนอน บัดแคร์ ขึ้นบรรทัดใหม่ แท็บแนวตั้ง หรืออักขระ ป้อนกระดาด
upper	อักขระตัวพิมพ์ใหญ่
xdigit	ตัวเลขฐานสิบหก

สมมติว่าคุณอยู่ในไดเรกทอรีที่มีไฟล์ต่อไปนี้:

```
a aa aax Alice b bb c cc
```

พิมพ์คำสั่งต่อไปนี้ที่พร้อมท์ C เซลล์:

```
ls [:lower:]
```

C เซลล์แสดงรายการชื่อไฟล์ทั้งหมดที่ขึ้นต้น ด้วยอักขระตัวพิมพ์เล็ก:

```
a aa aax b bb c cc
```

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับนิพจน์คลาสอักขระ ให้อ่านคำสั่ง `ed`

ตัวแปรสถานะแวดล้อมใน C เซลล์:

บางตัวแปรมีความพิเศษใน C เซลล์ จากตัวแปรดังกล่าว `argv`, `cwd`, `home`, `path`, `prompt`, `shell`, และ `status` มีการตั้งค่าโดย เซลล์เสมอ

ยกเว้นสำหรับตัวแปร `cwd` และ `status` การดำเนินการที่กำลังตั้งค่าโดยเซลล์จะเกิดขึ้นที่การเริ่มต้นเท่านั้น ตัวแปรข้างบน ทั้งหมดรักษาค่าติดตั้งของคุณไว้ยกเว้นว่าคุณรีเซ็ตค่าติดตั้งเหล่านั้น อย่างชัดเจน

คำสั่ง `csh` คัดลอกตัวแปรสถานะแวดล้อม `USER`, `TERM`, `HOME`, และ `PATH` เข้าในตัวแปร `csh`, `user`, `term`, `home`, และ `path` ตามลำดับ ค่าถูก คัดลอกกลับเข้าในสถานะแวดล้อมในทุกครั้งที่เราใช้ตัวแปรเซลล์ปกติ ตัวแปร `path` ไม่สามารถตั้งค่าในที่อื่น นอกเหนือจากในไฟล์ `.cshrc` เนื่องจากโปรเซสย่อย `csh` อิมพอร์ตนิยามพารามิเตอร์จากสถานะแวดล้อม และเอ็กซ์พอร์ตใหม่ถ้าเปลี่ยนแปลง

ตัวแปรต่อไปนี้มีความหมายพิเศษ:

ไอเท็ม	คำอธิบาย
<code>argv</code>	มีอาร์กิวเมนต์ที่ส่งผ่านไปยังเซลล์สคริปต์ พารามิเตอร์ ตำแหน่งถูกทดแทนจากตัวแปรนี้
<code>cdpath</code>	ระบุรายการของไดเรกทอรีอื่นที่จะค้นหา โดยคำสั่ง <code>chdir</code> หรือ <code>cd</code> เพื่อค้นหาไดเรกทอรีย่อย
<code>cwd</code>	ระบุชื่อพารามิเตอร์เต็มของไดเรกทอรีปัจจุบัน
<code>echo</code>	ตั้งค่าเมื่อใช้แฟล็กบรรทัดคำสั่ง <code>-x</code> เมื่อตั้งค่า ส่งผลให้แต่ละคำสั่งและอาร์กิวเมนต์ของคำสั่งมีการสะท้อนก่อนหน้าที่จะรันเพียงเล็กน้อย สำหรับคำสั่งที่ไม่ใช่คำสั่งในตัว การขยายทั้งหมดเกิดขึ้นก่อนหน้าการสะท้อน คำสั่งในตัว ถูกสะท้อนก่อนหน้าการทดแทนคำสั่ง และชื่อไฟล์ เนื่องจากการทดแทนเหล่านี้ สามารถเลือกทำได้

ไอเท็ม	คำอธิบาย
<i>histchars</i>	ระบุค่าตัวอักษรเพื่อเปลี่ยนอักขระที่ใช้ใน การทดแทนประวัติ ใช้อักขระแรกของค่าเป็นอักขระการทดแทนประวัติ ค่านี้จะแทนที่อักขระดีฟอลต์ ! อักขระที่สองของค่าจะแทนที่อักขระ ^ ในการทดแทนด้วย หมายเหตุ: การกำหนดค่า <i>histchars</i> เป็นอักขระที่ใช้ในคำสั่งหรือชื่อไฟล์อาจส่งผลให้เกิดการทดแทนประวัติโดยไม่ได้ตั้งใจ
<i>history</i>	มีค่าตัวเลขเพื่อควบคุมขนาดของรายการ ประวัติ คำสั่งใดๆ ที่มีการอ้างอิงภายในจำนวนเหตุการณ์ที่ได้รับอนุญาต จะไม่ถูกทิ้งไป ค่าใหญ่มากของตัวแปร <i>history</i> อาจส่งผลให้เซลล์ขาดหน่วยความจำ โดยไม่คำนึงว่ามีการตั้งค่าตัวแปรนี้หรือไม่ C เซลล์จะบันทึกคำสั่งล่าสุดที่รันไบนรายการประวัติ เสมอ
<i>home</i>	บ่งชี้ไดเรกทอรีโฮมของคุณที่เริ่มต้นจากสถานะแวดล้อม การขยายชื่อไฟล์ของอักขระเครื่องหมายการออกเสียง (~) อ้างอิง ตัวแปรนี้
<i>ignoreeof</i>	ระบุว่าเซลล์จะเว้นอักขระสิ้นสุดไฟล์ จากอุปกรณ์อินพุตที่เป็นเวิร์กสแตชัน การทำเช่นนี้ป้องกันเซลล์ไม่ให้ถูกทำลายโดยอุบัติเหตุเมื่อเซลล์อ่านอักขระสิ้นสุดไฟล์ (Ctrl-D)
<i>mail</i>	ระบุไฟล์ซึ่งเซลล์จะตรวจหาเมล การดำเนินการนี้ ทำหลังจากความสมบูรณ์ของคำสั่งแต่ละครั้งซึ่งส่งผลให้เกิดพร้อมต์ถ้าผ่านพ้นช่วงเวลาที่จะระบุ เซลล์แสดงข้อความ Mail in file ถ้าไฟล์มีอยู่โดยมีเวลาเขาถึงน้อยกว่าเวลา การเปลี่ยนแปลง หากค่าแรกของค่าตัวแปร <i>mail</i> เป็นตัวแรก ตัวแปรจะระบุช่วงเวลาการตรวจสอบเมลที่แตกต่างอื่น (ในหน่วยวินาที) ค่าดีฟอลต์คือ 600 (10 นาที) ถ้าคุณระบุไฟล์เมลหลายไฟล์ เซลล์จะแสดง ข้อความ New mail in file เมื่อมีเมล ในไฟล์ที่ระบุ
<i>noclobber</i>	ใส่ชื่อจำกัดเกี่ยวกับการเปลี่ยนทิศทางเอาต์พุต เพื่อให้แน่ใจว่า ไฟล์ไม่ถูกทำลายโดยอุบัติเหตุและการเปลี่ยนทิศทางนั้นผนวกเข้ากับไฟล์ที่มีอยู่
<i>noglob</i>	ยับยั้งการขยายชื่อไฟล์ ซึ่งมีประโยชน์สูงสุดในเซลล์สคริปต์ ที่ไม่ได้จัดการกับชื่อไฟล์ หรือเมื่อได้รับรายชื่อไฟล์ และไม่ต้องทำการขยายเพิ่มเติม
<i>nonomatch</i>	ระบุว่าไม่ส่งผลให้เกิดข้อผิดพลาดถ้าการขยายชื่อไฟล์ ไม่ตรงกับไฟล์ที่มีอยู่ใดๆ แต่ยังคงรูปแบบแบบดั้งเดิม แต่ยังคงมีข้อผิดพลาดถ้ารูปแบบแบบดั้งเดิมผิดพลาด
<i>notify</i>	ระบุว่าเซลล์ส่งการแจ้งเตือนถึงการเปลี่ยนแปลงในสถานะงาน แบบอะซิงโครนัส ค่าดีฟอลต์นำเสนอการเปลี่ยนแปลงสถานะก่อนหนทางการแสดง เซลล์พร้อมต์เพียงเล็กน้อยเท่านั้น
<i>path</i>	ระบุไดเรกทอรีซึ่งจะมีการค้นหาคำสั่งสำหรับการดำเนินการ ค่า null ระบุไดเรกทอรีปัจจุบัน หากไม่มีการตั้งค่าตัวแปร <i>path</i> สามารถรันได้เฉพาะชื่อพารามิเตอร์เท่านั้น พารามิเตอร์หาดีฟอลต์ (จากไฟล์ /etc/environment ที่ใช้ในระหว่างล็อกอิน) เป็นดังนี้: /usr/bin /etc /usr/sbin /usr/ucb /usr/bin/X11 /sbin เซลล์ที่ไม่มีทั้งแฟล็ก -c และ -t โดยปกติ hashes เนื้อหาของไดเรกทอรีในตัวแปร <i>path</i> หลังจากการอ่าน .cshrc และในแต่ละครั้งที่รีเซตตัวแปร <i>path</i> หากมีการเพิ่มคำสั่งใหม่ลงในไดเรกทอรีเหล่านี้ในขณะที่เซลล์ใช้งานอยู่ คุณต้องระบุคำสั่ง <i>rehash</i> มิฉะนั้น อาจไม่พบ คำสั่ง
<i>prompt</i>	ระบุสตริงที่แสดงขึ้นก่อนการอ่านแต่ละคำสั่ง จากอินพุตเวิร์กสแตชันแบบโต้ตอบ ถ้า ! ปรากฏขึ้นใน สตริง เครื่องหมายนี้จะถูกแทนที่โดยหมายเลขเหตุการณ์ปัจจุบัน ถ้าอักขระ ! อยู่ในสตริง อาจถึงซึ่งถูกใส่ไว้ในเครื่องหมายอัญประกาศเดี่ยวหรือคู่ อักขระ ! ต้องมีเครื่องหมาย \ นำหน้า ดีฟอลต์พร้อมต์สำหรับผู้ที่ไม่ใช่ผู้ใช้ที่ไม่มีสิทธิ์การใช้งานคือ % ดีฟอลต์พร้อมต์สำหรับผู้ใช้ที่มีสิทธิ์การใช้งานคือ #
<i>savehist</i>	ระบุค่าตัวเลขเพื่อควบคุมจำนวนรายการ ของรายการประวัติที่จัดเก็บไว้ในไฟล์ ~/.history เมื่อคุณล็อกเอาต์ คำสั่งใดๆ ที่อ้างอิงในจำนวนเหตุการณ์นี้จะถูกบันทึกไว้ในระหว่างสตาร์ทอัพ เซลล์จะอ่าน ~/.history เข้าในรายการ ประวัติ เพื่อให้สามารถบันทึกประวัติล็อกอินได้ ค่าใหญ่มากของตัวแปร <i>savehist</i> ทำให้สตาร์ทอัพเซลล์ช้าลง
<i>shell</i>	ระบุไฟล์ซึ่ง C เซลล์ตั้งอยู่ ข้อมูลนี้ใช้ในเซลล์ forking เพื่อตีความไฟล์ที่มีชุดบิตดำเนินการ แต่ไม่สามารถดำเนินการโดยระบบได้ ตัวแปรนี้มีการเริ่มต้นในโฮมของ C เซลล์
<i>status</i>	ระบุสถานะที่ส่งคืนโดยคำสั่งล่าสุด หากคำสั่ง สิ้นสุดลงอย่างผิดพลาดจะมีการเพิ่ม 0200 ในสถานะ คำสั่งในตัว ที่ไม่สำเร็จส่งคืนสถานะการออกเป็น 1 ส่วนคำสั่งในตัวที่สำเร็จ มีค่าสถานะเป็น 0
<i>time</i>	ควบคุมช่วงเวลาอัตโนมัติของคำสั่ง ถ้าตั้งค่าตัวแปรนี้ คำสั่งใดๆ ที่ใช้เวลามากกว่าจำนวนวินาที CPU ที่ระบุ จะแสดงบรรทัดของรีซอร์สที่ใช้เมื่อสิ้นสุดการดำเนินการ สำหรับข้อมูลเพิ่มเติม เกี่ยวกับดีฟอลต์เอาต์พุต ใหญ่ที่คำสั่งในตัว <i>time</i>
<i>verbose</i>	ตั้งค่าโดยแฟล็กบรรทัดคำสั่ง -v ตัวแปรนี้ส่งผลให้ คำของคำสั่งแต่ละคำสั่งแสดงขึ้นหลังจากการทดแทนประวัติ

การควบคุมงานใน C เซลล์:

เซลล์เชื่อมโยงหมายเลขงานเข้ากับแต่ละโปรเซส เซลล์ เก็บตารางของงานปัจจุบันและกำหนดเลขจำนวนเต็มค่าตัวให้กับตาราง

เมื่อคุณเริ่มทำงานในพื้นที่หลังด้วย ampersand (&) เซลล์จะพิมพ์บรรทัดที่คล้ายตัวอย่างต่อไปนี้:

```
[1] 1234
```

บรรทัดนี้บ่งชี้ว่าหมายเลขงานคือ 1 และงานประกอบด้วยโปรเซสเดียวที่มี ID โปรเซสเป็น 1234 ใช้คำสั่งในตัว `jobs` เพื่อดูตารางของงานปัจจุบัน

งานที่กำลังรันในพื้นที่หลังช่วงชิงอินพุต ถ้างานนั้นพยายามอ่านจากเวิร์กสเตชัน งานพื้นที่หลังยังสามารถจัดทำ เอาต์พุตสำหรับ เวิร์กสเตชันสลับกับเอาต์พุตของงานอื่น

คุณสามารถอ้างอิงงานในเซลล์ได้หลายวิธี ใช้อักขระเปอร์เซ็นต์ (%) เพื่อแนะนำชื่องาน ชื่อนี้ สามารถเป็นหมายเลขงานหรือชื่อคำสั่งที่เริ่มทำงาน ถ้าชื่อไม่ซ้ำกัน ตัวอย่างเช่น ถ้าโปรเซส `make` กำลังรันเป็นงาน 1 คุณสามารถอ้างอิงโปรเซสนั้นเป็น %1 คุณยังสามารถอ้างอิงเป็น %make ได้ด้วยถ้ามีงานที่พิกไว้เพียงงานเดียวที่มีชื่อซึ่งขึ้นต้นด้วยสตริง `make` คุณยังสามารถใช้ข้อมูลต่อไปนี้:

```
 $?String
```

เพื่อระบุงาน ที่มีชื่อซึ่งมีตัวแปร `String` ถ้ามีงานดังกล่าวเพียงงานเดียวเท่านั้น

เซลล์จะตรวจพบในทันทีในทุกเมื่อที่โปรเซสเปลี่ยนสถานะของตน หากงานถูกล็อกเพื่อไม่ให้มีความคืบหน้าเพิ่มเติม เซลล์จะส่งข้อความไปยังเวิร์กสเตชัน ข้อความนี้แสดงขึ้น เฉพาะหลังจากคุณกดปุ่ม Enter เท่านั้น อย่างไรก็ตาม ถ้ามีการตั้งค่าตัวแปรเซลล์ `notify` เซลล์จะออกใช้ข้อความที่บ่งชี้การเปลี่ยนแปลงในสถานะ ของงานพื้นที่หลังในทันที ใช้คำสั่งในตัว `notify` เพื่อทำเครื่องหมายโปรเซสเดียวเพื่อให้มีการรายงานการเปลี่ยนแปลงสถานะอย่างถูกต้อง โดยค่าดีฟอลต์ คำสั่ง `notify` ทำเครื่องหมายโปรเซสปัจจุบัน

รายการคำสั่งในตัว C เซลล์:

ข้อมูลต่อไปนี้คือคำสั่งในตัว C เซลล์

ไอเท็ม	คำอธิบาย
@	แสดงค่าของตัวแปรเซลล์ที่ระบุ
alias	แสดงสมนามที่ระบุหรือสมนามทั้งหมด
bg	วางงานปัจจุบันหรืองานที่ระบุไว้ในพื้นที่หลัง
break	ทำการรันต่อไปหลังการสิ้นสุดของคำสั่ง <code>foreach</code> หรือ <code>while</code> ล้อมรอบซึ่งใกล้เคียงที่สุด
breaksw	พักจากคำสั่ง <code>switch</code>
case	กำหนดเลเบลในคำสั่ง <code>switch</code>
cd	เปลี่ยนไดเรกทอรีปัจจุบันเป็นไดเรกทอรีที่ระบุ
chdir	เปลี่ยนไดเรกทอรีปัจจุบันเป็นไดเรกทอรีที่ระบุ
continue	ทำการดำเนินการของคำสั่ง <code>foreach</code> หรือ <code>while</code> ล้อมรอบซึ่งใกล้เคียงที่สุดต่อไป
default	เลเบลตัวพิมพ์ที่พอลต์ในคำสั่ง <code>switch</code>
dirs	แสดงไดเรกทอรีสแต็ก
echo	เขียนสตริงอักขระไปที่เอาต์พุตมาตรฐานของเซลล์
else	รันคำสั่งที่ตามหลัง <code>else</code> ที่สองใน ลำดับคำสั่ง <code>if (Expression) then ...else if (Expression2) then ... else ... endif</code>
end	แสดงการสิ้นสุดลำดับของคำสั่งที่นำหน้าโดยคำสั่ง <code>foreach</code>
endif	รันคำสั่งที่ตามหลังคำสั่ง <code>then</code> ที่สอง ในลำดับคำสั่ง <code>if (Expression) then ... else if (Expression2) then ... else ... endif</code>
endsw	ทำเครื่องหมายการสิ้นสุดของลำดับคำสั่ง <code>switch (String) case String : ... breaksw default : ... breaksw endsw</code> ลำดับคำสั่งนี้จับคู่เลเบลตัวพิมพ์แต่ละรายการกับค่าของ ตัวแปร <code>String</code> อย่างต่อเนื่อง การดำเนินการทำต่อไปหลังจาก คำสั่ง <code>endsw</code> ถ้ามีการดำเนินการคำสั่ง <code>breaksw</code> หรือถ้าไม่มีเลเบลตรงกันและไม่มีค่าดีฟอลต์

ไอเท็ม	คำอธิบาย
eval	อ่านค่าตัวแปรเป็นอินพุตไปยังเชลล์และดำเนินการคำสั่งที่ได้ในคอนเท็กซ์ของเชลล์ปัจจุบัน
exec	รันคำสั่งที่ระบุในตำแหน่งของเชลล์ปัจจุบัน
exit	ออกจากเชลล์ด้วยค่าของตัวแปรเชลล์สถานะ หรือค่าของนิพจน์ที่ระบุอย่างใดอย่างหนึ่ง
fg	นำงานปัจจุบันหรืองานที่ระบุไปยังพื้นหน้า โดยทำงานนั้นต่อไป ถ้างานหยุด
foreach	ตั้งค่างตัวแปร <i>Name</i> สำหรับแต่ละสมาชิกที่ระบุโดยตัวแปร <i>List</i> และลำดับของคำสั่งอย่างต่อเนื่อง จนกว่าถึงคำสั่ง <i>end</i>
glob	แสดงรายการโดยใช้ประวัติ ตัวแปร และส่วนขยายชื่อไฟล์
goto	รันต่อไปหลังจากบรรทัดที่ระบุ
hashstat	แสดงสถิติที่บ่งชี้ความสำเร็จของตาราง hash ที่คำสั่งระบุตำแหน่ง
history	แสดงรายการเหตุการณ์ประวัติ
if	รันคำสั่งที่ระบุถ้านิพจน์ที่ระบุเป็นจริง
jobs	แสดงรายการงานที่ใช้งานอยู่
kill	ส่งสัญญาณ TERM (ยุติ) หรือสัญญาณที่ระบุโดยตัวแปร <i>Signal</i> ไปยังงานหรือโปรเซสที่ระบุ
limit	จำกัดการใช้รีซอร์สที่ระบุโดยโปรเซสปัจจุบันและแต่ละโปรเซสที่สร้างขึ้น
login	สิ้นสุดเชลล์การล็อกอินและแทนที่เชลล์การล็อกอินด้วยอินสแตนซ์ของคำสั่ง /usr/sbin/login
logout	สิ้นสุดเชลล์การล็อกอิน
nice	ตั้งค่างลำดับความสำคัญของคำสั่งที่รันในเชลล์
nohup	ส่งผลให้กระบวนการไม่ตอบสนองสำหรับส่วนที่เหลือของโปรเซสเดออร์
notify	ส่งผลให้เชลล์แจ้งเตือนคุณแบบอะซิงโครนัสเมื่อสถานะของงานปัจจุบันหรืองานที่ระบุเปลี่ยนแปลง
onintr	ควบคุมการดำเนินการของเชลล์เมื่อเกิดการขัดจังหวะ
popd	แสดงไดเรกทอรีสแต็กและกลับไปยังไดเรกทอรีบนสุดใหม่
pushd	แลกเปลี่ยนอิลิเมนต์ของไดเรกทอรีสแต็ก
rehash	ส่งผลให้เกิดการคำนวณตาราง hash ภายในที่มีเนื้อหาของไดเรกทอรีในตัวแปรเชลล์พาทอีกครึ่ง
repeat	รันคำสั่งที่ระบุ ภายใต้ข้อจำกัดเดียวกันกับคำสั่ง if และจำนวนครั้งที่ระบุ
set	แสดงค่าของตัวแปรเชลล์ทั้งหมด
setenv	แก้ไขค่าของตัวแปรสภาวะแวดล้อมที่ระบุ
shift	เลื่อนตัวแปรที่ระบุไปทางซ้าย
source	อ่านคำสั่งที่ระบุโดยตัวแปร <i>Name</i>
stop	หยุดงานปัจจุบันหรืองานที่ระบุซึ่งกำลังรันในพื้นหลัง
suspend	หยุดเชลล์ราวกับว่าได้รับสัญญาณ STOP
switch	เริ่มต้นลำดับคำสั่ง switch (<i>String</i>) case <i>String</i> : ... breaksw default : ... breaksw endsw ลำดับคำสั่งนี้จับคู่เลเบลตัวพิมพ์แต่ละรายการกับค่าของตัวแปร <i>String</i> อย่างต่อเนื่อง ถ้าไม่มีเลเบลใดตรงกัน ก่อนพบดีฟอลต์เลเบล การดำเนินการจะเริ่มขึ้นหลังจากดีฟอลต์เลเบล
time	แสดงข้อมูลสรุปของเวลาที่ใช้โดยเชลล์และโปรเซสชาयन
umask	พิจารณาสิทธิ์ในไฟล์
unalias	ทิ้งสมนามทั้งหมดที่มีชื่อตรงกับตัวแปร <i>Pattern</i>
unhash	ปิดใช้งานการใช้ตาราง hash ภายในเพื่อระบุตำแหน่งโปรแกรมที่กำลังรัน
unlimit	ลบข้อจำกัดรีซอร์ส
unset	ลบตัวแปรทั้งหมดที่มีชื่อตรงกับตัวแปร <i>Pattern</i>
unsetenv	ลบตัวแปรทั้งหมดที่มีชื่อตรงกับตัวแปร <i>Pattern</i> ที่ระบุออกจากสภาวะแวดล้อม
wait	รอจนกว่าพื้นหลังทั้งหมด
while	ประเมินคำสั่งระหว่าง while และลำดับคำสั่ง end ที่ตรงกันในขณะที่นิพจน์ ซึ่งระบุโดยตัวแปร <i>Expression</i> ประเมินค่าที่ไม่ใช่ศูนย์

ข้อมูลที่เกี่ยวข้องมีดังต่อไปนี้:

Korn เชลล์

คำสั่ง ksh และ stty

คำสั่ง Korn เชลล์ alias, cd, export, fc, getopts, read, set, และ typeset

ไฟล์ /etc/passwd

เชลล์เป้าหมาย

คำสั่ง bsh หรือ Rsh, คำสั่ง login

คำสั่งพิเศษ เซลล์เป้าหมาย read

รูทีนย่อย setuid, รูทีนย่อย setgid

ไฟล์พิเศษ null

ไฟล์ environment, รูปแบบไฟล์ profile

C เซลล์

คำสั่ง csh, คำสั่ง ed

คำสั่งในตัว C เซลล์ alias, unalias, jobs, notify และ set

หลักการที่เกี่ยวข้อง:

“C เซลล์” ในหน้า 299

C เซลล์คือตัวแปลคำสั่งแบบโต้ตอบและภาษา การเขียนโปรแกรมคำสั่ง C เซลล์ใช้ไวยากรณ์ที่คล้ายกับภาษาการเขียนโปรแกรม C

“คำสั่งในตัว C เซลล์” ในหน้า 310

คำสั่งในตัวมีการรันภายในเซลล์ หากคำสั่งในตัวเกิดขึ้น เนื่องจากคอมไพเลอร์ใดๆ ของไฟฟ์ไลน์ ยกเว้นคอมไพเลอร์สุดท้าย คำสั่งจะรันในเซลล์ย่อย

การจัดการสัญญาณใน C เซลล์:

โดยปกติแล้ว C เซลล์จะเว้นสัญญาณออก งานที่กำลังรัน detached ไม่ได้รับผลกระทบจากสัญญาณที่สร้างขึ้นจากคีย์บอร์ด (INTERRUPT, QUIT, และ HANGUP)

สัญญาณอื่นมีค่าที่เซลล์ได้รับมาจากพาเรนต์ คุณสามารถควบคุมการจัดการสัญญาณ INTERRUPT และ TERMINATE ของเซลล์ในเซลล์ไพโรซีเตอร์ได้โดยใช้ onintr เซลล์การล็อกอินดักจับหรือละเว้น สัญญาณ TERMINATE ขึ้นอยู่กับวิธีการตั้งค่าเซลล์อื่นที่ไม่ใช่ เซลล์การล็อกอินส่งผ่านสัญญาณ TERMINATE ไปยังโปรเซสชายน ไม่มีกรณีใดที่สัญญาณ INTERRUPT ได้รับอนุญาตเมื่อเซลล์การล็อกอิน กำลังอ่านไฟล์ .logout

คำสั่ง C เซลล์:

คำสั่งแบบง่ายคือลำดับของคำที่แบ่งด้วยพื้นที่ว่างหรือ แท็บ คำ คือลำดับของอักขระหรือตัวเลข หรือทั้งสองอย่าง ซึ่งไม่มีพื้นที่ว่างและไม่มีเครื่องหมายอัฒภาค

นอกจากนี้ อักขระและอักขระคู่ต่อไปนี้จะทำให้เกิดคำเดี่ยว เมื่อใช้เป็นตัวแบ่งหรือตัวสิ้นสุดคำสั่ง:

```
&      |      ;  
&&     ||     <<     >>  
<      >     (      )
```

อักขระพิเศษเหล่านี้สามารถเป็นส่วนหนึ่งของคำอื่น อย่างไรก็ตาม เมื่อใส่เครื่องหมายทับขวา (\) นำหน้าอักขระพิเศษดังกล่าว จะป้องกันไม่ให้เซลล์ตีความ อักขระนั้นเป็นอักขระพิเศษ สตริงที่อยู่ในเครื่องหมาย ' ' หรือ " " (คู่ที่ตรงกัน ของอักขระอัฒภาค) หรือ backquotes ยังสามารถเป็นส่วนหนึ่งของคำได้ด้วย พื้นที่ว่าง อักขระแท็บ และอักขระพิเศษไม่ได้ทำให้เกิดคำแบ่ง เมื่อถูกใส่อยู่ในเครื่องหมายเหล่านี้ นอกจากนี้ คุณสามารถใส่อักขระบรรทัดใหม่ไว้ในเครื่องหมายเหล่านี้โดยการใส่เครื่องหมายทับขวา (\) นำหน้า

คำแรกในลำดับคำสั่งแบบง่าย (มีหมายเลข 0) โดยปกติ จะระบุชื่อของคำสั่ง คำที่เหลือจะถูกส่งผ่านไปยังคำสั่งนั้น โดยมีข้อยกเว้นสองสามข้อ ถ้าคำสั่งระบุไฟล์ปฏิบัติการที่เป็นโปรแกรมคอมไพล์ เซลล์จะรันโปรแกรมนั้นทันที ถ้าไฟล์มีการทำเครื่องหมายเป็นไฟล์ปฏิบัติการ แต่ไม่ใช่โปรแกรมคอมไพล์ เซลล์จะสมมติว่าเป็นเซลล์สคริปต์ ในกรณีนี้ เซลล์จะเริ่มต้นอีกอินสแตนซ์หนึ่งของตัวเอง (เซลล์ย่อย) เพื่ออ่านและดำเนินการคำสั่งที่รวมอยู่ในนั้น

คำสั่งในตัว C เซลล์:

คำสั่งในตัวมีการรันภายในเซลล์ หากคำสั่งในตัวเกิดขึ้น เนื่องจากคอมโพเนนต์ใดๆ ของไฟฟ์ไลน์ ยกเว้นคอมโพเนนต์สุดท้าย คำสั่งจะรันในเซลล์ย่อย

หมายเหตุ: ถ้าคุณป้อนคำสั่งจากพร้อมท์ C เซลล์ ระบบจะค้นหา คำสั่งในตัวลำดับแรก หากคำสั่งในตัวไม่มีอยู่ ระบบจะค้นหาไต่เรียกทอที่ระบุโดยตัวแปรเซลล์ *path* เพื่อหาคำสั่งระดับระบบ คำสั่งในตัว C เซลล์และคำสั่งระบบปฏิบัติการ บางคำสั่งมีชื่อเหมือนกัน อย่างไรก็ตาม คำสั่งเหล่านี้ไม่จำเป็นต้องทำงานในวิธีเดียวกัน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีการทำงานของคำสั่ง ให้ดูที่คำอธิบายคำสั่งที่เหมาะสม

ถ้าคุณรันเซลล์สคริปต์จากเซลล์ และบรรทัดแรกของเซลล์สคริปต์ เริ่มต้นด้วย `#!/ShellPathnameC` เซลล์จะรันเซลล์ที่ระบุในข้อคิดเห็นเพื่อประมวลผลสคริปต์ มิฉะนั้น เซลล์จะรันตีฟอลต์เซลล์ (เซลล์ที่ลิงก์ไปยัง `/usr/bin/sh`) ถ้ารันตามตีฟอลต์เซลล์ คำสั่งในตัว C เซลล์อาจไม่เป็นที่รู้จัก เมื่อต้องการรันคำสั่ง C เซลล์ให้ทำบรรทัดแรกของสคริปต์เป็น `#!/usr/bin/csh`

สิ่งอ้างอิงที่เกี่ยวข้อง:

“รายการคำสั่งในตัว C เซลล์” ในหน้า 307
ข้อมูลต่อไปนี้เป็นคำสั่งในตัว C เซลล์

คำอธิบายคำสั่ง C เซลล์:

C เซลล์แสดงคำสั่งในตัวต่อไปนี้

ไอเท็ม	คำอธิบาย
<code>alias [Name [WordList]]</code>	แสดงสมนามทั้งหมดถ้าคุณไม่ได้ระบุพารามิเตอร์ใด มิฉะนั้น คำสั่งแสดงสมนามสำหรับ <i>Name</i> ที่ระบุ ถ้ามีการระบุ <i>WordList</i> คำสั่งนี้จะกำหนดค่าของ <i>WordList</i> ให้กับสมนาม <i>Name Name</i> สมนามที่ระบุไม่สามารถเป็น <code>alias</code> หรือ <code>unalias</code>
<code>bg [%Job ...]</code>	วางงานปัจจุบันหรืองานที่ระบุโดย <i>Job</i> ไว้ใน พื้นหลัง โดยทำงานต่อไปถ้างานหยุด
<code>break</code>	ทำการรันต่อไปหลัง <code>end</code> ของคำสั่ง <code>foreach</code> หรือ <code>while</code> ล้อมรอบซึ่งใกล้เคียงที่สุด
<code>breaksw</code>	พักจากคำสั่ง <code>switch</code> ทำต่อไป หลังจากคำสั่ง <code>endsw</code>
<code>case Label:</code>	กำหนด <i>Label</i> ในคำสั่ง <code>switch</code>
<code>cd[Name]</code>	เทียบเท่ากับคำสั่ง <code>chdir</code> (โปรดดูที่ คำอธิบายต่อไปนี้)
<code>chdir [Name]</code>	เปลี่ยนไดเรกทอรีปัจจุบันเป็นไดเรกทอรีที่ระบุโดยตัวแปร <i>Name</i> ถ้าคุณไม่ได้ระบุ <i>Name</i> คำสั่งจะเปลี่ยนเป็นไดเรกทอรีโฮม ของคุณ หากค่าของตัวแปร <i>Name</i> ไม่ใช่ไดเรกทอรีย่อยของไดเรกทอรีปัจจุบันและไม่ได้ขึ้นต้นด้วย <code>/</code> , <code>./</code> , หรือ <code>../</code> เซลล์จะตรวจสอบแต่ละคอมโพเนนต์ของตัวแปรเซลล์ <i>cdpath</i> เพื่อความมีไดเรกทอรีย่อยซึ่งตรงกับตัวแปร <i>Name</i> อยู่หรือไม่ ถ้าตัวแปร <i>Name</i> เป็นตัวแปรเซลล์ที่มีค่าซึ่ง ขึ้นต้นด้วยเครื่องหมายทับ (<code>/</code>) เซลล์จะพยายามตรวจสอบเพื่อดูว่าเป็น ไดเรกทอรีหรือไม่ คำสั่ง <code>chdir</code> เทียบเท่ากับ คำสั่ง <code>cd</code>
<code>continue</code>	ทำการดำเนินการต่อไปเมื่อ <code>end</code> ของคำสั่ง <code>while</code> หรือ <code>foreach</code> ล้อมรอบซึ่งใกล้เคียงที่สุด
<code>default:</code>	เลเบลตัวพิมพ์ <code>default</code> ในคำสั่ง <code>switch default</code> ควรมาหลังจากเลเบล <code>case</code> อื่นทั้งหมด
<code>dirs</code>	แสดงไดเรกทอรีสแต็ก

ไอเท็ม	คำอธิบาย
echo	เขียนสตริงอักขระไปที่เอาต์พุตมาตรฐานของเซลล์
else	รันคำสั่งที่ตามหลัง else ที่สองใน ลำดับคำสั่ง if (Expression) then ...else if (Expression2) then ... else ... endif หมายเหตุ: คำสั่ง else เป็น คำสั่งในตัว csh เมื่อใช้ if (expr) then ..else ...endif ถ้า (expr) เป็นจริง จะมีการดำเนินการคำสั่งต่างๆ จนถึงคำสั่ง else ถ้า (expr) เป็นเท็จ จะมีการดำเนินการคำสั่งระหว่างคำสั่ง else และ endif ขอมูลใดๆ ในเครื่องหมายคำพูดเดี่ยวจะถูกใช้ตามตัวอักษรและไม่ตีความ
end	ตั้งค่าตัวแปร Name สำหรับแต่ละ สมาชิกที่ระบุโดยตัวแปร List และรันลำดับของ Commands อย่างต่อเนื่องระหว่าง foreach และ คำสั่ง end ที่ตรงกัน คำสั่ง foreach และ end ต้องปรากฏขึ้นโดยลำพังบนบรรทัดที่แยกกัน ใช้คำสั่ง continue เพื่อทำการวนซ้ำต่อไป และใช้คำสั่ง break เพื่อสิ้นสุดการวนซ้ำ ก่อนกำหนด เมื่ออ่านคำสั่ง foreach จาก เทอร์มินัล C เซลล์จะพร้อมด้วย ? เพื่อให้สามารถป้อน Commands ได้ คำสั่งภายในลูปที่พร้อมด้วย ? จะไม่ถูกใส่ไว้ในรายการประวัติ
endif	ถ้าตัวแปร Expression เป็นจริง ระบบจะรัน Commands ที่ตามหลังคำสั่ง then ลำดับแรก ถ้า else if Expression2 เป็น จริง ระบบจะรัน Commands ที่ตามหลังคำสั่ง then ที่สอง ถ้า else if Expression2 เป็น เท็จ ระบบจะรัน Commands ที่ตามหลัง else สามารถใช้คู่ else if จำนวนเท่าใดก็ได้ ต้องการคำสั่ง endif เพียง คำสั่งเดียวเท่านั้น เช็กเมนต์ else เป็นทางเลือก คำว่า else และ endif สามารถใช้ที่ตอนต้นของเส้นอินพุตเท่านั้น เช็กเมนต์ if ต้องปรากฏขึ้นโดยลำพังบนเส้นอินพุตของตนเองหรือหลังจากคำสั่ง else
endsw	จับคู่เลเบล ตัวพิมพ์ แต่ละรายการกับ ค่าของตัวแปร string อย่างต่อเนื่อง string เป็น คำสั่งและชื่อไฟล์ที่ขยายก่อน ใช้อักขระการจัดรูปแบบ *, ? และ [. . .] ในเลเบล ตัวพิมพ์ ซึ่งเป็นส่วนขยายตัวแปร ถ้าไม่มีเลเบลใดตรงกันก่อนพบ ดีฟอลต์ เลเบล การดำเนินการจะเริ่มขึ้นหลังจาก ดีฟอลต์ เลเบล เลเบลตัวพิมพ์ และ ดีฟอลต์ เลเบลต้องปรากฏขึ้นที่ตอนต้นของบรรทัด คำสั่ง breaksw ส่งผลให้การดำเนินการทำต่อไปหลังจากคำสั่ง endsw มิฉะนั้น การควบคุมอาจลดลงผ่านทางเลเบล ตัวพิมพ์ และ ดีฟอลต์ เลเบล ดังเช่น ในภาษาการเขียนโปรแกรม C หากไม่มีเลเบลตรงกันและไม่มี ดีฟอลต์ การดำเนินการจะทำต่อไปหลังจากคำสั่ง endsw
eval Parameter . . .	อ่านค่าของตัวแปร Parameter เป็นอินพุตไปยังเซลล์ และรันคำสั่งที่ได้ในคอนเท็กซ์ของเซลล์ปัจจุบัน ใช้คำสั่งนี้เพื่อรันคำสั่งที่สร้างขึ้นอันเป็นผลมาจาก การทดแทนคำสั่งหรือตัวแปร เนื่องจากเกิดการแจงส่วนก่อนการทดแทนเหล่านี้
exec Command	รัน Command ที่ระบุใน ตำแหน่งของเซลล์ปัจจุบัน
exit (Expression)	ออกจากเซลล์ด้วยค่าของตัวแปรเซลล์ status (ถ้าไม่ได้ระบุ Expression) หรือด้วยค่าของ Expression ที่ระบุ อย่างใดอย่างหนึ่ง
fg [%Job ...]	นำงานปัจจุบันหรืองานที่ระบุโดย Job ไปยัง พื้นหน้า โดยทำงานต่อไปถ้างานหยุด
foreach Name (List) Command. . .	ตั้งค่าตัวแปร Name สำหรับแต่ละสมาชิกที่ระบุ โดยตัวแปร List และลำดับของคำสั่งอย่างต่อเนื่อง จนกว่าถึงคำสั่ง end
glob List	แสดง List โดยใช้ประวัติ ตัวแปร และส่วนขยายชื่อไฟล์ วางอักขระ null ระหว่างคำและไม่รวม ปิดแคร์ที่ตอนท้าย
goto Word	รันต่อไปหลังจากบรรทัดที่ระบุโดยตัวแปร Word Word ที่ระบุคือชื่อไฟล์และคำสั่งซึ่งขยายเพื่อให้คำสั่งของรูปแบบที่ระบุโดยตัวแปร Label: เซลล์กรอกลับอินพุตมากที่สุดเท่าที่เป็นไปได้และค้นหาบรรทัดของ รูปแบบ Label: ซึ่งอาจมีพื้นที่ว่าง หรือแท็บนำหน้า
hashstat	แสดงสถิติที่บ่งชี้ความสำเร็จของตาราง hash ที่คำสั่งระบุตำแหน่ง
history [-r -h] [n]	แสดงรายการเหตุการณ์ประวัติ เหตุการณ์ที่เก่าสุดจะแสดงขึ้น ก่อน ถ้าคุณระบุจำนวน n เฉพาะจำนวนที่ระบุของเหตุการณ์ล่าสุดเท่านั้นจะแสดงขึ้น แฟล็ก -r ย้อนกลับ ลำดับในการแสดงเหตุการณ์เพื่อให้เหตุการณ์ล่าสุดแสดงขึ้น ก่อน แฟล็ก -h แสดงรายการประวัติโดยไม่มีหมายเลขนำ ใช้แฟล็กนี้เพื่อจัดทำไฟล์ที่เหมาะสมสำหรับการใช้กับแฟล็ก -h ของ คำสั่ง source

ไอเท็ม	คำอธิบาย
<code>if (Expression) Command</code>	รัน <i>Command</i> ที่ระบุ (รวมถึง อาร์กิวเมนต์ของคำสั่ง) ถ้า <i>Expression</i> ที่ระบุเป็นจริง การทดแทน ตัวแปร บนตัวแปร <i>Command</i> เกิดขึ้นก่อน ในเวลาเดียวกันกับส่วนที่เหลือของคำสั่ง <i>if Command</i> ที่ระบุต้อง เป็นคำสั่งแบบง่าย (ไม่ใช่ ไฟล์ไลน์ รายการคำสั่ง หรือรายการคำสั่งในวงเล็บ) หมายเหตุ: การเปลี่ยนทิศทางอินพุตและเอาต์พุต จะเกิดขึ้นแม้ว่าตัวแปร <i>Expression</i> เป็นเท็จ และไม่ ได้ดำเนินการ <i>Command</i>
<code>jobs [-l]</code>	แสดงรายการงานที่ใช้งานอยู่ ด้วยแฟล็ก <code>-l</code> (L ตัวพิมพ์เล็ก) คำสั่ง <code>jobs</code> จะแสดงรายการ IDs โปรเซสเพิ่มเติมจาก หมายเลขงานและชื่อ
<code>kill -l [[-Signal] % Job...!PID...]</code>	ส่งสัญญาณ TERM (ยุติ) หรือสัญญาณที่ระบุโดย <i>Signal</i> ไปยัง <i>Job</i> หรือ <i>PID</i> (โปรเซส) ที่ระบุ ระบุ สัญญาณโดยใช้หมายเลขหรือชื่อ (ตั้งที่กำหนดในไฟล์ <code>/usr/include/sys/signal.h</code> stripped ของคำสั่ง เดิมหน้า SIG) แฟล็ก <code>-l</code> (L ตัวพิมพ์เล็ก) แสดงรายการชื่อสัญญาณ
<code>limit [-h] [Resource [Max-Use]]</code>	จำกัดการใช้รีซอร์สที่ระบุโดยโปรเซสปัจจุบันและแต่ละ โปรเซสที่สร้างขึ้น ซัดจำกัดรีซอร์สโปรเซสมีการ กำหนดไว้ใน ไฟล์ <code>/etc/security/limits</code> รีซอร์สที่ควบคุมได้ คือเวลา central processing unit (CPU) ขนาดไฟล์ ขนาดข้อมูล ขนาดดัมพ์ หลัก และการใช้หน่วยความจำ ค่าสูงสุดที่ใช้ได้สำหรับรีซอร์สเหล่านี้มี การตั้งค่า ด้วยคำสั่ง <code>mkuser</code> เมื่อเพิ่มผู้ใช้ลงในระบบ และเปลี่ยนด้วยคำสั่ง <code>chuser</code> ซัดจำกัดมีการจัดหมวดหมู่เป็นแบบซอฟต์แวร์หรือฮาร์ด อย่างใดอย่างหนึ่ง ผู้ใช้อาจเพิ่มซัดจำกัดซอฟต์แวร์ของตนได้จนถึง เพดานที่กำหนดโดยซัดจำกัดฮาร์ด คุณต้องมีสิทธิ์ผู้ใช้ราก เพื่อเพิ่มซัดจำกัดซอฟต์แวร์ให้มากกว่าซัดจำกัดฮาร์ด หรือเพื่อเปลี่ยนซัดจำกัดฮาร์ด แฟล็ก <code>-h</code> แสดงซัดจำกัดฮาร์ดแทนซัดจำกัดซอฟต์แวร์ ถ้าไม่ได้รับพารามิเตอร์ <i>Max-Use</i> คำสั่ง <code>limit</code> จะแสดงซัดจำกัดปัจจุบัน ของรีซอร์สที่ระบุ ถ้าไม่ได้รับพารามิเตอร์ <i>Resource</i> คำสั่ง <code>limit</code> จะแสดงซัดจำกัดปัจจุบัน ของรีซอร์สทั้งหมด สำหรับข้อมูลเพิ่มเติมเกี่ยวกับรีซอร์สที่ควบคุม โดยคำสั่งย่อย <code>limit</code> ให้ดูที่รุ่นย่อย <code>getrlimit</code> , <code>setrlimit</code> , หรือ <code>vlimit</code> ใน <i>Technical Reference: Base Operating System and Extensions, Volume 1</i> พารามิเตอร์ <i>Max-Use</i> สำหรับเวลา CPU มีการระบุในรูปแบบ <code>hh:mm:ss</code> พารามิเตอร์ <i>Max-Use</i> สำหรับ รีซอร์สอื่นสามารถเลือกระบุ เป็นตัวเลขจุดลอยตัวหรือเลขจำนวนเต็มตามด้วย คาสเกล คาสเกลเป็น k หรือกิโลไบต์ (1024 ไบต์), m หรือเมกะไบต์ หรือ b หรือบล็อก (หน่วยที่ใช้โดยรูทีนย่อย <code>ulimit</code> ดัง อธิบาย ใน <i>Technical Reference: Base Operating System and Extensions, Volume 2</i>) หากคุณไม่ได้ ระบุคาสเกล จะใช้ k สำหรับรีซอร์สทั้งหมด สำหรับ ทั้งชื่อรีซอร์สและคาสเกล สามารถใช้คำเติมหน้าที่ไม่ ชัดเจนของชื่อได้ หมายเหตุ: คำสั่งนี้ จำกัดหน่วยความจำฟิสิกัล (การใช้หน่วยความจำ) ที่มีอยู่สำหรับโปรเซสเฉพาะถ้า มีการช่วงซึ่งหน่วยความจำระบบโดยโปรเซสที่ใช้งานอยู่อื่นเท่านั้น
<code>login</code>	สิ้นสุดเซสชันการล็อกอินและแทนที่เซสชันการล็อกอินด้วยอินสแตนซ์ของคำสั่ง <code>/usr/bin/login</code> นี้เป็นวิธี หนึ่งในวิธีการล็อกเอาต์ (รวมไว้เพื่อให้เข้ากันได้กับคำสั่ง <code>ksh</code> และ <code>bsh</code>)
<code>logout</code>	สิ้นสุดเซสชันการล็อกอิน ต้องใช้คำสั่งนี้ถ้ามีการตั้งค่าอ็อปชัน <code>ignoreeof</code>
<code>nice [+n] [Command]</code>	ถ้าไม่ได้รับค่า ให้ตั้งค่าลำดับความสำคัญของคำสั่ง ที่รันในเชลล์นี้เป็น 24 ถ้ามีการระบุแฟล็ก <code>+n</code> ให้ตั้ง ค่าลำดับความสำคัญบวกตัวเลขที่ระบุ ถ้ามีการระบุแฟล็ก <code>+n</code> และ <i>Command</i> ให้รัน <i>Command</i> ที่ ลำดับ ความสำคัญ 24 บวกตัวเลขที่ระบุ หากคุณมีสิทธิ์ผู้ใช้ราก คุณสามารถรันคำสั่ง <code>nice</code> ด้วยตัวเลขค่าลบได้ <i>Command</i> รันใน เชลล์ย่อยเสมอ และใช้ซัดจำกัดที่วางบนคำสั่งในคำสั่ง <code>if</code> แบบง่าย
<code>nohup [Command]</code>	ส่งผลให้ละเว้น การไม่ตอบสนอง สำหรับส่วนที่เหลือของ สคริปต์เมื่อไม่มีการระบุ <i>Command</i> ถ้ามีการ ระบุ <i>Command</i> ส่งผลให้ <i>Command</i> ที่ระบุซึ่งจะรันด้วย การไม่ตอบสนอง ถูกละเว้นไป เมื่อต้องการรัน ไฟล์ไลน์หรือรายการของคำสั่ง ให้วางไฟล์ไลน์หรือรายการไว้ใน เชลล์สคริปต์ ให้สิทธิ์ดำเนินการในสคริปต์ และใช้เชลล์สคริปต์เป็นค่าของ ตัวแปร <i>Command</i> โปรเซสทั้งหมดที่รันใน พื้นหลังด้วยเครื่องหมาย ampersand (&) จะได้รับการป้องกันอย่างมี ประสิทธิภาพจากการส่งสัญญาณ การไม่ตอบสนอง เมื่อคุณลี้ ออกเอาต์ อย่างไรก็ตาม โปรเซสเหล่านี้ ยังคงต้องถูกส่ง การไม่ตอบสนอง อย่างชัดเจนกว่าใช้คำสั่ง <code>nohup</code>
<code>notify [%Job...]</code>	ส่งผลให้เชลล์แจ้งเตือนคุณแบบอะซิงโครนัสเมื่อสถานะ ของงานปัจจุบันหรือ <i>Job</i> ที่ระบุเปลี่ยนแปลง โดยปกติ เชลล์แสดงการแจ้งเตือนก่อนหน้าที่จะแสดงเชลล์พร้อมต์เพียงเล็กน้อย คุณลักษณะนี้เป็นแบบ อัปเดตอัตโนมัติถ้ามีการตั้งค่าตัวแปรเชลล์ <code>notify</code>

ไอเท็ม	คำอธิบาย
<code>onintr [-l Label]</code>	ควบคุมการดำเนินการของเซลล์เมื่อเกิดการขัดจังหวะ ถ้าไม่ได้รับอาร์กิวเมนต์ ระบบจะเรียกคืนการดำเนินการดีฟอลต์ของเซลล์เมื่อเกิดการขัดจังหวะ ซึ่งสิ้นสุดเซลล์สคริปต์หรือกลับไปยังระดับอินพุตคำสั่ง ถ้ามีการระบุแฟล็ก -l ส่งผลให้การขัดจังหวะทั้งหมดถูกละเว้น ถ้ามีการระบุ <i>Label</i> ส่งผลให้เซลล์รันคำสั่ง <code>goto Label</code> เมื่อเซลล์ถูกขัดจังหวะหรือเมื่อโปรแกรมขยับสิ้นสุดลงเนื่องจาก การขัดจังหวะ ในทุกกรณี ถ้าเซลล์กำลังรันอย่างเป็นอิสระและการขัดจังหวะ ถูกละเว้น รูปแบบทั้งหมดของคำสั่ง <code>onintr</code> จะไม่มี ความหมาย การขัดจังหวะยังคงอยู่จนต่อไปโดยเซลล์และคำสั่งที่เรียกใช้ทั้งหมด
<code>popd [+n]</code>	แสดงไต่เรียกทอริสแตกและเปลี่ยนเป็นไต่เรียกทอริบสนุดใหม่ ถ้าคุณระบุตัวแปร <i>n</i> คำสั่งจะทิ้งรายการ n^{th} ในสแตก อิลิเมนต์ของไต่เรียกทอริสแตกถูกกำหนดหมายเลขจากด้านบนสุด โดยเริ่มต้นที่ 0
<code>pushd [+nName]</code>	แลกเปลี่ยนสองอิลิเมนต์บนสุดของไต่เรียกทอริสแตก โดยไม่มีอาร์กิวเมนต์ ด้วยตัวแปร <i>Name</i> คำสั่ง เปลี่ยนเป็นไต่เรียกทอริใหม่และกดไต่เรียกทอริปัจจุบันเก่า (ตั้งที่กำหนด ในตัวแปรเซลล์ <code>cwd</code>) ลงบน ไต่เรียกทอริสแตก ถ้าคุณระบุตัวแปร <i>n</i> คำสั่งจะหมุนคอมโพเนนต์ n^{th} ของไต่เรียกทอริสแตกไปเป็นอิลิเมนต์บนสุดและเปลี่ยนเป็นอิลิเมนต์นั้น สมาชิกของไต่เรียกทอริสแตก ถูกกำหนดหมายเลขจากด้านบนสุดโดยเริ่มต้นที่ 0
<code>rehash</code>	ส่งผลให้เกิดการคำนวณตาราง hash ภายในของเนื้อหา ของไต่เรียกทอริในตัวแปรเซลล์ <i>path</i> อีกครั้ง การดำเนินการนี้ เป็นสิ่งจำเป็นถ้ามีการเพิ่มคำสั่งใหม่ลงในไต่เรียกทอริในตัวแปรเซลล์ <i>path</i> ในขณะที่คุณลึกลง อยู่นอยู่ คำสั่ง <code>rehash</code> เป็นสิ่งจำเป็น เฉพาะถ้ามีการเพิ่มคำสั่งลงในไต่เรียกทอริของตนเองของผู้ใช้รายใด รายหนึ่ง หรือถ้าบางคนเปลี่ยนเนื้อหาของไต่เรียกทอริระบบอย่างใดอย่างหนึ่ง
<code>repeat Count Command</code>	รัน <i>Command</i> ที่ระบุ ภายในไต่ซ้ำจำกัดเดียวกันกับคำสั่งในคำสั่ง <code>if</code> แบบง่าย และจำนวนครั้งที่ระบุโดย <i>Count</i> หมายเหตุ: การเปลี่ยนทิศทาง I/O เกิดขึ้นเพียงครั้งเดียวเท่านั้น แม้ว่าตัวแปร <i>Count</i> เป็น 0
<code>set [[Name[n]] [= Word]] [Name = (List)]</code>	แสดงค่าของตัวแปรเซลล์ทั้งหมดเมื่อใช้โดยไม่มี อาร์กิวเมนต์ ตัวแปรที่มีค่าเป็นค่าเดียวกันมากกว่าหนึ่งค่า แสดงขึ้นเป็น รายการค่าในเครื่องหมายวงเล็บ ถ้ามีการระบุ <i>Name</i> อย่างเดียว <i>C</i> เซลล์จะตั้งค่าตัวแปร <i>Name</i> เป็นสตริง <code>null</code> มิฉะนั้น ตั้งค่า <i>Name</i> เป็นค่าของตัวแปร <i>Word</i> หรือตั้งค่าตัวแปร <i>Name</i> เป็นรายการ คำที่ระบุโดย ตัวแปร <i>List</i> เมื่อมีการระบุ <i>n</i> คอมโพเนนต์ n^{th} ของตัวแปร <i>Name</i> มีการตั้งค่าเป็นค่าของ ตัวแปร <i>Word</i> คอมโพเนนต์ n^{th} ต้องมีอยู่แล้ว ในทุกกรณี ค่าคือคำสั่งและชื่อไฟล์ที่ขยาย อาร์กิวเมนต์เหล่านี้ อาจมีการทำซ้ำเพื่อกำหนดหลายค่าในคำสั่ง <code>set</code> เดียว อย่างไรก็ตาม การขยายตัวแปรเกิดขึ้นสำหรับ อาร์กิวเมนต์ทั้งหมดก่อนค่าที่ตั้งใดๆ เกิดขึ้น
<code>setenv Name Value</code>	ตั้งค่าของตัวแปรสถานะแวดล้อมที่ระบุโดยตัวแปร <i>Name</i> เป็น <i>Value</i> สตริงเดียว ตัวแปรสถานะแวดล้อม ที่ใช้กันมากที่สุดคือ <i>USER</i> , <i>TERM</i> , <i>HOME</i> , และ <i>PATH</i> ถูกอิมพอร์ตโดยอัตโนมัติและ เอ็กซ์พอร์ตจาก ตัวแปร <i>C</i> เซลล์ <i>user</i> , <i>term</i> , <i>home</i> , และ <i>path</i> ไม่จำเป็นต้องใช้คำสั่ง <code>setenv</code> สำหรับตัวแปรเหล่านี้
<code>shift [Variable]</code>	เลื่อนสมาชิกของตัวแปรเซลล์ <i>argv</i> หรือ <i>Variable</i> ที่ระบุไปทางซ้าย ข้อผิดพลาดเกิดขึ้นถ้าตัวแปรเซลล์ <i>argv</i> หรือ <i>Variable</i> ที่ระบุ ไม่มีการตั้งค่า หรือมีค่าเป็นค่าน้อยกว่าหนึ่งค่า
<code>source[-h] Name</code>	อ่านคำสั่งที่เขียนในไฟล์ <i>Name</i> คุณสามารถจัดกลุ่มคำสั่ง <code>source</code> ได้อย่างไรก็ตาม ถ้าจัดกลุ่ม คำสั่งลึกลงไป เซลล์อาจขาดตัวอธิบายไฟล์ ข้อผิดพลาดใน คำสั่ง <code>source</code> ที่ระดับใดๆ ทำให้คำสั่ง <code>source</code> ที่จัดกลุ่มทั้งหมดสิ้นสุดลง โดยปกติ อินพุตในระหว่างคำสั่ง <code>source</code> ไม่มีการวางไว้ใน รายการประวัติ แฟล็ก <code>-h</code> ส่งผลให้วางคำสั่งไว้ใน รายการประวัติโดยไม่ดำเนินการคำสั่งนั้น
<code>stop [%Job ...]</code>	หยุดงานปัจจุบันหรือ <i>Job</i> ที่ระบุซึ่งกำลังรัน ในพื้นหลัง
<code>suspend</code>	หยุดเซลล์ราวกับว่าได้รับสัญญาณ STOP
<code>switch (string)</code>	เริ่มต้นลำดับคำสั่ง <code>switch (String) case String : ... breaksw default: ... breaksw endsw</code> ลำดับคำสั่งนี้ จับคู่เลเบลตัวพิมพ์แต่ละรายการกับค่าของ ตัวแปร <i>String</i> อย่างต่อเนื่อง ถ้าไม่มีเลเบลใดตรงกัน ก่อน พบดีฟอลต์เลเบล การดำเนินการจะเริ่มขึ้นหลังจากดีฟอลต์เลเบล

ไอเท็ม	คำอธิบาย
<code>time [Command]</code>	<p>คำสั่ง <code>time</code> ควบคุมช่วงเวลาอัตโนมัติ ของคำสั่ง ถ้าคุณไม่ได้ระบุตัวแปร <code>Command</code> คำสั่ง <code>time</code> จะแสดงข้อมูลสรุปของเวลาที่ใช้โดยเซลล์นี้ และชานของเซลล์ ถ้าคุณระบุคำสั่งพร้อมด้วยตัวแปร <code>Command</code> คำสั่งจะถูกจับเวลา จากนั้น เซลล์แสดงข้อมูลสรุปเวลา ตั้งอธิบายไว้ภายในตัวแปรเซลล์ <code>time</code> ถ้าจำเป็น เซลล์เพิ่มเติมจะถูกสร้างขึ้นเพื่อแสดงสถิติเวลา เมื่อคำสั่งเสร็จสมบูรณ์</p> <p>ตัวอย่างต่อไปนี้จะใช้ <code>time</code> พร้อมกับ คำสั่ง <code>sleep</code>:</p> <pre>time sleep</pre> <p>เอาต์พุต จากคำสั่งนี้มีลักษณะคล้ายกับตัวอย่างต่อไปนี้:</p> <pre>0.0u 0.0s 0:00 100% 44+4k 0+0io 0pf+0w</pre> <p>ฟิลด์ เอาต์พุตเป็นดังนี้:</p> <ul style="list-style-type: none"> ที่หนึ่ง จำนวนวินาทีของเวลา CPU ที่ใช้สำหรับโปรแกรมผู้ใช้ ที่สอง จำนวนวินาทีของเวลา CPU ที่ใช้โดยเคอร์เนลในนามของ โปรแกรมผู้ใช้ ที่สาม เวลาที่ผ่านไป (นาฬิกาติดผนัง) สำหรับคำสั่ง ที่สี่ เวลา CPU ผู้ใช้ทั้งหมดบวกเวลาระบบ เป็นเปอร์เซ็นต์ของเวลาที่ผ่านไป ที่ห้า จำนวนเฉลี่ยของหน่วยความจำแบบแบ่งใช้ที่ใช้ บวกจำนวนเฉลี่ยของพื้นที่ว่าง ข้อมูลแบบไม่แบ่งใช้ที่ใช้ ในหน่วยกิโลไบต์ ที่หก จำนวนบล็อกการดำเนินงานอินพุตและเอาต์พุต ที่เจ็ด ความผิดพลาดของเพจบวกจำนวนของการสับเปลี่ยน
<code>umask [Value]</code>	พิจารณาสิทธิในไฟล์ <code>Value</code> นี้ พร้อมกับสิทธิของการสร้างโปรแกรม เป็นตัวกำหนดสิทธิของไฟล์ เมื่อไฟล์ถูกสร้างขึ้น ค่าดีฟอลต์คือ 022 ค่าที่ตั้งปัจจุบันจะแสดงขึ้น ถ้าไม่มีการระบุ <code>Value</code>
<code>unalias * Pattern</code>	ทั้งสมนามทั้งหมดที่มีชื่อตรงกับตัวแปร <code>Pattern</code> สมนามทั้งหมดถูกลบออกโดยคำสั่ง <code>unalias *</code> การขาดสมนามไม่ทำให้เกิดข้อผิดพลาด
<code>unhash</code>	ปิดใช้งานการใช้ตาราง hash ภายในเพื่อระบุตำแหน่งโปรแกรมที่คำสั่งรัน
<code>unlimit [-h][Resource]</code>	<p>ลบขีดจำกัดบนตัวแปร <code>Resource</code> ถ้าไม่มีการระบุตัวแปร <code>Resource</code> ขีดจำกัดรีซอร์สทั้งหมด จะถูกลบออกให้ดูที่คำอธิบายของคำสั่ง <code>limit</code> สำหรับ รายชื่อของ <code>Resource</code></p> <p>แฟล็ก <code>-h</code> ลบ ขีดจำกัดฮาร์ดที่สอดคล้องกัน เฉพาะผู้ใช้ที่มีสิทธิผู้ใช้รากเท่านั้นสามารถเปลี่ยนขีดจำกัดฮาร์ดได้</p>
<code>unset * Pattern</code>	ลบตัวแปรทั้งหมดที่มีชื่อตรงกับตัวแปร <code>Pattern</code> ใช้ <code>unset *</code> เพื่อลบตัวแปรทั้งหมด การไม่ตั้งค่า ตัวแปรไม่ทำให้เกิดข้อผิดพลาด
<code>unsetenvPattern</code>	ลบตัวแปรทั้งหมดที่มีชื่อตรงกับ <code>Pattern</code> ที่ระบุออกจากสภาวะแวดล้อม (โปรดดูที่คำสั่งในตัว <code>setenv</code>)
<code>wait</code>	รอกานพื้นหลังทั้งหมด ถ้าเซลล์เป็นแบบโต้ตอบ <code>INTERRUPT</code> (โดยปกติคือลำดับคีย์ <code>Ctrl-C</code>) จะทำลายการรอคอย จากนั้น เซลล์แสดงชื่อและหมายเลขงานของงานทั้งหมดที่ทราบว่าจะอยู่
<code>while (Expression) Command. . . end</code>	ประเมิน <code>Commands</code> ระหว่าง <code>while</code> และ คำสั่ง <code>end</code> ที่ตรงกัน ในขณะที่นิพจน์ซึ่งระบุโดย ตัวแปร <code>Expression</code> ประเมินค่าที่ไม่ใช่ศูนย์ คุณสามารถใช้คำสั่ง <code>break</code> เพื่อสิ้นสุด และใช้คำสั่ง <code>continue</code> เพื่อทำการวนซ้ำ ก่อนกำหนดต่อไป คำสั่ง <code>while</code> และ <code>end</code> ต้องปรากฏขึ้นโดยลำดับบนเส้นอินพุตของคำสั่ง ถ้าอินพุตมาจากเทอร์มินัล พร้อมต์ที่เกิดขึ้น หลังจาก <code>while (Expression)</code> จะคล้ายกับคำสั่ง <code>foreach</code>

ไอเท็ม	คำอธิบาย
@ [Name[n] = Expression]	<p>แสดงค่าของตัวแปรเซลล์ทั้งหมดเมื่อใช้โดยไม่มี อาร์กิวเมนต์ มิฉะนั้น ตั้งค่าที่ระบุโดยตัวแปร Name เป็นค่าของตัวแปร Expression ถ้านิพจน์ มี <, >, &, หรืออักขระ ต้องวางนิพจน์ส่วนนี้ไว้ภายในวงเล็บ เมื่อมีการระบุ n คอมโพเนนต์ nth ของตัวแปร Name จะมีการตั้งค่าเป็นตัวแปร Expression ทั้งตัวแปร Name และคอมโพเนนต์ nth ต้องมีอยู่แล้ว</p> <p>มีตัวดำเนินการภาษา C เช่น *= และ += ช่องว่างที่คั่นตัวแปร Name ออกจาก ตัวดำเนินการกำหนดค่าเป็นทางเลือกอย่างไรก็ตาม ต้องใช้พื้นที่ว่างเพื่อคั่น คอมโพเนนต์ของตัวแปร Expression ถ้าไม่ใช้พื้นที่ว่าง คั่น จะถูกอ่านเป็นคำเดียว ตัวดำเนินการเติมหลังพิเศษ เครื่องหมายบวกคู่ (++) และยัติภังค์คู่ (--) จะเพิ่มและลดค่าของตัวแปร Name ตามลำดับ</p>

นิพจน์และตัวดำเนินการ C เซลล์:

คำสั่งในตัว @ และคำสั่ง exit, if, และ while ยอมรับนิพจน์ที่มีตัวดำเนินการคล้ายกับ ตัวดำเนินการของภาษา C โดยมีลำดับความสำคัญเหมือนกัน

ตัวดำเนินการที่มีอยู่มีดังต่อไปนี้:

ตัวดำเนินการ	ความหมาย
()	เปลี่ยนลำดับความสำคัญ
~	ส่วนประกอบ
!	การปฏิเสธ
*/%	คูณ ทหาร มอดุโล
+ -	บวก, ลบ
<< >	เลื่อนไปทางซ้าย, เลื่อนไปทางขวา
<=><>	ตัวดำเนินการสัมพันธ์
== != =~ !~	การเปรียบเทียบสตริง/การจับคู่รูปแบบ
&	bitwise AND
^	bitwise exclusive OR
	bitwise inclusive OR
&&	ตรรกะ AND
	ตรรกะ OR

ในรายการก่อนหน้านี้ ลำดับความสำคัญของตัวดำเนินการลดลงใน รายการ (จากซ้ายไปขวา จากบนลงล่าง)

หมายเหตุ: ตัวดำเนินการ + และ - มีความสัมพันธ์จากทางด้านขวา ตัวอย่างเช่น การประเมิน a + b - c มี การทำดังนี้:

$$a + (b - c)$$

ไม่ใช่:

$$(a + b) - c$$

ตัวดำเนินการ ==, !=, =~, และ !~ เปรียบเทียบอาร์กิวเมนต์เป็นสตริง ข้อมูลอื่นทั้งหมด ดำเนินงานเป็นตัวเลข ตัวดำเนินการ =~ และ !~ คล้ายกับ == และ != ยกเว้นว่า ด้านขวาสุดคือ รูปแบบ ซึ่งตรงกับตัวถูกดำเนินการด้านซ้ายสุด ลักษณะนี้ลดความจำเป็นในการใช้คำสั่ง switch ในเซลล์โพรซีเจอร์

นอกจากนี้ ยังมีตัวดำเนินการทางตรรกะ or (||) และ and (&&) ด้วย สามารถใช้ตัวดำเนินการทางตรรกะเพื่อตรวจสอบช่วงของตัวเลข ดังเช่นใน ตัวอย่างต่อไปนี้:

```
if ($#argv > 2 && $#argv < 7) then
```

ในตัวอย่างก่อนหน้านี้ จำนวนของอาร์กิวเมนต์ต้องมากกว่า 2 และ น้อยกว่า 7

สตริงที่ขึ้นต้นด้วยศูนย์ (0) ถือว่าเป็นตัวเลขฐานแปด Null หรืออาร์กิวเมนต์ที่ขาดไปถือว่าเป็น 0 สำหรับนิพจน์ทั้งหมดที่ส่งผลให้เกิด สตริงแสดงถึงตัวเลขฐานสิบ โปรดทราบว่าสองคอมโพเนนต์ของนิพจน์หนึ่งสามารถ ปรากฏขึ้นในค่าเดียวกันได้ ยกเว้นว่าสิ่งที่ถัดจากคอมโพเนนต์ของนิพจน์ มีความสำคัญทางวากยสัมพันธ์ต่อ parser (& | < > ()) คอมโพเนนต์ของนิพจน์ควรมีช่องว่างนำหน้าและตามหลัง

นอกจากนี้ สิ่งที่มีอยู่ในนิพจน์เป็นตัวถูกดำเนินการหลักคือการดำเนินการคำสั่งที่ใส่ไว้ในเครื่องหมายวงเล็บ () และการสอบถามไฟล์ของรูปแบบ (-operator Filename) โดยที่ operator คือ ข้อมูลอย่างใดอย่างหนึ่งต่อไปนี้:

ไอเท็ม	คำอธิบาย
r	สิทธิการอ่าน
w	สิทธิการเขียน
x	สิทธิการดำเนินการ
e	การมีอยู่
o	ความเป็นเจ้าของ
z	दानศูนย์
f	ไฟล์ปกติ
d	ไดเรกทอรี

Filename ที่ระบุคือคำสั่งและชื่อไฟล์แบบขยาย จากนั้น ทดสอบเพื่อดูว่ามีความสัมพันธ์ที่ระบุ กับผู้ใช้จริงหรือไม่ ถ้า Filename ไม่มีอยู่หรือไม่สามารถเข้าถึงได้ การสอบถามทั้งหมดจะส่งคืน false(0) หากคำสั่งรันได้สำเร็จ การสอบถามจะส่งคืนค่าเป็น true(1) มิฉะนั้น ถ้าคำสั่ง ล้มเหลว การสอบถามจะส่งคืนค่าเป็น false(0) หากต้องการ ข้อมูลสถานะโดยละเอียดเพิ่มเติม ให้รันคำสั่งภายนอกนิพจน์ แล้วตรวจสอบตัวแปรเซลล์ status

การทดแทนคำสั่งในเซลล์ C:

ใน การทดแทนคำสั่ง เซลล์ดำเนินการคำสั่งที่ระบุ และแทนที่คำสั่งนั้นด้วยเอาต์พุต

เมื่อต้องการทำการทดแทนคำสั่งใน C เซลล์ ให้ใส่ คำสั่งหรือสตริงคำสั่งไว้ในเครื่องหมาย backquotes (` `) โดยปกติ เซลล์ แบ่งเอาต์พุตจากคำสั่งออกเป็นค่าที่แยกต่างหากที่พื้นที่ว่าง แท็บ และอักขระบรรทัดใหม่ จากนั้น แทนที่คำสั่งดั้งเดิมด้วย เอาต์พุตนี้

ในตัวอย่างต่อไปนี้ เครื่องหมาย backquotes (` `) รอบ คำสั่ง date บ่งชี้ว่าเอาต์พุตของคำสั่ง จะถูกทดแทน:

```
echo The current date and time is: `date`
```

เอาต์พุตจากคำสั่งนี้อาจมีลักษณะคล้ายตัวอย่างต่อไปนี้:

```
The current date and time is: Wed Apr 8 13:52:14 CDT 1992
```

C เซลล์ทำการทดแทนคำสั่งที่เลือกบนอาร์กิวเมนต์ ของคำสั่งเซลล์ในตัว ซึ่งหมายความว่า เซลล์ไม่ได้ขยายส่วนของ นิพจน์ที่จะถูกประเมิน สำหรับคำสั่งที่ไม่ใช่คำสั่งในตัว เซลล์จะทดแทนชื่อคำสั่งแยกต่างหากจากรายการอาร์กิวเมนต์ การทดแทนเกิดขึ้นในชานี้ของเซลล์หลัก แต่เฉพาะหลังจากที่เซลล์ ทำการเปลี่ยนทิศทางอินพุตหรือเอาต์พุตแล้วเท่านั้น

ถ้าสตริงคำสั่งอยู่ในเครื่องหมาย " " เซลล์ จะถือว่าอักขระบรรทัดใหม่เท่านั้นเป็นตัวแบ่งคำ และรักษาพื้นที่ว่าง และแท็บไว้ภายในคำ ในทุกกรณี อักขระบรรทัดใหม่เดี่ยวสุดท้าย ไม่ได้บังคับให้เกิดคำใหม่

หลักการที่เกี่ยวข้อง:

“การทดแทนตัวแปรใน C เซลล์” ในหน้า 301

C เซลล์รักษาชุดของตัวแปร โดยแต่ละชุดมีค่ารายการค่า เป็นศูนย์หรือมากกว่านั้น บางตัวแปรเหล่านี้มีการตั้งค่าโดย เซลล์ หรืออ้างอิงโดยเซลล์ ตัวอย่างเช่น ตัวแปร `argv` เป็นรูปภาพของรายการตัวแปรเซลล์ และค่าที่ประกอบเป็นค่า ของตัวแปรนี้มีการอ้างอิงในวิธีพิเศษ

การดำเนินการคำสั่งที่ไม่ใช่คำสั่งในตัว C เซลล์:

เมื่อ C เซลล์กำหนดว่าคำสั่งไม่ใช่คำสั่งในตัวเซลล์ C เซลล์จะพยายามรันคำสั่งด้วยรูทีนย่อย `execv`

แต่ละค่าในตัวแปรเซลล์ `path` ระบุชื่อไดเรกทอรี ซึ่งเซลล์จะพยายามรันคำสั่ง หากไม่มีการระบุทั้งแฟล็ก `-c` และ `-t` เซลล์จะ hashes ชื่อในไดเรกทอรีเหล่านี้เข้าในตารางภายใน เซลล์ พยายามเรียกรูทีนย่อย `execv` บนไดเรกทอรี เฉพาะถ้ามีความเป็นไปได้ว่ามีคำสั่งอยู่ในไดเรกทอรีนั้น หากคุณปิด กลไกนี้ด้วยคำสั่ง `unhash` หรือกำหนดแฟล็ก `-c` หรือ `-t` ให้กับเซลล์ เซลล์จะเชื่อมต่อชื่อคำสั่งที่กำหนด เพื่อให้เป็นชื่อพาทของไฟล์ เซลล์ยังทำเช่นนี้ในทุกกรณีสำหรับ แต่ละไดเรกทอรีคอมโพเนนต์ของตัวแปร `path` ที่ไม่ได้ขึ้นต้นด้วยเครื่องหมายทับ (/) จากนั้น เซลล์จะพยายาม รันคำสั่ง

คำสั่งที่อยู่ในวงเล็บรันในเซลล์ย่อยเสมอ ตัวอย่างเช่น:

```
(cd ; pwd) ; pwd
```

แสดง ไดเรกทอรีโฮมโดยไม่เปลี่ยนที่ตั้งไดเรกทอรีปัจจุบัน อย่างไรก็ตาม คำสั่ง:

```
cd ; pwd
```

เปลี่ยนที่ตั้งไดเรกทอรีปัจจุบัน เป็นไดเรกทอรีโฮม คำสั่งที่อยู่ในวงเล็บมีการใช้มากที่สุด เพื่อป้องกันคำสั่ง `chdir` ไม่ให้แสดงผลกระทบต่อ เซลล์ปัจจุบัน

ถ้าไฟล์มีสิทธิการดำเนินการ แต่ไม่ใช่ไบนารีที่ปฏิบัติการได้ของ ระบบ เซลล์จะสมมติว่าเป็นไฟล์ที่มีคำสั่งเซลล์ และรันเซลล์ใหม่เพื่ออ่าน

ถ้ามีสมนามสำหรับเซลล์ คำของสมนามจะถูกเติมหน้า รายการอาร์กิวเมนต์เพื่อให้เป็นคำสั่งเซลล์ คำแรกของสมนาม ควรจะเป็นชื่อพาทแบบเต็มของเซลล์

การทดแทนประวัติใน C เซลล์:

การทดแทนประวัติอนุญาตให้คุณแก้ไขแต่ละคำจากคำสั่งก่อนหน้านี้ เพื่อสร้างคำสั่งใหม่ การทดแทนประวัติช่วยให้ง่ายในการทำซ้ำ คำสั่ง ทำซ้ำอาร์กิวเมนต์ของคำสั่งก่อนหน้านี้ในคำสั่งปัจจุบัน หรือแก้ไขข้อผิดพลาดการสะกดในคำสั่งก่อนหน้านี้ ด้วยการพิมพ์เพียงเล็กน้อยเท่านั้น

การทดแทนประวัติขึ้นต้นด้วยอักขระเครื่องหมายอัศเจรีย์ (!) และสามารถปรากฏขึ้นที่ได้ก็ได้บนบรรทัดคำสั่ง หากว่าไม่ได้จัดกลุ่ม (กล่าวอีกอย่างคือ การทดแทนประวัติไม่สามารถมี การทดแทนประวัติอื่น) คุณสามารถนำหน้า ! ด้วย \ เพื่อยกเลิกความหมายพิเศษของอัศเจรีย์ นอกจากนี้ ถ้าคุณใส่ ! ไว้ข้างหน้าพื้นที่ว่าง แท็บ อักขระบรรทัดใหม่, =, หรือ (การทดแทนประวัติ จะไม่เกิดขึ้น

การทดแทนประวัติยังเกิดขึ้นเมื่อคุณขึ้นต้นเส้นอินพุต ด้วยเครื่องหมาย carat (^) เซลล์สะท้อนเส้นอินพุตใดๆ ที่มีการทดแทนประวัติที่เวิร์กสแตชัน ก่อนเซลล์ดำเนินการเส้นอินพุต นั้น

หลักการที่เกี่ยวข้อง:

“การทดแทนสมนามใน C เซลล์” ในหน้า 300

สมนาม คือชื่อที่กำหนดให้กับคำสั่งหรือสตริงคำสั่ง C เซลล์อนุญาตให้คุณกำหนดสมนามและใช้สมนามเมื่อคุณจะทำคำสั่ง เซลล์เก็บรักษารายการของสมนามซึ่งคุณกำหนด

รายการประวัติสำหรับ C เซลล์:

รายการประวัติบันทึกคำสั่งที่เซลล์อ่านจากบรรทัดคำสั่ง ซึ่งประกอบด้วยคำสั่งแต่หนึ่งคำสั่งขึ้นไป การทดแทนประวัติแนะนำ ลำดับของคำสั่งอีกครั้งหนึ่งจากคำสั่งที่บันทึกไว้เหล่านี้เข้าไปในสตรีมอินพุต

ตัวแปรเซลล์ *history* ควบคุม ขนาดของรายการประวัติ คุณต้องตั้งค่าตัวแปรเซลล์ *history* ในไฟล์ `.cshrc` หรือบนบรรทัดคำสั่ง ด้วยคำสั่งใน `set` คำสั่งก่อนหน้านี้มีการเก็บรักษาไว้ เสมอ ไม่ว่าค่าของตัวแปร *history* เป็นอะไรก็ตาม คำสั่งในรายการ ประวัติมีการกำหนดลำดับหมายเลข โดยเริ่มต้นด้วย 1 คำสั่งในตัว *history* สร้างเอาต์พุตที่คล้ายกับตัวอย่าง ต่อไปนี้:

```
9 write michael
10 ed write.c
11 cat oldwrite.c
12 diff *write.c
```

เซลล์แสดงสตริงคำสั่งพร้อมด้วยหมายเลข เหตุการณ์ หมายเลขเหตุการณ์ปรากฏขึ้นที่ด้านซ้ายของคำสั่ง และแสดงถึง เวลาที่ ป้อนคำสั่งโดยสัมพันธ์กับคำสั่งอื่นในประวัติ โดยปกติไม่จำเป็นต้องใช้หมายเลขเหตุการณ์เพื่ออ้างอิงถึงเหตุการณ์ แต่คุณ สามารถมีหมายเลขเหตุการณ์ปัจจุบันที่แสดงขึ้นเป็นส่วนหนึ่งของพร้อมระบบได้ โดยการใส่เครื่องหมายอัศเจรีย์ (!) ใน พร้อมตสตริงที่กำหนดให้กับ ตัวแปรสภาวะแวดล้อม *PROMPT*

การอ้างอิงประวัติทั้งหมดประกอบด้วยข้อกำหนดคุณลักษณะเหตุการณ์ ตัวออกแบบคำ และตัวแก้ไขตั้งแต่หนึ่งตัวขึ้นไป ใน รูปแบบทั่วไปต่อไปนี้:

```
Event[.]Word[:Modifier[:Modifier]] . . .
```

หมายเหตุ: สามารถแก้ไขได้เพียงหนึ่งคำเท่านั้น ไม่อนุญาตให้ใช้สตริงที่มี พื้นที่ว่าง

ในตัวอย่างก่อนหน้านี้ของคำสั่งเอาต์พุต *history* หมายเลขเหตุการณ์ปัจจุบันคือ 13 ในตัวอย่างนี้ เหตุการณ์ก่อนหน้านี้มีการ อ้างอิง ดังต่อไปนี้:

ไอเท็ม	คำอธิบาย
!10	หมายเลขเหตุการณ์ 10
!-2	หมายเลขเหตุการณ์ 11 (เหตุการณ์ปัจจุบันลบ 2)
!d	ค่าของคำสั่งที่ขึ้นต้นด้วย d (หมายเลขเหตุการณ์ 12)
!?mic?	ค่าของคำสั่งที่มีสตริง mic (หมายเลขเหตุการณ์ 9)

รูปแบบเหล่านี้ที่ไม่มีการแก้ไขเพิ่มเติมเป็นเพียงการแนะนำคำของ เหตุการณ์ที่ระบุอีกครั้ง โดยค้นแต่ละรายการด้วยพื้นที่ว่าง หนึ่งช่อง เนื่องจากเป็น ตัวพิมพ์พิเศษ !! อ้างถึงคำสั่งก่อนหน้านี้ คำสั่ง !! ที่แสดงขึ้นโดยลำพัง บนเส้นอินพุตจะรีรันคำสั่งก่อน หน้านี้

ข้อกำหนดคุณลักษณะเหตุการณ์สำหรับ C เซลล์:

เมื่อต้องการเลือกคำจากเหตุการณ์ ให้ใส่เครื่องหมายจุดคู่ (:) ตามหลังข้อกำหนดคุณลักษณะเหตุการณ์และตัวออกแบบคำ อย่างใด อย่างหนึ่งต่อไปนี้ (ค่าของเส้นอินพุตมีการกำหนดลำดับหมายเลขโดยเริ่มต้นจาก 0)

ไอเท็ม	คำอธิบาย
0	ค่าแรก (ชื่อคำสั่ง)
n	อาร์กิวเมนต์ลำดับที่ n th
^	อาร์กิวเมนต์แรก
\$	อาร์กิวเมนต์สุดท้าย
%	การค้นหาที่ตรงกันโดยใช้สตริง? นำหน้า?
x-y	ช่วงของค่าตั้งแต่ค่าที่ x th ถึง ค่าที่ y th
-y	ช่วงของค่าตั้งแต่ค่าแรก (0) ถึงค่าที่ y th
*	อาร์กิวเมนต์แรกถึงอาร์กิวเมนต์สุดท้าย หรือไม่มีข้อมูลใดถ้ามีเพียงค่าเดียว (ชื่อคำสั่ง) อยู่ในเหตุการณ์
x*	อาร์กิวเมนต์ลำดับที่ x th ถึงอาร์กิวเมนต์สุดท้าย
x-	เหมือนกับ x* แต่ข้าม อาร์กิวเมนต์สุดท้าย

หากตัวออกแบบคำสั่งขึ้นต้นด้วย ^, \$, *, -, หรือ % คุณสามารถข้ามเครื่องหมายจุดคู่ที่คั่นระหว่างข้อกำหนดคุณลักษณะเหตุการณ์และตัวออกแบบคำสั่งได้ คุณยังสามารถใส่ลำดับของ ตัวแก้ไขต่อไปนี้ตามหลังตัวออกแบบคำสั่งที่เป็นทางเลือกได้ด้วย โดยแต่ละตัวมี เครื่องหมายจุดคู่หน้า:

ไอเท็ม	คำอธิบาย
h	ลบส่วนขยายชื่อพาราสั้นปลาย แต่คงส่วนหัวไว้
r	ลบคอมโพเนนต์ .xxx ส่วนปลาย แต่ปล่อยชื่อรากไว้
e	ลบออกทั้งหมดยกเว้นส่วนขยายส่วนปลาย .xxx
s/OldWord/NewWord/	ใช้ค่าของตัวแปร NewWord แทน ค่าของตัวแปร OldWord

ด้านซ้ายของการทดแทนไม่ใช่เป็นรูปแบบ ของสตริงที่รู้จักโดยโปรแกรมแก้ไข แต่เป็นคำยูนิตเดียวที่ไม่มีพื้นที่ว่าง โดยปกติเครื่องหมายทับ (/) จะคั่นระหว่าง คำดั้งเดิม (OldWord) และคำที่ใช้แทนที่ (NewWord) อย่างไรก็ตาม คุณสามารถใช้อักขระใดๆ เป็น ตัวคั่นได้ในตัวอย่างต่อไปนี้ การใช้ % เป็น ตัวคั่นอนุญาตให้รวม / ไว้ในคำได้:

s%/home/myfile%/home/yourfile%

เซลล์แทนที่เครื่องหมาย ampersand (&) ด้วยข้อความ OldWord ในตัวแปร NewWord ในตัวอย่างต่อไปนี้ /home/myfile กลายเป็น /temp/home/myfile

s%/home/myfile%/temp&%

เซลล์แทนที่ค่า null ในการทดแทน ด้วยการทดแทนล่าสุด หรือด้วยสตริงล่าสุดที่ใช้ในสแกน คอนเท็กซ์!?String? คุณสามารถข้าม ตัวคั่นส่วนปลาย (/) ได้ถ้ามีอักขระบรรทัดใหม่ตามหลังติดกัน ใช้ตัวแก้ไขต่อไปนี้เพื่อคั่นรายการประวัติ:

ไอเท็ม	คำอธิบาย
t	ลบคอมโพเนนต์ชื่อพารานำทั้งหมด ปล่อยส่วนปลายไว้
&	ทำซ้ำการทดแทนก่อนหน้านี้
g	ใช้การเปลี่ยนแปลงสากล นั่นคือทุกตำแหน่งที่เกิดขึ้นในแต่ละบรรทัด
p	แสดงคำสั่งใหม่ แต่ไม่ได้รับ
q	อ้างถึงคำที่ถูกทดแทน เพื่อป้องกันการทดแทนเพิ่มเติม
x	ทำหน้าที่เหมือนกับตัวแก้ไข q แต่แบ่งออกเป็นคำที่พื้นที่ว่าง แท็บ และอักขระบรรทัดใหม่

เมื่อใช้ตัวแก้ไขนำหน้า การเปลี่ยนแปลงจะใช้เฉพาะ กับค่าแรกที่แก้ไขได้เท่านั้น ยกเว้นว่าตัวแก้ไข g อยู่ก่อนหน้า ตัวแก้ไขที่เลือกไว้

หากคุณระบุการอ้างอิงประวัติโดยไม่มี ข้อกำหนดคุณลักษณะเหตุการณ์ (ตัวอย่างเช่น !\$) เซลล์จะใช้ คำสั่งก่อนหน้านี้เป็นเหตุการณ์ หากการอ้างอิงประวัติก่อนหน้านี้เกิดขึ้นบน บรรทัดเดียวกัน เซลล์จะทำซ้ำการอ้างอิงก่อนหน้านี้ ดังนั้น ลำดับต่อไปนี้จะให้อาร์กิวเมนต์แรกและอาร์กิวเมนต์สุดท้ายของคำสั่งที่ตรงกับ ?foo?

! ?foo? ^ !\$

การย่อพิเศษของการอ้างอิงประวัติ เกิดขึ้นเมื่ออักขระไม่วางเปล่าแรกของเส้นอินพุตเป็นเครื่องหมาย carat (^) ซึ่งเทียบเท่ากับ !:s^ แต่ช่วยให้สะดวกในการเขียนสั้นลง สำหรับการทดแทนข้อความของบรรทัดก่อนหน้านี้ คำสั่ง ^ |b^ lib แก้ไขการสะกดของ lib ในคำสั่ง

หากจำเป็น คุณสามารถใส่การทดแทนประวัติไว้ใน วงเล็บปีกกา {} เพื่อกันจากอักขระที่ตามมา ตัวอย่างเช่น ถ้าคุณต้องการใช้การอ้างอิงไปยังคำสั่ง:

```
ls -ld ~paul
```

เพื่อทำคำสั่ง:

```
ls -ld ~paula
```

ให้ใช้ โครงสร้างต่อไปนี้:

```
!{1}a
```

ในตัวอย่างนี้ !{1} จะค้นหาสตริงคำสั่งที่ขึ้นต้นด้วย l และ ผวนก a เข้าที่ตอนท้าย

การอ้างอิงด้วยเครื่องหมายัญประกาศเดี่ยวและคู่:

เพื่อป้องกันการตีความการทดแทนทั้งหมดหรือบางรายการเพิ่มเติม ให้ใส่สตริงไว้ในเครื่องหมายัญประกาศเดี่ยวและคู่

การใส่สตริงไว้ใน ' ' ป้องกันการตีความเพิ่มเติม ในขณะที่การใส่สตริงไว้ใน " " อนุญาตการขยายเพิ่มเติม ในทั้งสองกรณีข้อความที่ได้กลายเป็นข้อความทั้งหมดหรือบางส่วนของคำเดียว

การเปลี่ยนทิศทางอินพุตและเอาต์พุตใน C เซลล์:

ก่อน C เซลล์ดำเนินการคำสั่ง เซลล์จะสแกนบรรทัดคำสั่ง เพื่อหาอักขระการเปลี่ยนทิศทาง สัญลักษณ์พิเศษเหล่านี้สั่งให้เซลล์เปลี่ยนทิศทางอินพุต และเอาต์พุต

คุณสามารถเปลี่ยนทิศทางอินพุตและเอาต์พุตมาตรฐานของคำสั่ง โดยใช้คำสั่งไวยากรณ์ต่อไปนี้:

ไอเท็ม	คำอธิบาย
< File	เปิด File ที่ระบุ (ซึ่ง เป็นตัวแปร คำสั่ง และชื่อไฟล์ที่ขยายลำดับแรก) เป็นอินพุตมาตรฐาน
<< Word	อ่านอินพุตเซลล์จนถึงบรรทัดที่ตรงกับค่าของตัวแปร Word ตัวแปร Word ไม่อยู่ภายใต้การทดแทนตัวแปรชื่อไฟล์ หรือคำสั่ง แต่ละเส้นอินพุต ถูกเปรียบเทียบกับตัวแปร Word ก่อนทำการทดแทนใดๆ บนบรรทัด ยกเว้นอักขระการอ้างถึง (\, ", ' หรือ `) ปรากฏขึ้นในตัวแปร Word เซลล์จะทำการทดแทน ตัวแปรและคำสั่งบนบรรทัดแรกๆ เพื่อให้อักขระ \ สามารถอ้างถึงอักขระ \$, \, และ ` ได้ คำสั่งที่ถูกทดแทนมีพื้นที่ว่าง แท็บ และอักขระบรรทัดใหม่ทั้งหมดซึ่ง สงวนไว้ยกเว้นสำหรับอักขระบรรทัดใหม่สุดท้ายที่ถูกทิ้งไป ข้อความที่ได้ มีกรวางไว้ในไฟล์ชั่วคราวแบบนิรนาม ซึ่งถูกกำหนดให้กับคำสั่ง เป็นอินพุตมาตรฐาน
> File	ใช้ File ที่ระบุเป็น เอาต์พุตมาตรฐาน หากไม่มี File อยู่ ไฟล์จะถูกสร้างขึ้น หากมี File ไฟล์จะถูกตัดให้สั้นลง และเนื้อหาก่อนหน้านี้ จะหายไป ถ้ามีการตั้งค่าตัวแปรเซลล์ noclobber ต้องไม่มี File อยู่หรือเป็นไฟล์พิเศษอักขระ หรือส่งผลให้เกิดข้อผิดพลาด ลักษณะนี้ช่วย ป้องกันการทำลายไฟล์โดยอุบัติเหตุ ในกรณีนี้ ให้ใช้รูปแบบที่มี ! เพื่อยับยั้งการตรวจสอบนี้ File มีการขยาย ในวิธีเดียวกับชื่อไฟล์อินพุต < รูปแบบ & เปลี่ยนทิศทาง ทั้งเอาต์พุตมาตรฐานและข้อผิดพลาดมาตรฐานไปยัง File ที่ระบุ ตัวอย่างต่อไปนี้แสดงวิธีการเปลี่ยนทิศทางเอาต์พุตมาตรฐานไปยัง /dev/tty และ ข้อผิดพลาดมาตรฐานไปยัง /dev/null จำเป็นต้องใช้วงเล็บ เพื่อแยกแยะระหว่างเอาต์พุตมาตรฐานและข้อผิดพลาดมาตรฐาน
>!File	
>&File	
>&!File	% (find / -name vi -print > /dev/tty) >& /dev/null

ไอเท็ม	คำอธิบาย
>>File	ใช้ File ที่ระบุเป็น เอาต์พุตมาตรฐานเหมือนกับ > แต่ <i>ผนวก</i> เอาต์พุตเข้าที่ตอนท้ายของ File ถ้ามีการตั้งค่าตัวแปรเซลล์ <i>noclobber</i> จะส่งผลให้เกิดข้อผิดพลาดถ้า File ไม่มีอยู่ยกเว้นว่ามีการระบุรูปแบบอย่างใดอย่างหนึ่งที่มี ! มิฉะนั้น คล้ายกับ >
>>!File	
>>& File	
>>&! File	

คำสั่งได้รับสถานะแวดล้อมซึ่ง เซลล์ถูกเรียกใช้ ตามการเปลี่ยนแปลงโดยพารามิเตอร์อินพุต/เอาต์พุตและการมีอยู่ ของคำสั่ง เป็นไฟฟ์ไลน์ ดังนั้น สิ่งที่ไม่เหมือนกับบางเซลล์ก่อนหน้านี้คือ คำสั่ง ที่รับจากเซลล์สคริปต์ไม่มีสิทธิเข้าถึงข้อความของคำสั่ง โดยค่าดีฟอลต์ แต่ได้รับอินพุตมาตรฐานดั้งเดิมของเซลล์แทน ใช้ลไก << เพื่อแสดงข้อมูลออนไลน์ ซึ่งช่วยให้ไฟล์ คำสั่งเซลล์ สามารถทำหน้าที่เป็นคอมโพเนนต์ของไฟฟ์ไลน์ และยังช่วยให้ เซลล์บล็อกอ่านอินพุตของเซลล์ได้ด้วย โปรดทราบว่าอินพุต มาตรฐานดีฟอลต์ ของคำสั่งที่รับอย่างเป็นทางการไม่ถูกเปลี่ยนเป็นไฟล์ /dev/null วาง แต่อินพุตมาตรฐานยังคงเป็นอินพุต มาตรฐานดั้งเดิมของเซลล์

เมื่อต้องการเปลี่ยนทิศทางข้อผิดพลาดมาตรฐานผ่านทาง ไฟฟ์พร้อมกับเอาต์พุตมาตรฐาน ให้ใช้รูปแบบ | & แทน | อย่างเดียว

การควบคุมความต่อเนื่องใน C เซลล์:

เซลล์มีคำสั่งซึ่งสามารถใช้เพื่อควบคุมความต่อเนื่องของการควบคุม ในไฟล์คำสั่ง (เซลล์สคริปต์) และ (ในวิธีที่จำกัดแต่มี ประโยชน์) จากอินพุตบรรทัดคำสั่งเซลล์ คำสั่งทั้งหมดเหล่านี้ดำเนินงานโดยการบังคับให้เซลล์ ทำซ้ำ หรือข้าม ในอินพุตของ เซลล์

คำสั่ง `foreach`, `switch`, และ `while` และรูปแบบ `if-then-else` ของคำสั่ง `if` กำหนดว่า คีย์เวิร์ดสำคัญต้องปรากฏขึ้นในคำสั่งเดี่ยว แบบบางยบบนเส้นอินพุต

หากอินพุตเซลล์ไม่สามารถค้นหาได้ เซลล์จะบัฟเฟอร์อินพุต ในทุกเมื่อที่กำลังอ่านลูปและค้นหาบัฟเฟอร์ภายในเพื่อทำการ อ่านซ้ำที่ใช้โดยลูป ภายในขอบเขตที่ได้รับอนุญาต สามารถย้อนกลับไปยัง `gotos` บนอินพุตซึ่งคุณไม่สามารถค้นหาได้

การรักษาความปลอดภัยของระบบปฏิบัติการ

เป้าหมายของการรักษาความปลอดภัยของคอมพิวเตอร์คือการป้องกันข้อมูลที่จัดเก็บไว้บนระบบคอมพิวเตอร์

การรักษาความปลอดภัยข้อมูลมีเป้าหมายต่อไปนี้:

ไอเท็ม	คำอธิบาย
บูรณภาพ	ค่าของข้อมูลทั้งหมดขึ้นอยู่กับความถูกต้องของข้อมูลนั้น หากทำการเปลี่ยนแปลงที่ไม่ได้รับอนุญาตในข้อมูล ข้อมูลนี้จะมีค่าน้อยลง หรือไม่มีค่าใดๆ เลย
ความเป็นส่วนตัว	ค่าของข้อมูลจำนวนมากขึ้นอยู่กับการรักษาความลับของข้อมูลนั้น
การมีอยู่	ข้อมูลต้องมีอยู่พร้อม

การวางแผนและนำนโยบายด้านความปลอดภัยของคุณไปใช้ เป็นสิ่งที่มี ประโยชน์ก่อนคุณเริ่มต้นใช้ระบบ นโยบายด้านความ ปลอดภัยเป็นสิ่งที่ใช้เวลานานมาก ถ้าต้องเปลี่ยนในภายหลัง ดังนั้นการวางแผนล่วงหน้าสามารถช่วยประหยัดเวลาได้มาก

การระบุและการพิสูจน์ตัวตน

การระบุและการพิสูจน์ตัวตนสร้างเอกลักษณ์ของคุณ

คุณต้องล็อกอินเข้าสู่ระบบ คุณระบุชื่อผู้ใช้และรหัสผ่าน ถ้ามีหนึ่งแอคเคาต์ (ในระบบที่มีการรักษาความปลอดภัย แอคเคาต์ทั้งหมดควรมีรหัสผ่าน มิฉะนั้นจะไม่ถูกต้อง) หากรหัสผ่านถูกต้อง คุณจะล็อกอินเข้าสู่แอคเคาต์นั้นได้ และคุณได้รับสิทธิเข้าถึงและเอกสิทธิ์ของแอคเคาต์

เนื่องจากรหัสผ่านเป็นเพียงเครื่องป้องกันเพียงอย่างเดียวสำหรับแอคเคาต์ของคุณ จึงควรเลือกและรักษาหัสผ่านอย่างรอบคอบ ความพยายามในการเจาะเข้าสู่ระบบจำนวนมากเริ่มต้นด้วยความพยายามคาดเดารหัสผ่าน ระบบปฏิบัติการนำเสนอการป้องกันรหัสผ่านที่สำคัญโดยการจำกัดเก็บรหัสผ่านผู้ใช้ไว้แยกต่างหากจากข้อมูลอื่นของผู้ใช้ รหัสผ่านที่เข้ารหัสและข้อมูลเกี่ยวกับความปลอดภัยอื่นๆ ของผู้ใช้มีการจัดเก็บไว้ในไฟล์ /etc/security/passwd ไฟล์นี้ ควรเข้าถึงได้โดยผู้รักรากเท่านั้น ด้วยสิทธิเข้าถึงที่จำกัดนี้ใน รหัสผ่านที่เข้ารหัส ผู้โจมตีไม่สามารถแกะรหัสผ่านด้วย โปรแกรมที่เพียงแต่ลองใช้รหัสผ่านที่เป็นไปได้หรือมีแนวโน้มทั้งหมด

อย่างไรก็ตาม ยังคงสามารถคาดเดารหัสผ่านโดยการพยายามล็อกอินเข้าสู่ระบบซ้ำๆ หากรหัสผ่านเป็นชื่อทั่วไปหรือไม่ได้เปลี่ยนบ่อยๆ ความพยายามดังกล่าว อาจสำเร็จได้ง่าย

ID ผู้ใช้สำหรับล็อกอิน

ระบบปฏิบัติการสามารถระบุผู้ใช้โดยใช้ ID ผู้ใช้สำหรับล็อกอิน

ID ผู้ใช้สำหรับล็อกอินช่วยให้ระบบสามารถติดตามการดำเนินการของผู้ใช้ทั้งหมด ไปยังต้นทางได้ หลังจากผู้ใช้ล็อกอินเข้าสู่ระบบและก่อนจะรันโปรแกรมผู้ใช้แรกเริ่ม ระบบจะตั้งค่า ID ล็อกอินของโปรเซสให้กับ ID ผู้ใช้ที่พบในฐานข้อมูลผู้ใช้ โปรเซสในลำดับต่อมาทั้งหมดในระหว่างล็อกอินเซสชันจะมีการติดแท็กด้วย ID นี้ แท็กเหล่านี้แสดงร่องรอยของกิจกรรมทั้งหมดที่ทำโดย ID ผู้ใช้สำหรับล็อกอิน

คุณสามารถรีเซ็ต ID ผู้ใช้ที่บังคับใช้, ID ผู้ใช้จริง, ID กลุ่ม ที่บังคับใช้, ID กลุ่มจริง, และ ID กลุ่มเสริม ในระหว่าง เซสชัน แต่คุณไม่สามารถเปลี่ยน ID ผู้ใช้สำหรับล็อกอิน

เทอร์มินัลที่ไม่ได้ใส่ใจ

ระบบทุกระบบอาจได้รับอันตรายได้ถ้าเทอร์มินัลมีการล็อกอินทิ้งไว้ และไม่ได้ใส่ใจ ปัญหาร้ายแรงที่สุดเกิดขึ้นเมื่อผู้จัดการระบบปล่อยเทอร์มินัลที่ไม่ได้ใส่ใจ ทิ้งไว้โดยที่เปิดใช้งานด้วยสิทธิการใช้งานรากแล้ว โดยทั่วไป ผู้ใช้ควรล็อกเอาต์ทุกครั้งที่ปล่อยเทอร์มินัลทิ้งไว้

คุณสามารถบังคับให้เทอร์มินัลล็อกเอาต์หลังจากไม่มีกิจกรรมช่วงเวลาหนึ่งโดย ตั้งค่าพารามิเตอร์ TMOUT และ TIMEOUT ในไฟล์ /etc/profile พารามิเตอร์ TMOUT ทำงานใน ksh (Korn) เชลล์ และพารามิเตอร์ TIMEOUT ทำงานใน bash (เชลล์เป้าหมาย)

ตัวอย่างต่อไปนี้จะนำมาจากไฟล์ .profile บังคับให้เทอร์มินัลล็อกเอาต์หลังจากที่ไม่มีกิจกรรมนานหนึ่งชั่วโมง:

```
T0=3600
echo "Setting Autologout to $T0"
TIMEOUT=$T0
TMOUT=$T0
export TIMEOUT TMOUT
```

หมายเหตุ: คุณสามารถยกเลิกค่า TMOUT และ TIMEOUT ในไฟล์ /etc/profile โดยการระบุค่าที่แตกต่างอื่น ในไฟล์ .profile ในไดเรกทอรีโฮมของคุณ

หลักการที่เกี่ยวข้อง:

“การทดแทนตัวแปรในเชลล์เป้าหมาย” ในหน้า 295
เชลล์เป้าหมายอนุญาตให้คุณทำการทดแทนตัวแปรได้
สิ่งอ้างอิงที่เกี่ยวข้อง:

“การทดแทนพารามิเตอร์ใน Korn เชลล์หรือ POSIX เชลล์” ในหน้า 252
Korn เชลล์หรือ POSIX เชลล์อนุญาตให้คุณทำการทดแทนพารามิเตอร์ได้

ความเป็นเจ้าของไฟล์และกลุ่มผู้ใช้

โดยแรกเริ่ม เจ้าของไฟล์มีการระบุโดย ID ผู้ใช้ของบุคคล ที่สร้างไฟล์

เจ้าของไฟล์กำหนดบุคคลที่สามารถอ่าน เขียน (แก้ไข) หรือดำเนินการ ไฟล์ได้ ความเป็นเจ้าของสามารถเปลี่ยนได้โดยใช้คำสั่ง **chown**

ทุก ID ผู้ใช้มีการกำหนดให้กับกลุ่มพร้อมกับ ID กลุ่มที่ไม่ซ้ำกัน ผู้จัดการระบบ สร้างกลุ่มของผู้ใช้เมื่อตั้งค่าระบบ เมื่อมีการสร้างไฟล์ใหม่ ระบบปฏิบัติการจะกำหนดสิทธิอนุญาตให้กับ ID ผู้ใช้ที่สร้างไฟล์ ให้กับ ID กลุ่มที่มีเจ้าของไฟล์ และให้กับกลุ่มที่เรียกว่า **others** ซึ่งประกอบด้วยผู้ใช้อื่นทั้งหมด คำสั่ง **id** แสดง ID ผู้ใช้ (UID), ID กลุ่ม (GID), และชื่อของกลุ่มทั้งหมดซึ่งคุณเป็นสมาชิก

ในรายการไฟล์ (เช่นรายการที่แสดงโดยคำสั่ง **ls**) กลุ่มของผู้ใช้มีการแสดงแทนในลำดับต่อไปนี้เสมอ: ผู้ใช้ กลุ่ม และอื่นๆ หากคุณต้องการค้นหาชื่อกลุ่มของคุณ คำสั่ง **groups** แสดง กลุ่มทั้งหมดของ ID ผู้ใช้

การเปลี่ยนความเป็นเจ้าของไฟล์หรือไดเรกทอรี:

ใช้คำสั่ง **chown** เพื่อเปลี่ยนเจ้าของไฟล์

เมื่อคุณระบุอ็อปชัน **-R** คำสั่ง **chown** จะเรียงลำดับจากมากไปน้อยแบบวนซ้ำโดยใช้โครงสร้าง ไดเรกทอรีตั้งแต่ไดเรกทอรีที่ระบุ เมื่อพบลิงก์สัญลักษณ์ จะมีการเปลี่ยนความเป็นเจ้าของของไฟล์หรือไดเรกทอรีซึ่งชี้โดยลิงก์นั้น แต่ไม่เปลี่ยนความเป็นเจ้าของของลิงก์สัญลักษณ์

หมายเหตุ: เฉพาะผู้ใช้รากเท่านั้น สามารถเปลี่ยนเจ้าของของไฟล์อื่นได้ ข้อผิดพลาด ไม่แสดงขึ้นเมื่อระบุอ็อปชัน **-f**

ตัวอย่าง เช่น เมื่อต้องการเปลี่ยนเจ้าของไฟล์ `program.c` ให้พิมพ์ดัง ต่อไปนี้:

```
chown jim program.c
```

ขณะนี้ สิทธิเข้าถึงของผู้ใช้ สำหรับไฟล์ `program.c` ใช้กับ jim เนื่องจากเจ้าของ jim สามารถใช้คำสั่ง **chmod** เพื่ออนุญาตหรือปฏิเสธผู้ใช้รายอื่น ในการเข้าถึงไฟล์ `program.c`

ให้ดูที่คำสั่ง **chown** สำหรับ ไวยากรณ์ทั้งหมด

โหมดการเข้าถึงไฟล์และไดเรกทอรี:

ทุกไฟล์มีเจ้าของ สำหรับไฟล์ใหม่ ผู้ใช้ที่สร้างไฟล์เป็นเจ้าของของไฟล์นั้น เจ้าของกำหนด โหมดการเข้าถึง ให้กับไฟล์ โหมดการเข้าถึงมอบสิทธิการอ่าน การแก้ไข หรือการดำเนินการในไฟล์ให้แก่ผู้ใช้ระบบรายอื่น เฉพาะเจ้าของไฟล์หรือผู้ใช้ที่มีสิทธิการใช้งานรากเท่านั้น สามารถเปลี่ยนโหมดการเข้าถึงของไฟล์ได้

ผู้ใช้งานออกเป็นสามคลาสดังนี้: ผู้ใช้/เจ้าของ, กลุ่ม, และบุคคลอื่นทั้งหมด มีการมอบสิทธิเข้าถึงให้กับคลาสผู้ใช้งานเหล่านี้ในชุดที่เป็นส่วนผสม ระหว่างสามโหมดคือ: อ่าน เขียน หรือดำเนินการ เมื่อมีการสร้างไฟล์ใหม่ สิทธิอนุญาตดีฟอลต์คือ อ่าน เขียน และดำเนินการสำหรับผู้สร้างไฟล์ ผู้ใช้อีกสองกลุ่มมีสิทธิอ่านและดำเนินการ ตารางต่อไปนี้ แสดงโหมดการเข้าถึงไฟล์ดีฟอลต์สำหรับกลุ่มผู้ใช้งานทั้ง สามคลาส:

ไอเท็ม	คำอธิบาย		
คลาส	อ่าน	เขียน	ดำเนินการ
เจ้าของ	ใช่	ใช่	ใช่
กลุ่ม	ใช่	ไม่ใช่	ใช่
อื่นๆ	ใช่	ไม่ใช่	ใช่

ระบบกำหนดบุคคลที่มีสิทธิอนุญาตและระดับ สิทธิอนุญาตของบุคคลนั้นสำหรับแต่ละกิจกรรมเหล่านี้ โหมดการเข้าถึงมีการแสดงแทนโดยใช้ ทั้งสัญลักษณ์และตัวเลขในระบบปฏิบัติการ

หลักการที่เกี่ยวข้อง:

“ไตรีกทอรี” ในหน้า 516

ไตรีกทอรีคือชนิดที่ไม่ซ้ำกันของไฟล์ซึ่งมีเฉพาะ ข้อมูลที่จำเป็นสำหรับการเข้าถึงไฟล์หรือไตรีกทอรีอื่น ส่งผลให้ไตรีกทอรีใช้พื้นที่ว่างน้อยกว่าไฟล์ชนิดอื่น

“ชนิดของไฟล์” ในหน้า 209

ชนิดของไฟล์ที่ระบบรู้จักคือ ปกติ, ไตรีกทอรี, หรือ พิเศษ อย่างไรก็ตาม โดยอย่างหนึ่ง อย่างไรก็ตาม ระบบปฏิบัติการใช้ชนิดพื้นฐานเหล่านี้ในหลาย รูปแบบที่แตกต่างกันไป

การแสดงแทนโหมดการเข้าถึงโดยใช้สัญลักษณ์:

โหมดการเข้าถึงมีการแสดงแทนโดยใช้สัญลักษณ์

ไอเท็ม คำอธิบาย

r บังชี้สิทธิการอ่าน ซึ่งอนุญาตให้ผู้ใช้ดูเนื้อหาของไฟล์ได้

w บังชี้สิทธิการเขียน ซึ่งอนุญาตให้ผู้ใช้แก้ไขเนื้อหาของไฟล์ได้

x บังชี้สิทธิการดำเนินการสำหรับไฟล์ปฏิบัติการ (ไฟล์ปกติที่มีโปรแกรม) สิทธิการดำเนินการหมายความว่าโปรแกรมสามารถรันได้ สำหรับไตรีกทอรี สิทธิการดำเนินการหมายความว่าสามารถค้นหาเนื้อหาของไตรีกทอรีได้

โหมดการเข้าถึงสำหรับไฟล์หรือไตรีกทอรีมีการแสดงแทน โดยอักขระเก้าตัว สามอักขระแรกแสดงแทนสิทธิอนุญาต เจ้าของ ปัจจุบัน สามอักขระชุดที่สองแสดงแทนสิทธิอนุญาต กลุ่ม ปัจจุบัน และสามอักขระชุดที่สามแสดงแทนค่าติดตั้งปัจจุบันของสิทธิอนุญาต อื่นๆ ยัติกัณฑ์ (-) ในชุดเก้าอักขระ บ่งชี้ว่าไม่มีการให้สิทธิอนุญาต ตัวอย่างเช่น ไฟล์ที่มีการตั้งค่าโหมดการเข้าถึงเป็น rwxr-xr-x ให้สิทธิการอ่านและดำเนินการแก่ทั้งสามกลุ่ม และให้สิทธิการเขียนแก่เจ้าของไฟล์เท่านั้น นี่เป็นการแสดงแทนโดยใช้ สัญลักษณ์ของค่าติดตั้งดีฟอลต์

คำสั่ง ls เมื่อใช้พร้อม กับ แฟล็ก -l (L ตัวพิมพ์เล็ก) จะแสดงรายการโดยละเอียดของไตรีกทอรี ปัจจุบัน 10 อักขระแรกในรายการ ls -l แสดง ชนิดไฟล์และสิทธิอนุญาตสำหรับแต่ละกลุ่มทั้งหมดสามกลุ่ม คำสั่ง ls -l ยังแสดงรายการเจ้าของและกลุ่มที่เชื่อมโยงกับแต่ละไฟล์และไตรีกทอรีด้วย

อักขระแรกบ่งชี้ชนิดของไฟล์ อีกเก้า อักขระที่เหลือมีข้อมูลสิทธิอนุญาตในไฟล์สำหรับแต่ละ คลาสของผู้ใช้จากทั้งหมดสามคลาส มีการใช้สัญลักษณ์ต่อไปนี้เพื่อแสดงแทน ชนิดของไฟล์:

- ไอเท็ม คำอธิบาย
- ไฟล์ปกติ
- d โดเร็กทอรี
- b บล็อกไฟล์พิเศษ
- c อักขระไฟล์พิเศษ
- p ไฟล์พิเศษ
- l ลิงก์สัญลักษณ์
- s ซ็อกเก็ต

ตัวอย่างเช่น นี่เป็นรายการ ls -l ตัวอย่าง:

```
-rwxrwxr-x 2 janet acct 512 Mar 01 13:33 january
```

ที่นี้ ยัติภังค์แรก (-) บ่งชี้ไฟล์ปกติ แก้อักขระถัดไป (rwxrwxr-x) แสดงแทนโหมดการเข้าถึง ผู้ใช้ กลุ่ม และอื่นๆ ดังอธิบายไว้ข้างบน janet เป็นเจ้าของไฟล์ และ acct เป็นชื่อกลุ่มของ Janet 512 คือขนาดไฟล์ ในหน่วยไบต์ Mar 01 13:33 คือวันที่และเวลาการแก้ไขครั้งล่าสุด และ january คือชื่อไฟล์ 2 บ่งชี้จำนวนลิงก์ ที่มีอยู่ไปยังไฟล์

การแสดงแทนโหมดการเข้าถึงโดยใช้ตัวเลข:

โดยใช้ตัวเลข สิทธิการอ่านมีการแสดงแทนด้วยค่า 4, สิทธิการเขียน มีการแสดงแทนด้วยค่า 2, และสิทธิการดำเนินการมีการแสดงแทนด้วย ค่า 1 ค่าทั้งหมดระหว่าง 1 ถึง 7 แสดงแทนโหมดการเข้าถึง สำหรับแต่ละกลุ่ม (ผู้ใช้ กลุ่ม และอื่นๆ)

ตารางต่อไปนี้แสดงค่าตัวเลข สำหรับระดับการเข้าถึงแต่ละระดับ:

ค่าทั้งหมด	อ่าน	เขียน	ดำเนินการ
0	-	-	-
1	-	-	1
2	-	2	-
3	-	2	1
4	4	-	-
5	4	-	1
6	4	2	-
7	4	2	1

เมื่อมีการสร้างไฟล์ขึ้น โหมดการเข้าถึงไฟล์ดีฟอลต์ คือ 755 ซึ่งหมายความว่าผู้ใช้มีสิทธิการอ่าน เขียน และ ดำเนินการ (4+2+1=7), กลุ่มมีสิทธิการอ่านและดำเนินการ (4+1=5), และบุคคลอื่นทั้งหมดมีสิทธิการอ่านและดำเนินการ (4+1=5) เมื่อต้องการเปลี่ยนโหมดสิทธิเข้าถึงสำหรับไฟล์ที่คุณเป็นเจ้าของ ให้รันคำสั่ง **chmod** (change mode)

การแสดงข้อมูลกลุ่ม:

ใช้คำสั่ง **ls -l group** เพื่อแสดงแอตทริบิวต์ ของกลุ่มทั้งหมดบนระบบ (หรือของกลุ่มที่ระบุ) หากไม่สามารถอ่านแอตทริบิวต์ตั้ง แต่หนึ่งรายการขึ้นไป คำสั่ง **ls -l group** จะแสดงข้อมูลมากที่สุดเท่าที่เป็นไปได้

ข้อมูลแอตทริบิวต์แสดงขึ้นเป็นนิยาม **Attribute=Value** โดยแต่ละค่ามีการคั่นด้วยช่องว่าง

1. เมื่อต้องการแสดงรายการกลุ่มทั้งหมดบนระบบ ให้พิมพ์ดังต่อไปนี้:

```
lsgroup ALL
```

ระบบแสดงแต่ละกลุ่ม, ID กลุ่ม และผู้ใช้ทั้งหมดในกลุ่มในรายการคล้ายกับตัวอย่างต่อไปนี้:

```
system 0      arne,pubs,ctw,geo,root,chucka,noer,su,dea,backup,build,janice,denise
staff  1      john,ryan,flynn,daveb,jzitt,glover,maple,ken,gordon,mbrady
bin    2      root,bin
sys    3      root,su,bin,sys
```

2. เมื่อต้องการแสดงแอตทริบิวต์เฉพาะสำหรับกลุ่มทั้งหมด ให้ทำอย่างใดอย่างหนึ่ง ต่อไปนี้:

- คุณสามารถแสดงรายการแอตทริบิวต์ในรูปแบบ *Attribute=Value* ที่ค้น ด้วยช่องว่าง นี้เป็นลักษณะดีฟอลต์ ตัวอย่างเช่น เมื่อต้องการแสดงรายการ ID และผู้ใช้ของกลุ่มทั้งหมดบนระบบ ให้พิมพ์ดังต่อไปนี้:

```
lsgroup -a id users ALL | pg
```

รายการ คล้ายกับดังต่อไปนี้แสดงขึ้น:

```
system id=0 users=arne,pubs,ctw,geo,root,chucka,noer,su,dea,backup,build
staff id=1 users=john,ryan,flynn,daveb,jzitt,glover,maple,ken
```

- คุณยังสามารถแสดงรายการข้อมูลในรูปแบบ stanza ได้ด้วย ตัวอย่างเช่น เมื่อต้องการแสดงรายการ ID และผู้ใช้ของกลุ่มทั้งหมดบนระบบในรูปแบบ stanza ให้พิมพ์ดังต่อไปนี้:

```
lsgroup -a -f id users ALL | pg
```

รายการ คล้ายกับดังต่อไปนี้แสดงขึ้น:

```
system:
  id=0
  users=pubs,ctw,geo,root,chucka,noer,su,dea,backup,build
```

```
staff:
  id=1
  users=john,ryan,flynn,daveb,jzitt,glover,maple,ken
```

```
bin:
  id=2
  users=root,bin
```

```
sys:
  id=3
  users=root,su,bin,sys
```

3. เมื่อต้องการแสดงแอตทริบิวต์ทั้งหมดสำหรับกลุ่มเฉพาะ คุณสามารถใช้ลักษณะ อย่างใดอย่างหนึ่งจากสองลักษณะเพื่อแสดงแอตทริบิวต์เฉพาะสำหรับกลุ่มทั้งหมด

- คุณสามารถแสดงรายการแต่ละแอตทริบิวต์ในรูปแบบ *Attribute=Value* ที่ค้น ด้วยช่องว่าง นี้เป็นลักษณะดีฟอลต์ ตัวอย่างเช่น เมื่อต้องการแสดงรายการ แอตทริบิวต์ทั้งหมดสำหรับระบบกลุ่ม ให้พิมพ์ดังต่อไปนี้:

```
lsgroup system
```

รายการ คล้ายกับดังต่อไปนี้แสดงขึ้น:

```
system id=0 users=arne,pubs,ctw,geo,root,chucka,noer,su,dea,backup,build,janice,denise
```

- คุณยังสามารถแสดงรายการข้อมูลในรูปแบบ stanza ได้ด้วย ตัวอย่างเช่น เมื่อต้องการแสดงรายการแอตทริบิวต์ทั้งหมดสำหรับกลุ่ม bin ในรูปแบบ stanza ให้พิมพ์ดังต่อไปนี้:

```
lsgroup -f system
```

รายการ คล้ายกับดังต่อไปนี้แสดงขึ้น:

```
system:
  id=0   users=arne,pubs,ctw,geo,root,chucka,noer,su,dea,backup,build,janice,denise
```

4. เมื่อต้องการแสดงรายการแอตทริบิวต์เฉพาะสำหรับกลุ่มเฉพาะ ให้พิมพ์ ดังต่อไปนี้:

```
lsgroup -a Attributes Group
```

ตัวอย่างเช่น เมื่อต้องการแสดงรายการ ID และผู้ใช้สำหรับกลุ่ม bin ให้พิมพ์ดังต่อไปนี้:

```
lsgroup -a id users bin
```

รายการ คล้ายกับดังต่อไปนี้แสดงขึ้น:

```
bin id=2 users=root,bin
```

ให้ดูที่คำสั่ง `lsgroup` สำหรับไวยากรณ์ทั้งหมด

การเปลี่ยนสิทธิอนุญาตในไฟล์หรือไดเรกทอรี:

ใช้คำสั่ง `chmod` เพื่อเปลี่ยนสิทธิอนุญาต ในไฟล์ของคุณ

1. เมื่อต้องการเพิ่มชนิดของสิทธิอนุญาตลงในไฟล์ chap1 และ chap2 ให้พิมพ์ดังต่อไปนี้:

```
chmod g+w chap1 chap2
```

คำสั่งนี้เพิ่มสิทธิอนุญาตการเขียนสำหรับสมาชิกกลุ่มลงในไฟล์ chap1 และ chap2

2. เมื่อต้องการทำการเปลี่ยนแปลงสิทธิอนุญาตหลายรายการพร้อมกันในไดเรกทอรี mydir ให้พิมพ์ดังต่อไปนี้:

```
chmod go-w+x mydir
```

คำสั่งนี้ปฏิเสธไม่ให้สิทธิ (-) แก่สมาชิกกลุ่ม (g) และบุคคลอื่น (o) ในการสร้างหรือลบไฟล์ (w) ในไดเรกทอรี mydir และอนุญาต (+) ให้สมาชิกกลุ่มและบุคคลอื่นค้นหาไดเรกทอรี mydir หรือใช้ (x) ในชื่อพาธ คำสั่งนี้เทียบเท่ากับลำดับคำสั่ง ดังต่อไปนี้:

```
chmod g-w mydir
chmod o-w mydir
chmod g+x mydir
chmod o+x mydir
```

3. เมื่อต้องการอนุญาตเฉพาะเจ้าของให้ใช้เซสล์ไพรซีเดอร์ที่ชื่อ cmd เป็น คำสั่ง ให้พิมพ์ดังต่อไปนี้:

```
chmod u=rwx,go= cmd
```

คำสั่งนี้ให้สิทธิอ่าน เขียน และสิทธิการใช้งานแก่ผู้ใช้ที่เป็นเจ้าของไฟล์ (u=rwx) รวมทั้งปฏิเสธกลุ่มหรือผู้อื่นในการให้สิทธิเข้าถึง cmd ไม่ว่าวิธีใด (go=)

4. เมื่อต้องการใช้รูปแบบโหมดตัวเลขของคำสั่ง `chmod` เพื่อเปลี่ยนสิทธิอนุญาตของไฟล์ text ให้พิมพ์ ดังต่อไปนี้:

```
chmod 644 text
```

คำสั่งนี้ ตั้งค่าสิทธิการอ่านและเขียนสำหรับเจ้าของ และยังตั้งค่าโหมดอ่านอย่างเดียวสำหรับ กลุ่มและผู้อื่น

ให้ดูที่คำสั่ง `chmod` สำหรับไวยากรณ์ทั้งหมด

รายการควบคุมการเข้าถึง

การควบคุมการเข้าถึงประกอบด้วยรีซอร์สข้อมูลที่ป้องกันซึ่ง ระบุว่าใครสามารถได้รับมอบสิทธิเข้าถึงในรีซอร์สดังกล่าว

ระบบปฏิบัติการอนุญาตให้ใช้คุณลักษณะ `need-to-know` หรือการรักษาความปลอดภัยได้ตามที่เห็นสมควร เจ้าของรีซอร์สข้อมูลสามารถมอบสิทธิการอ่านหรือการเขียนในรีซอร์สของตน ให้แก่ผู้ใช้รายอื่นได้ จากนั้น ผู้ใช้ที่ได้รับมอบสิทธิการเข้าถึงในรีซอร์สสามารถโอนย้ายสิทธิดังกล่าวไปให้ผู้อื่นได้ การรักษาความปลอดภัยนี้ช่วยให้เกิดการไหลของข้อมูลที่ ควบคุมโดยผู้ใช้ในระบบ โดยเจ้าของรีซอร์สข้อมูลจะเป็นผู้กำหนด สิทธิเข้าถึงในอ็อบเจกต์

ผู้ใช้มีสิทธิเข้าถึงระดับผู้ใช้เฉพาะในอ็อบเจกต์ที่เป็นของตนเองเท่านั้น โดยปกติ ผู้ใช้ได้รับสิทธิการอนุญาตกลุ่มหรือสิทธิการอนุญาตดีฟอลต์สำหรับ รีซอร์ส ภารกิจหลักในการจัดการควบคุมการเข้าถึงคือ การกำหนด ความเป็นสมาชิกในกลุ่มของผู้ใช้ เนื่องจากความเป็นสมาชิกเหล่านี้เป็นตัวกำหนด สิทธิเข้าถึงของผู้ใช้ในไฟล์ที่ผู้ใช้ไม่ได้เป็นเจ้าของ

รายการควบคุมการเข้าถึงสำหรับอ็อบเจกต์ระบบไฟล์:

โดยปกติ อ็อบเจกต์ระบบไฟล์มีการเชื่อมโยงกับ Access Control List (ACL) ซึ่งโดยปกติแล้ว ประกอบด้วยชุดของ Access Control Entries (ACEs) แต่ละ ACE กำหนดเอกลักษณ์และสิทธิเข้าถึงที่เกี่ยวข้อง

เมื่อต้องการรักษารายการควบคุมการเข้าถึงไว้ ให้ใช้คำสั่ง `aclget`, `acledit`, `aclput` และ `aclconvert`

โปรดทราบว่า โดยปกติแล้ว ACL มีการจัดเก็บและจัดการบนสื่อบันทึกโดย ระบบไฟล์ฟิสิคัล (physical file system หรือ PFS) ระบบปฏิบัติการ AIX มีโครงสร้างพื้นฐานสำหรับระบบไฟล์ฟิสิคัลเพื่อสนับสนุน และจัดการชนิด ACL หลายแบบ ระบบไฟล์ JFS2 ที่จัดส่งมาพร้อมกับ AIX สนับสนุน ACL สองชนิด:

- AIXC
- NFS4

ระบบไฟล์ก่อนหน้านี้นับสนับสนุนเฉพาะชนิด AIXC ACL ดังเช่นใน AIX รีลีสก่อนหน้านี้ ชนิด ACL เหล่านี้มีการอธิบายโดยละเอียดใน *Security*

ชนิดรายการควบคุมการเข้าถึง AIXC:

ชนิด AIXC (AIX Classic) ACL นำเสนอลักษณะการทำงาน ACL ดังที่กำหนดไว้ในรีลีสก่อนหน้าของ AIX ชนิด ACL นี้ประกอบด้วยบิตโหมดพื้นฐานปกติและสิทธิอนุญาตแบบขยาย (ACEs)

ด้วยสิทธิอนุญาตแบบขยาย คุณสามารถอนุญาตหรือปฏิเสธการเข้าถึงไฟล์ของ ผู้ใช้แต่ละรายหรือกลุ่มได้โดยไม่ต้องเปลี่ยนสิทธิอนุญาตพื้นฐาน

หมายเหตุ: AIXC ACL สำหรับไฟล์หนึ่ง ต้องมีขนาดไม่เกินกว่าหนึ่งหน้าหน่วยความจำ (ประมาณ 4096 ไบต์)

คำสั่ง `chmod` ในโหมดตัวเลข (ที่มีสัญลักษณ์ฐานแปด) สามารถตั้งค่าสิทธิอนุญาตพื้นฐานและแอ็ททริบิวต์ได้ รุทีนย่อย `chmod` ซึ่งคำสั่งเรียก จะปิดใช้งานสิทธิอนุญาตแบบขยาย สิทธิอนุญาตแบบขยาย ถูกปิดใช้งานถ้าคุณใช้โหมดตัวเลขของคำสั่ง `chmod` บนไฟล์ที่มี ACL โหมดสัญลักษณ์ของคำสั่ง `chmod` ไม่ได้ปิดใช้งานสิทธิอนุญาตแบบขยายเมื่อ ACL ที่เชื่อมโยงเป็นชนิด AIXC สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ โหมดตัวเลขและสัญลักษณ์ ให้อ้างอิงที่คำสั่ง `chmod` สำหรับข้อมูล เกี่ยวกับคำสั่ง `chmod` ให้ดูที่ `chmod`

สิทธิอนุญาตพื้นฐาน

สิทธิอนุญาตพื้นฐานเฉพาะ AIXC ACL คือโหมดการเข้าถึงไฟล์แบบดั้งเดิมที่กำหนดให้กับเจ้าของไฟล์ กลุ่มไฟล์ และผู้ใช้อื่น โหมดการเข้าถึงคือการอ่าน (r), การเขียน (w), และการดำเนินการ/ค้นหา (x)

หมายเหตุ: สิทธิอนุญาตพื้นฐานชนิด AIXC ACL จะเหมือนกับบิตโหมดไฟล์ที่จัดเก็บไว้ในส่วนหัว inode ของอ็อบเจกต์ระบบไฟล์ นั่นคือ ข้อมูลในบิตโหมดพื้นฐานเหมือนกับค่าที่ส่งคืน โดยระบบไฟล์เมื่อทำ stat บนอ็อบเจกต์ระบบไฟล์

ในรายการควบคุมการเข้าถึง สิทธิอนุญาตพื้นฐานอยู่ในรูปแบบต่อไปนี้ โดยที่แสดงพารามิเตอร์ Mode เป็น rwx (โดยใช้เครื่องหมายติดลบ (-) แทนที่สิทธิอนุญาตที่ไม่ได้ระบุแต่ละสิทธิ):

```
base permissions:
  owner(name): Mode
  group(group): Mode
  others: Mode
```

แอตทริบิวต์

สามารถเพิ่มแอตทริบิวต์ลงในรายการควบคุมการเข้าถึงได้สามแอตทริบิวต์:

setuid (SUID)

บิตโหมด Set-user-ID แอตทริบิวต์นี้ตั้งค่า IDs ผู้ใช้ที่มีประสิทธิภาพและบันทึกไว้ ของโปรเซสเป็น ID เจ้าของของไฟล์ในการดำเนินการ

setgid (SGID)

บิตโหมด Set-group-ID แอตทริบิวต์นี้ตั้งค่า IDs กลุ่มที่มีประสิทธิภาพและบันทึกไว้ ของโปรเซสเป็น ID กลุ่มของไฟล์ในการดำเนินการ

savetext (SVTX)

บันทึกข้อความในรูปแบบไฟล์ข้อความ

แอตทริบิวต์ข้างบนมีการเพิ่มในรูปแบบต่อไปนี้:

```
attributes: SUID, SGID, SVTX
```

สิทธิอนุญาตแบบขยาย

สิทธิอนุญาตแบบขยาย AIXC ACL ช่วยให้เจ้าของไฟล์สามารถกำหนดสิทธิเข้าถึงไฟล์นั้นได้ แม่นยำมากขึ้น สิทธิอนุญาตแบบขยายแก้ไขสิทธิอนุญาตไฟล์พื้นฐาน (เจ้าของ กลุ่ม อื่นๆ) โดยการอนุญาต ปฏิเสธ หรือระบุโหมดการเข้าถึงสำหรับแต่ละ บุคคล กลุ่ม หรือชุดของผู้ใช้และกลุ่ม สิทธิอนุญาตมีการแก้ไข ผ่านทางการใช้คีย์เวิร์ด

คีย์เวิร์ดการอนุญาต ปฏิเสธ และการระบุ มีการกำหนดไว้ดังนี้:

อนุญาต

มอบสิทธิที่ระบุในไฟล์ให้แก่ผู้ใช้หรือกลุ่ม

ปฏิเสธ จำกัดผู้ใช้หรือกลุ่มไม่ให้ใช้สิทธิที่ระบุในไฟล์

ระบุ กำหนดสิทธิเข้าถึงไฟล์อย่างแม่นยำสำหรับผู้ใช้หรือกลุ่ม

หากผู้ใช้ได้รับการกำหนดสิทธิ์เข้าถึงเฉพาะโดยคีย์เวิร์ดปฏิเสธหรือ ระบุ อย่างใดอย่างหนึ่ง ไม่มีรายการอื่นสามารถยกเลิกการปฏิเสธสิทธิ์เข้าถึงนั้นได้

ต้องระบุ คีย์เวิร์ดที่เปิดใช้งานใน ACL เพื่อให้สิทธิ์อนุญาตแบบขยาย มีผลบังคับใช้ ค่าดีฟอลต์คือคีย์เวิร์ดปิดใช้งาน

ใน AIX ACL สิทธิ์อนุญาตแบบขยาย อยู่ในรูปแบบต่อไปนี้:

```
extended permissions:
  enabled | disabled
  permit  Mode  UserInfo...:
  deny    Mode  UserInfo...:
  specify Mode  UserInfo...:
```

ใช้บรรทัดแยกต่างหากสำหรับ แต่ละรายการอนุญาต ปฏิเสธ หรือระบุ พารามิเตอร์ Mode มีการแสดงเป็น rwx (โดยใช้เครื่องหมายยัติภังค์ (-) แทนที่สิทธิ์ อนุญาตที่ไม่ได้ระบุแต่ละสิทธิ์) พารามิเตอร์ UserInfo มีการแสดงเป็น u:UserName หรือ g:GroupName หรือชุดที่คั่นด้วยเครื่องหมายคอมมาของ u:UserName และ g:GroupName

หมายเหตุ: หากมีการระบุชื่อผู้ใช้นั้นชื่อในรายการ ไม่สามารถใช้รายการนั้น ในการตัดสินใจควบคุมการเข้าถึงได้ เนื่องจากโปรเซสมี ID ผู้ใช้เพียง ID เดียวเท่านั้น

ชนิดรายการควบคุมการเข้าถึง NFS4:

ระบบไฟล์ JFS2 ใน AIX ยัง สนับสนุนชนิด NFS4 ACL ด้วยการนำ ACL นี้ไปใช้เป็นไปตามคำนิยาม ACL ตามที่ระบุใน RFC ที่เกี่ยวข้องกับโปรโตคอล NFS4 เวอร์ชัน 4

ACL นี้แนะนำการควบคุมที่ละเอียดมากขึ้นในสิทธิ์เข้าถึงและยัง แนะนำคุณลักษณะต่างๆ เช่น การสืบทอด NFS4 ACL ประกอบด้วยอาร์เรย์ของ ACEs แต่ละ ACE กำหนดสิทธิ์เข้าถึงสำหรับเอกลักษณ์ ตามที่กำหนดไว้ใน RFC คอมโพเนนต์หลักของ NFS4 ACE มีดังนี้:

```
struct nfsace4 {
    acetype4      type;
    aceflag4      flag;
    acemask4      access_mask;
    utf8str_mixed who;
};
```

โดยที่:

ชนิด ตัวพยางค์ที่กำหนดชนิดของ ACE รายละเอียด เช่น ACE นี้อนุญาตการเข้าถึงหรือปฏิเสธการเข้าถึงมีการกำหนดที่นี่

แฟล็ก ตัวพยางค์ที่อธิบายลักษณะการสืบทอดของ ACE กำหนดว่า ACE นี้ใช้ได้กับอ็อบเจ็กต์ระบบไฟล์ หรือชายน์ของอ็อบเจ็กต์ หรือทั้งสองอย่าง

access_mask

ตัวพยางค์ที่กำหนดสิทธิ์เข้าถึงต่างๆ ที่เป็นไปได้ สิทธิ์ที่กำหนดไว้รวมถึง อ่าน เขียน ดำเนินการ สร้าง ลบ สร้างชายน์ ลบชายน์ เป็นต้น

ใคร สตริงที่สิ้นสุดด้วย null นี้กำหนดเอกลักษณ์ของบุคคลซึ่ง ACE นี้จะใช้ โปรดทราบว่าสำหรับแต่ละ RFC ขนาดของสตริงนี้ไม่มีข้อจำกัด และคำนิยามแบบกว้างๆ ช่วยให้สามารถกำหนดโดเมนภายในเครือข่าย NFS เวอร์ชัน 4 เพื่อจัดการควบคุมการเข้าถึงได้ โดยธรรมชาติ (เวลาส่วนใหญ่) AIX ไม่ได้ตีความสตริงนี้ และแต่ละ ACE มีการเชื่อมโยงกับ

เอกลักษณ์ที่ AIX เข้าใจได้ (เช่น uid หรือ gid) คาดกันว่าระบบไฟล์ NFS เวอร์ชัน 4 จะตีความสตริงเหล่านี้เมื่อจำเป็น เพื่อแปลงสตริงเป็น IDs ผู้ใช้หรือกลุ่มที่ OS เข้าใจได้ AIX เข้าใจสตริงใดเป็นพิเศษบางรายการซึ่งกำหนดไว้ใน RFC เท่านั้น

ใน AIX ให้ใช้คำสั่ง `aclget`, `acledit`, `aclput` และ `aclconvert` เพื่อจัดการ NFS4 ACLs

หมายเหตุ: คำสั่ง `chmod` ทุกชนิดจะลบ ACL ของไฟล์

ตัวอย่างรายการควบคุมการเข้าถึงสำหรับ AIXC:

ข้อมูลต่อไปนี้เป็นตัวอย่างของ AIXC access control list (ACL)

ข้อมูลต่อไปนี้เป็นตัวอย่างของ AIXC ACL:

```
attributes: SUID
base permissions:
  owner (frank): rw-
  group (system): r-x
  others      : ---
extended permissions:
  enabled
  permit  rw-  u:dhs
  deny    r--  u:chas,  g:system
  specify r--  u:john,  g:gateway, g:mail
  permit  rw-  g:account, g:finance
```

ส่วนประกอบของ ACL และความหมายมีดังนี้:

- บรรทัดแรกบ่งชี้ว่าบิต `setuid` เปิดอยู่
- บรรทัดถัดไปซึ่งแนะนำสิทธิการอนุญาตพื้นฐาน เป็นทางเลือก
- อีกสามบรรทัดถัดไประบุสิทธิการอนุญาตพื้นฐาน ชื่อเจ้าของและกลุ่ม ในเครื่องหมายวงเล็บมีไว้เป็นข้อมูลเท่านั้น การเปลี่ยนชื่อเหล่านี้ไม่ได้ เปลี่ยนเจ้าของไฟล์หรือกลุ่มไฟล์ เฉพาะคำสั่ง `chown` และคำสั่ง `chgrp` เท่านั้นสามารถเปลี่ยนแอ็ททริบิวต์ไฟล์เหล่านี้ได้ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งเหล่านี้ ให้อูที่ `chown` และ `chgrp`
- บรรทัดถัดไปซึ่งแนะนำสิทธิการอนุญาตแบบขยาย เป็นทางเลือก
- บรรทัดถัดไปบ่งชี้ว่าสิทธิการอนุญาตแบบขยายที่ตามมา มีการเปิดใช้งาน
- สี่บรรทัดสุดท้ายเป็นรายการแบบขยาย
- รายการแบบขยายอันดับแรกมอบสิทธิการอ่าน `dhs (r)` และการเขียน (`w`) ในไฟล์ให้แก่ผู้ใช้
- รายการแบบขยายที่สองปฏิเสธสิทธิการอ่าน (`r`) สำหรับผู้ใช้ `chas` เฉพาะ ถ้าผู้ใช้เป็นสมาชิกของกลุ่ม `system`
- รายการแบบขยายที่สามระบุว่า トラบใดที่ผู้ใช้ `john` เป็นสมาชิกของทั้งกลุ่ม `gateway` และกลุ่ม `mail` ผู้ใช้รายนี้จะมีสิทธิการอ่าน (`r`) หากผู้ใช้ `john` ไม่ได้เป็น สมาชิกของทั้งสองกลุ่ม จะไม่ใช่สิทธิอนุญาตที่ขยายออกนี้
- รายการแบบขยายอันดับสุดท้ายอนุญาตให้ผู้ใช้ใดๆ ใน ทั้ง กลุ่ม `account` และกลุ่ม `finance` มีสิทธิการอ่าน (`r`) และการเขียน (`w`)

หมายเหตุ: สามารถใช้รายการแบบขยายได้มากกว่าหนึ่งรายการ กับหนึ่งโปรเซส โดยที่โหนดข้อจำกัดมีความสำคัญเหนือกว่า โหมดการอนุญาต

สำหรับข้อมูลเพิ่มเติมและไวยากรณ์ทั้งหมด ให้อูที่คำสั่ง `acledit` ใน *Commands Reference, Volume 1*

การอนุญาตให้เข้าถึงรายการควบคุมการเข้าถึง:

เจ้าของรีซอร์สข้อมูลเป็นผู้รับผิดชอบในการจัดการกับ สิทธิการเข้าถึง รีซอร์สได้รับการปกป้องโดยบิตสิทธิ์อนุญาต ซึ่งรวมอยู่ในโหมดของอ็อบเจกต์

สำหรับ AIX ACL บิตสิทธิ์อนุญาต กำหนดสิทธิ์อนุญาตการเข้าถึงที่มอบให้แก่เจ้าของอ็อบเจกต์ กลุ่ม ของอ็อบเจกต์ และ ดีพอลต์คลาส others ชนิด AIX ACL สนับสนุนโหมด การเข้าถึงที่แตกต่างกันสามโหมด (การอ่าน การเขียน และการดำเนินการ) ซึ่งสามารถมอบแยกต่างหากกัน

เมื่อผู้ใช้ล็อกอินเข้าสู่แอคเคาต์ (โดยใช้คำสั่ง login หรือ su) IDs ผู้ใช้และ IDs กลุ่มที่กำหนดให้กับแอคเคาต์นั้นจะถูกเชื่อมโยงกับโปรเซสของผู้ใช้ IDs เหล่านี้กำหนดสิทธิ์การเข้าถึงของกระบวนการ

สำหรับไฟล์ ไดรฟ์ทอริฟฟ์ที่ระบุ และอุปกรณ์ (ไฟล์พิเศษ) ที่มี AIX ACL ซึ่งเชื่อมโยง การเข้าถึงได้รับ อนุญาตดังนี้:

- สำหรับแต่ละ access control entry (ACE) ใน access control list (ACL) รายการตัวระบุจะถูกเปรียบเทียบกับตัวระบุของโปรเซส หากมีข้อมูล ตรงกัน โปรเซสจะได้รับสิทธิ์อนุญาตและข้อจำกัดที่กำหนดไว้ สำหรับรายการนั้น โลกจำลองเนียน สำหรับทั้งสิทธิ์อนุญาตและข้อจำกัดมีการคำนวณ สำหรับแต่ละรายการที่ตรงกันใน ACL หากโปรเซสที่ร้องขอไม่ตรงกับรายการใดๆ ใน ACL โปรเซสนั้นจะได้รับสิทธิ์อนุญาตและข้อจำกัด ของรายการดีพอลต์
- ถ้าโหมดการเข้าถึงที่ร้องขอได้รับอนุญาต (รวมอยู่ในยูเนียนของ สิทธิ์อนุญาต) และไม่ถูกจำกัด (รวมอยู่ในยูเนียนของข้อจำกัด) จะมีการมอบสิทธิ์เข้าถึงให้ มีฉะนั้น การเข้าถึงจะถูกปฏิเสธ

ยิ่งไปกว่านั้น สำหรับชนิด AIX ACL รายการตัวระบุของ ACL จะตรงกับโปรเซสถ้าตัวระบุทั้งหมดใน รายการตรงกับชนิดที่สอดคล้องกันของตัวระบุที่มีประสิทธิภาพสำหรับโปรเซส ที่ร้องขอ ตัวระบุ USER-type ที่ตรงกันเท่ากับ ID ผู้ใช้ที่มีประสิทธิภาพ ของโปรเซส และตัวระบุ GROUP-type จะตรงกันถ้าเท่ากับ ID กลุ่มที่มี ประสิทธิภาพของโปรเซสหรือ IDs กลุ่มเสริม อย่างใดอย่างหนึ่ง ตัวอย่างเช่น ACE ที่มีรายการตัวระบุเช่นดังต่อไปนี้:

```
USER:fred, GROUP:philosophers, GROUP:software_programmer
```

จะจับคู่กระบวนการเข้ากับ ID ผู้ใช้ที่มีประสิทธิภาพเป็น fred และ ชุดกลุ่มของ:

```
philosophers, philanthropists, software_programmer, doc_design
```

แต่จะไม่จับคู่กระบวนการที่มี ID ผู้ใช้ที่มีประสิทธิภาพเป็น fred และ ชุดกลุ่มของ:

```
philosophers, iconoclasts, hardware_developer, graphic_design
```

โปรดทราบว่า ACE ที่มีรายการตัวระบุต่อไปนี้จะตรงกันสำหรับ ทั้งสองโปรเซส:

```
USER:fred, GROUP:philosophers
```

กล่าวอีกอย่างหนึ่งคือ รายการตัวระบุในฟังก์ชัน ACE คือชุดของเงื่อนไข ที่ต้องรักษาไว้เพื่อให้ได้รับมอบสิทธิ์การเข้าถึงที่ระบุ

กลไกการควบคุมการเข้าถึงแบบรอบคอบช่วยให้สามารถควบคุมการเข้าถึง รีซอร์สข้อมูลได้อย่างมีประสิทธิภาพ และนำเสนอ การป้องกันแยกต่างหากสำหรับ การรักษาความลับและบูรณภาพของข้อมูล กลไกการควบคุมการเข้าถึงที่ควบคุม โดยเจ้าของ มีประสิทธิภาพเหมือนกับกลไกที่ควบคุมโดยผู้ใช้นั้น ผู้ใช้ทุกรายต้อง เข้าใจวิธีการมอบและปฏิเสธสิทธิ์การเข้าถึง และวิธีการตั้งค่าสิทธิ์ ดังกล่าว

โปรดทราบว่าอ็อบเจกต์ระบบไฟล์ที่มีชนิด NFS4 ACL ที่เชื่อมโยง การตรวจสอบ สิทธิ์เข้าถึงพิจารณาจากข้อมูล ACEs ต่างๆ ซึ่งประกอบขึ้นเป็น ACL สำหรับเซต อัปเดตแต่ละข้อใน RFC ที่เกี่ยวข้องกับโปรโตคอล NFS เวอร์ชัน 4 การจับคู่เอกลักษณ์

ทำโดยการเปรียบเทียบข้อมูล ID ผู้ใช้ หรือ ID กลุ่ม หรือสตริง who พิเศษที่กำหนดไว้ใน ACE กับหลักฐานของโปรเซส หากมีรายการตรงกัน สิทธิเข้าถึงที่ร้องขอจะถูกตรวจสอบกับสิทธิเข้าถึงที่กำหนดไว้ใน ACE ถ้ามีสิทธิเข้าถึงใดๆ ได้รับอนุญาต สิทธิเหล่านั้นจะถูกนำออกมา และทำการเปรียบเทียบกับ ACE ถัดไปต่อไป โปรเซสที่ดำเนินไปอย่างต่อเนื่อง จนกว่า ACL อย่างใดอย่างหนึ่งจะสิ้นสุด หรือพบสิทธิเข้าถึงทั้งหมด หรือ สิทธิเข้าถึงที่ร้องขอใดๆ ถูกปฏิเสธ ขั้นตอนต่อไปนี้คือการตรวจสอบการเข้าถึงในกรณีของอ็อบเจกต์ระบบไฟล์ที่มี NFS4 ACL เชื่อมโยง:

1. สำหรับแต่ละ access control entry (ACE) ใน access control list (ACL) รายการตัวระบุจะถูกเปรียบเทียบกับตัวระบุของโปรเซส การตรวจสอบเอกลักษณ์รวมถึง ID ผู้ใช้หรือ ID กลุ่มที่กำหนดไว้ใน ACE นอกจากนี้ ถ้าเอกลักษณ์ มีการกำหนดเป็น พิเศษ ด้วยสตริง เช่น OWNER@ การตรงกันจะเกิดขึ้นถ้ากระบวนการเรียกทำโดยเจ้าของไฟล์ หากมีข้อมูล ตรงกัน โปรเซสจะได้รับสิทธิเข้าถึงที่กำหนดไว้สำหรับรายการนั้น มิฉะนั้น จะทำ ACE ถัดไปต่อไป
2. สิทธิเข้าถึงที่ร้องขอถูกเปรียบเทียบกับสิทธิเข้าถึงที่ดึงข้อมูลมาจาก รายการ ACE หากสิทธิเข้าถึงที่ร้องขอใดๆ ถูกปฏิเสธอย่างชัดเจน โดย ACE โปรเซสการตรวจสอบการเข้าถึงจะสิ้นสุดลง และโปรเซส ที่ร้องขอจะถูกปฏิเสธการเข้าถึง
3. ถ้าสิทธิเข้าถึงที่ร้องขอตรงรายการตรงกับ ACE สิทธิเข้าถึงเหล่านั้น จะถูกนำออกมาจากรายการสิทธิเข้าถึงที่ร้องขอ และทำการเปรียบเทียบกับ ACE ถัดไปต่อไป
4. ถ้าสิทธิเข้าถึงที่ร้องขอทั้งหมดตรงกับ ACEs การเข้าถึงที่ร้องขอ จะได้รับอนุญาต
5. ถ้า ACL สิ้นสุดลงก่อนที่สิทธิเข้าถึงที่ร้องขอทั้งหมดจะได้รับการแก้ไข การเข้าถึงจะถูกปฏิเสธ

โปรดทราบว่านอกเหนือจากการตรวจสอบการเข้าถึงตามข้อมูลชนิด ACL แล้ว ระบบไฟล์ฟิสิกส์แต่ละระบบยังอาจเลือกที่จะนำเสนอการเข้าถึงอ็อบเจกต์ระบบไฟล์ ตามข้อมูลเอกลักษณ์ที่ติดต่อกับ ตัวอย่างเช่น เจ้าของอาจมีสิทธิการแก้ไข ACL เป็นอย่างน้อยเสมอ โดยไม่คำนึงถึงสิทธิเข้าถึง ACL ที่มีอยู่ โปรเซสที่มี ID ผู้ใช้เป็น 0 รู้จักกันว่าเป็นโปรเซสผู้ใช้ราก โดยทั่วไปแล้ว โปรเซส เหล่านี้ได้รับมอบสิทธิการเข้าถึงทั้งหมด อย่างไรก็ตาม ถ้าโปรเซสผู้ใช้ราก ร้องขอสิทธิการดำเนินการในโปรแกรม จะมีการมอบสิทธิให้เฉพาะถ้าสิทธิ การดำเนินการมีการมอบให้แก่ผู้ใช้อย่างน้อยหนึ่งรายเท่านั้น

การตรวจสอบสิทธิการเข้าถึงทั้งหมดสำหรับอ็อบเจกต์เหล่านี้มีการทำที่ระดับ การเรียกกระบวนการเมื่อเข้าถึงอ็อบเจกต์ก่อน เนื่องจากอ็อบเจกต์ System V Interprocess Communication (SVIPC) มีการเข้าถึงแบบไม่มีสถานะ ดังนั้นจึงทำการตรวจสอบในทุกการเข้าถึง อย่างไรก็ตาม อาจเป็นไปได้ที่มีการทำการตรวจสอบโดยระบบไฟล์ฟิสิกส์ ในเวลาเปิดอ็อบเจกต์ระบบไฟล์ ไม่ใช่ในเวลาที่ดำเนินการอ่าน หรือเขียน สำหรับอ็อบเจกต์ที่มีชื่อระบบไฟล์ จำเป็นต้องสามารถ จำแนกชื่อของอ็อบเจกต์จริงได้ ชื่อมีการจำแนกแบบสัมพันธ์ (ไปยังไคเร็กทอรีการทำงานของโปรเซส) หรือแบบสัมบูรณ์ (ไปยังไคเร็กทอรีรากของโปรเซส) การจำแนกชื่อทั้งหมดเริ่มต้นโดยการค้นหาข้อมูลอย่างใดอย่างหนึ่งเหล่านี้

คำสั่งสำหรับการแสดงข้อมูลควบคุมการเข้าถึง (คำสั่ง `aclget`):

คำสั่ง `aclget` แสดงข้อมูลควบคุมการเข้าถึง ของไฟล์ ข้อมูลที่คุณเห็นได้แก่ แอ็ททริบิวต์ สิทธิอนุญาตพื้นฐาน และสิทธิอนุญาตแบบขยาย

ตัวอย่างเช่น เมื่อต้องการแสดงข้อมูลควบคุมการเข้าถึง ของไฟล์ `status` ให้พิมพ์ดังต่อไปนี้:

```
aclget status
```

ข้อมูลควบคุมการเข้าถึงที่แสดงขึ้นได้แก่ รายการของแอ็ททริบิวต์ สิทธิอนุญาตพื้นฐาน และสิทธิอนุญาตแบบขยาย

ให้ดูที่คำสั่ง `aclget` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

หลักการที่เกี่ยวข้อง:

“ตัวอย่างและคำอธิบายรายการควบคุมการเข้าถึง”

ข้อมูลต่อไปนี้เป็นตัวอย่างและคำอธิบายของรายการควบคุมการเข้าถึง (access control lists หรือ ACLs)

การตั้งค่าข้อมูลควบคุมการเข้าถึง (คำสั่ง `aclput`):

เมื่อต้องการตั้งค่าข้อมูลควบคุมการเข้าถึงสำหรับไฟล์ให้ใช้คำสั่ง `aclput`

หมายเหตุ: รายการควบคุมการเข้าถึงสำหรับไฟล์ต้องมีขนาดไม่เกินหนึ่งหน้า หน่วยความจำ (ประมาณ 4096 ไบต์)

โปรดดูตัวอย่างต่อไปนี้:

ตัวอย่าง เช่น เมื่อต้องการตั้งค่าข้อมูลควบคุมการเข้าถึงสำหรับไฟล์สถานะด้วยข้อมูลควบคุมการเข้าถึงที่จัดเก็บไว้ในไฟล์ `acldefs` ให้พิมพ์ดังต่อไปนี้:

```
aclput -i acldefs status
```

เมื่อต้องการตั้งค่าข้อมูลควบคุมการเข้าถึงสำหรับไฟล์สถานะด้วยข้อมูลเดียวกันกับที่ใช้สำหรับไฟล์ `plans` ให้พิมพ์ดังต่อไปนี้:

```
aclget plans | aclput status
```

สำหรับข้อมูลเพิ่มเติมและไวยากรณ์ทั้งหมดให้ดูที่คำสั่ง `aclput` ใน *Commands Reference, Volume 1*

ตัวอย่างและคำอธิบายรายการควบคุมการเข้าถึง:

ข้อมูลต่อไปนี้เป็นตัวอย่างและคำอธิบายของรายการควบคุมการเข้าถึง (access control lists หรือ ACLs)

ข้อมูลต่อไปนี้เป็นตัวอย่างของ ACL:

```
attributes: SUID
base permissions:
  owner (frank): rw-
  group (system): r-x
  others      : ---
extended permissions:
  enabled
  permit  rw-  u:dhs
  deny    r--  u:chas,  g:system
  specify r--  u:john,  g:gateway, g:mail
  permit  rw-  g:account, g:finance
```

ส่วนประกอบของ ACL และความหมายมีดังต่อไปนี้:

- บรรทัดแรกบ่งชี้ว่าบิต `setuid` เปิดอยู่
- บรรทัดถัดไปซึ่งแนะนำสิทธิการอนุญาตพื้นฐาน เป็นทางเลือก
- อีกสามบรรทัดถัดไประบุสิทธิการอนุญาตพื้นฐาน ชื่อเจ้าของและกลุ่ม ในเครื่องหมายวงเล็บมีไว้เป็นข้อมูลเท่านั้น การเปลี่ยนชื่อเหล่านี้ไม่ได้เปลี่ยนเจ้าของไฟล์หรือกลุ่มไฟล์ เฉพาะคำสั่ง `chown` และคำสั่ง `chgrp` เท่านั้นที่สามารถเปลี่ยนแอตทริบิวต์ไฟล์เหล่านี้ได้ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งเหล่านี้ให้ดูที่ `chown` และ `chgrp`
- บรรทัดถัดไปซึ่งแนะนำสิทธิการอนุญาตแบบขยาย เป็นทางเลือก
- บรรทัดถัดไปบ่งชี้ว่าสิทธิการอนุญาตแบบขยายที่ตามมา มีการเปิดใช้งาน

- สื่อบรรทัดสุดท้ายเป็นรายการแบบขยาย รายการแบบขยายอันดับแรก มอบสิทธิการอ่าน dhs (r) และการเขียน (w) ในไฟล์ให้แก่ผู้ใช้
- รายการแบบขยายที่สองปฏิเสธสิทธิการอ่าน (r) สำหรับผู้ใช้ chas เฉพาะ ถ้าผู้ใช้เป็นสมาชิกของกลุ่ม system
- รายการแบบขยายที่สามระบุว่า ทราบใดที่ผู้ใช้ john เป็นสมาชิกของทั้งกลุ่ม gateway และกลุ่ม mail ผู้ใช้จะมีสิทธิการอ่าน (r) หากผู้ใช้ john ไม่ได้เป็น สมาชิกของทั้งสองกลุ่ม จะไม่ใช้สิทธิอนุญาตที่ขยายออกนี้
- รายการแบบขยายอันดับสุดท้ายอนุญาตให้ผู้ใช้ใดๆ ใน ทั้ง กลุ่ม account และกลุ่ม finance มีสิทธิการอ่าน (r) และการเขียน (w)

หมายเหตุ: สามารถใช้รายการแบบขยายได้มากกว่าหนึ่งรายการ กับหนึ่งโปรเซส โดยที่โหมดข้อจำกัดมีความสำคัญเหนือกว่า โหมดการอนุญาต

ให้ดูที่คำสั่ง `acledit` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

หลักการที่เกี่ยวข้อง:

“คำสั่งสำหรับการแสดงข้อมูลควบคุมการเข้าถึง (คำสั่ง `aclget`)” ในหน้า 333

คำสั่ง `aclget` แสดงข้อมูลควบคุมการเข้าถึง ของไฟล์ ข้อมูลที่คุณเห็นได้แก่ แอ็ททริบิวต์ สิทธิอนุญาตพื้นฐาน และสิทธิอนุญาตแบบขยาย

งานที่เกี่ยวข้อง:

“การแก้ไขข้อมูลควบคุมการเข้าถึง (คำสั่ง `acledit`)”

ใช้คำสั่ง `acledit` เพื่อเปลี่ยนข้อมูลควบคุมการเข้าถึง ของไฟล์ คำสั่งแสดงข้อมูลควบคุมการเข้าถึงปัจจุบัน และอนุญาตให้เจ้าของไฟล์เปลี่ยนข้อมูลนั้นได้

การแก้ไขข้อมูลควบคุมการเข้าถึง (คำสั่ง `acledit`):

ใช้คำสั่ง `acledit` เพื่อเปลี่ยนข้อมูลควบคุมการเข้าถึง ของไฟล์ คำสั่งแสดงข้อมูลควบคุมการเข้าถึงปัจจุบัน และอนุญาตให้เจ้าของไฟล์เปลี่ยนข้อมูลนั้นได้

ก่อนทำการเปลี่ยนแปลงใดๆ อย่างถาวร คำสั่งจะถามว่าคุณต้องการดำเนินการต่อไปหรือไม่ สำหรับข้อมูลเกี่ยวกับคำสั่ง `acledit` ให้ดูที่ `acledit`

หมายเหตุ: ตัวแปรสภาวะแวดล้อม `EDITOR` ต้อง ถูกระบุโดยใช้ชื่อพาธแบบสมบูรณ์ มิฉะนั้นคำสั่ง `acledit` จะล้มเหลว

ข้อมูลควบคุมการเข้าถึงที่แสดงขึ้นเป็นข้อมูลเฉพาะชนิด ACL เท่านั้น และมีรายการของแอ็ททริบิวต์ สิทธิการอนุญาตพื้นฐาน และสิทธิการอนุญาตแบบขยาย

ตัวอย่างเช่น เมื่อต้องการแก้ไขข้อมูลควบคุมการเข้าถึง ของไฟล์ `plans` ให้พิมพ์ดังต่อไปนี้:

```
acledit plans
```

ให้ดูที่คำสั่ง `acledit` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

หลักการที่เกี่ยวข้อง:

“ตัวอย่างและคำอธิบายรายการควบคุมการเข้าถึง” ในหน้า 334

ข้อมูลต่อไปนี้เป็นตัวอย่างเป็นตัวอย่างและคำอธิบายของรายการควบคุมการเข้าถึง (access control lists หรือ ACLs)

การล็อกเทอร์มินัลของคุณ (คำสั่ง lock หรือ xlock)

ใช้คำสั่ง lock เพื่อล็อกเทอร์มินัลของคุณ คำสั่ง lock ร้องขอรหัสผ่านของคุณ อ่านรหัสผ่าน และ ร้องขอรหัสผ่านเป็นครั้งที่สอง เพื่อตรวจสอบ

ในระหว่างดำเนินการ คำสั่งจะล็อกเทอร์มินัลและไม่ปลดล็อก จนกว่าคำสั่งได้รับรหัสผ่านเป็นครั้งที่สอง ค่าตีพอลต์ไทม์เอาต์คือ 15 นาที แต่สามารถเปลี่ยนค่านี้ได้ด้วยแฟล็ก -Number

หมายเหตุ: หาก อินเทอร์เน็ตของคุณเป็น AIXwindows ให้ใช้คำสั่ง xlock ในลักษณะเดียวกัน

ตัวอย่าง เช่น เมื่อต้องการล็อกเทอร์มินัลของคุณภายใต้การควบคุมรหัสผ่าน ให้พิมพ์ดังต่อไปนี้:

```
lock
```

คุณจะได้รับ การพร้อมตัวให้อ่านรหัสผ่านเป็นครั้งที่สองเพื่อให้ระบบสามารถตรวจสอบรหัสผ่านได้ หากไม่ได้ป้อนรหัสผ่านซ้ำภายใน 15 นาที คำสั่งจะไทม์เอาต์

เมื่อต้องการสำรอง เทอร์มินัลภายใต้การควบคุมรหัสผ่านด้วยช่วงเวลาไทม์เอาต์นาน 10 นาที ให้พิมพ์ดังต่อไปนี้:

```
lock -10
```

โปรดดูที่คำสั่ง lock หรือ xlock in *Commands Reference* สำหรับ ไวยากรณ์ทั้งหมด

ข้อมูลสรุปคำสั่งสำหรับความปลอดภัยของไฟล์และระบบ

ข้อมูลต่อไปนี้เป็นคำสั่งสำหรับระบบไฟล์และความปลอดภัย

ไอเท็ม	คำอธิบาย
acldit	แก้ไขข้อมูลควบคุมการเข้าถึงของไฟล์
aclget	แสดงข้อมูลควบคุมการเข้าถึงของไฟล์
aclput	ตั้งค่าข้อมูลควบคุมการเข้าถึงของไฟล์
chmod	เปลี่ยนโหมดสิทธิ์อนุญาต
chown	เปลี่ยนผู้ใช้ที่เชื่อมโยงกับไฟล์
lock	สำรองเทอร์มินัล
lsgrupp	แสดงแอดทริบิวต์ของกลุ่ม
xlock	ล็อกจอแสดงผล X โคลงจนกว่าป้อนรหัสผ่าน

สภาวะแวดล้อมผู้ใช้

ชื่อล็อกอินแต่ละชื่อมีสภาวะแวดล้อมระบบของตนเอง

สภาวะแวดล้อมของระบบเป็นพื้นที่ซึ่งใช้จัดเก็บข้อมูล ที่ใช้กันทั่วไปในโปรเซสทั้งหมดซึ่งกำลังรันอยู่ในเซสชัน คุณสามารถใช้ได้หลาย คำสั่งเพื่อแสดงข้อมูลเกี่ยวกับระบบ

ไฟล์สภาวะแวดล้อมผู้ใช้และโปรซีเตอร์การกำหนดเอง

ไฟล์และโปรซีเตอร์เหล่านี้ช่วยผู้ใช้กำหนดสภาวะแวดล้อมของระบบเอง

ไฟล์สตาร์ทอัพระบบ

ไอเท็ม	คำอธิบาย
/etc/profile	ไฟล์ระบบที่มีคำสั่งซึ่งระบบดำเนินการ เมื่อคุณล็อกอิน
/etc/environment	ไฟล์ระบบที่มีตัวแปรซึ่งระบุสถานะแวดล้อมพื้นฐาน สำหรับโปรเซสทั้งหมด
\$HOME/.profile	ไฟล์ในไดเรกทอรีโฮมที่มีคำสั่งซึ่งยกเลิกระบบ /etc/profile เมื่อคุณล็อกอิน สำหรับข้อมูลเพิ่มเติม ให้ดูที่ไฟล์ .profile
\$HOME/.env	ไฟล์ในไดเรกทอรีโฮมที่ยกเลิกระบบ /etc/environment และมีตัวแปรซึ่งระบุสถานะแวดล้อมพื้นฐานสำหรับโปรเซสทั้งหมด สำหรับข้อมูลเพิ่มเติม ให้ดูที่ไฟล์ .env

ไฟล์สตาร์ทอัพ AIXwindows

ไอเท็ม	คำอธิบาย
\$HOME/.xinitrc	ไฟล์ในไดเรกทอรีโฮมที่ควบคุมหน้าต่างและแอปพลิเคชันซึ่งเริ่มต้นขึ้นเมื่อคุณเริ่มต้น AIXwindows สำหรับข้อมูลเพิ่มเติม ให้ดูที่ไฟล์ .xinitrc
\$HOME/.Xdefaults	ไฟล์ในไดเรกทอรีโฮมที่ควบคุมลักษณะภาพ หรือการทำงานของรีซอร์ส AIXwindows สำหรับข้อมูลเพิ่มเติม ให้ดูที่ “ไฟล์ .Xdefaults” ในหน้า 347
\$HOME/.mwmrc	ไฟล์ในไดเรกทอรีโฮมที่กำหนดการผูกคีย์ การผูกปุ่มเมาส์ และนิยามเมนูสำหรับตัวจัดการหน้าต่าง สำหรับข้อมูลเพิ่มเติม ให้ดูที่ “ไฟล์ .mwmrc” ในหน้า 348

โปรซีเตอร์กำหนดเอง

ไอเท็ม	คำอธิบาย
PS1	พร้อมระบบปกติ
PS2	พร้อมระบบอินพุตเพิ่มเติม
PS3	พร้อมระบบบราว
chfont	เปลี่ยนฟอนต์ที่ใช้โดยจอแสดงผลเมื่อสตาร์ทระบบ
stty	ตั้งค่ารีเซต และรายงานพารามิเตอร์การดำเนินงานเวิร์กสเตชัน

รายการอุปกรณ์ระบบ (คำสั่ง lscfg)

เมื่อต้องการแสดงชื่อที่ตั้ง และคำอธิบายของอุปกรณ์แต่ละรายการที่พบในคอนฟิกูเรชันปัจจุบัน ให้ใช้คำสั่ง lscfg รายการมีการเรียงลำดับตามที่ตั้งอุปกรณ์

ตัวอย่างเช่น เมื่อต้องการแสดงรายการอุปกรณ์ที่กำหนดคอนฟิกในระบบ ที่พร้อมให้พิมพ์ดังต่อไปนี้:

```
lscfg
```

ระบบแสดงเอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```
INSTALLED RESOURCE LIST
```

The following resources are installed on your machine.

+/- = Added/Deleted from Diagnostic Test List.

* = NOT Supported by Diagnostics.

```
Model Architecture: chrp
```

```
Model Implementation: Multiple Processor, PCI bus
```

```
+ sysplanar0 00-00 CPU Planar
+ fpa0 00-00 Floating Point Processor
+ mem0 00-0A Memory Card
```

```

+ mem1      00-0B      Memory Card
+ ioplanar0 00-00      I/O Planar
+ rs2320    00-01      RS232 Card
+ tty0      00-01-0-01   RS232 Card Port
- tty1      00-01-0-02   RS232 Card Port
..
..
..

```

รายการอุปกรณ์ไม่ได้เรียงลำดับตามที่ตั้งอุปกรณ์เพียงอย่างเดียว มีการเรียงลำดับตาม ลำดับชั้นพาวเวอร์/ชาแนลด้วย หากพาวเวอร์ตีมัลติชาแนล ชาแนลจะมีการ เรียงลำดับตามที่ตั้งอุปกรณ์ ถ้าชาแนลมีที่ตั้งอุปกรณ์เหมือนกัน ชาแนลดังกล่าวจะแสดงชั้นในลำดับซึ่งซอฟต์แวร์ได้รับมา เมื่อต้องการแสดงข้อมูลเกี่ยวกับอุปกรณ์เฉพาะ คุณสามารถใช้แฟล็ก -l ตัวอย่างเช่น เมื่อต้องการแสดงข้อมูลเกี่ยวกับอุปกรณ์ `sysplanar0` ที่พร้อมด ให้พิมพ์ดังต่อไปนี้:

```
lscfg -l sysplanar0
```

ระบบแสดงเอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```

DEVICE      LOCATION    DESCRIPTION
sysplanar0  00-00      CPU Planar

```

คุณยังสามารถใช้คำสั่ง `lscfg` เพื่อแสดงข้อมูลผลิตภัณฑ์ที่สำคัญมาก (VPD) เช่น หมายเลขชิ้นส่วน หมายเลขประจำผลิตภัณฑ์ และระดับ การเปลี่ยนแปลงทางวิศวกรรม สำหรับบางอุปกรณ์ VPD มีการรวบรวมและเพิ่มลงใน คอนฟิกูเรชันระบบโดยอัตโนมัติ แต่ในบางอุปกรณ์ ต้องพิมพ์ VPD ด้วยตนเอง ME ที่นำหน้าข้อมูลบ่งชี้ว่า เป็นข้อมูลที่พิมพ์ด้วยตนเอง

ตัวอย่างเช่น เมื่อต้องการแสดงรายการ VPD ของอุปกรณ์ที่กำหนดคอนฟิกในระบบ ที่พร้อมด ให้พิมพ์ดังต่อไปนี้:

```
lscfg -v
```

ระบบแสดงเอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```

INSTALLED RESOURCE LIST WITH VPD
The following resources are installed in your machine.
  Model Architecture: chrp
  Model Implementation: Multiple Processor, PCI bus
sysplanar0  00-00  CPU Planar
  Part Number.....342522
  EC Level.....254921
  Serial Number.....353535
fpa0  00-00  Floating Point Processor
mem0  00-0A  Memory Card
  EC Level.....990221
.
.
.

```

ให้ดูที่คำสั่ง `lscfg` ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงชื่อคอนโซล

เมื่อต้องการเขียนชื่อของอุปกรณ์คอนโซลปัจจุบันไปยังเอาต์พุตมาตรฐาน (โดยปกติคือหน้าจอของคุณ) ให้ใช้คำสั่ง `lscons`

ตัวอย่างเช่น ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
lscons
```

ระบบ แสดงเอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```
/dev/lft0
```

ให้ดูที่คำสั่ง `lscons` สำหรับ ไวยากรณ์ทั้งหมด

การแสดงชื่อเทอร์มินัล (คำสั่ง `tty`)

เมื่อต้องการแสดงชื่อของเทอร์มินัล ให้ใช้คำสั่ง `tty`

ตัวอย่างเช่น ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
tty
```

ระบบแสดงข้อมูล คล้ายกับตัวอย่างต่อไปนี้:

```
/dev/tty06
```

ในตัวอย่างนี้ `tty06` คือชื่อเทอร์มินัล และ `/dev/tty06` คือ ไฟล์อุปกรณ์ที่มีอินเตอร์เฟซไปยังเทอร์มินัลนี้

ให้ดูที่คำสั่ง `tty` ใน *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงรายการจอแสดงผลที่มีอยู่ (คำสั่ง `lsdisp`)

เมื่อต้องการแสดงรายการจอแสดงผลที่มีอยู่บนระบบในปัจจุบัน โดยแสดง ชื่อการระบุจอแสดงผล หมายเลขสล็อต จอแสดงผล และคำอธิบาย ของแต่ละจอแสดงผล ให้ใช้คำสั่ง `lsdisp`

ตัวอย่างเช่น เมื่อต้องการแสดงรายการจอแสดงผลที่มีอยู่ทั้งหมด ให้พิมพ์ดังต่อไปนี้:

```
lsdisp
```

ข้อมูล ต่อไปนี้เป็นตัวอย่างของเอาต์พุต รายการแสดงในลำดับจากน้อยไปมาก ตามหมายเลขสล็อต

Name	Slot	Name	Description
ppr0	00-01	POWER_G4	Midrange Graphics Adapter
gda0	00-03	colorgda	Color Graphics Display Adapter
ppr1	00-04	POWER_Gt3	Midrange Entry Graphics Adapter

ให้ดูที่คำสั่ง `lsdisp` ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงรายการฟอนต์ที่มีอยู่ (คำสั่ง `lsfont`)

เมื่อต้องการแสดงรายการของฟอนต์ที่มีอยู่บนจอแสดงผล ให้ใช้คำสั่ง `lsfont`

ตัวอย่างเช่น เมื่อต้องการแสดงรายการฟอนต์ที่มีอยู่ทั้งหมดบนจอแสดงผล ให้พิมพ์ดังต่อไปนี้:

```
lsfont
```

ข้อมูลต่อไปนี้เป็นตัวอย่างของ เอาต์พุต ที่แสดงตัวระบุฟอนต์ ชื่อไฟล์ ขนาด glyph และการเข้ารหัส ฟอนต์:

FONT ID	FILE NAME	GLYPH SIZE	FONT ENCODING
0	Erg22.iso1.snf	12x30	IS08859-1
1	Erg11.iso1.snf	8x15	IS08859-1

ให้ดูที่คำสั่ง **lsfont** ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงผลรายการซอฟต์แวร์คีย์บอร์ดแม่พิมพ์ปัจจุบัน (คำสั่ง **lskbd**)

เมื่อต้องการแสดงชื่อพารามิเตอร์ของซอฟต์แวร์คีย์บอร์ดแม่พิมพ์ปัจจุบัน ที่โหลดเข้าในระบบ ให้ใช้คำสั่ง **lskbd**

ตัวอย่างเช่น เมื่อต้องการแสดงรายการคีย์บอร์ดแม่พิมพ์ปัจจุบัน ให้พิมพ์ดังต่อไปนี้:

```
lskbd
```

ข้อมูล ต่อไปนี้เป็นตัวอย่างของรายการที่แสดงโดยคำสั่ง **lskbd**:

```
The current software keyboard map = /usr/lib/nls/loc/C.lftkeymap
```

การแสดงผลรายการผลิตภัณฑ์ซอฟต์แวร์ที่มีอยู่ (คำสั่ง **lsipp**)

เมื่อต้องการแสดงข้อมูลเกี่ยวกับผลิตภัณฑ์ซอฟต์แวร์ที่มีอยู่ในระบบ ให้ใช้คำสั่ง **lsipp**

ตัวอย่างเช่น เมื่อต้องการแสดงรายการผลิตภัณฑ์ซอฟต์แวร์ทั้งหมดในระบบ ที่พร้อมระบบ ให้พิมพ์ดังต่อไปนี้:

```
lsipp -l -a
```

ข้อมูล ต่อไปนี้เป็นตัวอย่างของเอาต์พุต:

Fileset	Level	State	Description
Path: /usr/lib/objrepos			
X11_3d.gl.dev.obj	APPLIED		AIXwindows/3D GL Development Utilities
Fonts			
X11fnt.o1dX.fnt	APPLIED		AIXwindows Miscellaneous X Fonts
X11mEn_US.msg	APPLIED		AIXwindows NL Message files
.			
.			
.			

หาก รายการยาวเกินไป ส่วนบนสุดอาจหลุดจากหน้าจอ เมื่อต้องการแสดง รายการหนึ่งหน้า (หน้าจอ) ในแต่ละครั้ง ให้พิมพ์คำสั่ง **lsipp** เข้ากับคำสั่ง **pg** ที่พร้อมให้พิมพ์ดังต่อไปนี้:

```
lsipp -l -a | pg
```

ให้ดูที่คำสั่ง **lsipp** ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

รายการของการกำหนดคีย์ควบคุมสำหรับเทอร์มินัลของคุณ (คำสั่ง `stty`)

เมื่อต้องการแสดงค่าที่ตั้งเทอร์มินัลให้ใช้คำสั่ง `stty` โปรดสังเกตเป็นพิเศษว่าเทอร์มินัลใช้คีย์ใดเป็นคีย์ควบคุม

ตัวอย่างเช่น ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
stty -a
```

ระบบแสดงข้อมูล คล้ายกับตัวอย่างต่อไปนี้:

```
.  
. .  
intr = ^C; quit = ^\; erase = ^H; kill = ^U; eof = ^D;  
eol = ^@ start = ^Q; stop = ^S; susp = ^Z; dsusp = ^Y;  
reprint = ^R discard = ^O; werase = ^W; lnext = ^V  
. . .
```

ในตัวอย่างนี้ บรรทัดเช่น `intr = ^C`; `quit = ^\`; `erase = ^H`; แสดง ค่าที่ตั้งคีย์ควบคุมของคุณ คีย์ `^H` คือปุ่ม Backspace และมีการตั้งค่าเพื่อทำฟังก์ชันการลบ

หากรายการยาวเกินไป ส่วนบนสุดอาจหลุดจากหน้าจอ เมื่อต้องการแสดง รายการหนึ่งหน้า (หน้าจอ) ในแต่ละครั้ง ให้พิมพ์คำสั่ง `stty` เข้ากับคำสั่ง `pg` ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
stty -a | pg
```

ให้ดูที่คำสั่ง `stty` ใน *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

หลักการที่เกี่ยวข้อง:

“การยกเลิกกระบวนการพื้นหน้า” ในหน้า 158

หากคุณเริ่มต้นกระบวนการพื้นหน้า แล้วตัดสินใจว่าคุณไม่ต้องการ ให้กระบวนการนั้นเสร็จสิ้น คุณสามารถยกเลิกได้โดยกด INTERRUPT ซึ่งโดยปกติเป็น `Ctrl-C` หรือ `Ctrl-Backspace`

การแสดงผลการตัวแปรสถานะแวดล้อม (คำสั่ง `env`)

เมื่อต้องการแสดงตัวแปรสถานะแวดล้อมปัจจุบัน ให้ใช้คำสั่ง `env` ตัวแปรสถานะแวดล้อมที่สามารถเข้าถึงโปรเซสของคุณทั้งหมดได้เรียกว่า *ตัวแปรสากล*

ตัวอย่างเช่น เมื่อต้องการแสดงรายการตัวแปรสถานะแวดล้อมทั้งหมดและ ค่าที่เชื่อมโยง ให้พิมพ์ดังต่อไปนี้:

```
env
```

ข้อมูล ต่อไปนี้เป็นตัวอย่างของเอาต์พุต:

```
TMPDIR=/usr/tmp  
myid=denise  
LANG=en_US  
UNAME=barnard  
PAGER=/bin/pg
```

```

VISUAL=vi
PATH=/usr/ucb:/usr/lpp/X11/bin:/bin:/usr/bin:/etc:/u/denise:/u/denise/bin:/u/bin1
MAILPATH=/usr/mail/denise?denise has mail !!!
MAILRECORD=/u/denise/.Outmail
EXINIT=set beautify noflash nomesg report=1 showmode showmatch
EDITOR=vi
PSCH=>
HISTFILE=/u/denise/.history
LOGNAME=denise
MAIL=/usr/mail/denise
PS1=denise@barnard:${PWD}>
PS3=#
PS2=>
epath=/usr/bin
USER=denise
SHELL=/bin/ksh
HISTSIZE=500
HOME=/u/denise
FCEDIT=vi
TERM=1ft
MAILMSG=**YOU HAVE NEW MAIL. USE THE mail COMMAND TO SEE YOUR PWD=/u/denise
ENV=/u/denise/.env

```

ถ้ารายการยาวเกินไป ส่วนบนสุด จะหลุดจากหน้าจอ เมื่อต้องการแสดง รายการหนึ่งหน้า (หน้าจอ) ในแต่ละครั้ง ให้พิมพ์คำสั่ง **env** เข้ากับคำสั่ง **pg** ที่พร้อมด ให้พิมพ์ดังต่อไปนี้:

```
env | pg
```

ให้ดูที่คำสั่ง **env** ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงผลตัวแปรสภาวะแวดล้อม (คำสั่ง **printenv**)

เมื่อต้องการแสดงค่าของตัวแปรสภาวะแวดล้อม ให้ใช้คำสั่ง **printenv**

หากคุณระบุพารามิเตอร์ **Name** ระบบจะพิมพ์ เฉพาะค่าที่เชื่อมโยงกับตัวแปรซึ่งคุณร้องขอเท่านั้น หากคุณไม่ได้ระบุ พารามิเตอร์ **Name** คำสั่ง **printenv** จะแสดง ตัวแปรสภาวะแวดล้อมปัจจุบันทั้งหมด โดยแสดงหนึ่งลำดับ **Name = Value** ต่อบรรทัด

ตัวอย่างเช่น เมื่อต้องการค้นหาค่าติดตั้งปัจจุบันของตัวแปรสภาวะแวดล้อม **MAILMSG** ให้พิมพ์ดังต่อไปนี้:

```
printenv MAILMSG
```

คำสั่ง ส่งคืนค่าของตัวแปรสภาวะแวดล้อม **MAILMSG** ตัวอย่างเช่น:

```
YOU HAVE NEW MAIL
```

ให้ดูที่คำสั่ง **printenv** ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

ภาษาสองทาง (คำสั่ง **aixterm**)

คำสั่ง **aixterm** สนับสนุนภาษาอาร์บิกและฮิบรู ซึ่งเป็นภาษาสองทาง

ภาษาสองทางมีความสามารถในการอ่านและ เขียนในสองทิศทาง: จากซ้ายไปขวา และจากขวาไปซ้าย คุณสามารถทำงาน กับแอปพลิเคชันภาษาอาร์บิกและฮิบรูโดยการเปิดหน้าต่าง ที่ระบุโลแคลภาษาอาร์บิกหรือฮิบรู

ให้ดูที่คำสั่ง `aixterm` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

ข้อมูลสรุปคำสั่งสำหรับสภาวะแวดล้อมผู้ใช้และข้อมูลระบบ

ข้อมูลต่อไปนี้ เป็นคำสั่งสำหรับสภาวะแวดล้อมผู้ใช้และข้อมูลระบบ

ไอเท็ม	คำอธิบาย
<code>aixterm</code>	ช่วยให้คุณสามารถทำงานกับภาษาสองทาง
<code>env</code>	แสดงสภาวะแวดล้อมปัจจุบันหรือตั้งค่าสภาวะแวดล้อม สำหรับการดำเนินการคำสั่ง
<code>lscfg</code>	แสดงข้อมูลการวิเคราะห์เกี่ยวกับอุปกรณ์
<code>lscons</code>	แสดงชื่อของคอนโซลปัจจุบัน
<code>lsdisp</code>	แสดงรายการจอแสดงผลที่มีอยู่ในปัจจุบันบนระบบ
<code>lsfont</code>	แสดงรายการฟอนต์ที่พร้อมใช้งานสำหรับจอแสดงผล
<code>lskbd</code>	แสดงรายการคีย์บอร์ดแม่พิมพ์ซึ่งโหลดในปัจจุบันในระบบ
<code>lslpp</code>	แสดงรายการผลิตภัณฑ์ซอฟต์แวร์
<code>printenv</code>	แสดงค่าของตัวแปรสภาวะแวดล้อม
<code>stty</code>	แสดงค่าที่ตั้งระบบ
<code>tty</code>	แสดงชื่อพารามิเตอร์เพิ่มเติมของเทอร์มินัลของคุณ

การกำหนดสภาวะแวดล้อมผู้ใช้เอง

ระบบปฏิบัติการจัดเตรียมคำสั่งและไฟล์การเริ่มต้นจำนวนมาก ที่ช่วยให้คุณกำหนดลักษณะการทำงานและลักษณะปรากฏของสภาวะแวดล้อม ผู้ใช้ของคุณได้เอง

คุณยังสามารถกำหนดดีฟอลต์รีซอร์สบางส่วนของแอปพลิเคชันซึ่งคุณใช้ บนระบบได้เองด้วย คำดีฟอลต์มีการเริ่มต้นโดยโปรแกรมที่สตาร์ทออฟ เมื่อคุณเปลี่ยนค่าดีฟอลต์ คุณต้องออก แล้วรีสตาร์ทโปรแกรมเพื่อให้ คำดีฟอลต์ใหม่มีผลบังคับใช้

หากต้องการข้อมูลเพิ่มเติมเกี่ยวกับการกำหนดลักษณะการทำงานและลักษณะปรากฏของ Common Desktop Environment เอง ให้ดูที่ *Common Desktop Environment 1.0: คู่มือผู้ใช้และผู้ดูแลระบบขั้นสูง*

ไฟล์สตาร์ทอัพระบบ:

เมื่อคุณล็อกอิน เซลล์กำหนดสภาวะแวดล้อมผู้ใช้ของคุณหลังจาก การอ่านไฟล์การเริ่มต้นซึ่งคุณตั้งค่าไว้ ลักษณะของ สภาวะแวดล้อมผู้ใช้ถูกกำหนดโดยค่าที่กำหนดให้กับตัวแปร สภาวะแวดล้อมของคุณ คุณรักษาสภาวะแวดล้อมนี้จนกว่าคุณล็อกเอาต์ออกจากระบบ

เซลล์ใช้ไฟล์โปรไฟล์สองชนิดเมื่อคุณล็อกอินเข้าสู่ ระบบปฏิบัติการ เซลล์ประเมินคำสั่งที่มีอยู่ในไฟล์ จากนั้นดำเนินการ คำสั่ง เพื่อตั้งค่าสภาวะแวดล้อมของระบบ ไฟล์มีฟังก์ชันคล้ายกัน ยกเว้นว่าไฟล์ `/etc/profile` ควบคุมตัวแปรโปรไฟล์ สำหรับผู้ใช้ทั้งหมดบนระบบ ในขณะที่ไฟล์ `.profile` อนุญาตให้คุณกำหนดสภาวะแวดล้อมของคุณเอง

อันดับแรก เซลล์รันคำสั่งเพื่อตั้งค่าสภาวะแวดล้อมของระบบในไฟล์ `/etc/environment` จากนั้น ประเมินคำสั่งที่มีอยู่ในไฟล์ `/etc/profile` หลังจากรันไฟล์เหล่านี้แล้ว ระบบตรวจสอบเพื่อดูว่าคุณมีไฟล์ `.profile` อยู่ใน ไดเรกทอรีโฮมของคุณหรือไม่ ถ้ามีไฟล์ `.profile` อยู่ ระบบจะรันไฟล์นี้ ไฟล์ `.profile` จะระบุ ถ้ามีไฟล์สภาวะแวดล้อมด้วย ถ้ามีไฟล์สภาวะแวดล้อม (โดยปกติชื่อว่า `.env`) ระบบจะรันไฟล์นี้และตั้งค่า ตัวแปรสภาวะแวดล้อมของคุณ

ไฟล์ `/etc/environment`, `/etc/profile`, และ `.profile` รันหนึ่งครั้งในเวลาล็อกอิน ในทางตรงกันข้าม ไฟล์ `.env` รันในทุกครั้งที่คุณเปิดเซลล์ใหม่หรือหน้าต่าง

ไฟล์ /etc/environment:

ไฟล์แรกซึ่งระบบปฏิบัติการใช้ในเวลาล็อกอินคือไฟล์ /etc/environment ไฟล์ /etc/environment มีตัวแปรที่ระบุสถานะแวดล้อมพื้นฐานสำหรับโปรเซสทั้งหมด

เมื่อโปรเซสใหม่เริ่มขึ้น รูทีนย่อย exec จัดทำอาร์เรย์ของสตริงที่มีอยู่ในรูปแบบ Name=Value อาร์เรย์ของสตริงนี้เรียกว่า *สภาวะแวดล้อม* แต่ละชื่อถูกกำหนดโดยสตริงอย่างใดอย่างหนึ่งที่เรียกว่า *ตัวแปรสภาวะแวดล้อม* หรือ *ตัวแปรเชลล์* รูทีนย่อย exec ช่วยให้สามารถตั้งค่าทั้ง สภาวะแวดล้อมได้ในครั้งเดียว

เมื่อคุณล็อกอิน ระบบกำหนดตัวแปรสภาวะแวดล้อมจากไฟล์ /etc/environment ก่อนการอ่านโปรไฟล์ล็อกอินของคุณที่ชื่อ .profile ตัวแปร ต่อไปนี้ประกอบขึ้นเป็นสภาวะแวดล้อมพื้นฐาน:

ไอเท็ม	คำอธิบาย
HOME	ชื่อพารามิเตอร์ของล็อกอินของผู้ใช้หรือไดเรกทอรี HOME โปรแกรม login ตั้งค่านี้เป็นชื่อที่ระบุในไฟล์ /etc/passwd
LANG	ชื่อ locale ที่ใช้อยู่ในปัจจุบัน ตัวแปร LANG มีการตั้งค่า ครั้งแรกในไฟล์ /etc/profile ในเวลาติดตั้ง
NLS_PATH	ชื่อพารามิเตอร์ของแค็ตตาล็อกข้อความ
LOCPATH	ชื่อพารามิเตอร์ของที่ตั้งของตาราง National Language Support
PATH	ลำดับของไดเรกทอรีที่คำสั่ง เช่น sh, time, nice และ nohup, ค้นหาเมื่อค้นหาคำสั่งซึ่งมีชื่อพารามิเตอร์ไม่สมบูรณ์
TZ	ข้อมูลโซนเวลา ตัวแปรสภาวะแวดล้อม TZ มีการตั้งค่าครั้งแรกโดยไฟล์ /etc/profile ซึ่งเป็นโปรไฟล์ ล็อกอินระบบ

สำหรับข้อมูลรายละเอียดเกี่ยวกับไฟล์ /etc/environment ให้ดูที่ *Files Reference*

ไฟล์ /etc/profile:

ไฟล์ลำดับที่สองซึ่งระบบปฏิบัติการใช้ในเวลาล็อกอินคือไฟล์ /etc/profile

ไฟล์ /etc/profile ควบคุมตัวแปรที่พอลต์ทั่วทั้งระบบ เช่น:

- ตัวแปรเอ็กซ์พอร์ต
- มาสก์การสร้างไฟล์ (umask)
- ชนิดเทอร์มินัล
- ข้อความเมลเพื่อบ่งชี้เวลาที่ได้รับเมลใหม่

ผู้ดูแลระบบกำหนดคอนฟิกไฟล์ /etc/profile สำหรับผู้ใช้ทั้งหมดบนระบบ เฉพาะผู้ดูแลระบบเท่านั้นสามารถเปลี่ยนไฟล์นี้ได้

ตัวอย่างต่อไปนี้ เป็นไฟล์ /etc/profile ปกติ:

```
#Set file creation mask
umask 022
#Tell me when new mail arrives
MAIL=/usr/mail/$LOGNAME
#Add my /bin directory to the shell search sequence
PATH=/usr/bin:/usr/sbin:/etc::
```

```
#Set terminal type
TERM=ift
#Make some environment variables global
export MAIL PATH TERM
```

สำหรับข้อมูลรายละเอียดเกี่ยวกับไฟล์ `/etc/profile` ให้อ่านที่ *Files Reference*

ไฟล์ `.profile`:

ไฟล์ `.profile` มีอยู่ในไดเรกทอรีโฮม (`$HOME`) ของคุณและอนุญาตให้คุณกำหนดสภาวะแวดล้อมการทำงาน เองได้

เนื่องจากไฟล์ `.profile` ถูกซ่อนไว้ให้ใช้คำสั่ง `ls -a` เพื่อแสดงไฟล์

หลังจากโปรแกรม `login` เพิ่มตัวแปร `LOGNAME` (ชื่อ ล็อกอิน) และ `HOME` (ไดเรกทอรีล็อกอิน) ลงใน สภาวะแวดล้อมแล้ว จะมีการดำเนินการคำสั่งในไฟล์ `$HOME/.profile` ถ้ามีไฟล์อยู่ ไฟล์ `.profile` มีโปรไฟล์แต่ละโปรไฟล์ของคุณที่จะยกเลิก ตัวแปรที่ตั้งค่าไว้ในไฟล์ `/etc/profile` ไฟล์ `.profile` มีการใช้บ่อยเพื่อตั้งค่าตัวแปรสภาวะแวดล้อมที่เอ็กซ์พอร์ตและ โหมดเทอร์มินัล คุณสามารถกำหนดสภาวะแวดล้อมของคุณเองโดยการแก้ไขไฟล์ `.profile` ใช้ไฟล์ `.profile` เพื่อควบคุม ค่าที่พอลดต่อไปนี้:

- เซลล์ที่จะเปิด
- ลักษณะปรากฏของพร้อมต์
- เสี่ยงคีย์บอร์ด

ตัวอย่างต่อไปนี้เป็นไฟล์ `.profile` ปกติ:

```
PATH=/usr/bin:/etc:/home/bin1:/usr/lpp/tps4.0/user::
epath=/home/gsc/e3:
export PATH epath
csh
```

ตัวอย่างนี้กำหนดตัวแปรพาสสองตัว (`PATH` และ `epath`) เอ็กซ์พอร์ตตัวแปรนั้น และเปิด C เซลล์ (`csh`)

คุณยังสามารถใช้ไฟล์ `.profile` (หรือถ้าไม่มีอยู่ไฟล์ `/etc/profile`) เพื่อกำหนดตัวแปรเซลล์การล็อกอิน ได้ด้วย คุณยังสามารถกำหนดสภาวะแวดล้อมเซลล์อื่นเองได้ด้วย ตัวอย่างเช่น ใช้ไฟล์ `.cshrc` และไฟล์ `.kshrc` เพื่อกำหนด C เซลล์และ Korn เซลล์เอง ตามลำดับ เมื่อเซลล์แต่ละชนิด เริ่มต้นขึ้น

ไฟล์ `.env`:

ไฟล์ลำดับที่สี่ซึ่งระบบปฏิบัติการใช้ในเวลาล็อกอินคือไฟล์ `.env` ถ้า `.profile` ของคุณมีบรรทัดต่อไปนี้: `export ENV=$HOME/.env`

ไฟล์ `.env` อนุญาตให้คุณกำหนดตัวแปรสภาวะแวดล้อมการทำงาน แต่ละตัวของตนเอง เนื่องจากไฟล์ `.env` ถูกซ่อนไว้ให้ใช้ คำสั่ง `ls -a` เพื่อแสดงไฟล์ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `ls` ให้อ่านที่ `ls` ไฟล์ `.env` มีตัวแปรสภาวะแวดล้อมผู้ใช้แต่ละตัวซึ่ง ยกเลิกตัวแปร ที่กำหนดไว้ในไฟล์ `/etc/environment` คุณสามารถกำหนดตัวแปรสภาวะแวดล้อม เองได้ตามต้องการโดยการแก้ไขไฟล์ `.env`

ตัวอย่างต่อไปนี้เป็นไฟล์ `.env` ปกติ:

```

export myid=`id | sed -n -e 's/).*$/' -e 's/^.*(//p'`
#set prompt: login & system name & path
if [ $myid = root ]
    then typeset -x PSCH='#:\${PWD}> '
        PS1="#:\${PWD}> "
    else typeset -x PSCH='>'
        PS1="\$LOGNAME@\$UNAME:\${PWD}> "
        PS2=">"
        PS3="#?"
fi
export PS1 PS2 PS3
#setup my command aliases
alias ls="/bin/ls -CF" \
    d="/bin/ls -Fal | pg" \
    rm="/bin/rm -i" \
    up="cd .."

```

หมายเหตุ: เมื่อแก้ไขไฟล์ .env ตรวจสอบให้แน่ใจว่าตัวแปรสถานะแวดล้อมที่สร้างขึ้นใหม่ไม่ขัดแย้งกับตัวแปรมาตรฐาน เช่น MAIL, PS1, PS2, และ IFS

ไฟล์สตาร์ทอัพ AIXwindows:

ระบบคอมพิวเตอร์ที่แตกต่างกันมีวิธีการเริ่มต้นเซิร์ฟเวอร์ X และ AIXwindows ที่แตกต่างกัน

เนื่องจากระบบคอมพิวเตอร์ที่แตกต่างกันมีวิธีการเริ่มต้นเซิร์ฟเวอร์ X และ AIXwindows ที่แตกต่างกัน ดังนั้นโปรดปรึกษาคู่มือและระบบของคุณเพื่อศึกษาวิธีการเริ่มต้น โดยปกติ เซิร์ฟเวอร์ X และ AIXwindows มีการเริ่มต้น จากเซลล์สคริปต์ที่รันโดยอัตโนมัติเมื่อคุณล็อกอิน อย่างไรก็ตาม คุณอาจพบว่าคุณต้องเริ่มต้นเซิร์ฟเวอร์ X หรือ AIXwindows หรือทั้งสองอย่าง

ถ้าคุณล็อกอินและพบว่าจอแสดงผลของคุณทำงานเป็นเทอร์มินัลเดียวโดยไม่มีหน้าต่างแสดงขึ้น คุณสามารถเริ่มต้นเซิร์ฟเวอร์ X โดยพิมพ์ดังต่อไปนี้:

```
xinit
```

หมายเหตุ: ก่อนป้อนคำสั่งนี้ ตรวจสอบให้แน่ใจว่าตัวชี้วางอยู่ใน หน้าต่างที่มีพร้อมระบบ

หากคำสั่งนี้ไม่ได้เริ่มต้นเซิร์ฟเวอร์ X ให้ตรวจสอบกับผู้ดูแลระบบของคุณ เพื่อให้แน่ใจว่าพารามิเตอร์ค้นหาของคุณมีไดเรกทอรี X11 ซึ่งมีโปรแกรมที่ปฏิบัติการได้ พารามิเตอร์ที่เหมาะสมอาจแตกต่างกันไปในแต่ละระบบ

ถ้าคุณล็อกอินและพบหน้าต่างที่ไม่มีกรอบตั้งแต่หนึ่งบานขึ้นไป คุณสามารถเริ่มต้น AIXwindows Window Manager โดยพิมพ์ ดังต่อไปนี้:

```
mwm &
```

เนื่องจาก AIXwindows อนุญาต การกำหนดเองทั้งโดยโปรแกรมเมอร์ที่เขียนแอปพลิเคชัน AIXwindows และโดยผู้ใช้ คุณจึงอาจพบว่าปุ่มเมาส์หรือฟังก์ชันอื่นไม่ทำงาน ดังที่คุณคาดหวังจากการอ่านเอกสารคู่มือนี้ คุณสามารถรีเซ็ต สถานะแวดล้อม AIXwindows เป็น ลักษณะการทำงานดีฟอลต์โดยกดปุ่มต่อไปนี้ค้างไว้:

Alt-Ctrl-Shift-!

คุณสามารถกลับไปยังลักษณะการทำงานที่กำหนดเองได้โดยกดปุ่มนี้ตามลำดับ อีกครั้ง หากระบบของคุณไม่อนุญาตให้กดชุดปุ่มนี้ คุณยังสามารถเรียกคืนลักษณะการทำงานดีฟอลต์จากเมนูรากดีฟอลต์ได้

ไฟล์ *.xinitrc*:

คำสั่ง *xinit* ใช้ไฟล์เชลล์สคริปต์ที่กำหนดเองได้ ซึ่งแสดงรายการโปรแกรม X Client ที่จะเริ่มต้น ไฟล์ *.xinitrc* ใน ไดเรกทอรี โห้ยมควบคุมหน้าต่างและแอ็พพลิเคชันที่จะเริ่มต้นเมื่อ คุณเริ่มต้น AIXwindows

คำสั่ง *xinit* ทำงานกับเชลล์สคริปต์ในลำดับ ต่อไปนี้:

1. อันดับแรก คำสั่ง *xinit* ค้นหาตัวแปรสถานะแวดล้อม *\$XINTRC* เพื่อเริ่มต้น AIXwindows
2. หากไม่พบตัวแปรสถานะแวดล้อม *\$XINTRC* คำสั่ง *xinit* จะค้นหาเชลล์สคริปต์ *\$HOME/.xinitrc*
3. หากไม่พบเชลล์สคริปต์ *\$HOME/.xinitrc* คำสั่ง *xinit* จะเริ่มต้นเชลล์สคริปต์ */usr/lib/X11/\$LANG/xinitrc*
4. หากไม่พบ */usr/lib/X11/\$LANG/xinitrc* คำสั่งจะค้นหา เชลล์สคริปต์ */usr/lpp/X11/defaults/\$LANG/xinitrc* หากไม่พบสคริปต์นั้น คำสั่งจะค้นหาเชลล์สคริปต์ */usr/lpp/X11/defaults/xinitrc*
5. เชลล์สคริปต์ *xinitrc* เริ่มต้นคำสั่ง เช่น *mwm* (AIXwindows Window Manager), *aixterm*, และคำสั่ง *xclock*

คำสั่ง *xinit* ทำการดำเนินงานต่อไปนี้:

- เริ่มต้นเซิร์ฟเวอร์ X บนจอแสดงผลปัจจุบัน
- ตั้งค่าตัวแปรสถานะแวดล้อม *\$DISPLAY*
- รันไฟล์ *xinitrc* เพื่อเริ่มต้นโปรแกรม X Client

ตัวอย่างต่อไปนี้แสดงส่วนของไฟล์ *xinitrc* ซึ่ง คุณสามารถกำหนดเองได้:

```
# This script is invoked by /usr/lpp/X11/bin/xinit
.
.
.
#*****
# Start the X clients. Change the following lines to      *
# whatever command(s) you desire! *
# The default clients are an analog clock (xclock), an lft *
# terminal emulator (aixterm), and the Motif Window Manager *
# (mwm). *
#*****
exec mwm
```

ไฟล์ *.Xdefaults*:

หากคุณทำงานในอินเทอร์เฟซ AIXwindows คุณสามารถกำหนด อินเทอร์เฟซนี้เองโดยใช้ไฟล์ *.Xdefaults* AIXwindows อนุญาตให้คุณระบุการกำหนดค่าตามความชอบสำหรับลักษณะที่มองเห็นได้ เช่น สีและฟอนต์

หลายๆ ลักษณะของระบบปฏิบัติการ Windows ตามรูปลักษณะและพฤติกรรมของแอปพลิเคชันถูกควบคุมโดยชุดตัวแปรที่ เรียกว่า *resources* ลักษณะที่มองเห็นได้หรือการทำงานของรีซอร์สถูกกำหนดโดยค่า ที่กำหนดให้ คำสำหรับรีซอร์สมีอยู่หลาย ชนิด ตัวอย่างเช่น รีซอร์สที่ควบคุมสีสามารถได้รับการกำหนดค่าที่กำหนดไว้ล่วงหน้า เช่น *DarkSlateBlue* หรือ *Black* รีซอร์สที่ ระบุมิติจะได้รับการกำหนดค่าตัวเลข บางรีซอร์ส ใช้คำบูลีน (จริง หรือ เท็จ)

หากคุณไม่มีไฟล์ `.Xdefaults` อยู่ในไดเรกทอรีโฮมของคุณ คุณสามารถสร้างขึ้นโดยใช้โปรแกรมแก้ไขข้อความใดๆ หลังจากคุณมีไฟล์นี้ อยู่ในไดเรกทอรีโฮมแล้ว คุณสามารถตั้งค่ารีซอร์สในไฟล์ได้ตามต้องการไฟล์ดีฟอลต์ ตัวอย่างที่เรียกว่า `Xdefaults.tmp1` อยู่ในไดเรกทอรี `/usr/lpp/X11/defaults`

ตัวอย่างต่อไปนี้จะแสดงส่วนของไฟล์ `.Xdefaults` ปกติ:

```
*AutoRaise: on
*DeIconifyWarp: on
*warp:on
*TitleFont:andysans12
*scrollBar: true
*font: Rom10.500
Mwm*menu*foreground: black
Mwm*menu*background: CornflowerBlue
Mwm*menu*RootMenu*foreground: black
Mwm*menu*RootMenu*background: CornflowerBlue
Mwm*icon*foreground: grey25
Mwm*icon*background: LightGray
Mwm*foreground: black
Mwm*background: LightSkyBlue
Mwm*bottomShadowColor: Blue1
Mwm*topShadowColor: CornflowerBlue
Mwm*activeForeground: white
Mwm*activeBackground: Blue1
Mwm*activeBottomShadowColor: black
Mwm*activeTopShadowColor: LightSkyBlue
Mwm*border: black
Mwm*highlight:white

aixterm.foreground: green
aixterm.background: black
aixterm.fullcursor: true
aixterm.ScrollKey: on
aixterm.autoRaise: true
aixterm.autoRaiseDelay: 2
aixterm.boldFont:Rom10.500
aixterm.geometry: 80x25
aixterm.iconFont: Rom8.500
aixterm.iconStartup: false
aixterm.jumpScroll: true
aixterm.reverseWrap: true
aixterm.saveLines: 500
aixterm.scrollInput: true
aixterm.scrollKey: false
aixterm.title: AIX
```

ไฟล์ `.mwmrc`:

คุณลักษณะส่วนใหญ่ซึ่งคุณต้องการกำหนดเองสามารถตั้งค่าได้โดยใช้รีซอร์สในไฟล์ `.Xdefaults` อย่างไรก็ตาม การผูกคีย์ การผูกปุ่มเมาส์ และนิยามเมนูสำหรับตัวจัดการหน้าต่างมีการระบุไว้ในไฟล์ `.mwmrc` เสริม ซึ่งมีการอ้างอิงโดยรีซอร์สในไฟล์ `.Xdefaults`

หากคุณไม่มีไฟล์ `.mwmrc` อยู่ในไดเรกทอรีโฮมของคุณ คุณสามารถคัดลอกไฟล์ได้ดังนี้:

```
cp /usr/lib/X11/system.mwmrc .mwmrc
```

เนื่องจากไฟล์ .mwmrc ยกเลิกการบังคับใช้ทั่วทั้งระบบของไฟล์ system.mwmrc ข้อกำหนดคุณลักษณะของคุณ จึงไม่รบกวนข้อกำหนดคุณลักษณะของผู้ใช้อื่น

ตัวอย่างต่อไปนี้แสดงส่วนของไฟล์ system.mwmrc ปกติ:

```
# DEFAULT mwm RESOURCE DESCRIPTION FILE (system.mwmrc)
#
# menu pane descriptions
#
# Root Menu Description

Menu RootMenu
{ "Root Menu"      f.title
  no-label         f.separator
  "New Window"    f.exec "aixterm &"
  "Shuffle Up"    f.circle_up
  "Shuffle Down"  f.circle_down
  "Refresh"       f.refresh
  no-label         f.separator
  "Restart"       f.restart
  "Quit"          f.quit_mwm
}

# Default Window Menu Description

Menu DefaultWindowMenu MwmWindowMenu
{ "Restore"  _R  Alt<Key>F5      f.normalize
  "Move"    _M  Alt<Key>F7      f.move
  "Size"    _S  Alt<Key>F8      f.resize
  "Minimize" _n  Alt<Key>F9      f.minimize
  "Maximize" _x  Alt<Key>F10    f.maximize
  "Lower"   _L  Alt<Key>F3      f.lower
  no-label  f.separator
  "Close"   _C  Alt<Key>F4      f.kill
}

# no acclerator window menu
Menu NoAccWindowMenu
{
  "Restore"  _R  f.normalize
  "Move"    _M  f.move
  "Size"    _S  f.resize
  "Minimize" _n  f.minimize
  "Maximize" _x  f.maximize
  "Lower"   _L  f.lower
  no-label  f.separator
  "Close"   _C  f.kill
}

Keys DefaultKeyBindings
{
  Shift<Key>Escape      icon|window      f.post_wmenu
  Meta<Key>space        icon|window      f.post_wmenu
  Meta<Key>Tab          root|icon|window f.next_key
  Meta Shift<Key>Tab    root|icon|window f.prev_key
}
```

```

Meta<Key>Escape          root|icon|window  f.next_key
Meta Shift<Key>Escape    root|icon|window  f.prev_key
Meta Ctrl Shift<Key>exclam root|icon|window  f.set_behavior
}
#
# button binding descriptions
#
Buttons DefaultButtonBindings
{
<Btn1Down>          frame|icon          f.raise
<Btn3Down>          frame|icon          f.post_wmenu
<Btn1Down>          root                f.menu  RootMenu
<Btn3Down>          root                f.menu  RootMenu
Meta<Btn1Down>      icon|window         f.lower
Meta<Btn2Down>      window|icon         f.resize
Meta<Btn3Down>      window              f.move
}

Buttons PointerButtonBindings
{
<Btn1Down>          frame|icon          f.raise
<Btn2Down>          frame|icon          f.post_wmenu
<Btn3Down>          frame|icon          f.lower
<Btn1Down>          root                f.menu  RootMenu
Meta<Btn2Down>      window|icon         f.resize
Meta<Btn3Down>      window|icon         f.move
}

#
# END OF mwm RESOURCE DESCRIPTION FILE
#

```

การเอ็กซ์พอร์ตตัวแปรเชลล์ (คำสั่ง export shell):

ตัวแปรเชลล์แบบ *โลคัล* คือตัวแปรที่รู้จักโดย เชลล์ที่สร้างตัวแปรนั้นเท่านั้น หากคุณเริ่มต้นเชลล์ใหม่ เชลล์ใหม่จะไม่รู้จักตัวแปรของเชลล์เก่า ถ้าคุณต้องการให้เชลล์ใหม่ซึ่งคุณเปิด ใช้ตัวแปรจากเชลล์เก่า ให้เอ็กซ์พอร์ตตัวแปรเพื่อทำให้ตัวแปรนั้นเป็นแบบ *สากล*

คุณสามารถใช้คำสั่ง `export` เพื่อทำให้ตัวแปร *โลคัล* เป็นตัวแปรแบบสากล เมื่อต้องการทำให้ตัวแปรเชลล์แบบ *โลคัล* เป็นตัวแปรแบบสากล โดยอัตโนมัติ ให้เอ็กซ์พอร์ตตัวแปรนั้นในไฟล์ `.profile` ของคุณ

หมายเหตุ: ตัวแปร สามารถถูกเอ็กซ์พอร์ตลงไปยังเชลล์ชายนได้ แต่ไม่สามารถเอ็กซ์พอร์ตขึ้นไปยัง เชลล์พารেন্ট

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการทำให้ตัวแปรเชลล์ *โลคัล* `PATH` เป็นแบบสากล ให้พิมพ์ดังต่อไปนี้:

```
export PATH
```
- เมื่อต้องการแสดงรายการตัวแปรที่เอ็กซ์พอร์ตทั้งหมดของคุณ ให้พิมพ์ดังต่อไปนี้:

```
export
```

ระบบ แสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```

DISPLAY=unix:0
EDITOR=vi
ENV=$HOME/.env
HISTFILE=/u/denise/.history
HISTSIZ=500
HOME=/u/denise
LANG=En_US
LOGNAME=denise
MAIL=/usr/mail/denise
MAILCHECK=0
MAILMSG=**YOU HAVE NEW MAIL.
USE THE mail COMMAND TO SEE YOUR MAILPATH=/usr/mail/denise?denise has mail !!!
MAILRECORD=/u/denise/.Outmail
PATH=/usr/ucb:/usr/lpp/X11/bin:/bin:/usr/bin:/etc:/u/denise:/u/denise/bin:/u/bin1
PWD=/u/denise
SHELL=/bin/ksh

```

การเปลี่ยนดีฟอลต์ฟอนต์ (คำสั่ง chfont):

เมื่อต้องการเปลี่ยนดีฟอลต์ฟอนต์ที่สตาร์ทอัพระบบ ให้ใช้คำสั่ง **chfont** หรือ **smit ฟอนต์พาเลต** คือไฟล์ที่ระบบใช้เพื่อกำหนดและระบุฟอนต์ที่มีอยู่

หมายเหตุ: เมื่อต้องการรันคำสั่ง **chfont** คุณต้องมีสิทธิ์การใช้งานราก

คำสั่ง chfont

ดูตัวอย่างวิธีการใช้คำสั่ง **chfont** ต่อไปนี้:

- เมื่อต้องการเปลี่ยนฟอนต์ที่ใช้งานอยู่เป็นฟอนต์ลำดับที่ห้าในฟอนต์พาเลต ให้พิมพ์ดังต่อไปนี้:
chfont -a5
- เมื่อต้องการเปลี่ยนฟอนต์เป็นตัวเอน, roman, และตัวหนาในขนาดเดียวกัน ให้พิมพ์ดังต่อไปนี้:
chfont -n /usr/lpp/fonts/It114.snf /usr/lpp/fonts/B1d14.snf /usr/lpp/fonts/Rom14.snf

ให้ดูที่คำสั่ง **chfont** ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

คำสั่ง smit

นอกจากนี้ สามารถรันคำสั่ง **chfont** โดยใช้ **smit** ได้ด้วย

เมื่อต้องการเลือกฟอนต์ที่ใช้งานอยู่ ให้พิมพ์ดังต่อไปนี้:

```
smit chfont
```

เมื่อต้องการเลือกฟอนต์พาเลต ให้พิมพ์ดังต่อไปนี้:

```
smit chfontpl
```

การเปลี่ยนคีย์ควบคุม (คำสั่ง stty):

เมื่อต้องการเปลี่ยนคีย์ที่เทอร์มินัลของคุณใช้เป็นคีย์ควบคุม ให้ใช้ คำสั่ง **stty**

การเปลี่ยนแปลงในคีย์ควบคุมของคุณยังคงมีผลบังคับใช้จนกว่าคุณล็อกเอาต์ เมื่อต้องการทำการเปลี่ยนแปลงถาวรให้ใส่การเปลี่ยนแปลงไว้ในไฟล์ `.profile`

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการกำหนด Ctrl-Z เป็นคีย์ขัดจังหวะให้พิมพ์ดังต่อไปนี้:

```
stty intr ^Z
```

ต้องแน่ใจว่าวางอักขระช่องว่างไว้ระหว่าง `intr` และ `^Z`

- เมื่อต้องการรีเซ็ตคีย์ควบคุมทั้งหมดกลับเป็นค่าดีฟอลต์ให้พิมพ์ดังต่อไปนี้:

```
stty sane
```

- เมื่อต้องการแสดงค่าที่ตั้งปัจจุบันของคุณให้พิมพ์ดังต่อไปนี้:

```
stty -a
```

ให้ดูที่คำสั่ง `stty` ใน *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

การเปลี่ยนพร้อมระบบของคุณ:

คุณสามารถเปลี่ยนพร้อมระบบของคุณได้

เซลล์ของคุณใช้ตัวแปรพร้อมต่อไปนี้:

ไอเท็ม	คำอธิบาย
<code>PS1</code>	พร้อมที่ใช้เป็นพร้อมระบบปกติ
<code>PS2</code>	พร้อมที่ใช้เมื่อเซลล์คาดหวังอินพุตเพิ่มเติม
<code>PS3</code>	พร้อมที่ใช้เมื่อคุณมีสิทธิ์การใช้งานราก

คุณสามารถเปลี่ยนอักขระพร้อมใดๆ ของคุณได้โดยการเปลี่ยนค่าของตัวแปรเซลล์ การเปลี่ยนพร้อมของคุณยังคงมีผลบังคับใช้จนกว่าคุณล็อกเอาต์ เมื่อต้องการทำการเปลี่ยนแปลงถาวรให้ใส่การเปลี่ยนแปลงไว้ในไฟล์ `.env`

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการแสดงค่าปัจจุบันของตัวแปร `PS1` ให้พิมพ์ ดังต่อไปนี้:

```
echo "prompt is $PS1"
```

ระบบ แสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
prompt is $
```

- เมื่อต้องการเปลี่ยนพร้อมเป็น `Ready>` ให้พิมพ์ดังต่อไปนี้:

```
PS1="Ready> "
```

- เมื่อต้องการเปลี่ยนพร้อมการต่อเนื่องเป็น `Enter more->` ให้พิมพ์ ดังต่อไปนี้:

```
PS2="Enter more->"
```

- เมื่อต้องการเปลี่ยนพร้อมรากเป็น `Root->` ให้พิมพ์ดังต่อไปนี้:

```
PS3="Root-> "
```

การอ้างอิงระบบ BSD

ภาคผนวกนี้จัดทำขึ้นสำหรับผู้ดูแลระบบที่มีความคุ้นเคยกับระบบปฏิบัติการ 4.3 BSD UNIX หรือ System V ข้อมูลนี้อธิบายความแตกต่างและความคล้ายกันระหว่างระบบดังกล่าวและ AIX

หัวข้อที่อธิบายในภาคผนวกนี้มีดังนี้:

แนวคิด BSD

ก่อนคุณเริ่มต้นทำงานกับการกระจายซอฟต์แวร์ของ Berkeley (BSD) คุณต้องเข้าใจความแตกต่างบางอย่างระหว่าง BSD และ AIX

บทนำเกี่ยวกับ AIX for BSD system managers:

ข้อมูลต่อไปนี้เป็นเคล็ดลับที่ช่วยผู้จัดการระบบการกระจายซอฟต์แวร์ของ Berkeley (BSD) เริ่มต้นจัดการ AIX

- เริ่มต้นโดยการล็อกอินเป็นรากที่กราฟิกคอนโซล
- ทำการจัดการระบบจากคอนโซลระบบจนกว่าคุณคุ้นเคยกับระบบ เป็นการง่ายกว่าที่จะทำงานจากคอนโซลระบบแทนรีโมต เทอร์มินัล หลังจากคุณคุ้นเคยกับระบบแล้ว คุณสามารถทำงานแบบรีโมต จาก xterm หรือเทอร์มินัล ASCII ได้
- ใช้ประโยชน์จากสิ่งอำนวยความสะดวก AIX หลายอย่าง สำหรับภารกิจการจัดการระบบ ซึ่งรวมถึง:
 - System Management Interface Tool (SMIT) SMIT นำเสนออินเตอร์เฟซระหว่าง ผู้จัดการระบบและคำสั่งคอนพิกูเรชันและการจัดการ SMIT สามารถช่วยผู้จัดการระบบ ทำภารกิจการจัดการระบบส่วนใหญ่ได้
 - Object Data Manager (ODM) ODM นำเสนอวิธีที่เข้าถึงอ็อบเจกต์ จากฐานข้อมูล ODM ฐานข้อมูล ODM มีข้อมูลคอนพิกูเรชันอุปกรณ์
 - System Resource Controller (SRC) SRC นำเสนอการเข้าถึงและการควบคุม daemons และรีซอร์สระบบอื่นผ่านทางอินเตอร์เฟซเดียว

หลักการที่เกี่ยวข้อง:

“คอนพิกูเรชันของอุปกรณ์จำนวนมาก” ในหน้า 423

อุปกรณ์รวมถึงฮาร์ดแวร์คอมพิวเตอร์ เช่น เครื่องพิมพ์ ไดรฟ์ อะแดปเตอร์ บัส และส่วนแนบ ตลอดจนอุปกรณ์เทียม เช่น ไฟล์พิเศษ ขอบผิดพลาดและไฟล์พิเศษ null ไดรเวอร์อุปกรณ์ตั้งอยู่ในไดเรกทอรี /usr/lib/drivers

“System Resource Controller” ในหน้า 203

System Resource Controller (SRC) นำเสนอชุดของคำสั่ง และวิธีที่ย่อยที่ทำให้ผู้จัดการระบบและโปรแกรมเมอร์สามารถสร้างและควบคุมระบบย่อยได้ง่ายขึ้น

ความแตกต่างที่สำคัญระหว่าง 4.3 BSD และ AIX:

ต่อไปนี้เป็นข้อมูลสรุปความแตกต่างที่สำคัญระหว่างระบบ AIX และ 4.3 BSD

บน AIX daemons เครื่องข่ายเริ่มต้นขึ้นจากไฟล์ /etc/rc.tcpip ไม่ใช่ไฟล์ /etc/rc.local เซลล์สคริปต์ /etc/rc.tcpip มีการเรียกใช้จากไฟล์ /etc/inittab ไม่ใช่ไฟล์ /etc/rc

หาก System Resource Controller (SRC) กำลังรันอยู่ TCP/IP daemons จะรันภายใต้การควบคุม SRC ถ้าคุณไม่ต้องการให้ TCP/IP daemons รันภายใต้การควบคุม SRC ให้ใช้พารามิเตอร์ `smit setbootup_option` เพื่อเปลี่ยนระบบเป็นคอนพิกูเรชัน BSD-style rc

ฟังก์ชันการจัดการเครือข่ายเหล่านี้ที่มีอยู่บน 4.3 BSD ได้รับการสนับสนุนจาก AIX:

- ความสะดวกในการบันทึก SYSLOG ระดับเคอร์เนล
- สิทธิเข้าถึง UNIX โดเมน ซ็อกเก็ต

หน่วยเก็บข้อมูลคอนฟิกูเรชัน

โดยปกติ 4.3 BSD จัดเก็บข้อมูลคอนฟิกูเรชันไว้ใน ไฟล์ ASCII ส่วนของข้อมูลที่เกี่ยวข้องมีการเก็บไว้บนบรรทัดเดียวกัน และสามารถทำการประมวลผลเร็กคอร์ด (การเรียงลำดับและการค้นหา) บนตัวไฟล์ ASCII เอง เร็กคอร์ดสามารถมีความยาวแตกต่างกันและสิ้นสุดด้วยการป้อนบรรทัด 4.3 BSD นำเสนอเครื่องมือ การแปลงไฟล์ ASCII ขนาดใหญ่บางไฟล์เป็นรูปแบบฐานข้อมูล (dbm) ฟังก์ชันไลบรารีที่เกี่ยวข้องค้นหาคู่ของไฟล์ dbm ถ้ามีอยู่แต่จะค้นหาไฟล์ ASCII ดั้งเดิมถ้าไม่พบไฟล์ dbm

ข้อมูล คอนฟิกูเรชันบางส่วนสำหรับ AIX มีการจัดเก็บไว้ในไฟล์ ASCII แต่มักอยู่ในรูปแบบ stanza Stanza คือชุดของชิ้นส่วนข้อมูลที่เกี่ยวข้องซึ่งจัดเก็บไว้ในกลุ่มของหลายบรรทัด ชิ้นส่วนข้อมูล แต่ละชิ้นมีเลเบลเพื่อให้สามารถเข้าใจเนื้อหาของไฟล์ได้มากขึ้น

AIX ยัง สนับสนุนเวอร์ชัน dbm ของรหัสผ่านและข้อมูลผู้ช่วย ยิ่งไปกว่านั้น ไฟล์ /etc/passwd, /etc/group, และ /etc/inittab เป็นตัวอย่างของไฟล์สำหรับ AIX ซึ่งข้อมูลมีการจัดเก็บไว้ใน รูปแบบดั้งเดิม ไม่ใช่รูปแบบ stanza

ข้อมูล คอนฟิกูเรชันอื่นของ AIX มีการจัดเก็บไว้ในไฟล์ที่เก็บรักษาโดย Object Data Manager (ODM) System Management Interface Tool (SMIT) สามารถจัดการและแสดงข้อมูลในไฟล์ ODM หรือคุณสามารถใช้คำสั่ง ODM โดยตรงเพื่อดูไฟล์เหล่านี้ เมื่อต้องการเคียวรีไฟล์ ODM ให้ใช้คำสั่งต่อไปนี้:

- **odmget**
- **odmshow.**

คำสั่ง ODM ต่อไปนี้เปลี่ยนไฟล์ ODM:

- **odmadd**
- **odmcreate**
- **odmdrop**
- **odmchange**
- **odmdelete.**

ข้อควรสนใจ: การเปลี่ยนไฟล์ ODM อย่างไม่ถูกต้องอาจส่งผลให้ ระบบล้มเหลว และอาจทำให้คุณไม่สามารถรีสตาบ์ระบบได้ ใช้เฉพาะคำสั่ง ODM โดยตรงบนไฟล์ ODM เมื่อคำสั่งเฉพาะภารกิจ อย่างเช่น คำสั่งที่สร้างโดย SMIT ทำงานไม่สำเร็จ

การจัดการคอนฟิกูเรชัน

เมื่อระบบที่รัน AIX เริ่มต้น ขึ้น ชุดของคำสั่งเฉพาะคอนฟิกูเรชันจะถูกเรียกใช้โดย Configuration Manager คำสั่งเฉพาะคอนฟิกูเรชันเหล่านี้เรียกว่า *เมธอด* เมธอดระบุอุปกรณ์ระบบและอัปเดตไฟล์ ODM ที่เหมาะสมในไดเรกทอรี /etc/objrepos

ไฟล์พิเศษ อุปกรณ์ใน /dev โดยตรงไม่มีการติดตั้งไว้ล่วงหน้า ไฟล์พิเศษบางไฟล์ เช่น ไฟล์สำหรับฮาร์ดดิสก์มีการสร้างขึ้นโดยอัตโนมัติ ในระหว่างโปรเซสคอนฟิกูเรชันสตาร์ทอัพ ไฟล์พิเศษอื่นๆ เช่น ไฟล์สำหรับเทอร์มินัล ASCII ต้องถูกสร้างขึ้นโดยผู้ดูแลระบบโดยใช้เมนู SMIT Devices ข้อมูลนี้มีการรักษาไว้ใน ODM สำหรับการใช้งานในภายหลังโดยระบบ

การจัดการดิสก์

ใน AIX ดิสก์ไดร์ฟมีการอ้างอิงเป็น *ฟิสิคัลวอลุ่ม* พาร์ติชันมีการอ้างอิงเป็น *โลจิคัลวอลุ่ม* ใน 4.3 BSD ฟิสิคัลวอลุ่มหนึ่งสามารถมีได้หลายโลจิคัลวอลุ่ม อย่างไรก็ตาม ไม่เหมือนกับ 4.3 BSD คือโลจิคัลวอลุ่มหนึ่งใน AIX สามารถทอดข้ามหลายฟิสิคัล วอลุ่ม เมื่อต้องการทำเช่นนี้ คุณต้องจัดทำหลายฟิสิคัลวอลุ่มใน *กลุ่ม วอลุ่ม* และสร้างโลจิคัลวอลุ่มบนกลุ่มวอลุ่ม

คำสั่ง ใน AIX ที่ใช้สำหรับระบบไฟล์ และการจัดการวอลุ่มรวมถึง:

- **crfs**
- **varyonvg**
- **varyoffvg**
- **lsvg**
- **importvg**
- **exportvg.**

และยังมีคำสั่ง 4.3 BSD ต่อไปนี้:

- **mkfs**
- **fsck**
- **fsdb**
- **mount**
- **umount.**

ความแตกต่างระหว่างคำสั่งเหล่านี้สำหรับ 4.3 BSD และสำหรับ AIX มีการอธิบายไว้ใน “ระบบไฟล์สำหรับผู้จัดการระบบ BSD 4.3” ในหน้า 373

4.3 BSD เก็บรักษารายการของระบบไฟล์ไว้ในไฟล์ /etc/fstab AIX เก็บรักษา stanza ของ แต่ละระบบไฟล์ไว้ในไฟล์ /etc/filesystems

คำสั่ง tn3270

คำสั่ง **tn3270** เป็นลิงก์ไปยังคำสั่ง **telnet** แต่ใช้ไฟล์ /etc/map3270 และค่าตัวแปรสภาวะแวดล้อม **TERM** ปัจจุบัน เพื่อนำเสนอการแม็ปคีย์บอร์ด 3270 ดังนั้น คำสั่ง **tn3270** จึงดำเนินการเหมือนกับเวอร์ชัน BSD ทุกประการ
ถ้าคุณ ต้องการเปลี่ยนลำดับ escape จากค่าดีฟอลต์ที่ใช้โดยคำสั่ง **tn3270**, **telnet**, หรือ **tn** ให้ตั้งค่าตัวแปรสภาวะแวดล้อม **TNESC** ก่อนเริ่มต้นคำสั่งเหล่านี้

คำสั่งใหม่

เพื่อจัดการระบบคอนฟิกูเรชันและการจัดการดิสก์ใหม่ AIX มี คำสั่งประมาณ 150 รายการที่เป็นคำสั่งใหม่สำหรับผู้ดูแลระบบ 4.3 BSD

สตาร์ทอัพ

AIX สนับสนุนการระบุ และคอนฟิกูเรชันของอุปกรณ์ฮาร์ดแวร์ ในลำดับต่อมา สตาร์ทอัพโปรเซสมีความแตกต่างจาก ระบบ 4.3 BSD เป็นอย่างมาก นอกเหนือจากเคอร์เนล รูปภาพของระบบไฟล์บูต และข้อมูลคอนฟิกูเรชัน อุปกรณ์พื้นฐานก่อนหน้าจะถูกโหลดไปยัง RAM ดิสก์ในระยะแรกของสตาร์ทอัพ ข้อมูลคอนฟิกูเรชันที่พอเพียงจะถูกโหลดและตรวจสอบเพื่ออนุญาตการเข้าถึงโลจิคัลวอลุ่ม อุปกรณ์พื้นที่การเลื่อนหน้า มีการระบุสำหรับเคอร์เนล และมีการตรวจสอบระบบไฟล์รากฮาร์ดดิสก์ ในเวลานี้ระบบปฏิบัติการจะเปลี่ยนระบบไฟล์รากจาก RAM ดิสก์เป็นฮาร์ดดิสก์และทำสตาร์ทอัพโปรซีเตอร์ให้สมบูรณ์ รวมถึง การกำหนดคอนฟิกูเรชันอื่น

สิทธิของผู้ใช้

4.3 BSD และเวอร์ชันของระบบปฏิบัติการ AT&T UNIX ก่อนหน้า SVR4 จัดเก็บข้อมูลการพิสูจน์ตัวตนผู้ใช้ทั้งหมด รวมถึง รหัสผ่านที่เข้ารหัส ไว้ในไฟล์ `/etc/passwd` โดยดั้งเดิม ไฟล์ `/etc/passwd` สามารถอ่านได้โดยทุกคน

บนระบบ SVR4 รหัสผ่านที่เข้ารหัสถูกเอาออกจากไฟล์ `/etc/passwd` และจัดเก็บไว้ในไฟล์ `/etc/shadow` เฉพาะผู้ใช้ที่มีสิทธิการใช้งานราก และโปรแกรมที่เชื่อถือได้เท่านั้น (เช่น โปรแกรม `/bin/login`) สามารถอ่านไฟล์ `/etc/shadow` ได้

AIX จัดเก็บรหัสผ่านที่เข้ารหัสไว้ใน ไฟล์ `/etc/security/passwd` ไฟล์อื่นในไดเรกทอรี `/etc/security` คือไฟล์ `user` และ `limits` สามไฟล์ เหล่านี้กำหนดวิธีที่ผู้ใช้ได้รับอนุญาตให้เข้าถึงระบบ (เช่น การใช้คำสั่ง `rlogin` หรือ `telnet`) และขีดจำกัดรีซอร์สของผู้ใช้ (เช่น ขนาดไฟล์และพื้นที่แอดเดรส)

การพิมพ์

คำสั่งการพิมพ์ 4.3 BSD ส่วนใหญ่ได้รับการสนับสนุน โดยมีความแตกต่างเล็กน้อย ความแตกต่างอย่างหนึ่งคือไฟล์ `/etc/qconfig` เป็นไฟล์คอนฟิกูเรชันใน AIX

ระบบการพิมพ์บรรทัดสำหรับ AIX สามารถทำงานร่วมกับระบบการพิมพ์บรรทัด 4.3 BSD ทั้งสำหรับการส่งงานพิมพ์ไปยังระบบ 4.3 BSD และสำหรับการพิมพ์งานที่ส่งมาจากระบบ 4.3 BSD

เชลล์

AIX สนับสนุน เชลล์เป้าหมาย, C เชลล์และ Korn เชลล์ ชื่อพาธแบบเต็มสำหรับโปรแกรม เชลล์เป้าหมายคือ `/bin/bsh` ไฟล์ `/bin/sh` คือฮาร์ดลิงก์ไปยังไฟล์ `/bin/ksh` ไฟล์นี้สามารถเปลี่ยนได้โดยผู้ดูแลระบบ

AIX ไม่สนับสนุน `setuid` หรือ `setgid` สำหรับ เชลล์สคริปต์ในเชลล์ใดๆ

หมายเหตุ:

1. AIX ไม่มีเชลล์สคริปต์ที่พึ่งพา `/bin/sh` ใดๆก็ตาม เชลล์สคริปต์จำนวนมากจากระบบอื่นพึ่งพา `/bin/sh` ในการเป็นเชลล์เป้าหมาย
2. แม้ว่าเชลล์เป้าหมายและ Korn เชลล์คล้ายกัน แต่ Korn เชลล์ไม่ใช่ ซูเปอร์เซตที่สมบูรณ์ของเชลล์เป้าหมาย

สิ่งอ้างอิงที่เกี่ยวข้อง:

“คำสั่งสำหรับผู้ดูแลระบบของผู้จัดการระบบ BSD 4.3” ในหน้า 369

รายการต่อไปนี้แสดงคำสั่งที่ใช้เฉพาะสำหรับการจัดการ สภาวะแวดล้อมของ AIX

ตารางการเปรียบเทียบไฟล์สำหรับ 4.3 BSD, SVR4, และ AIX:

ตารางต่อไปนี้เปรียบเทียบชื่อไฟล์และฟังก์ชันระหว่าง 4.3 BSD, SVR4, และ AIX

ตารางที่ 6 1. File Comparison Table

ไฟล์ 4.3 BSD	ไฟล์ SVR4	ไฟล์สำหรับ AIX	ฐานข้อมูล	ชนิด (odm/dbm)
L-Devices	อุปกรณ์	อุปกรณ์	ไม่	
L-dialcodes	Dialcodes	Dialcodes	ไม่	
L.cmds	สิทธิ์อนุญาต	สิทธิ์อนุญาต	ไม่	
L.sys	ระบบ	ระบบ	ไม่	
USERFILE	สิทธิ์อนุญาต	สิทธิ์อนุญาต	ไม่	
นามแฝง	เมล/ไฟล์ชื่อ	นามแฝง	aliasesDB/DB	dbm
fstab	vfstab	filesystems	ไม่	
ftputers	ftputers	ftputers	ไม่	
gettytab		N/A		
กลุ่ม	กลุ่ม	กลุ่ม	ไม่	
โฮสต์	โฮสต์	โฮสต์	ไม่	
hosts.equiv	hosts.equiv	hosts.equiv	ไม่	
inetd.conf	inetd.conf	inetd.conf	ไม่	
map3270	N/A	map3270	ไม่	
motd	motd	motd	ไม่	
mtab	mnttab	N/A	ไม่	
named.boot	named.boot	named.boot	ไม่	
named.ca		named.ca	ไม่	
named.hosts		named.data (โปรดดูที่หมายเหตุ)	ไม่	
named.local		named.local	ไม่	
named.pid	named.pid	named.pid	ไม่	
named.rev		named.rev	ไม่	
เครือข่าย	เครือข่าย	เครือข่าย	ไม่	
passwd	passwd	passwd	ไม่	
printcap	qconfig	qconfig		
โปรโตคอล		โปรโตคอล	ไม่	
รีโมต	รีโมต	รีโมต	ไม่	
resolv.conf	resolv.conf	resolv.conf	ไม่	
sendmail.cf	sendmail.cf	sendmail.cf	sendmail.cfDB	ไม่ทั้งสองอย่าง

ตารางที่ 61. File Comparison Table (ต่อ)

ไฟล์ 4.3 BSD	ไฟล์ SVR4	ไฟล์สำหรับ AIX	ฐานข้อมูล	ชนิด (odm/dbm)
เชอร์วิส		เชอร์วิส	ไม่	
เชลล์	เชลล์	N/A		
stab		N/A		
syslog.conf		syslog.conf	ไม่	
syslog.pid		syslog.pid	ไม่	
termcap	terminfo	terminfo		
ttys	ttys	N/A	yes	odm
ชนิด		N/A		
utmp	utmp	utmp		
vfont		N/A		
vgrindefs		vgrindefs		
wtmp	wtmp	wtmp		

หมายเหตุ: ชื่อไฟล์ named.ca, named.hosts, named.local, และ named.rev เป็นชื่อที่ผู้ใช้สามารถกำหนดได้ในไฟล์ named.boot อย่างไรก็ตาม ชื่อเหล่านี้เป็นชื่อที่ใช้สำหรับไฟล์เหล่านี้ในเอกสารคู่มือสำหรับ AIX

การแก้ไขชื่อและแอดเดรส:

รoutines ย่อย `gethostbyname` และ `gethostbyaddr` ในไลบรารี `libc` ให้การสนับสนุน Domain Name Service, Network Information Services (NIS ซึ่งเรียกอย่างเป็นทางการว่า Yellow Pages) และฐานข้อมูล `/etc/hosts`

ถ้ามีไฟล์ `/etc/resolv.conf` เนมเซิร์ฟเวอร์ จะถูกตรวจสอบก่อนเสมอ ถ้าชื่อไม่ได้รับการแก้ไขและ NIS กำลังรัน NIS จะถูกตรวจสอบ หาก NIS ไม่ได้รันอยู่ไฟล์ `/etc/hosts` จะถูกตรวจสอบ

เอกสารคู่มือทางออนไลน์และคำสั่ง `man` สำหรับผู้จัดการระบบ BSD 4.3:

AIX สนับสนุนคำสั่ง `man-k`, `apropos`, และ `whatis` แต่ฐานข้อมูลที่ใช้โดยคำสั่งเหล่านี้ต้อง ถูกสร้างขึ้นก่อนด้วยคำสั่ง `catman-w`

คำสั่ง `man` ค้นหาหน้าข้อความ flat ในไฟล์ `/usr/man/cat?` ก่อน ถัดไป ค้นหาที่จัดรูปแบบ `nroff` ในไฟล์ `/usr/man/man?` สามารถเพิ่มแมนเพจใหม่ลงในข้อความ flat หรือรูปแบบ `nroff`

หมายเหตุ:

- หน้าข้อความคำสั่ง `man` ไม่มีการจัดเตรียมให้พร้อมทั้ง ระบบ คำสั่ง `catman` สร้างฐานข้อมูลจากหน้าข้อความ เหล่านี้ หน้าเหล่านี้สามารถเป็นหน้าข้อความ flat ที่จัดเก็บไว้ในไฟล์ `/usr/man/cat?` หรือหน้าที่จัดรูปแบบ `nroff` ซึ่งจัดเก็บไว้ในไฟล์ `/usr/man/man?`
- ต้องติดตั้งโปรแกรมที่มีไลเซนส์การจัดรูปแบบข้อมูลสำหรับคำสั่ง `nroff` เพื่อให้มีอยู่สำหรับคำสั่ง `man` เพื่ออ่านแมนเพจที่จัดรูปแบบ `nroff`

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งเหล่านี้ให้ดูที่ `man`, `apropos`, `whatis`, และ `catman`

NFS และ NIS (เรียกอย่างเป็นทางการว่า Yellow Pages) สำหรับผู้จัดการระบบ BSD 4.3:

ข้อมูลต่อไปนี้อธิบาย NFS และ NIS สำหรับผู้จัดการระบบ BSD 4.3

Network File System (NFS) และ Network Information Services (NIS) daemons มีการเริ่มต้นจากไฟล์ `/etc/rc.nfs` อย่างไรก็ตาม ก่อนสามารถเริ่มต้น NFS และ NIS daemons ได้ ต้องเริ่มต้น `portmap` daemon ในไฟล์ `/etc/rc.tcpip` โดยค่าตีพอลต์ไฟล์ `/etc/rc.nfs` ไม่มีการเรียกใช้โดยไฟล์ `/etc/inittab` หากคุณเพิ่มบรรทัดลงในไฟล์ `/etc/inittab` เพื่อเรียกใช้สคริปต์ `/etc/rc.nfs` ไฟล์ควรถูกเรียกใช้หลังจากสคริปต์ `/etc/rc.tcpip`

ถ้า NIS ใช้งานอยู่ให้รวมรายการรากก่อนหน้ารายการ `++:` (เครื่องหมาย บวก, จุดคู่, จุดคู่) ในไฟล์ `/etc/passwd` และ รายการระบบก่อนหน้ารายการ `++:` ในไฟล์ `/etc/group` การทำเช่นนี้ช่วยให้ผู้ดูแลระบบสามารถล็อกอินเป็นรากและทำการเปลี่ยนแปลงได้ ถ้าระบบไม่สามารถสื่อสารกับเซิร์ฟเวอร์ NIS

NFS สามารถกำหนดคอนฟิกโดยใช้พาด่วน SMIT, `smit nfs` เมนู The และ SMIT อ้างอิงถึง NIS (เดิมเรียกว่า Yellow Pages) ว่าเป็น NIS คำสั่ง NFS และ NIS จำนวนมากสามารถพบได้ในไดเรกทอรี `/etc` และ `/usr/etc`

บางสภาวะแวดล้อม NFS ใช้คำสั่ง `arch` เพื่อระบุตระกูลเครื่องและชนิดของเครื่อง ตัวอย่างเช่น ถ้าคุณ กำลังใช้ IBM® RS/6000 ให้ระบุตัวระบุ `power` สำหรับตระกูล (CPU) และตัวระบุ `ibm6000` สำหรับชนิด (เครื่อง)

รหัสผ่านผู้ใช้สำหรับผู้จัดการระบบ BSD 4.3:

เมื่อคุณใช้คำสั่ง `/bin/passwd` สำหรับ AIX ในฐานะผู้ใช้งาน คุณจะได้รับการพร้อมท์ให้ป้อนรหัสผ่านผู้ใช้รากปัจจุบัน

ตัวอย่างของการใช้คำสั่ง `/bin/passwd` เป็นดังนี้:

```
# passwd cslater
Changing password for "cslater"
Enter root's Password or
cslater's Old password:
cslater's New password:
Re-enter cslater's
new password:
#
```

เวอร์ชัน 4.3 BSD ไม่ได้พร้อมท์ให้ป้อนรหัสผ่านผู้ใช้ราก ปัจจุบัน ตัวอย่างของเวอร์ชัน 4.3 BSD เป็นดังนี้:

```
# passwd cslater
New password:
Retype new password:
#
```

การจัดการ BSD

มีหลายคำสั่งสำหรับ BSD ซึ่งคุณสามารถใช้เพื่อประเมินประสิทธิภาพ พิมพ์ และจัดการกับระบบของคุณ

การบัญชีสำหรับผู้จัดการระบบ BSD 4.3:

ไฟล์การบัญชีในไดเรกทอรี `/usr/lib/acct` และเครื่องมือการรายงานกิจกรรมระบบในไดเรกทอรี `/usr/lib/sa` สำหรับ AIX เหมือนกับที่มีอยู่ใน AT&T System V Release 4 (SVR4) ซึ่งมีส่วนเพิ่มของยูทิลิตี้การบัญชี 4.3 BSD

คำสั่งการบัญชีจำนวนมากอยู่ในไดเรกทอรี /usr/lib/acct เมื่อต้องการเริ่มต้นบัญชีระบบให้ใช้คำสั่ง /usr/lib/acct/startup หากไม่ได้เริ่มต้นการบัญชี คำสั่ง เช่น lastcomm(1) ไม่สามารถ ส่งคืนข้อมูลได้

AIX นำเสนอ ยูทิลิตี้การบัญชี 4.3 BSD เหล่านี้:

ไอเท็ม	คำอธิบาย
last(1)	บ่งชี้ล็อกอินล่าสุดของผู้ใช้และเทอร์มินัล
lastcomm(1)	แสดงคำสั่งล่าสุดที่ดำเนินการในลำดับย้อนกลับ
acct(3)	เปิดใช้งานและปิดใช้งานบัญชีกระบวนการ
ac(8)	การบัญชีล็อกอิน
accton(8)	เปิดหรือปิดบัญชีระบบ
sa(8)	รักษาไฟล์บัญชีระบบโดยทั่วไป

AIX ยัง นำเสนอคำสั่งการบัญชี System V Interface Definition (SVID) Issue II และฟังก์ชันไลบรารีเหล่านี้ด้วย:

ไอเท็ม	คำอธิบาย
acctcms(1)	จัดทำข้อมูลสรุปการใช้คำสั่งจากเรกคอร์ดการบัญชี
acctcom(1)	แสดงข้อมูลสรุปเรกคอร์ดบัญชีกระบวนการที่เลือก
acctcon1(1)	แปลงเรกคอร์ดล็อกอิน/ล็อกออฟเป็นเรกคอร์ดเซสชัน
acctcon2(1)	แปลงเรกคอร์ดล็อกอิน/ล็อกออฟเป็นเรกคอร์ดการบัญชีทั้งหมด
acctdisk(1)	สร้างเรกคอร์ดการบัญชีทั้งหมดจากคำสั่งเอาต์พุต diskusg(1)
acctmerg(1)	ผสานไฟล์การบัญชีทั้งหมดเข้าในไฟล์ระหว่างกลาง
accton(1)	เปิดการบัญชี
acctprc1(1)	ประมวลผลข้อมูลการบัญชีจากคำสั่ง acct(3)
acctprc2(1)	ประมวลผลเอาต์พุตของคำสั่ง acctprc1(1) เป็นเรกคอร์ด การบัญชีทั้งหมด
acctwtmp(1)	จัดการเรกคอร์ดการบัญชีเวลาเชื่อมต่อ
chargefee(1)	เรียกเก็บไปยังชื่อล็อกอิน
ckpacct(1)	ตรวจสอบขนาดของไฟล์ /usr/adm/pacct
diskusg(1)	สร้างข้อมูลการบัญชีดีสก์
dodisk(1)	ทำการบัญชีดีสก์
fwtmp(1)	แปลงเรกคอร์ดไบนารี (ไฟล์ wtmp) เป็นรูปแบบ ASCII

หมายเหตุ: ไฟล์ wtmp อยู่ในไดเรกทอรี /var/adm

lastlogin(1)	อัปเดตวันที่ล่าสุดซึ่งแต่ละบุคคลล็อกอิน
monacct(1)	สร้างไฟล์ข้อมูลสรุปรายเดือน
prctmp(1)	พิมพ์ไฟล์เรกคอร์ดเซสชันที่จัดทำโดยคำสั่ง acctcon1(1)
prdaily(1)	จัดรูปแบบรายงานข้อมูลการบัญชีของเมื่อวานนี้
prtacct(1)	จัดรูปแบบและพิมพ์ไฟล์การบัญชีทั้งหมดใดๆ
runacct(1)	รันการบัญชีรายวัน
shutacct(1)	เรียกโดยการปิดระบบเพื่อหยุดการบัญชีและบันทึกเหตุผล
startup(1)	เรียกโดยการเริ่มต้นระบบเพื่อเริ่มต้นการบัญชี
turnacct(1)	เปิดหรือปิดบัญชีกระบวนการ
wtmpfix(1)	แก้ไขการประทับเวลา/วันที่ในไฟล์โดยใช้รูปแบบ wtmp

สำเนาสำรองสำหรับผู้จัดการระบบ BSD 4.3:

ผู้จัดการระบบ BSD 4.3 สามารถทำสำเนาสำรองข้อมูลได้

คำสั่ง tar และ cpio สามารถย้ายข้อมูลระหว่าง ระบบได้ คำสั่ง tar ของ AIX เข้ากันได้ไม่เต็มที่กับคำสั่ง 4.3 BSD tar คำสั่ง tar ของ AIX ต้องการแฟล็ก -B (การบล็อกอินพุต) ถ้าคำสั่งอ่านจากไฟฟ์ คำสั่ง AT&T cpio เข้ากันได้กับเวอร์ชันนี้

AIX สามารถ อ่านและเขียนในรูปแบบคำสั่ง dump และ restore ตัวอย่างเช่น คำสั่ง backup ของ AIX ที่มี ไวยากรณ์:

```
backup -Ouf Device Filesystemname
```

เหมือนกับ คำสั่ง 4.3 BSD **dump** ที่มีไวยากรณ์:

```
dump 0uf Device Filesystemname
```

ในลักษณะคล้ายกัน คำสั่ง **restore** ของ AIX ที่มีไวยากรณ์:

```
restore -mivf Device
```

เหมือนกับ คำสั่ง 4.3 BSD **restore** ที่มีไวยากรณ์:

```
restore ivf Device
```

AIX ยังมีคำสั่ง 4.3 BSD **rdump** และ **rrestore** ด้วย ความแตกต่างเพียงอย่างเดียว ระหว่างสองเวอร์ชันคือ สำหรับ AIX แต่ละอาร์กิวเมนต์ต้องมี - (เครื่องหมายขีด) นำหน้า ตัวอย่างเช่น คำสั่ง ต่อไปนี้:

```
rdump -0 -f orca:/dev/rmt0 /dev/hd2
```

เทียบ เทากับคำสั่ง 4.3 BSD:

```
rdump 0f orca:/dev/rmt0 /dev/hd2
```

คำสั่ง **backup** ของ AIX ที่มี ไวยากรณ์ต่อไปนี้:

```
backup -0f /dev/rmt0 /dev/hd2
```

เทียบ เทากับคำสั่ง 4.3 BSD **dump** ที่มีไวยากรณ์นี้:

```
dump 0f /dev/rmt0 /dev/hd2
```

การสนับสนุนเทป SCSI ที่ไม่ใช่ของ IBM

AIX ไม่สนับสนุน เทปไดรฟ์ SCSI ที่ไม่ใช่ของ IBM โดยตรง อย่างไรก็ตาม คุณสามารถเพิ่มส่วนหัวและอินเตอร์เฟซของคุณเองที่ใช้ไดรเวอร์ IBM SCSI ได้

หลักการที่เกี่ยวข้อง:

“สำเนาสำรองระบบ” ในหน้า 23

หลังจากใช้ระบบของคุณแล้ว ข้อควรพิจารณาถัดไปของคุณควรจะเป็น การทำสำเนาสำรองระบบไฟล์ ไตรีกทอรี และไฟล์ หาก你做สำเนาสำรองระบบไฟล์ คุณสามารถเรียกคืนไฟล์หรือระบบไฟล์ในกรณีที่เกิดข้อผิดพลาดได้ มีเมธอดที่แตกต่างกันสำหรับการทำสำเนาสำรองข้อมูล

ข้อมูลที่เกี่ยวข้อง:

การเพิ่มอุปกรณ์ที่ไม่สนับสนุนลงในระบบ

สตาร์ทอัพสำหรับผู้จัดการระบบ BSD 4.3:

ข้อมูลต่อไปนี้อธิบายสตาร์ทอัพระบบ AIX สำหรับผู้จัดการระบบ BSD 4.3

บนระบบ 4.3 BSD โปรแกรม **init** คือขั้นตอนสุดท้ายในสตาร์ทอัพโพธิ์เซเตอร์ บทบาทหลักของโปรแกรม **init** คือการสร้างโปรเซสสำหรับแต่ละเทอร์มินัลพอร์ตที่มีอยู่ เทอร์มินัลพอร์ตที่มีอยู่ สามารถพบได้โดยการอ่านไฟล์ `/etc/ttys`

บน System V โปรแกรม **init** มีการ เริ่มต้นที่การเริ่มต้นระบบ โปรเซส **init** เริ่มต้น โปรเซสตามรายการในไฟล์ `/etc/inittab`

AIX ปฏิบัติตาม โพรซีเจอร์การเริ่มต้น System V คุณสามารถแก้ไขไฟล์ /etc/inittab โดยการแก้ไขไฟล์โดยตรง โดยใช้คำสั่ง **telinit** หรือโดยใช้คำสั่ง ต่อไปนี้:

ไอเท็ม	คำอธิบาย
chitab(1)	เปลี่ยนเร็กคอร์ดในไฟล์ /etc/inittab
lsitab(1)	แสดงรายการเร็กคอร์ดในไฟล์ /etc/inittab
mkitab(1)	จัดทำเร็กคอร์ดในไฟล์ /etc/inittab
rmitab(1)	ลบเร็กคอร์ดในไฟล์ /etc/inittab

การเปลี่ยนแปลงที่ทำในไฟล์ /etc/inittab มีผลบังคับใช้เมื่อรีบูตระบบครั้งถัดไป หรือเมื่อรันคำสั่ง **telinit q**

การค้นหาและการตรวจสอบไฟล์สำหรับผู้จัดการระบบ BSD 4.3:

ข้อมูลต่อไปนี้เป็นรายการของคำสั่งไฟล์ BSD ที่ AIX สนับสนุน

AIX สนับสนุน คำสั่งไฟล์ 4.3 BSD ต่อไปนี้:

- **which**
- **whereis**
- **what**
- **file.**

AIX ไม่สนับสนุนไวยากรณ์ 4.3 BSD ค้นหาตัวน ของคำสั่ง **find** ณ เวลานี้ ไม่มีฟังก์ชันการแทนที่ เซลล์สคริปต์ **ffind** ต่อไปนี้สามารถใช้เพื่อจำลองการทำงาน:

```
#!/bin/bsh
PATH=/bin
for dir in /bin /etc /lib /usr
do
find $dir -print | egrep $1
done
```

ไวยากรณ์สำหรับสคริปต์ **ffind** คือ:

```
ffind Filename
```

พื้นที่การเลื่อนหน้าสำหรับผู้จัดการระบบ BSD 4.3:

คำสั่งต่อไปนี้จะช่วยในการจัดการพื้นที่การเลื่อนหน้า (เรียกอีกอย่างว่า พื้นที่สลับ)

ไอเท็ม	คำอธิบาย
chps(1)	เปลี่ยนแอตทริบิวต์ของพื้นที่การเลื่อนหน้า
lsps(1)	แสดงรายการแอตทริบิวต์ของพื้นที่การเลื่อนหน้า
mkps(1)	เพิ่มพื้นที่การเลื่อนหน้าเพิ่มเติมลงในระบบ
rmpps(1)	ลบพื้นที่การเลื่อนหน้าออกจากระบบ
swapoff(1)	ยกเลิกการใช้พื้นที่การเลื่อนหน้าตั้งแต่หนึ่งพื้นที่ขึ้นไป
swapon(1)	ระบุอุปกรณ์เพิ่มเติมสำหรับการเลื่อนหน้าและการสลับ

ถ้าต้องการพื้นที่การเลื่อนหน้าจำนวนมาก ให้อ้างอิงถึงคู่มือการเลื่อนหน้าหนึ่งรายการสำหรับแต่ละฮาร์ดดิสก์ ซึ่งช่วยให้สามารถจัดตารางเวลาการเลื่อนหน้า บนหลายดิสก์ได้

การเปลี่ยนดีฟอลต์สตาร์ทอัพเพื่ออนุญาตคอนฟิกูเรชัน 4.3 BSD ASCII:

คุณสามารถจัดการอินเตอร์เฟซเครือข่ายสำหรับ AIX โดยใช้ไฟล์ SMIT และ ODM หรือโดยใช้ไฟล์คอนฟิกูเรชัน 4.3 BSD ASCII

เมื่อต้องการจัดการอินเตอร์เฟซเครือข่ายโดยใช้ไฟล์คอนฟิกูเรชัน 4.3 BSD ASCII ให้ uncomment คำสั่งในไฟล์ /etc/rc.net ข้างใต้หัวข้อ:

```
# Part II - Traditional  
Configuration
```

จากนั้น ถ้าคุณต้องการคอนฟิกูเรชัน ไฟล์ flat และการสนับสนุน SRC ให้แก้ไขไฟล์ /etc/rc.net และ uncomment คำสั่ง **hostname**, **ifconfig**, และ **route** ด้วยพารามิเตอร์ที่เหมาะสม

ถ้าคุณต้องการคอนฟิกูเรชันไฟล์ flat โดยไม่มีการสนับสนุน SRC ให้ใช้พาด่วน **smit setbootup_option** เพื่อเปลี่ยนระบบ เป็นคอนฟิกูเรชัน BSD-style rc อีพชัณนี้กำหนดคอนฟิกูเรชันเพื่อใช้ไฟล์ /etc/rc.bsdnet เมื่อสตาร์ทอัพ คุณยังต้องแก้ไขไฟล์ /etc/rc.bsdnet และ uncomment คำสั่ง **hostname**, **ifconfig**, และ **route** ด้วยพารามิเตอร์ที่เหมาะสมด้วย

อีพชัณเพิ่มเติมสำหรับคำสั่ง ifconfig และ netstat:

ข้อมูลต่อไปนี้เป็นรายการของอีพชัณเพิ่มเติมสำหรับ คำสั่ง **ifconfig** และ **netstat**

คำสั่ง **ifconfig** สำหรับ AIX มีอีพชัณเพิ่มเติมต่อไปนี้:

mtu ตัวแปร **mtu** ระบุหน่วยการส่งผ่านค่าสูงสุด (MTU) ที่ใช้บนเครือข่ายโลคัล (และโลคัล subnets) และ MTU ที่ใช้สำหรับเครือข่ายรีโมต เมื่อต้องการเพิ่มความเข้ากันได้กับอีเทอร์เน็ตและเครือข่ายอื่นให้มากที่สุด ให้ตั้งค่าพอลด์ **mtu** ของทั้ง Token-Ring และอีเทอร์เน็ต เป็น 1500

allcast แฟล็ก **allcast** ตั้งค่ากลยุทธ์การกระจาย Token-Ring การตั้งค่า แฟล็ก **allcast** ช่วยเพิ่มประสิทธิภาพการเชื่อมต่อผ่านทางบริดจ์ Token-Ring การล้างแฟล็ก **allcast** (โดยการระบุ -allcast) ลดการจราจรส่วนเกินบนวงแหวนให้เหลือน้อยที่สุด

คำสั่ง **netstat** สำหรับ AIX มีแฟล็ก **-v** คำสั่ง **netstat -v** พิมพ์สถิติไดเรกเตอร์ เช่น จำนวนไบต์ที่ส่งผ่าน จำนวนข้อผิดพลาด การส่งผ่าน จำนวนไบต์ที่ได้รับ และจำนวนข้อผิดพลาดการได้รับ สำหรับข้อมูลเพิ่มเติม เกี่ยวกับคำสั่ง **ifconfig** และ **netstat** ให้ดูที่ **ifconfig** และ **netstat**

คำสั่งการจัดการเครือข่ายเพิ่มเติม:

คำสั่งเพิ่มเติมต่อไปนี้ได้รับการสนับสนุนบน AIX

ไอเท็ม securetcpip	คำอธิบาย เซลล์สคริปต์ securetcpip เปิดใช้งานโหมดการเข้าถึงที่ควบคุม ซึ่งนำเสนอความปลอดภัยเครือข่ายที่พัฒนาขึ้น โหมดนี้ไม่อนุญาตให้รันโปรแกรม TCP/IP ที่ไม่ปลอดภัยหลายโปรแกรม เช่น โปรแกรม tftp , rcp , rlogin , และ rsh และยังจำกัดการใช้ไฟล์ .netrc ด้วย
gated no	คำสั่ง gated นำเสนอการสนับสนุน MIB สำหรับ SNMP คำสั่ง no ตั้งค่า อ็อพชันเครือข่ายที่รวมถึง: dogticks ตั้งค่าความละเอียดของตัวจับเวลาสำหรับรูทีน ifwatchdog subnetsarelocal กำหนดว่าแพ็กเก็ตแอดเดรสอยู่บนเครือข่ายโลคัลหรือไม่ ipsendredirects ระบุว่าเคอร์เนลควรส่งสัญญาณเปลี่ยนทิศทางหรือไม่ ipforwarding ระบุว่าเคอร์เนลควรส่งต่อแพ็กเก็ตหรือไม่ tcp_ttl ระบุ time-to-live สำหรับโปรโตคอล Transmission Control Protocol (TCP) udp_ttl ระบุ time-to-live สำหรับโปรโตคอล User Datagram Protocol (UDP) maxttl ระบุ time-to-live สำหรับโปรโตคอล Routing Information Protocol (RIP) ipfragttl ระบุ time-to-live สำหรับแฟร็กเมนต์ Internet Protocol (IP) lowclust ระบุลายนํ้าตาสำหรับพูล mbuf คลัสเตอร์ lowmbuf ระบุลายนํ้าตาสำหรับพูล mbuf thewall ระบุจำนวนหน่วยความจำสูงสุดที่จัดสรรไปยังพูล mbuf และ mbuf คลัสเตอร์ arpt_killc ระบุเวลาในหน่วยนาฬิกาที่ก่อนรายการ Address Resolution Protocol (ARP) ที่สมบูรณ์และไม่ได้ใช้งานอยู่ จะถูกลบออก
iptrace ipreport	คำสั่ง iptrace นำเสนอการติดตามแพ็กเก็ตระดับอินเทอร์เฟซสำหรับอินเทอร์เน็ทโปรโตคอล คำสั่ง ipreport จัดรูปแบบการติดตามในรูปแบบที่มนุษย์อ่านได้ ตัวอย่างของการใช้คำสั่งนี้เป็นดังต่อไปนี้: <pre>iptrace -i en0 /tmp/iptrace.log # kill iptrace daemon kill `ps ax grep iptrace awk '{ print \$1 }'` ipreport /tmp/iptrace.log more</pre>

การอิมพอร์ตไฟล์รหัสผ่าน BSD 4.3:

คุณสามารถอิมพอร์ตไฟล์รหัสผ่าน BSD 4.3 เข้าใน AIX

เมื่อต้องการอิมพอร์ตไฟล์รหัสผ่าน BSD 4.3 ให้ปฏิบัติตามขั้นตอนต่อไปนี้:

1. คัดลอกไฟล์รหัสผ่าน BSD 4.3 ไปยังไฟล์ **/etc/passwd** และป้อน:

```
pwdck -y ALL
```
2. อัปเดตไฟล์ **/etc/security/limits** ด้วย **null stanza** สำหรับผู้ใช้ใหม่ใดๆ คำสั่ง **usrck** ทำเช่นนี้ แต่การใช้คำสั่ง **usrck** อาจส่งผลให้เกิดปัญหาได้ ยกเว้นว่ามี การอิมพอร์ตไฟล์ **/etc/group** พร้อมกับไฟล์ **/etc/passwd** สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง **usrck** ให้ดูที่ **usrck**
ข้อควรสนใจ: ถ้าไฟล์ **/etc/security/limits** ถูกแก้ไข สแต็กต้องไม่เกินกว่า 65,536 ไบต์ ถ้าเกิน การรันคำสั่ง **usrck** อาจส่งผลให้เกิดปัญหาได้ ให้เปลี่ยนขนาดสแต็กเป็น 65,536 และรันคำสั่ง **usrck** อีกครั้ง

3. รันคำสั่ง `grpck` และ `usrck` เพื่อตรวจสอบแอ็ททริบิวต์กลุ่มและผู้ใช้

การแก้ไขไฟล์รหัสผ่านสำหรับผู้จัดการระบบ BSD 4.3:

ข้อมูลต่อไปนี้อธิบายวิธีการเปลี่ยนรายการในไฟล์รหัสผ่านและวิธีการจัดการรหัสผ่านบน AIX ในลักษณะ BSD 4.3

ใน AIX มีการจัดเตรียมคำสั่ง `lsuser`, `mkuser`, `chuser`, และ `rmuser` สำหรับการจัดการกับรหัสผ่าน คำสั่งเหล่านี้ทั้งหมดสามารถใช้ได้โดยการรัน SMIT อย่างไรก็ตาม คำสั่งทั้งหมดเหล่านี้จัดการกับผู้ใช้เพียงหนึ่งรายเท่านั้นในแต่ละครั้ง

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่งเหล่านี้ ให้ดูที่ `lsuser`, `mkuser`, `chuser`, และ `rmuser`

หมายเหตุ: การใช้โปรแกรมแก้ไข เพื่อเปลี่ยนรายการชื่อผู้ใช้หลายรายการในเวลาเดียวกันจำเป็นต้องแก้ไขหลายไฟล์พร้อมกัน เนื่องจากรหัสผ่านมีการจัดเก็บไว้ในไฟล์ `/etc/security/passwd` ข้อมูลการพิสูจน์ตัวตนมีการจัดเก็บไว้ในไฟล์ `/etc/security/user` และข้อมูลผู้ใช้ที่เหลือมีการจัดเก็บไว้ในไฟล์ `/etc/passwd`

AIX ไม่สนับสนุนคำสั่ง `vipw` แต่สนับสนุนคำสั่ง `mkpasswd` อย่างไรก็ตาม คุณยังคงสามารถจัดการกับรหัสผ่านใน AIX ในลักษณะ BSD 4.3 ได้ใช้โปรแกรมต่อไปนี้:

1. วางไฟล์รหัสผ่าน BSD 4.3 ไว้ในไฟล์ `/etc/shadow`

2. เปลี่ยนสิทธิ์อนุญาตในไฟล์โดยการป้อน:

```
chmod 000 /etc/shadow
```

3. วางเชลล์สคริปต์ `vipw` ต่อไปนี้ในไดเรกทอรี `/etc`:

```
-----  
----  
#!/bin/bash  
#  
# vipw. Uses pwdck for now. May use usrck someday  
#  
PATH=/bin:/usr/bin:/etc:/usr/ucb # Add to this if your editor is  
# some place else  
if [ -f /etc/ptmp ] ; then  
    echo "/etc/ptmp exists. Is someone else using vipw?"  
    exit 1  
fi  
if [ ! -f `which "$EDITOR" | awk '{ print $1 }'` ] ; then  
    EDITOR=vi  
fi  
cp /etc/shadow /etc/ptmp  
if (cmp /etc/shadow /etc/ptmp) ; then  
    $EDITOR /etc/ptmp  
else  
    echo cannot copy shadow to ptmp  
    exit 1  
fi  
if (egrep "^root:" /etc/ptmp >/dev/null) ; then  
    cp /etc/ptmp /etc/shadow ; cp /etc/ptmp /etc/passwd  
    chmod 000 /etc/passwd /etc/shadow  
    pwdck -y ALL 2>1 >/dev/null # return code 114 may change  
    rc=$?  
    if [ $rc -eq 114 ] ; then  
        chmod 644 /etc/passwd
```

```

        rm -f /etc/passwd.dir /etc/passwd.pag
mkpasswd /etc/passwd
        # update /etc/security/limits, or ftp
        # will fail
    else
        pwdck -y ALL
    fi
else
    echo bad entry for root in ptmp
fi
rm /etc/ptmp
-----

```

4. หากคุณใช้เชลล์สคริปต์ **vipw** หรือคำสั่ง **mkpasswd** ควรตระหนักว่า SMIT และคำสั่ง **mkuser**, **chuser**, และ **rmuser** ไม่ต้องใช้คำสั่ง **mkpasswd** คุณต้องรัน:

```
mkpasswd /etc/passwd
```

เพื่ออัปเดตไฟล์ **/etc/passwd.dir** และ **/etc/passwd.pag**

ข้อควรสนใจ: การเริ่มต้นตัวแปร **IFS** และ คำสั่ง **trap** ช่วยเสริมเมธอดทั่วไปบางวิธีซึ่งใช้เพื่ออุดช่องโหว่ด้านความปลอดภัยที่ได้รับมาจากคุณลักษณะ **setuid** อย่างไรก็ตาม เชลล์สคริปต์ **vipw** และ **passwd** มีไว้สำหรับสภาวะแวดล้อมที่ค่อนข้างเปิดซึ่งความเข้ากันได้เป็นข้อควรพิจารณาที่สำคัญ ถ้าคุณต้องการสภาวะแวดล้อมที่ปลอดภัยมากขึ้น ให้ใช้เฉพาะคำสั่งมาตรฐานสำหรับ AIX

5. วางเชลล์สคริปต์ **passwd** ต่อไปนี้ไว้ในไดเรกทอรี **/usr/ucb**:

```

-----
#!/bin/ksh
#
# matches changes to /etc/security/passwd file with changes to
#/etc/shadow
#
IFS=" "
PATH=/bin
trap "exit 2" 1 2 3 4 5 6 7 8 10 12 13 14 15 16 17 18 21 22 \
    23 24 25 27 28 29 30 31 32 33 34 35 36 60 61 62
if [ -n "$1" ]; then
    USERNAME=$1
else
    USERNAME=$LOGNAME
fi
if [ -f /etc/ptmp ]; then
    echo password file busy
    exit 1
fi
    trap "rm /etc/ptmp; exit 3" 1 2 3 4 5 6 7 8 10 12 13 \
        14 15 16 17 18 21 22 23 24 25 27 28 29 30 31 \
        32 33 34 35 36 60 61 62
if (cp /etc/security/passwd /etc/ptmp) ; then
    chmod 000 /etc/ptmp else
    rm -f /etc/ptmp exit 1
fi
if ( /bin/passwd $USERNAME ) ; then
    PW=`awk ' BEGIN { RS = " " }'

```

```

        $1 == user { print $4 } ' user="$USERNAME:" \
/etc/security/passwd `
else
    rm -f /etc/ptmp
    exit 1
fi
rm -f /etc/ptmp
awk -F: ' $1 == user { print $1:"pw":$3 ":"$4":"$5":"$6":"$7 }
        $1 != user { print $0 }' user="$USERNAME" pw="$PW" \
        /etc/shadow > /etc/ptmp
chmod 000 /etc/ptmp
mv -f /etc/ptmp /etc/shadow
-----

```

6. เปลี่ยนสิทธิอนุญาตในสคริปต์ `passwd` โดยการป้อน:

```
chmod 4711 /usr/ucb/passwd
```

7. ตรวจสอบให้แน่ใจว่าตัวแปรสถานะแวดล้อม `PATH` ของผู้ใช้แต่ละ รายระบุว่า จะค้นหาไดเรกทอรี `/usr/ucb` ก่อน ไดเรกทอรี `/bin`

การประเมินประสิทธิภาพและการปรับสำหรับผู้จัดการระบบ BSD 4.3

ข้อมูลต่อไปนี้อธิบายแอตทริบิวต์อุปกรณ์ AIX และการประเมินประสิทธิภาพและการปรับ

อุปกรณ์ทุกรายการบน AIX มีแอตทริบิวต์ที่เชื่อมโยงด้วย เมื่อต้องการดูแอตทริบิวต์อุปกรณ์ให้ป้อน:

```
lsattr -E -l devicename
```

แอตทริบิวต์ใดๆ ที่มีค่า True สามารถแก้ไขได้ด้วยคำสั่ง:

```
chdev -l devicename -a attr=value
```

ข้อควรสนใจ: การเปลี่ยนพารามิเตอร์อุปกรณ์อย่างไม่ถูกต้องสามารถทำให้ระบบของคุณเสียหายได้

โดยค่าดีฟอลต์จำนวนโปรเซสสูงสุดต่อผู้ใช้ คือ 40 ค่าดีฟอลต์อาจต่ำเกินไปสำหรับผู้ใช้ที่มีหน้าต่างเปิดอยู่พร้อมกัน เป็นจำนวนมาก คำสั่งต่อไปนี้สามารถใช้เพื่อเปลี่ยนค่าทั้งระบบได้:

```
hdev -l sys0 -a maxuproc=100
```

ตัวอย่าง นี้เปลี่ยนจำนวนสูงสุดเป็น 100 ค่าใหม่มีการตั้งค่าหลังจากรีสตาร์ท ระบบแล้ว

เมื่อต้องการดูค่าที่ตั้งปัจจุบันของแอตทริบิวต์นี้และแอตทริบิวต์ ระบบอื่น ให้พิมพ์:

```
lsattr -E -l sys0
```

แอตทริบิวต์ `maxmbuf` ไม่ได้รับการสนับสนุนในปัจจุบัน โดยเซอร์วิส `mbuf`

AIX สนับสนุน คำสั่ง `vmstat` และ `iostat` แต่ไม่สนับสนุน คำสั่ง `sysstat` และโหลดเฉลี่ย สำหรับข้อมูลเพิ่มเติม เกี่ยวกับคำสั่งเหล่านี้ ให้ดูที่ `vmstat` และ `iostat`

เครื่องพิมพ์สำหรับผู้จัดการระบบ BSD 4.3

ระบบปฏิบัติการ AIX สนับสนุนระบบย่อยการพิมพ์ 2 แบบคือ: 4.3 BSD และ System V

ลักษณะ System V ของระบบย่อยการพิมพ์ใช้คำสั่ง System V รหัส 4, คิว, และไฟล์ และมีการจัดการในวิธีเดียวกัน ย่อหน้าต่อไปนี้อธิบายสิ่งที่คุณต้องทราบเพื่อจัดการกับ ลักษณะ 4.3 BSD ของระบบย่อยการพิมพ์ คุณควบคุมระบบย่อย ที่จะใช้งานโดยใช้ SMIT สามารถมีระบบย่อยที่ใช้งานอยู่ได้เพียงหนึ่งระบบเท่านั้น ในแต่ละครั้ง

การพิมพ์มีการจัดการโดยโปรแกรมและคอนฟิกรูเรชัน ในไดเรกทอรี /usr/lpd การออกแบบ คอนฟิกรูเรชัน กลไกการจัดการคิว และ daemon โปรแกรมของ 4.3 BSD และระบบย่อย เครื่องพิมพ์สำหรับ AIX แตกต่างกันอย่างใดก็ตาม ทั้งสองระบบใช้โปรโตคอล lpd สำหรับการพิมพ์รีโมต ทั้งสองระบบใช้ /etc/hosts.lpd ถ้ามีอยู่หรือ /etc/host.equiv ถ้าไม่มีอยู่ ระบบย่อย เครื่องพิมพ์สำหรับ AIX นำเสนอ gateway ไปยังระบบย่อยเครื่องพิมพ์ 4.3 BSD ดังนั้นระบบที่ใช้ AIX จึงสามารถส่งงานพิมพ์ไปยังระบบ 4.3 BSD และยอมรับงานพิมพ์ที่ส่งมาโดยระบบ 4.3 BSD ได้

ไฟล์ /etc/printcap ของ 4.3 BSD ไม่มีอยู่ใน AIX ไฟล์นี้ถูกเป็นการรวมกันระหว่างคอนฟิกรูเรชันตัวเก็บพักกับฐานข้อมูลความสามารถในการพิมพ์ ผู้ใช้ต้องเข้าใจรูปแบบและคีย์เวิร์ดของไฟล์ printcap เพื่อตั้งค่าเครื่องพิมพ์อย่างถูกต้อง

ไฟล์ /etc/qconfig ของ AIX มีข้อมูล คอนฟิกรูเรชันตัวเก็บพักอย่างเดียว ความสามารถของเครื่องพิมพ์ถูกกำหนดไว้ในฐานข้อมูล ODM ที่กำหนดไว้ล่วงหน้าหรือฐานข้อมูลกำหนดเอง คุณสามารถใช้คำสั่ง mkvirprt (ทำเครื่องพิมพ์เสมือน) เพื่อกำหนดความสามารถของเครื่องพิมพ์เฉพาะ ในระบบ

เมื่อต้องการทำให้เครื่องพิมพ์ lp0 มีอยู่สำหรับการพิมพ์บนรีโมตโฮสต์ viking ให้วางข้อมูลต่อไปนี้ในไฟล์ /etc/printcap ของระบบ 4.3 BSD:

```
lp0|Print on remote printer attached to
viking:Z
:lp=:rm=viking:rp=lp:st=/usr/spool/lp0d
```

เมื่อต้องการทำเช่นเดียวกันใน AIX ให้วางข้อมูลต่อไปนี้ในไฟล์ /etc/qconfig:

```
lp0:
    device = dlp0
    host = viking
    rq = lp
dlp0:
    backend = /usr/lib/lpd/rembak
```

AIX สนับสนุน คำสั่งเครื่องพิมพ์และฟังก์ชันไลบรารีต่อไปนี้:

ไอเท็ม	คำอธิบาย
cancel(1)	ยกเลิกการรอคอยไปยังเครื่องพิมพ์บรรทัด
chqueuedev(1)	เปลี่ยนชื่ออุปกรณ์คิวเครื่องพิมพ์หรือฟลิตเตอร์
chvirprt(1)	เปลี่ยนค่าแอตทริบิวต์ของเครื่องพิมพ์เสมือน
disable(1)	ปิดใช้งานคิวเครื่องพิมพ์
enable(1)	เปิดใช้งานคิวเครื่องพิมพ์
hplj(1)	ประมวลผลเอาต์พุต troff ในภายหลังสำหรับ HP LaserJetII ที่มีคาร์ทริดจ์ K
ibm3812(1)	ประมวลผลเอาต์พุต troff ในภายหลังสำหรับ IBM 3812 Mod 2 Pageprinter
ibm3816(1)	ประมวลผลเอาต์พุต troff ในภายหลังสำหรับ IBM 3816 Pageprinter
ibm5587G(1)	ประมวลผลเอาต์พุต troff ในภายหลังสำหรับ IBM 5587G ที่มีคาร์ทริดจ์ 32x32/24x24
lp(1)	ส่งการรอคอยไปยังเครื่องพิมพ์บรรทัด
lpr(1)	จัดคิวงานพิมพ์
lprm(1)	ลบงานออกจากคิวการเก็บพักของเครื่องพิมพ์บรรทัด
lpstat(1)	แสดงข้อมูลสถานะของเครื่องพิมพ์บรรทัด
lpstat(1)	สร้างรูปแบบระลอกของเครื่องพิมพ์บรรทัด
lsallqdev(1)	แสดงรายการชื่ออุปกรณ์คิวเครื่องพิมพ์ที่กำหนดคอนฟิกทั้งหมดภายในคิว
lsvirprt(1)	เปลี่ยนค่าแอตทริบิวต์ของเครื่องพิมพ์เสมือน

ไอเท็ม	คำอธิบาย
mkque(1)	เพิ่มคิวเครื่องพิมพ์ลงในระบบ
mkqudev(1)	เพิ่มอุปกรณ์คิวเครื่องพิมพ์ลงในระบบ
mkvirprt(1)	จัดทำเครื่องพิมพ์เสมือน
pac(1)	จัดเตรียมเรีกคอร์ดการบัญชีเครื่องพิมพ์/พล็อตเตอร์
pio(1)	Print Job Manager สำหรับ backend เครื่องพิมพ์
pioburst(1)	สร้างเพจการส่งเป็นชุด (เพจส่วนหัวและส่วนท้าย) สำหรับเอาต์พุต เครื่องพิมพ์
piocmdout(3)	รูทีนย่อยที่ส่งเอาต์พุตแอสกีวีตสตรีงสำหรับตัวจัดรูปแบบเครื่องพิมพ์
piodigest(1)	Digests ค่าแอสกีวีตสำหรับนิยามเครื่องพิมพ์เสมือนและ คลังข้อมูล
pioexit(3)	รูทีนย่อยที่ออกจากตัวจัดรูปแบบเครื่องพิมพ์
pioformat(1)	ควบคุมตัวจัดรูปแบบเครื่องพิมพ์
piofquote(1)	แปลงอักขระควบคุมบางตัวที่มีปลายทางเป็นเครื่องพิมพ์ PostScript
piogetstr(3)	รูทีนย่อยที่ดึงข้อมูลแอสกีวีตสตรีงสำหรับตัวจัดรูปแบบ เครื่องพิมพ์
piogetvals(3)	รูทีนย่อยที่เริ่มต้นตัวแปรฐานข้อมูล แอสกีวีตเครื่องพิมพ์ สำหรับ ตัวจัดรูปแบบเครื่องพิมพ์
piomsgout(3)	รูทีนย่อยที่ส่งข้อความจากตัวจัดรูปแบบเครื่องพิมพ์
pioout(1)	โปรแกรมอินเตอร์เฟซไทรเวอร์อุปกรณ์ของ backend เครื่องพิมพ์
pioprefdef(1)	สร้างนิยามสุดริมข้อมูลเครื่องพิมพ์ที่กำหนดไว้ล่วงหน้า
proff(1)	จัดรูปแบบข้อความสำหรับเครื่องพิมพ์ด้วยสุดริมข้อมูลเครื่องพิมพ์ส่วนบุคคล
qadm(1)	ทำการจัดการระบบสำหรับการเก็บพักเครื่องพิมพ์
qconfig(4)	กำหนดคอนฟิกการระบบการจัดคิวเครื่องพิมพ์
qstatus(1)	แสดงสถานะเครื่องพิมพ์ของระบบการเก็บพักการพิมพ์
restore(3)	เรียกคืนเครื่องพิมพ์เป็นสภาพดีพอลต์
rmque(1)	ลบคิวเครื่องพิมพ์ออกจากระบบ
rmqudev(1)	ลบอุปกรณ์คิวเครื่องพิมพ์หรือพล็อตเตอร์ออกจากระบบ
rmvirprt(1)	ลบเครื่องพิมพ์เสมือน
splp(1)	เปลี่ยนหรือแสดงค่าที่ตั้งไทรเวอร์เครื่องพิมพ์
xpr(1)	จัดรูปแบบไฟล์ต้นทางสำหรับเอาต์พุตไปยังเครื่องพิมพ์

ข้อมูลที่เกี่ยวข้อง:

ภาพรวมของเครื่องพิมพ์สำหรับการจัดการระบบ

คำสั่งสำหรับผู้ดูแลระบบของผู้จัดการระบบ BSD 4.3

รายการต่อไปนี้แสดงคำสั่งที่ใช้เฉพาะสำหรับการจัดการ สภาวะแวดล้อมของ AIX

ไอเท็ม	คำอธิบาย
bosboot(1)	เริ่มต้นอุปกรณ์บูต
bootlist(1)	เปลี่ยนรายการของอุปกรณ์บูต (หรือการจัดลำดับของอุปกรณ์เหล่านี้ ในรายการ) ที่มีอยู่บนระบบ
cfgmgr(1)	กำหนดคอนฟิกอุปกรณ์โดยการรันโปรแกรมในไดเรกทอรี /etc/methods
chcons(1)	เปลี่ยนทิศทางคอนโซลระบบไปยังอุปกรณ์หรือไฟล์ โดยมีผลบังคับใช้ ในสตาร์ทอัพครั้งถัดไป
chdev(1)	เปลี่ยนลักษณะของอุปกรณ์
chdisp(1)	เปลี่ยนจอแสดงผลที่ใช้โดยระบบย่อย low-function terminal (LFT)
checkkw(1)	จัดเตรียมข้อความความกว้างคงที่สำหรับคำสั่ง troff
checkq(1)	ตรวจสอบเอกสารที่จัดรูปแบบด้วยแม่โคร memorandum
checkmm(1)	ตรวจสอบเอกสารที่จัดรูปแบบด้วยแม่โคร memorandum
checknr(1)	ตรวจสอบไฟล์ nroff และ troff
chfont(1)	เปลี่ยนแปลงดีพอลต์ฟอนต์ที่เลือกต่อบูต
chfs(1)	เปลี่ยนแอสกีวีตของระบบไฟล์
chgroup(1)	เปลี่ยนแอสกีวีตสำหรับกลุ่ม
chgrpmm(1)	เปลี่ยนผู้ดูแลระบบหรือสมาชิกของกลุ่ม
chhwkbd(1)	เปลี่ยนแอสกีวีตคีย์บอร์ด low function terminal (LFT) ที่จัดเก็บไว้ในฐานข้อมูล Object Data Manager (ODM)
chitab(1)	เปลี่ยนแปลงเรีกคอร์ดในไฟล์ /etc/inittab
chkbd(1)	เปลี่ยนดีพอลต์คีย์บอร์ดแม่ที่ใช้โดย low-function terminal (LFT) ที่สตาร์ทอัพระบบ
chkey(1)	เปลี่ยนคีย์การเข้ารหัสของคุณ
chlang	ตั้งค่าตัวแปรสภาวะแวดล้อม LANG ในไฟล์ /etc/environment สำหรับล็อกอินถัดไป
chlicense(1)	การให้ไลเซนส์ผู้ใช้มีอยู่สองชนิดคือ แบบคงที่ และแบบลอยตัว การให้ไลเซนส์แบบคงที่ถูกรับประกันเสมอ และจำนวนไลเซนส์สามารถ เปลี่ยนได้โดยใช้แฟล็ก -u การให้ไลเซนส์แบบลอยตัวสามารถ เปิดใช้งานหรือปิดใช้งาน (เปิดหรือปิด) โดยใช้แฟล็ก -f

ไอเอ็ม	คำอธิบาย
chlv(1)	เปลี่ยนลักษณะของไลจิคัลวอลุ่ม
chnamsv(1)	เปลี่ยนคอนฟิกูเรชันเซอวิสช็อบน TCP/IP บนโฮสต์
chprtsv(1)	เปลี่ยนคอนฟิกูเรชันเซอวิสการพิมพ์บนเครื่องไคลเอ็นต์หรือ เซิร์ฟเวอร์
chps(1)	เปลี่ยนแอ็ททริบิวต์ของพื้นที่การเล็อนหนา
chpv(1)	เปลี่ยนคุณสมบัติของฟิลิคัลวอลุ่ม ในกลุ่มวอลุ่ม
chque(1)	เปลี่ยนชื่อคิว
chquedev(1)	เปลี่ยนชื่ออุปกรณ์คิวเครื่องพิมพ์หรือพล็อดเตอร์
chssys(1)	เปลี่ยนนิยามระบบย่อยในอ็อบเจ็กต์คลาส ระบบย่อย
chtcb(1)	เปลี่ยนหรือเค็ยวรีแอ็ททริบิวต์พื้นฐานการค่านวมที่เช็ถือได้ของ ไฟล์
chtz	เปลี่ยนข้อมูลโซนเวลาของระบบ
chuser(1)	เปลี่ยนแอ็ททริบิวต์สำหรับผู้ใช้ที่ระบุ
chvfs(1)	เปลี่ยนรายการในไฟล์ /etc/vfs
chvg(1)	ตั้งคาคคุณสมบัติของกลุ่มวอลุ่ม
chvirprt(1)	เปลี่ยนคาคแอ็ททริบิวต์ของเครื่องพิมพ์เสมือน
crfs(1)	เพิ่มระบบไฟล์
crvfs(1)	สร้างรายการในไฟล์ /etc/vfs
exportvg(1)	เอ็กซ์พอร์ตนิยามของกลุ่มวอลุ่มจากชุด ของฟิลิคัลวอลุ่ม
extndvg(1)	เพิ่มฟิลิคัลวอลุ่มในกลุ่มวอลุ่ม
grpck(1)	ตรวจสอบความถูกต้องของนิยามกลุ่ม
importvg(1)	อิมพอร์ตนิยามของกลุ่มวอลุ่มใหม่จากชุด ของฟิลิคัลวอลุ่ม
lsallq(1)	แสดงรายการชื่อของคิวที่กำหนดคอนฟิกทั้งหมด
lsallqdev(1)	แสดงรายการชื่ออุปกรณ์คิวเครื่องพิมพ์และพล็อดเตอร์ที่กำหนดคอนฟิกทั้งหมด ภายในคิวที่ระบุ
lsdisp(1)	แสดงรายการจ้อแสดงผลที่มีอยู่ในปัจจุบันบนระบบ
lsfont(1)	แสดงรายการฟอนต์ที่พร้อมใช้งานสำหรับจ้อแสดงผล
lsfs(1)	แสดงลักษณะของระบบไฟล์
lsgroup(1)	แสดงแอ็ททริบิวต์ของกลุ่ม
lsitab(1)	แสดงรายการเร็กคอร์ดในไฟล์ /etc/inittab
lskbd(1)	แสดงรายการค็ยบอร์ดแม็พที่มีอยู่ในปัจจุบันสำหรับระบบย่อย low-function terminal (LFT)
lslicense(1)	แสดงจำนวนไลเซนส์คงที่และสถานะของการไถ่ไลเซนส์ แบบลอยตัว
lslpp(1)	แสดงรายการผลิตภัณฑ์โปรแกรมที่เป็นทางเลือก
lsnamsv(1)	แสดงข้อมูลเซอวิสช็อบที่จัดเก็บไว้ในฐานข้อมูล
lsprtsv(1)	แสดงข้อมูลเซอวิสการพิมพ์ที่จัดเก็บไว้ในฐานข้อมูล
lsps	แสดงรายการพื้นที่การเล็อนหนาและแอ็ททริบิวต์
lsque(1)	แสดงชื่อ stanza คิว
lsquedev(1)	แสดงชื่อ stanza อุปกรณ์
lssrc(1)	เรียกสถานะของระบบย่อย กลุ่มของระบบย่อย หรือ เซิร์ฟเวอร์ย่อย
lsuser(1)	แสดงแอ็ททริบิวต์ของแอ็คเคาต์ผู้ใช้
lsvfs(1)	แสดงรายการในไฟล์ /etc/vfs
mkcatdefs(1)	ประมวลผลไฟล์แหล่งที่มาของข้อความล่งหนา
runcat(1)	ไพพ์ข้อมูลเอาต์พุตจากคำสั่ง mkcatdefs ไปยังคำสั่ง gencat
mkdev(1)	เพิ่มอุปกรณ์ลงในระบบ
mkfont(1)	เพิ่มฟอนต์โคัดที่เชื่อมโยงกับจ้อแสดงผลลงในระบบ
mkfontdir(1)	สร้างไฟล์ fonts.dir จากไดเรกทอรีของไฟล์ฟอนต์
mkgroup(1)	สร้างกลุ่มใหม่
mkitab(1)	จัดทำเร็กคอร์ดในไฟล์ /etc/inittab
mklv(1)	สร้างไลจิคัลวอลุ่ม
mklvcopy(1)	เพิ่มสำเนาลงในไลจิคัลวอลุ่ม
mknamsv(1)	กำหนดคอนฟิกเซอวิสช็อบน TCP/IP บนโฮสต์สำหรับไคลเอ็นต์
mknotify(1)	เพิ่มนิยามเมรอดแจงเตือนลงในคลาสอ็อบเจ็กต์แจงเตือน
mkprtsv(1)	กำหนดคอนฟิกเซอวิสการพิมพ์บน TCP/IP บนโฮสต์
mkps(1)	เพิ่มพื้นที่การเล็อนหนาเพิ่มเติมลงในระบบ
mkque(1)	เพิ่มคิวเครื่องพิมพ์ลงในระบบ
mkquedev(1)	เพิ่มอุปกรณ์คิวเครื่องพิมพ์ลงในระบบ
mkserver(1)	เพิ่มนิยามเซิร์ฟเวอร์ย่อยลงในคลาสอ็อบเจ็กต์เซิร์ฟเวอร์ย่อย
mkssys(1)	เพิ่มนิยามระบบย่อยลงในคลาสอ็อบเจ็กต์ระบบย่อย
mksysb	ทำสำเนาสำรองระบบไฟล์ที่ติดตั้งในกลุ่มวอลุ่ม rootvg สำหรับการติดตั้งใหม่ ในภายหลัง
mkszfile	บันทึกขนาดของระบบไฟล์ที่ติดตั้งในกลุ่มวอลุ่ม rootvg สำหรับการติดตั้งใหม่
mktcpip(1)	กำหนดคาคที่จำเป็นสำหรับการเริ่มต้น TCP/IP บนโฮสต์

ไอเท็ม	คำอธิบาย
mkuser(1)	สร้างแอคเคาต์ผู้ใช้ใหม่
mkuser.sys(1)	กำหนดแอคเคาต์ผู้ใช้ใหม่เอง
mkvg(1)	สร้างกลุ่มวอลุ่ม
mkvirprt(1)	จัดทำเครื่องพิมพ์เสมือน
odmadd(1)	เพิ่มอ็อบเจ็กต์ลงในคลาสอ็อบเจ็กต์ที่สร้างขึ้น
odmchange(1)	เปลี่ยนเนื้อหาของอ็อบเจ็กต์ที่เลือกในคลาสอ็อบเจ็กต์ที่ระบุ
odmcreate(1)	จัดทำไฟล์ .c (ซอร์ส) และ .h (รวม) ที่จำเป็นสำหรับการพัฒนาแอปพลิเคชัน ODM และสร้างคลาสอ็อบเจ็กต์ว่าง
odmdelete(1)	ลบอ็อบเจ็กต์ที่เลือกออกจากคลาสอ็อบเจ็กต์ที่ระบุ
odmdrop(1)	เอาคลาสอ็อบเจ็กต์ออก
odmget(1)	ดึงข้อมูลอ็อบเจ็กต์จากคลาสอ็อบเจ็กต์ที่ระบุ และวางไว้ในไฟล์อินพุต odmadd
odmshow(1)	แสดงนิยามคลาสอ็อบเจ็กต์บนหน้าจอ
pwdck(1)	ตรวจสอบความถูกต้องของข้อมูลการพิสูจน์ตัวตนโลคัล
redefinevg	กำหนดชุดของฟิลิควอลุ่มของกลุ่มวอลุ่มที่กำหนดอีกครั้ง ในฐานข้อมูลคอนฟิกูเรชันอุปกรณ์
reducevg(1)	ลบฟิลิควอลุ่มออกจากกลุ่มวอลุ่ม เมื่อฟิลิควอลุ่มทั้งหมด ถูกลบออกจากกลุ่มวอลุ่ม กลุ่มวอลุ่มจะถูกลบออก
reorgvg(1)	จัดระเบียบการจัดสรรพาร์ติชันฟิลิควอลุ่มอีกครั้งสำหรับกลุ่ม วอลุ่ม
restbase(1)	เรียกคืนข้อมูลแบบกำหนดเองจากอิมเมจเปิดเครื่องใหม่
rmddl(1)	ลบเดลตาออกจากไฟล์ Source Code Control System (SCCS)
rmdev(1)	ลบอุปกรณ์ออกจากระบบ
rmf(1)	ลบไฟล์เดอรัลและข้อความที่อยู่ในไฟล์เดอรัลนั้น
rmfs(1)	ลบระบบไฟล์
rmgroup(1)	ลบกลุ่ม
rmitab(1)	ลบเร็กคอร์ดในไฟล์ /etc/inittab
rmlv(1)	ลบไลจิคัลวอลุ่มออกจากกลุ่มวอลุ่ม
rmlvcopy(1)	ลบสำเนาออกจากไลจิคัลวอลุ่ม
rmm(1)	ลบข้อความ
rmnamsv(1)	ยกเลิกการกำหนดคอนฟิกเซอรัลส์ชื่อบน TCP/IP บนโฮสต์
rmnotify(1)	ลบนิยามเมธอดแจ้งเตือนออกจากคลาสอ็อบเจ็กต์แจ้งเตือน
rmprtvs(1)	ยกเลิกกำหนดคอนฟิกเซอรัลส์การพิมพ์บนเครื่องโคไลเอ็นต์หรือเซิร์ฟเวอร์
rmpps(1)	ลบพื้นที่การเลื่อนหนออกจากระบบ
rmque(1)	ลบคิวเครื่องพิมพ์ออกจากระบบ
rmqudev(1)	ลบอุปกรณ์คิวเครื่องพิมพ์หรือพล็อตเดอรัลออกจากระบบ
rmserver(1)	ลบนิยามเซิร์ฟเวอร์ย่อยออกจากคลาสอ็อบเจ็กต์เซิร์ฟเวอร์ย่อย
rmssys(1)	ลบนิยามระบบย่อยออกจากคลาสอ็อบเจ็กต์ระบบย่อย
rmuser(1)	ลบแอคเคาต์ผู้ใช้
rmvfs(1)	ลบรายการในไฟล์ /etc/vfs
rmvirprt(1)	ลบเครื่องพิมพ์เสมือน
savebase(1)	บันทึกข้อมูลอุปกรณ์แบบกำหนดเองพื้นฐานใน ODM บนอุปกรณ์ บูต
swapoff(1)	ยกเลิกการเรียกใช้พื้นที่การเลื่อนหนตั้งแต่หนึ่งพื้นที่ขึ้นไป
swapon(1)	ระบุอุปกรณ์เพิ่มเติมสำหรับการเลื่อนหนและการสลับ
syncvg(1)	ซิงโครไนซ์สำเนาไลจิคัลวอลุ่มที่ไม่ใช่สำเนาปัจจุบัน
usrck(1)	ตรวจสอบความถูกต้องของนิยามผู้ใช้
varyoffvg(1)	ยกเลิกการเรียกใช้กลุ่มวอลุ่ม
varyonvg(1)	เรียกใช้กลุ่มวอลุ่ม

หลักการที่เกี่ยวข้อง:

“ความแตกต่างที่สำคัญระหว่าง 4.3 BSD และ AIX” ในหน้า 353

ต่อไปนี้เป็นข้อมูลสรุปความแตกต่างที่สำคัญระหว่างระบบ AIX และ 4.3 BSD

Cron สำหรับผู้จัดการระบบ BSD 4.3

cron daemon สำหรับระบบปฏิบัติการนี้คล้ายกับ System V Release 2 cron daemon

รายการในไฟล์ /etc/inittab จะเริ่มต้น cron daemon

อุปกรณ์สำหรับผู้จัดการระบบ BSD 4.3

ข้อมูลต่อไปนี้อธิบายอุปกรณ์สำหรับผู้จัดการระบบ BSD 4.3

อุปกรณ์บนระบบ 4.3 BSD สามารถเข้าถึงได้โดยแอฟพลิเคชัน เฉพาะถ้า:

- อุปกรณ์มีการติดตั้งทางกายภาพและทำงานอยู่
- ไตรเวอร์ของอุปกรณ์อยู่ในเคอร์เนล
- มีไฟล์พิเศษอุปกรณ์ของอุปกรณ์อยู่ในไดเรกทอรี /dev

อุปกรณ์บน AIX สามารถเข้าถึงได้โดยแอฟพลิเคชันเฉพาะถ้า:

- อุปกรณ์มีการติดตั้งทางกายภาพและทำงานอยู่
- ไตรเวอร์ของอุปกรณ์อยู่ในเคอร์เนลหรือในส่วนขยายเคอร์เนลที่โหลด
- มีไฟล์พิเศษอุปกรณ์ของอุปกรณ์อยู่ในไดเรกทอรี /dev
- ฐานข้อมูลอ็อบเจกต์ในไดเรกทอรี /etc/objrepos มีรายการของอุปกรณ์ที่ตรงกับฟิสิคัลคอนฟิกูเรชัน

โปรแกรมเฉพาะอุปกรณ์ที่เรียกว่า *เมธอด* และอยู่ในไดเรกทอรี /etc/methods เก็บรักษาฐานข้อมูลอ็อบเจกต์ไว้ เมธอดถูกเรียกใช้โดย Configuration Manager (ที่เข้าถึงโดยใช้คำสั่ง `cfgmgr`) หรือคำสั่งอื่น

หากแอฟพลิเคชันโปรแกรมไม่สามารถเข้าถึงอุปกรณ์ได้อีกต่อไป นั่นอาจหมายความว่าฮาร์ดแวร์บกพร่อง หรือฐานข้อมูลคอนฟิกูเรชัน ในไดเรกทอรี /etc/objrepos เสียหาย

คำสั่ง `cfgmgr` ประมวลผล ฐานข้อมูลคอนฟิกูเรชันในไดเรกทอรี /etc/objrepos และมีการประมวลผล ในเวลาสตาร์ทอัพโดยคำสั่ง `cfgmgr` (Configuration Manager)

Pseudocode ข้างล่างแสดงตรรกะ Configuration Manager :

```
/* Main */
While there are rules in the Config_Rules database
{
    Get the next rule and execute it
    Capture stdout from the last execution
    Parse_Output(stdout)
}

/* Parse Output Routine */
/* stdout will contain a list of devices found */
Parse_OutPut(stdout)
{
    While there are devices left in the list
    {
        Lookup the device in the database
        if (!defined)
            Get define method from database and execute
        if (! configured)
        {
            Get config method from database and execute
            Parse_Output(stdout)
        }
    }
}
```

UUCP สำหรับผู้จัดการระบบ BSD 4.3

ตารางต่อไปนี้แสดงรายการคำสั่ง UUCP และไฟล์ต่างๆ

ไอเท็ม	คำอธิบาย
Dialers(4)	แสดงรายการโมเด็มที่ใช้สำหรับลิงก์การสื่อสารแบบรีโมต BNU
Maxuuxqts(4)	จำกัดจำนวนอินสแตนซ์ของ BNU <code>uuxqt</code> daemons ที่สามารถรันได้
Permissions(4)	ระบุสิทธิอนุญาตในคำสั่ง BNU สำหรับระบบรีโมต
Poll(4)	ระบุเวลาที่โปรแกรม BNU ควรสำรวจความคิดเห็นระบบรีโมต
Systems(4)	แสดงรายการคอมพิวเตอร์รีโมตซึ่งระบบโลคัลสามารถสื่อสารด้วยได้
rmail(1)	จัดการรีโมตเมลที่ได้รับผ่านทาง BNU
uuccheck(1)	ตรวจหาไฟล์และไดเรกทอรีที่ต้องการโดย BNU
uuclean(1)	ลบไฟล์ออกจากไดเรกทอรีเก็บพัก BNU
uucleanup(1)	ลบไฟล์ที่เลือกออกจากไดเรกทอรีเก็บพัก BNU
uucpadm(1)	ป้อนข้อมูลคอนฟิกูเรชัน BNU พื้นฐาน
uudemon.admin(1)	แสดงข้อมูลเป็นครั้งคราวเกี่ยวกับสถานะของการโอนย้ายไฟล์ BNU
uudemon.cleanu(1)	ทำความสะอาดไดเรกทอรีเก็บพัก BNU และล็อกไฟล์
uudemon.hour(1)	เริ่มต้นการเรียกการขนส่งไฟล์ไปยังระบบรีโมตโดยใช้โปรแกรม BNU
uudemon.poll(1)	สำรวจความคิดเห็นระบบที่แสดงรายการอยู่ในไฟล์ BNU Poll
uulog(1)	แสดงข้อมูลเกี่ยวกับกิจกรรมการโอนย้ายไฟล์ BNU บนระบบ
uupoll(1)	บังคับใช้การสำรวจความคิดเห็นของระบบ BNU รีโมต
uuq(1)	แสดงคิวงาน BNU และลบงานที่ระบออกจากคิว
uusnap(1)	แสดงสถานะของผู้ติดต่อ BNU กับระบบรีโมต
uustat(1)	รายงานสถานะและนำเสนอการควบคุมที่จำกัดบนการดำเนินงาน BNU

AIX ยังมี คำสั่ง 4.3 BSD `uuencode` และ `uudecode` ด้วย อย่างไรก็ตาม ไม่สนับสนุนคำสั่ง HDB `uugetty` สำหรับข้อมูล เกี่ยวกับคำสั่งเหล่านี้ให้ดูที่ `uuencode` และ `uudecode`

ข้อมูลที่เกี่ยวข้อง:

โครงสร้างไฟล์ BNU และไดเรกทอรี

ระบบไฟล์สำหรับผู้จัดการระบบ BSD 4.3

มีการใช้คำสั่งที่คล้ายกันเพื่อติดตั้งและถอนติดตั้งระบบไฟล์

AIX ใช้ไฟล์ `/etc/filesystem` เพื่อแสดงข้อมูลอุปกรณ์ระบบไฟล์ และมีคำสั่งที่คล้ายกันสำหรับการติดตั้ง และการถอนติดตั้งระบบไฟล์

ไฟล์ `/etc/filesystems` และไฟล์ `/etc/fstab`:

ระบบ 4.3 BSD จัดเก็บรายการของอุปกรณ์บล็อกและจุดต่อไว้ในไฟล์ `/etc/fstab` ระบบ SVR4 จัดเก็บข้อมูลอุปกรณ์บล็อกและจุดต่อไว้ในไฟล์ `/etc/vfstab` AIX จัดเก็บข้อมูลอุปกรณ์บล็อกและจุดต่อไว้ในไฟล์ `/etc/filesystems`

คำสั่ง `crfs`, `chfs`, และ `rmfs` อัปเดตไฟล์ `/etc/filesystems`

ผู้ดูแลระบบ 4.3 BSD อาจสนใจในตัวแปร `check` ในไฟล์ `/etc/filesystems` สามารถตั้งค่าตัวแปร `check` เป็นค่า `True`, `False`, หรือเป็นตัวเลข ตัวอย่างเช่น คุณสามารถระบุ `check=2` ในไฟล์ `/etc/filesystems` ตัวเลขระบุการส่งผ่านของคำสั่ง `fsck` ที่จะตรวจสอบระบบไฟล์นี้ พารามิเตอร์ `check` ตรงกับฟิลด์ที่ห้าในเรกคอร์ดไฟล์ `/etc/fstab`

ไม่มีพารามิเตอร์ความถี่เต็มพินในไฟล์ `/etc/filesystems`

การสนับสนุนระบบไฟล์บน AIX:

AIX สนับสนุนระบบไฟล์หลายระบบ

AIX สนับสนุน ดิสก์โควต้า

AIX ไม่อนุญาตให้ติดตั้งดิสก์เป็นระบบไฟล์

ไวยากรณ์ของคำสั่ง **mount** และ **umount** สำหรับ AIX แตกต่างจาก 4.3 BSD และแตกต่างจากเวอร์ชัน SVR4 ของคำสั่งเหล่านี้ คำสั่งที่จะติดตั้ง และถอนติดตั้งระบบไฟล์ทั้งหมดพร้อมกันแสดงขึ้นสำหรับระบบทั้งหมดสามารถพบในตารางต่อไปนี้:

คำสั่ง **mount** และ **umount**

ฟังก์ชัน	ไวยากรณ์สำหรับระบบปฏิบัติการนี้	ไวยากรณ์ 4.3 BSD	ไวยากรณ์ SVR4
ติดตั้งระบบไฟล์ทั้งหมด	mount all	mount -a	mountall
ถอนติดตั้งระบบไฟล์ทั้งหมด	umount all	umount -a	umountall

โปรดดูที่ “ระบบไฟล์” ในหน้า 464 สำหรับข้อมูลเพิ่มเติม

เทอร์มินัลสำหรับผู้จัดการระบบ BSD 4.3

ข้อมูลต่อไปนี้อธิบายเทอร์มินัลสำหรับผู้จัดการระบบ BSD 4.3

โดยดั้งเดิม ผู้จัดการระบบ 4.3 BSD เปิดใช้งานหรือปิดใช้งาน เทอร์มินัลพอร์ตโดยการแก้ไขไฟล์ `/etc/tty` และการส่งสัญญาณ HUP ไปยังโปรแกรม `init`

AIX จัดเก็บข้อมูลเทอร์มินัลพอร์ตไว้ใน ODM และเริ่มต้นเทอร์มินัลเมื่อโปรแกรม `init` อ่านไฟล์ `/etc/inittab` ใน AIX ให้ใช้อินเตอร์เฟซ SMIT เพื่อกำหนดคอนฟิกเทอร์มินัลพอร์ต

ไม่มีการแม็พวงที่ระหว่างพอร์ตและชื่อไฟล์พิเศษอุปกรณ์ในไดเรกทอรี `/dev` ส่งผลในเวลาต่อมาทำให้ผู้จัดการระบบสับสนว่าอะไรเป็นสิ่งที่ใหม่ซึ่ง AIX จะกำหนดคอนฟิกพอร์ตให้เมื่อใช้ SMIT พอร์ตอนุกรมเชิงระนาบตัวแรก (ทางกายภาพ มีเลเบลเป็น `s1`) มีการอ้างอิงเป็นที่ตั้ง `00-00-S1`, อะแดปเตอร์ `sa0`, และพอร์ต `s1` ในเมนู SMIT พอร์ตอนุกรมเชิงระนาบตัวที่สอง (ทางกายภาพมีเลเบลเป็น `s2`) มีการอ้างอิงเป็นที่ตั้ง `00-00-S2`, อะแดปเตอร์ `sa1`, และพอร์ต `s2`

ใช้คำสั่ง `penable` และ `pdisable` เพื่อเปิดใช้งานและปิดใช้งาน พอร์ต

termcap และ terminfo:

เช่นเดียวกับ System V ระบบปฏิบัติการนี้ใช้รายการ terminfo ในไฟล์ `/usr/lib/terminfo/?./*`

ผู้ใช้ที่มีระบบ 4.3 BSD อาจพบว่า คำสั่งต่อไปนี้ไม่มีประโยชน์:

captainfo(1)

แปลงไฟล์ termcap เป็นไฟล์ terminfo

tic(1)

แปลงไฟล์ terminfo จากซอร์สเป็นรูปแบบที่คอมไพล์

ระบบปฏิบัติการนี้มีซอร์สสำหรับรายการ terminfo จำนวนมาก ซอร์สบางส่วนเหล่านี้อาจต้องคอมไพล์โดยใช้คำสั่ง tic ไฟล์ termcap มีการจัดเตรียมไว้ให้ในไฟล์ /lib/libtermcap/termcap.src

การเปลี่ยนทิศทางอินพุตและเอาต์พุต

ระบบปฏิบัติการ AIX อนุญาตให้คุณจัดวางอินพุตและเอาต์พุต (I/O) ของข้อมูลเข้าและออกจากระบบของคุณได้ โดยใช้คำสั่ง I/O และสัญลักษณ์เฉพาะ

คุณสามารถควบคุมอินพุตโดยการระบุที่ตั้งซึ่งจะรวบรวมข้อมูล ตัวอย่างเช่น คุณสามารถระบุให้อ่านอินพุตในขณะที่ป้อนข้อมูลบนคีย์บอร์ด (อินพุต มาตรฐาน) หรือให้อ่านอินพุตจากไฟล์ คุณสามารถควบคุมเอาต์พุตโดยการระบุ ตำแหน่งที่จะแสดงหรือจัดเก็บข้อมูล คุณสามารถระบุให้เขียนข้อมูลเอาต์พุต ไปยังหน้าจอ (เอาต์พุตมาตรฐาน) หรือให้เขียนไปยังไฟล์

เนื่องจาก AIX เป็นระบบปฏิบัติการหลายภารกิจ จึงได้รับการออกแบบมาเพื่อจัดการกับโปรเซสเป็นชุด ผสมกัน

หลักการที่เกี่ยวข้อง:

“การจัดการไฟล์” ในหน้า 212

มีหลายวิธีในการทำงานกับไฟล์บนระบบของคุณ โดยปกติ คุณสร้างไฟล์ข้อความด้วยโปรแกรมแก้ไขข้อความ

“คำสั่งสำหรับการแสดงเนื้อหาไฟล์ (คำสั่ง pg, more, page, และ cat)” ในหน้า 217

คำสั่ง pg, more, และ page ช่วยให้คุณสามารถดูเนื้อหาของไฟล์และควบคุมความเร็วในการแสดงไฟล์ ของคุณได้

“การเปลี่ยนทิศทางอินพุตและเอาต์พุตใน Korn เซลล์หรือ POSIX เซลล์” ในหน้า 255

ก่อน Korn เซลล์ดำเนินการคำสั่ง เซลล์จะสแกนบรรทัดคำสั่ง เพื่อหาอักขระการเปลี่ยนทิศทาง สัญลักษณ์พิเศษเหล่านี้สั่งให้เซลล์เปลี่ยนทิศทางอินพุตและเอาต์พุต

ไฟล์อินพุตมาตรฐาน เอาต์พุตมาตรฐาน และข้อผิดพลาดมาตรฐาน

เมื่อคำสั่งเริ่มต้นรัน โดยปกติ คำสั่งคาดหวังว่าไฟล์ต่อไปนี้จะเปิดอยู่แล้ว: อินพุตมาตรฐาน เอาต์พุตมาตรฐาน และข้อผิดพลาดมาตรฐาน (บางครั้งเรียกว่า *เอาต์พุตข้อผิดพลาด* หรือเอาต์พุต *การวินิจฉัย*)

ตัวเลขซึ่งเรียกว่า *ตัวอธิบายไฟล์* มีการเชื่อมโยง กับแต่ละไฟล์เหล่านี้ ดังนี้:

ไอเท็ม	คำอธิบาย
ตัวอธิบายไฟล์ 0	อินพุตมาตรฐาน
ตัวอธิบายไฟล์ 1	เอาต์พุตมาตรฐาน
ตัวอธิบายไฟล์ 2	เอาต์พุตข้อผิดพลาดมาตรฐาน (การวินิจฉัย)

โดยปกติ โปรเซสซายันได้รับสืบทอดไฟล์เหล่านี้มาจาก พาเรนต์ โดยแรกเริ่ม ทั้งสามไฟล์เหล่านี้มีการกำหนดให้กับเวิร์กสเตชัน (0 ให้กับ คีย์บอร์ด, 1 และ 2 ให้กับจอแสดงผล) เซลล์อนุญาตให้เปลี่ยนทิศทางไฟล์เหล่านี้ไปยัง ที่อื่นได้ ก่อนจะส่งผ่านการควบคุมไปยังคำสั่ง

เมื่อคุณป้อนคำสั่ง ถ้าไม่มีการกำหนดชื่อไฟล์ คีย์บอร์ดของคุณจะเป็น *อินพุต มาตรฐาน* ซึ่งบางครั้งแสดงเป็น *stdin* เมื่อคำสั่งเสร็จสิ้น ผลลัพธ์ จะแสดงขึ้นบนจอภาพ

จอภาพของคุณเป็น *เอาต์พุตมาตรฐาน* ซึ่งบางครั้งแสดงเป็น *stdout* โดยค่าดีฟอลต์ คำสั่งใช้อินพุตจากอินพุตมาตรฐานและส่งผลลัพธ์ ไปยังเอาต์พุตมาตรฐาน

ข้อความแสดงข้อผิดพลาดถูกกำหนดทิศทางไปยังข้อผิดพลาดมาตรฐาน ซึ่งบางครั้งแสดงเป็น *stderr* โดยค่าดีฟอลต์ นี่เป็นหน้าจอของคุณ

การดำเนินการตีฟอลต์ของอินพุตและเอาต์พุตเหล่านี้สามารถเปลี่ยนแปลงได้ คุณสามารถใช้ไฟล์ เป็นอินพุตและเขียนผลลัพธ์ของคำสั่งไปยังไฟล์ การดำเนินการนี้เรียกว่า *การเปลี่ยนทิศทาง อินพุต/เอาต์พุต*

เอาต์พุตจากคำสั่ง ซึ่งโดยปกติ ไปยังอุปกรณ์จอแสดงผล สามารถถูก เปลี่ยนทิศทางไปยังไฟล์แทนได้ การดำเนินการนี้เรียกว่า *การเปลี่ยนทิศทางเอาต์พุต* ซึ่งเป็นประโยชน์เมื่อคุณมีเอาต์พุตจำนวนมากที่อ่านได้ยากบน หน้าจอ หรือเมื่อคุณต้องการวางไฟล์ไว้ด้วยกันเพื่อสร้างเป็นไฟล์ที่ใหญ่ขึ้น

อินพุตสำหรับคำสั่ง ซึ่งโดยปกติมาจากคีย์บอร์ด สามารถถูกเปลี่ยนทิศทางให้มาจากไฟล์ได้เช่นกัน แม้ว่าจะมีการใช้ไม่มากเท่ากับการเปลี่ยนทิศทางเอาต์พุต การดำเนินการนี้เรียกว่า *การเปลี่ยนทิศทางอินพุต* การเปลี่ยนทิศทางอินพุตช่วยให้คุณจัดเตรียมไฟล์ ล่วงหน้าจากนั้นให้คำสั่งอ่านไฟล์

การเปลี่ยนทิศทางเอาต์พุตมาตรฐาน

เมื่อมีการเพิ่มสัญกรณ์ `> filename` ที่ตอนท้ายของคำสั่ง เอาต์พุตของคำสั่งจะถูกเขียนลงในชื่อไฟล์ ที่ระบุ สัญลักษณ์ `>` รู้จักกันว่าเป็น *ตัวดำเนินการเปลี่ยนทิศทางเอาต์พุต*

คำสั่งใดๆ ที่ส่งเอาต์พุตผลลัพธ์ของคำสั่งไปยังหน้าจอ สามารถถูกกลับทิศทางเอาต์พุตไปยังไฟล์ได้

การเปลี่ยนทิศทางเอาต์พุตไปยังไฟล์

เอาต์พุตของโปรแกรมสามารถถูกเปลี่ยนทิศทางไปยังไฟล์ได้โดยพิมพ์ คำสั่งตามด้วยตัวดำเนินการเปลี่ยนทิศทางเอาต์พุตและชื่อไฟล์

ตัวอย่างเช่น เมื่อต้องการเปลี่ยนทิศทางผลลัพธ์ของคำสั่ง `who` ไปยังไฟล์ที่ชื่อ `users` ให้พิมพ์ดังต่อไปนี้:

```
who > users
```

หมายเหตุ: หากไฟล์ `users` มีอยู่แล้ว ไฟล์จะถูกลบออกและแทนที่ ยกเว้นว่ามีการระบุอ็อปชัน `noclobber` ของคำสั่งในตัว `set ksh` (Korn เชลล์) หรือ `csk` (C เชลล์)

เมื่อต้องการดูเนื้อหาของไฟล์ `users` ให้พิมพ์ดังต่อไปนี้:

```
cat users
```

รายการ คล้ายกับดังต่อไปนี้แสดงขึ้น:

```
denise 1ft/0 May 13 08:05
marta pts/1 May 13 08:10
endrica pts/2 May 13 09:33
```

การเปลี่ยนทิศทางเอาต์พุตเพื่อผนวกเข้ากับไฟล์

เมื่อมีการเพิ่มสัญกรณ์ `>> filename` ที่ตอนท้ายของคำสั่ง เอาต์พุตของคำสั่งจะถูกผนวกเข้ากับชื่อไฟล์ ที่ระบุ แทนการเขียนทับบนข้อมูลที่มีอยู่ สัญลักษณ์ `>>` รู้จักกันว่าเป็น *ตัวดำเนินการการเปลี่ยนทิศทางต่อท้าย*

ตัวอย่างเช่น เมื่อต้องการผนวก `file2` เข้ากับ `file1` ให้พิมพ์ดังต่อไปนี้:

```
cat file2 >> file1
```

หมายเหตุ: หากไฟล์ `file1` ไม่มีอยู่ ไฟล์จะถูกสร้างขึ้น ยกเว้นว่ามีการระบุ อ็อปชัน `noclobber` ของคำสั่งในตัว `set ksh` (Korn เชลล์) หรือ `csk` (C เชลล์)

การสร้างไฟล์ข้อความด้วยการเปลี่ยนทิศทางจากคีย์บอร์ด

คำสั่ง `cat` ซึ่งใช้งานโดยลำพังจะใช้ข้อมูลใดๆ ที่คุณพิมพ์จากคีย์บอร์ดเป็นอินพุต คุณสามารถเปลี่ยนทิศทางอินพุตนี้ไปยังไฟล์ได้

กด `Ctrl-D` บนบรรทัดใหม่เพื่อส่งสัญญาณการสิ้นสุดของข้อความ

ที่พร้อมระบบ ให้พิมพ์ดังต่อไปนี้:

```
cat > filename
This is a test.
^D
```

การต่อไฟล์ข้อความ

คุณสามารถรวมหลายไฟล์เข้าเป็นหนึ่งไฟล์ได้ การรวมหลายไฟล์ เข้าเป็นหนึ่งไฟล์เรียกว่า *การต่อ*

ตัวอย่างต่อไปนี้สร้าง `file4` ซึ่งประกอบด้วย `file1`, `file2`, และ `file3` ที่ผนวกเข้าด้วยกันในลำดับข้างล่าง

โปรดดูตัวอย่างต่อไปนี้:

- ที่พร้อมระบบ ให้พิมพ์ดังต่อไปนี้:

```
cat file1 file2 file3 > file4
```

- ตัวอย่างต่อไปนี้แสดงข้อผิดพลาดทั่วไปเมื่อต่อไฟล์:

```
cat file1 file2 file3 > file1
```

ข้อควรสนใจ: ในตัวอย่างนี้ คุณอาจคาดหวังให้คำสั่ง `cat` ผนวกเนื้อหาของ `file1`, `file2`, และ `file3` เข้าเป็น `file1` คำสั่ง `cat` สร้างไฟล์เอาต์พุตขึ้นก่อน แล้วลบเนื้อหาของ `file1` ออกจริงจากนั้นผนวก `file2` และ `file3` เข้ากับ `file1`

การเปลี่ยนทิศทางอินพุตมาตรฐาน

เมื่อมีการเพิ่มสัญกรณ์ `< filename` ที่ตอนท้ายของคำสั่ง อินพุตของคำสั่งจะถูกอ่านจากชื่อไฟล์ ที่ระบุ สัญลักษณ์ `<` รู้จักกันว่า *เป็นตัวดำเนินการเปลี่ยนทิศทาง อินพุต*

หมายเหตุ: สามารถเปลี่ยนทิศทางได้เฉพาะอินพุตของคำสั่งที่โดยปกติใช้อินพุต จากคีย์บอร์ด

ตัวอย่างเช่น เมื่อต้องการส่งไฟล์ `letter1` เป็นข้อความไปยังผู้ใช้ `denise` โดยใช้คำสั่ง `mail` ให้พิมพ์ดังต่อไปนี้:

```
mail denise < letter1
```

การทิ้งเอาต์พุตด้วยไฟล์ `/dev/null`

ไฟล์ `/dev/null` เป็นไฟล์พิเศษ ไฟล์นี้ มีคุณสมบัติเฉพาะคือ ไฟล์ว่างเสมอ ข้อมูลใดๆ ที่ส่งไปยัง `/dev/null` จะ ถูกทิ้งไป นี้เป็นคุณลักษณะที่มีประโยชน์เมื่อคุณรันโปรแกรมหรือคำสั่งที่ สร้างเอาต์พุตซึ่งคุณต้องการละเว้น

ตัวอย่างเช่น คุณมีโปรแกรมที่ชื่อ `myprog` ซึ่งยอมรับ อินพุตจากหน้าจอและสร้างข้อความในขณะที่กำลังรันซึ่ง คุณไม่ต้องการเห็นบนหน้าจอของคุณ เมื่อต้องการอ่านอินพุตจากไฟล์ `myscript` และ ทิ้งข้อความเอาต์พุตมาตรฐาน ให้พิมพ์ดังต่อไปนี้:

```
myprog < myscript >/dev/null
```

ในตัวอย่างนี้ `myprog` ใช้ไฟล์ `myscript` เป็นอินพุต และเอาต์พุตมาตรฐานทั้งหมดถูกทิ้งไป

การเปลี่ยนทิศทางข้อผิดพลาดมาตรฐานและเอาต์พุตอื่น

นอกเหนือจากอินพุตมาตรฐานและเอาต์พุตมาตรฐานแล้ว คำสั่งมักจะจัดทำเอาต์พุตชนิดอื่น เช่น ข้อผิดพลาด หรือข้อความสถานะ ที่รู้จักกันว่าเป็นเอาต์พุตการวิเคราะห์ เช่นเดียวกับเอาต์พุตมาตรฐานคือ เอาต์พุตข้อผิดพลาด มาตรฐานมีการเขียนไปย้งหน้าจอยกเว้นว่าถูกเปลี่ยนทิศทาง

เมื่อต้องการเปลี่ยนทิศทางข้อผิดพลาดมาตรฐานหรือเอาต์พุตอื่น ให้ใช้ตัวอธิบายไฟล์ *ตัวอธิบายไฟล์* เป็นตัวเลขที่เชื่อมโยงกับไฟล์ I/O แต่ละไฟล์ซึ่งคำสั่งใช้ตามปกติ นอกจากนี้ยังสามารถระบุตัวอธิบายไฟล์ เพื่อเปลี่ยนทิศทางอินพุตมาตรฐานและเอาต์พุตมาตรฐาน หมายเลข ต่อไปนี้เชื่อมโยงกับอินพุตมาตรฐาน เอาต์พุต และ ข้อผิดพลาด:

ไอเท็ม	คำอธิบาย
0	อินพุตมาตรฐาน (คีย์บอร์ด)
1	เอาต์พุตมาตรฐาน (จอแสดงผล)
2	ข้อผิดพลาดมาตรฐาน (จอแสดงผล)

เมื่อต้องการเปลี่ยนทิศทางเอาต์พุตข้อผิดพลาดมาตรฐาน ให้พิมพ์ตัวอธิบายไฟล์หมายเลข 2 ข้างหน้า เอาต์พุต หรือผนวกสัญลักษณ์การเปลี่ยนทิศทาง (> หรือ >>) และชื่อไฟล์หลังจากสัญลักษณ์ ตัวอย่างเช่น คำสั่งต่อไปนี้ ใช้เอาต์พุตข้อผิดพลาดมาตรฐานจากคำสั่ง `cc` ซึ่งใช้ในการคอมไพล์ไฟล์ `testfile.c` และผนวกเอาต์พุตนั้นเข้ากับตอนท้ายของไฟล์ `ERRORS`:

```
cc testfile.c 2 >> ERRORS
```

เอาต์พุตชนิดอื่นยังสามารถเปลี่ยนทิศทางได้เช่นกันโดยใช้ตัวอธิบายไฟล์ ตั้งแต่ 0 ถึง 9 ตัวอย่างเช่น ถ้า คำสั่ง `cmd` เขียนเอาต์พุตไปยังตัวอธิบายไฟล์ 9 คุณสามารถเปลี่ยนทิศทางเอาต์พุตนั้นไปยังไฟล์ `savedata` ด้วยคำสั่งต่อไปนี้:

```
cmd 9> savedata
```

หากคำสั่งเขียนมากกว่าหนึ่งเอาต์พุต คุณสามารถเปลี่ยนทิศทางแต่ละ เอาต์พุตอย่างเป็นอิสระจากกัน สมมติว่าคำสั่งกำหนดทิศทางเอาต์พุตมาตรฐานไปยัง ตัวอธิบายไฟล์ 1, กำหนดทิศทางเอาต์พุตข้อผิดพลาดมาตรฐานไปยังตัวอธิบายไฟล์ 2, และสร้างไฟล์ข้อมูลบนตัวอธิบายไฟล์ 9 บรรทัดคำสั่งต่อไปนี้ จะเปลี่ยนทิศทางแต่ละเอาต์พุตเหล่านั้นไปยังไฟล์ที่แตกต่างกัน:

```
command > standard 2> error 9> data
```

การเปลี่ยนทิศทางเอาต์พุตไปยังเอกสารอินไลน์อินพุต (here)

คุณสามารถเปลี่ยนทิศทางเอาต์พุตไปยังเอกสารอินไลน์อินพุต (here)

ถ้าคำสั่งอยู่ในรูปแบบต่อไปนี้:

```
command << eofstring
```

และ `eofstring` เป็นสตริงใดๆ ที่ไม่มีอักขระการจับคู่รูปแบบ เชลล์จะใช้บรรทัดในลำดับต่อมาเป็น อินพุตมาตรฐานของ `command` จนกว่า เชลล์อ่านบรรทัดที่มี `eofstring` (อาจมีอักขระแท็บ หนึ่งหรือหลายอักขระนำหน้า) เพียงอย่างเดียว บรรทัดระหว่าง `eofstring` ลำดับแรก และที่สองมักเรียกกันว่า เอกสารอินไลน์อินพุต หรือ เอกสาร here หากมียัติภังค์ (-) ตามหลังต่อจากอักขระการเปลี่ยนทิศทาง << เชลล์จะ strip อักขระแท็บนำจากแต่ละบรรทัดของเอกสาร here ก่อน ส่งผ่านบรรทัดไปยัง `command`

เชลล์ สร้างไฟล์ชั่วคราวที่มีเอกสาร here และทำการทดแทน ตัวแปรและคำสั่งในเนื้อหาก่อนการส่งผ่านไฟล์ ไปยังคำสั่ง เชลล์ทำการจับคู่รูปแบบบนชื่อไฟล์ที่เป็นส่วนหนึ่งของ บรรทัดคำสั่งในการทดแทนคำสั่ง เมื่อต้องการห้ามการทดแทนทั้งหมด ให้ใส่อักขระใดๆ ของ `eofstring` ไว้ในเครื่องหมายอัฒภาค:

```
command << \eofstring
```

เอกสาร **here** มีประโยชน์มากเป็นพิเศษสำหรับข้อมูลอินพุตจำนวนเล็กน้อยที่สามารถวางในเชลล์พรอซีเดอร์ท่ได้สะดวกกว่าการเก็บไว้ในไฟล์แยกต่างหาก (เช่น สคริปต์โปรแกรมแก้ไข) ตัวอย่างเช่น คุณสามารถพิมพ์ดังต่อไปนี้:

```
cat <<- xyz
  This message will be shown on the
  display with leading tabs removed.
xyz
```

หลักการที่เกี่ยวข้อง:

“การเปลี่ยนทิศทางอินพุตและเอาต์พุตใน Korn เชลล์หรือ POSIX เชลล์” ในหน้า 255

ก่อน Korn เชลล์ดำเนินการคำสั่ง เชลล์จะสแกนบรรทัดคำสั่ง เพื่อหาอักขระการเปลี่ยนทิศทาง สัญกรณ์พิเศษเหล่านี้สั่งให้เชลล์เปลี่ยนทิศทางอินพุตและเอาต์พุต

การเปลี่ยนทิศทางเอาต์พุตโดยใช้ไพพ์และตัวกรอง

คุณสามารถเชื่อมต่อคำสั่งตั้งแต่สองคำสั่งขึ้นไปเพื่อให้สามารถใช้เอาต์พุต มาตรฐานของคำสั่งหนึ่งเป็นอินพุตมาตรฐานของคำสั่งอื่น ชุดของ คำสั่งที่เชื่อมต่อในวิธีนี้รู้จักกันว่าเป็น *ไพพ์ไลน์*

การเชื่อมต่อที่รวมคำสั่งเข้าด้วยกันรู้จักกันว่า เป็น *ไพพ์* ไพพ์มีประโยชน์เนื่องจากช่วยให้คุณรวมคำสั่งวัตถุประสงค์เดียว หลายคำสั่งเข้าเป็นคำสั่งที่ประสิทธิภาพเพียงคำสั่งเดียว คุณสามารถกำหนดทิศทางเอาต์พุต จากคำสั่งหนึ่งให้กลายเป็นอินพุตของคำสั่งอื่นได้โดยใช้ไพพ์ไลน์ คำสั่งเชื่อมต่อกันด้วยสัญลักษณ์ไพพ์ (|)

เมื่อ คำสั่งใช้อินพุตจากคำสั่งอื่น แก้ไข และส่งผลลัพธ์ ไปยังเอาต์พุตมาตรฐาน เรียกกันว่าเป็น *ตัวกรอง* ตัวกรองสามารถใช้โดยลำพังได้ แต่มีประโยชน์มากขึ้นเมื่อใช้ในไพพ์ไลน์ ตัวกรองที่ใช้กันมากที่สุด มีดังนี้:

- sort
- more
- pg

โปรดดูตัวอย่างต่อไปนี้:

- คำสั่ง **ls** เขียนเนื้อหาของไดเรกทอรีปัจจุบัน ไปยังหน้าจอในหนึ่งสตรีมข้อมูลที่เลื่อนจอภาพได้ เมื่อมีการแสดงข้อมูลมากกว่าหนึ่งหน้าจอ ข้อมูลบางส่วนจะหายไปจากหน้าจอ เมื่อต้องการควบคุมเอาต์พุตเพื่อให้เนื้อหาแสดงขึ้น ทีละหน้าจอ คุณสามารถใช้ไพพ์ไลน์เพื่อกำหนดทิศทางเอาต์พุตของคำสั่ง **ls** ไปยังคำสั่ง **pg** ซึ่งควบคุมรูปแบบของเอาต์พุต บนหน้าจอ ตัวอย่างเช่น พิมพ์ดังต่อไปนี้:

```
ls | pg
```

ในตัวอย่าง นี้ เอาต์พุตของคำสั่ง **ls** กลายเป็น อินพุตของคำสั่ง **pg** กด Enter เพื่อดู หน้าจอถัดไป

ไพพ์ไลน์ดำเนินงานในทิศทางเดียวเท่านั้น (จากซ้ายไปขวา) แต่ละ คำสั่งในไพพ์ไลน์รันเป็นโปรเซสแยกต่างหาก และโปรเซสทั้งหมดสามารถรัน ในเวลาเดียวกัน โปรเซสจะหยุดชั่วคราวเมื่อไม่มีอินพุตจะอ่าน หรือเมื่อ ไพพ์กับโปรเซสถัดไปเต็ม

- อีกตัวอย่างหนึ่งของการใช้ไพพ์คือไพพ์กับคำสั่ง **grep** คำสั่ง **grep** ค้นหาไฟล์เพื่อหาบรรทัดที่มี สตรีมของบางรูปแบบ เมื่อต้องการแสดงไฟล์ทั้งหมดของคุณที่สร้างหรือแก้ไข ในเดือนกรกฎาคม ให้พิมพ์ดังต่อไปนี้:

```
ls -l | grep Jul
```

ในตัวอย่าง นี้ เอาต์พุตของคำสั่ง **ls** กลายเป็น อินพุตของคำสั่ง **grep**

การแสดงเอาต์พุตโปรแกรมและการคัดลอกไปยังไฟล์ (คำสั่ง tee)

คำสั่ง `tee` ที่ใช้กับไฟฟ์ อ่านอินพุตมาตรฐาน จากนั้นเขียนเอาต์พุตของโปรแกรมไปยังเอาต์พุตมาตรฐานและคัดลอกเอาต์พุตเข้าในไฟล์ที่ระบุในเวลาเดียวกัน ใช้คำสั่ง `tee` เพื่อดูเอาต์พุตของคุณในทันทีและในเวลาเดียวกัน จัดเก็บเอาต์พุตไว้ในอนาคต

ตัวอย่างเช่น พิมพ์ดังต่อไปนี้:

```
ps -ef | tee program.ps
```

คำสั่งนี้แสดงเอาต์พุตมาตรฐานของคำสั่ง `ps -ef` บน อุปกรณ์แสดงผล และในเวลาเดียวกัน บันทึกสำเนาของเอาต์พุตนั้นไว้ในไฟล์ `program.ps` หากมีไฟล์ `program.ps` อยู่แล้ว ไฟล์จะถูกลบออก และแทนที่ยกเว้นว่ามีการระบุอ็อปชัน `noclobber` ของคำสั่งในตัว `set`

ตัวอย่างเช่น เมื่อต้องการดูและบันทึกเอาต์พุตจาก คำสั่งไว้ในไฟล์ที่มีอยู่:

```
ls -l | tee -a program.ls
```

คำสั่งนี้แสดงเอาต์พุตมาตรฐานของ `ls -l` ที่อุปกรณ์แสดงผล และในเวลาเดียวกัน แนบสำเนาของเอาต์พุตนั้นไว้ที่ตอนท้ายของไฟล์ `program.ls`

ระบบ แสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้ และไฟล์ `program.ls` มีข้อมูลเดียวกัน:

```
-rw-rw-rw- 1 jones staff 2301 Sep 19 08:53 161414
-rw-rw-rw- 1 jones staff 6317 Aug 31 13:17 def.rpt
-rw-rw-rw- 1 jones staff 5550 Sep 10 14:13 try.doc
```

ให้ดูที่คำสั่ง `tee` ใน *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

การล้างหน้าจอของคุณ (คำสั่ง clear)

ใช้คำสั่ง `clear` เพื่อทำหน้าจอของข้อความและอินพุต คีย์บอร์ดให้ว่าง

ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
clear
```

ระบบ ล้างหน้าจอและแสดงพร้อมท์

ส่งข้อความไปยังเอาต์พุตมาตรฐาน

ใช้คำสั่ง `echo` เพื่อแสดงข้อความบน หน้าจอ

ตัวอย่างเช่น เมื่อต้องการเขียนข้อความไปยัง เอาต์พุตมาตรฐาน ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
echo Please insert diskette . . .
```

ข้อความต่อไปนี้แสดงขึ้น:

```
Please insert diskette . . .
```

ตัวอย่างเช่น เมื่อต้องการใช้คำสั่ง `echo` พร้อมด้วยอักขระการจับคู่รูปแบบ ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
echo The back-up files are: *.bak
```

ระบบ แสดงข้อความ The back-up files are: ตามด้วย ชื่อไฟล์ในไดเรกทอรีปัจจุบันที่ลงท้ายด้วย .bak

การผนวกข้อความหนึ่งบรรทัดเข้ากับไฟล์ (คำสั่ง echo)

ใช้คำสั่ง echo ที่ใช้กับตัวดำเนินการการเปลี่ยนทิศทางต่อท้าย เพื่อเพิ่มข้อความหนึ่งบรรทัดลงในไฟล์

ตัวอย่างเช่น ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
echo Remember to back up mail files by end of week.>>notes
```

คำสั่งนี้เพิ่มข้อความ Remember to back up mail files by end of week. ที่ตอนท้ายของไฟล์ notes

การคัดลอกหน้าจอของคุณไปยังไฟล์ (คำสั่ง capture และ script)

ใช้คำสั่ง capture ซึ่งเลียนแบบเทอร์มินัล VT100 เพื่อคัดลอกข้อมูลทุกอย่างที่พิมพ์บนเทอร์มินัลของคุณไปยังไฟล์ซึ่งคุณระบุ
ใช้คำสั่ง script เพื่อคัดลอกข้อมูลทุกอย่างที่พิมพ์บนเทอร์มินัลของคุณ ไปยังไฟล์ซึ่งคุณระบุ โดยไม่ต้องเลียนแบบเทอร์มินัล VT100

ทั้งสองคำสั่งมีประโยชน์สำหรับการพิมพ์เรกคอร์ดของไดอะล็อกเทอร์มินัล

ตัวอย่าง เช่น เมื่อต้องการดักจับหน้าจอของเทอร์มินัลในขณะที่เลียนแบบ VT100 ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
capture screen.01
```

ระบบแสดงข้อมูล คล้ายกับตัวอย่างต่อไปนี้:

```
Capture command is started. The file is screen.01.  
Use ^P to dump screen to file screen.01.  
You are now emulating a vt100 terminal.  
Press Any Key to continue.
```

หลังจากป้อน ข้อมูลและดัมพ์เนื้อหาหน้าจอแล้ว ให้หยุดคำสั่ง capture โดยกด Ctrl-D หรือพิมพ์ exit และกด Enter ระบบแสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
Capture command is complete. The file is screen.01.  
You are NO LONGER emulating a vt100 terminal.
```

ใช้คำสั่ง cat เพื่อแสดงเนื้อหาของไฟล์ของคุณ

ตัวอย่างเช่น เมื่อต้องการดักจับหน้าจอ ของเทอร์มินัล ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
script
```

ระบบแสดงข้อมูล คล้ายกับตัวอย่างต่อไปนี้:

```
Script command is started. The file is typescript.
```

ขณะนี้ ข้อมูลทุกอย่าง ที่แสดงขึ้นบนหน้าจอถูกคัดลอกไปยังไฟล์ typescript

เมื่อต้องการ หยุดคำสั่ง script ให้กด Ctrl-D หรือพิมพ์ exit และ กด Enter ระบบแสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
Script command is complete. ไฟล์คือ typescript
```

ใช้คำสั่ง cat เพื่อแสดงเนื้อหาของไฟล์ของคุณ

โปรดดูที่ คำสั่ง `capture` และ `script` in *Commands Reference* สำหรับไวยากรณ์ทั้งหมด

คำสั่งในการแสดงข้อความในตัวอักษรขนาดใหญ่บนหน้าจอของคุณ (คำสั่ง `banner`)

คำสั่ง `banner` แสดงอักขระ ASCII บนหน้าจอของคุณในตัวอักษรขนาดใหญ่

แต่ละบรรทัดในเอาต์พุตสามารถมีความยาวได้สูงสุด 10 ตำแหน่ง (หรือ อักขระตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็ก)

ตัวอย่างเช่น ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
banner GOODBYE!
```

ระบบแสดง GOODBYE! ใน ตัวอักษรขนาดใหญ่บนหน้าจอของคุณ

ข้อมูลสรุปคำสั่งสำหรับการเปลี่ยนทิศทางอินพุตและเอาต์พุต

ข้อมูลต่อไปนี้เป็นคำสั่งสำหรับการเปลี่ยนทิศทางอินพุตและเอาต์พุต

ไอเท็ม	คำอธิบาย
>	“การเปลี่ยนทิศทางเอาต์พุตมาตรฐาน” ในหน้า 376
<	“การเปลี่ยนทิศทางอินพุตมาตรฐาน” ในหน้า 377
>>	“การเปลี่ยนทิศทางเอาต์พุตเพื่อผนวกเข้ากับไฟล์” ในหน้า 376
	“การเปลี่ยนทิศทางเอาต์พุตโดยใช้ไฟฟ์และตัวกรอง” ในหน้า 379
<code>banner</code>	เขียนสตริงอักขระ ASCII ในตัวอักษรตัวใหญ่ไปยัง เอาต์พุตมาตรฐาน
<code>capture</code>	อนุญาตให้พิมพ์หน้าจอเทอร์มินัลไปยังไฟล์
<code>clear</code>	ล้างหน้าจอเทอร์มินัล
<code>echo</code>	เขียนสตริงอักขระไปยังเอาต์พุตมาตรฐาน
<code>script</code>	อนุญาตให้คัดลอกอินพุตและเอาต์พุตเทอร์มินัลไปยัง ไฟล์
<code>tee</code>	แสดงเอาต์พุตมาตรฐานของโปรแกรมและ คัดลอกไปยังไฟล์

การกู้คืนเคอร์เนล AIX

เริ่มต้นด้วย AIX 6.1 เคอร์เนลสามารถเลือกที่จะกู้คืนจากข้อผิดพลาดในรูทีนที่เลือกไว้ เพื่อหลีกเลี่ยง สัญญาณระบบขาดหายที่ไม่ได้วางแผนไว้

การกู้คืนเคอร์เนลปิดใช้งานโดยค่าดีฟอลต์ หากมีการเปิดใช้งานการกู้คืนเคอร์เนล ระบบอาจหยุดชั่วคราวในระหว่างการดำเนินการกู้คืนเคอร์เนล โดยทั่วไป เวลานี้น้อยกว่าสองวินาที การดำเนินการต่อไปนี้จะเกิดขึ้นในทันที หลังจากการดำเนินการกู้คืนเคอร์เนล:

- คอนโซลระบบแสดงข้อความต่อไปนี้:

```
-----  
A kernel error recovery action has occurred. A recovery log  
has been logged in the system error log.  
-----
```

- AIX เพิ่มรายการลงใน บันทึกข้อผิดพลาด คุณสามารถส่งข้อมูลบันทึกข้อผิดพลาดไปยัง IBM เพื่อขอรับเซอรัลได้ ในลักษณะคล้ายกับการส่งข้อมูล เนื่องจากทั้งระบบไม่ทำงาน ข้อมูลต่อไปนี้เป็นตัวอย่างรายการบันทึกข้อผิดพลาด การกู้คืน:

```
LABEL:          RECOVERY  
Date/Time:      Fri Feb 16 14:04:17 CST 2007  
Type:           INFO  
Resource Name:  RMGR  
Description  
Kernel Recovery Action
```

```

Detail Data
Live Dump Base Name
RECOV_20070216200417_0000
Function Name
w_clear
FRR Name
w_init_clear_frr
Symptom String
273
EEEE00009627A072
F10001001B18BBC0
w_clear+D0
wdog0030+288
test_index+4C
Recovery Log Data
0001 0000 0000 0000 F000 0000 2FFC AEBO 0000 0111 0000 0000 0000 0021 25BC
8000 0000 0002 9032 EEEE 0000 9627 A072 F100 0100 1B18 BBC0 0000 0000 0000 0000
0000 0001 0000 0000 0006 0057 D2FF 8C00 0001 0148 0500 0000 8000 0000 0002 9032
.....

```

- AIX สร้างดัมพ์ไฟล์ข้อมูลจากดัมพ์ไฟล์ที่ตั้งอยู่ในไดเรกทอรี /var/adm/ras/livedump โดย คำติฟอลต์ และไฟล์มีชื่อว่า **RECOV_timestamp_sequence** โดยที่ *timestamp* ระบุ เวลาที่เกิดการกู้คืนเคอร์เนล และ *sequence* ระบุ จำนวนครั้งที่เรียกใช้การกู้คืนเคอร์เนล คุณสามารถส่งข้อมูลดัมพ์ไฟล์ไปยัง IBM เพื่อขอรับเซอร์วิสได้ในลักษณะคล้ายกับการส่งข้อมูล เนื่องจากทั้งระบบไม่ทำงาน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับดัมพ์ไฟล์ให้ดูที่ ดัมพ์ไฟล์ใน *Kernel Extensions and Device Support Programming Concepts*

ข้อควรระวัง: บางฟังก์ชันอาจหายไปหลังจากการกู้คืนเคอร์เนล แต่ระบบปฏิบัติการยังคงอยู่ในสภามั่นคง หากจำเป็น ให้ปิดและรีสตาร์ทระบบของคุณเพื่อเรียกคืนฟังก์ชันที่หายไป

ข้อควรพิจารณาเกี่ยวกับหน่วยความจำและตัวประมวลผล

AIX รักษาข้อมูลสถานะของการกู้คืนเคอร์เนลในระหว่างการดำเนินงานเคอร์เนล mainline เมื่อ เปิดใช้งานการกู้คืนเคอร์เนล ต้องการคำแนะนำตัวประมวลผลเพิ่มเติม เพื่อรักษาข้อมูลไว้ และต้องการหน่วยความจำเพิ่มเติมเพื่อบันทึกข้อมูล การดำเนินการนี้ ส่งผลกระทบท่อการใช้ตัวประมวลผลน้อยมาก การใช้หน่วยความจำเพิ่มเติมสามารถกำหนดได้ โดยใช้สมการต่อไปนี้ โดยที่ *maxthread* คือ จำนวนเธรดสูงสุดที่กำลังรันอยู่บนระบบ และ *procnum* คือ จำนวนตัวประมวลผล:

$$\text{memory required} = 4 \text{ KB} \times \text{maxthread} + 128 \text{ KB} \times \text{procnum}$$

ดังแสดงในตัวอย่างต่อไปนี้ ระบบที่มี 16 ตัวประมวลผลและเธรดสูงสุด 1000 เธรดใช้หน่วยความจำเพิ่มเติมจำนวน 6304 KB:

$$4 \times 1000 + 128 \times 16 = 6304 \text{ KB}$$

การเปิดใช้งานและการปิดใช้งานการกู้คืนเคอร์เนล

คุณสามารถเปิดใช้งานหรือปิดใช้งานการกู้คืนเคอร์เนลได้จากอินเทอร์เฟซพาร์ SMIT

เมื่อต้องการเปิดใช้งาน หรือปิดใช้งานการกู้คืนเคอร์เนล ให้ใช้พาร์ SMIT ต่อไปนี้:

การกำหนดปัญหา > การกู้คืน เคอร์เนล > เปลี่ยนสถานะการกู้คืนเคอร์เนล > เปลี่ยน สถานะการกู้คืนเคอร์เนลในบูตถัดไป

คุณยังสามารถแสดงสถานะการกู้คืนเคอร์เนลปัจจุบัน โดยใช้พาร์ SMIT ต่อไปนี้:

การจัดการอุปกรณ์

คุณสามารถใช้คำสั่งเพื่อจัดการอุปกรณ์ต่างๆ ที่มีอยู่ใน AIX อุปกรณ์บางรายการ ซึ่งคุณสามารถจัดการได้คือ Logical Volume Manager, ระบบไฟล์ เทปไดรฟ์ และเครื่องพิมพ์

Logical Volume Manager

ชุดของคำสั่งระบบปฏิบัติการ รุทีนย่อยไลบรารี และ เครื่องมืออื่นที่ช่วยให้คุณสร้างและควบคุมหน่วยเก็บโลจิคัลวอลุ่มได้ เรียกว่า Logical Volume Manager (LVM)

LVM ควบคุมดิสก์หรือสื่อบันทึกข้อมูลระหว่างมุมมอง โลจิคัล แบบง่าย และยึดหยุ่นของพื้นที่วางหน่วยเก็บกับ ฟิสิคัล ดิสก์จริง LVM ทำเช่นนี้โดยใช้เลเยอร์ของโคตไดรเวอร์อุปกรณ์ที่รันบนไดรเวอร์อุปกรณ์ ดิสก์แบบดั้งเดิม

LVM ประกอบด้วยไดรเวอร์อุปกรณ์โลจิคัลวอลุ่ม (LVDD) และไลบรารี อินเตอร์เฟซรูทีนย่อย LVM ไดรเวอร์อุปกรณ์โลจิคัลวอลุ่ม (LVDD) เป็นไดรเวอร์อุปกรณ์เทียมที่จัดการและประมวลผล I/O ทั้งหมด ไดรเวอร์นี้แปล โลจิคัลแอดเดรสเป็นฟิสิคัลแอดเดรสและส่งคำร้องขอ I/O ไปยังไดรเวอร์ อุปกรณ์เฉพาะ *ไลบรารีอินเตอร์เฟซรูทีนย่อย LVM* มีรูทีน ที่ใช้โดยคำสั่งการจัดการระบบเพื่อทำภารกิจจัดการระบบ สำหรับโลจิคัลและฟิสิคัลวอลุ่มของระบบ

ข้อมูลที่เกี่ยวข้อง:

ภาพรวมของการเขียนโปรแกรมโลจิคัลวอลุ่ม

การทำความเข้าใจกับไดรเวอร์อุปกรณ์โลจิคัลวอลุ่ม

แนวคิด Logical Volume Manager

ก่อนคุณสามารถเริ่มต้นใช้ Logical Volume Manager คุณต้องเข้าใจ กลไกและคำศัพท์พื้นฐาน

โปรเซสแปรผัน:

โปรเซสแปรผันเป็นกลไกหนึ่งที่ LVM ใช้ เพื่อให้มั่นใจว่ากลุ่มวอลุ่มพร้อมใช้งานและมีข้อมูล ล่าสุด

คำสั่ง `varyonvg` และ `varyoffvg` เรียกใช้หรือยกเลิกเรียกใช้ (ทำให้พร้อมใช้งานหรือไม่พร้อมใช้งาน) กลุ่มวอลุ่ม ซึ่งคุณกำหนดให้กับระบบ กลุ่มวอลุ่มต้องถูก `varied on` ก่อน ระบบสามารถเข้าถึงได้ ในระหว่างโปรเซสแปรผัน LVM จะอ่านข้อมูลการจัดการจากฟิสิคัลวอลุ่มที่กำหนดไว้ในกลุ่มวอลุ่ม ข้อมูลการจัดการนี้ ซึ่งรวมถึง volume group descriptor area (VGDA) และ volume group status area (VGSA) มีการจัดเก็บไวบนแต่ละฟิสิคัลวอลุ่มของกลุ่มวอลุ่ม

VGDA มีข้อมูลที่อธิบายการแม็พของพาร์ติชันฟิสิคัลกับ พาร์ติชันโลจิคัลสำหรับแต่ละโลจิคัลวอลุ่มในกลุ่มวอลุ่ม ตลอดจนข้อมูลที่สำคัญมาก อื่นๆ รวมถึงการประทับเวลา VGSA มีข้อมูล เช่น พาร์ติชันฟิสิคัลที่เก่า และฟิสิคัลวอลุ่มที่ขาดหายไป (นั่นคือไม่มีอยู่หรือใช้ไม่ได้) เมื่อพยายามดำเนินงานแปรผัน บนกลุ่มวอลุ่ม

หากการดำเนินงานแปรผันไม่สามารถเข้าถึงฟิสิคัลวอลุ่มตั้งแต่หนึ่งรายการขึ้นไป ที่กำหนดไว้ในกลุ่มวอลุ่ม คำสั่งจะแสดงชื่อของฟิสิคัลวอลุ่มทั้งหมดซึ่ง กำหนดสำหรับกลุ่มวอลุ่มนั้นและสถานะ ซึ่งช่วยให้คุณตัดสินใจได้ว่า ควรจะ vary-off กลุ่มวอลุ่มนี้หรือไม่

หลักการที่เกี่ยวข้อง:

“การมีอยู่สูงในกรณีของความล้มเหลวดิสก์” ในหน้า 440
เมธอดหลักที่ใช้เพื่อป้องกันความล้มเหลวดิสก์เกี่ยวข้องกับ ค่าติดตั้งคอนฟิกูเรชันโลจิคัลวอลุ่ม เช่น การมีร์เรอร์

องค์ประกอบ:

องค์ประกอบเป็นกลไกหนึ่งที่ LVM ใช้เพื่อให้มั่นใจว่า กลุ่มวอลุ่มพร้อมใช้งานและมีข้อมูลล่าสุด

องค์ประกอบเป็นส่วนหนึ่งของจำนวนของ Volume Group Descriptor Areas และ Volume Group Status Areas (VGDA/VGSA) ที่ใช้งานอยู่ องค์ประกอบช่วยให้มั่นใจถึงบูรณภาพข้อมูล ของพื้นที่ VGDA/VGSA ในกรณีที่ดิสก์ล้มเหลว แต่ละฟิลิคัลดิสก์ในกลุ่มวอลุ่มมีอย่างน้อยหนึ่ง VGDA/VGSA เมื่อสร้างกลุ่มวอลุ่มขึ้น บนดิสก์เดียว โดยแรกเริ่ม กลุ่มวอลุ่มจะมีสองพื้นที่ VGDA/VGSA อยู่บนดิสก์ หากกลุ่มวอลุ่มมีสองดิสก์ ดิสก์หนึ่งยังคงมีสอง VGDA/VGSA พื้นที่ แต่อีกดิสก์หนึ่งมีหนึ่ง VGDA/VGSA เมื่อกลุ่มวอลุ่มมีตั้งแต่สามดิสก์ขึ้นไป แต่ละดิสก์จะได้รับจัดสรรเพียงหนึ่ง VGDA/VGSA เท่านั้น

องค์ประกอบสูญหายไปเมื่อดิสก์อย่างน้อยครั้งหนึ่ง (หมายถึงพื้นที่ VGDA/VGSA) สามารถอ่านได้โดย LVM ในกลุ่มวอลุ่มที่มีสองดิสก์ ถ้า ดิสก์ที่มีเพียง VGDA/VGSA เดียวสูญหายไป องค์ประกอบยังคงมีอยู่เนื่องจาก สองจากสามพื้นที่ VGDA/VGSA ยังคงสามารถเข้าถึงได้ ถ้าดิสก์ที่มีสองพื้นที่ VGDA/VGSA สูญหายไป คำสั่งนี้จะไม่เป็นจริงอีกต่อไป ส่วนประกอบดิสก์ของกลุ่มวอลุ่ม ยิ่งมาก โอกาสการสูญหายขององค์ประกอบเมื่อดิสก์หนึ่งล้มเหลว ยิ่งน้อยลง

เมื่อองค์ประกอบสูญหาย กลุ่มวอลุ่มจะ varies off ตัวเองเพื่อไม่ให้ LVM เข้าถึง ดิสก์อีกต่อไป ลักษณะนี้ป้องกันไม่ให้มีดิสก์ I/O เพิ่มเติมใน กลุ่มวอลุ่มนั้น เพื่อไม่ให้ข้อมูลสูญหายหรือมีการเขียนข้อมูลเมื่อเปิดปัญหา ทางกายภาพ นอกจากนั้น ผลของ vary-off คือผู้ใช้จะได้รับแจ้ง ในบันทึกข้อผิดพลาดว่า เกิดข้อผิดพลาดฮาร์ดแวร์ขึ้นและต้องทำการเซอร์วิส

อย่างไรก็ตาม มีกรณีที่ควรดำเนินการกลุ่มวอลุ่มต่อไป แม้ว่าองค์ประกอบสูญหาย ในกรณีเหล่านี้ การตรวจสอบองค์ประกอบ อาจถูกปิด สำหรับกลุ่มวอลุ่ม กลุ่มวอลุ่มชนิดนี้เรียกว่า *กลุ่มวอลุ่มที่ไม่ใช่ของค์ประกอบ* กรณีทั่วไปที่สุดสำหรับกลุ่มวอลุ่มที่ไม่ใช่ของค์ประกอบเกิดขึ้น เมื่อมีการมีร์เรอร์โลจิคัลวอลุ่ม เมื่อดิสก์สูญหาย ข้อมูลจะไม่สูญหายถ้าสำเนาของโลจิคัลวอลุ่มอยู่บนดิสก์ที่ไม่ได้ถูก ปิดใช้งานและสามารถเข้าถึงได้ อย่างไรก็ตาม อาจมีอินสแตนซ์ในกลุ่มวอลุ่มที่ไม่ใช่ของค์ประกอบ ทั้งที่มีร์เรอร์หรือไม่ได้มีร์เรอร์ เมื่อข้อมูล (รวมถึงสำเนา) อยู่บน ดิสก์ที่ไม่มีอยู่ในอินสแตนซ์ดังกล่าว ข้อมูลอาจไม่สามารถเข้าถึงได้ แม้ว่ากลุ่มวอลุ่ม varied on ต่อไป

หลักการที่เกี่ยวข้อง:

“การแปลงกลุ่มวอลุ่มเป็นสถานะไม่ใช่ของค์ประกอบ” ในหน้า 387

คุณสามารถเปลี่ยนกลุ่มวอลุ่มเป็นสถานะไม่ใช่ของค์ประกอบเพื่อให้ข้อมูล มีอยู่อย่างต่อเนื่องแม้ว่าไม่มีองค์ประกอบ

มีร์เรอร์พูล:

มีร์เรอร์พูลทำให้สามารถแบ่งฟิลิคัลวอลุ่มของ กลุ่มวอลุ่มออกเป็นพูลแยกต่างหากได้

มีร์เรอร์พูลประกอบด้วยฟิลิคัลวอลุ่มตั้งแต่หนึ่งรายการขึ้นไป แต่ละ ฟิลิคัลวอลุ่มสามารถเป็นสมาชิกของหนึ่งมีร์เรอร์พูลเท่านั้นในแต่ละครั้ง เมื่อ สร้างโลจิคัลวอลุ่ม สามารถกำหนดแต่ละสำเนาของโลจิคัลวอลุ่มที่กำลังสร้าง ให้กับมีร์เรอร์พูลได้ สำเนาโลจิคัลวอลุ่มที่ถูกกำหนดให้กับ มีร์เรอร์พูลจะจัดสรรพาร์ติชันจากฟิลิคัลวอลุ่มใน มีร์เรอร์พูลนั้นเท่านั้น ลักษณะนี้ช่วยให้สามารถจำกัดดิสก์ ที่สำเนาโลจิคัลวอลุ่มสามารถใช้ได้ หากไม่มีมีร์เรอร์พูล วิธีเดียวในการ จำกัดฟิลิคัลวอลุ่มที่จะใช้สำหรับการจัดสรรเมื่อ สร้างหรือขยายโลจิคัลวอลุ่มคือ การใช้ไฟล์แม่พิมพ์ ดังนั้น การใช้มีร์เรอร์พูลจึงช่วยให้โปรเซสง่ายขึ้นอย่างมาก มีร์เรอร์พูลสามารถสร้างขึ้น โดยใช้คำสั่ง `extendvg` หรือคำสั่ง `chpv`

คุณต้องระบุชื่อมีร์เรอร์พูลเมื่อคุณสร้างมีร์เรอร์พูลใหม่ ชื่อมีร์เรอร์พูลต้องสอดคล้องกับกฎต่อไปนี้:

- สามารถมีอักขระตัวอักษรและตัวเลขเท่านั้น
- ต้องน้อยกว่าหรือเท่ากับ 15 อักขระ
- ต้องไม่ซ้ำกันในกลุ่มวอลุ่ม

หลังจากใช้มีร์เรอร์พูลในกลุ่มวอลุ่มแล้ว จะไม่สามารถอิมพอร์ตกลุ่มวอลุ่มเข้าในเวอร์ชันของ AIX ที่ไม่สนับสนุนมีร์เรอร์พูล ซึ่งรวมถึงเวอร์ชันใดๆ ของ AIX ก่อนหน้า 6.1.1.0 นอกจากนี้เพื่อใช้มีร์เรอร์พูลกับ LVM โหมดเกิดขึ้นพร้อมกันที่เพิ่มประสิทธิภาพ โหนดทั้งหมดในคลัสเตอร์ต้องสนับสนุนมีร์เรอร์พูล

ข้อจำกัดของมีร์เรอร์พูล

ข้อจำกัดของมีร์เรอร์พูลสามารถนำไปใช้เพื่อเพิ่มข้อจำกัดที่เข้มงวดมากขึ้นในการใช้มีร์เรอร์พูล ข้อจำกัดของ มีร์เรอร์พูล สามารถมีค่าอย่างใดอย่างหนึ่งจากสามค่าต่อไปนี้:

- off เมื่อข้อจำกัดของมีร์เรอร์พูลมีการตั้งค่าเป็น off จะไม่มี ข้อจำกัดใดๆ เกี่ยวกับการใช้มีร์เรอร์พูล นี้เป็นค่าดีฟอลต์
- on เมื่อข้อจำกัดของมีร์เรอร์พูลมีการตั้งค่าเป็น on สำเนาโลจิคัลวอลุ่ม แต่ละรายการที่สร้างขึ้นในกลุ่มวอลุ่มต้องถูกกำหนดให้กับ มีร์เรอร์พูล
- super เมื่อข้อจำกัดของมีร์เรอร์พูลมีการตั้งค่าเป็น super จะใช้ข้อจำกัดต่อไปนี้:
 - โลคัลและรีโมตฟิสิคัลวอลุ่มไม่สามารถเป็นสมาชิกของมีร์เรอร์พูล เดียวกัน

หมายเหตุ: สำหรับข้อมูลเพิ่มเติมเกี่ยวกับโลคัลและรีโมตฟิสิคัลวอลุ่มให้อ้างอิงที่เอกสารคู่มือ HACMP/XD GLVM

- สามารถมีได้สูงสุดสามมีร์เรอร์พูลในหนึ่งกลุ่มวอลุ่ม
- แต่ละมีร์เรอร์พูลต้องมีสำเนาของแต่ละโลจิคัลวอลุ่มใน กลุ่มวอลุ่มอย่างน้อยหนึ่งสำเนา

Geographic Logical Volume Manager:

Geographic Logical Volume Manager (GLVM) ช่วยให้คุณสามารถ รักษาสำเนา มีร์เรอร์ของข้อมูลของคุณไว้ที่ที่ตั้งทาง ภูมิศาสตร์ระยะไกล

GLVM สามารถช่วยปกป้องธุรกิจของคุณจากภัยพิบัติได้โดยการมีร์เรอร์ข้อมูลที่ สำคัญมากไว้บนไซต์การกู้คืนภัยพิบัติรีโมต หากภัยพิบัติ เช่น อัคคีภัยหรืออุทกภัย ทำลายข้อมูลที่ไซต์การผลิตของคุณ คุณจะมีสำเนาสำรอง ของข้อมูลอยู่ที่ไซต์การกู้คืน ภัยพิบัติ

ข้อมูลมีการมีร์เรอร์บนเครือข่าย TCP/IP มาตรฐาน ไซต์การผลิตและ ไซต์การกู้คืนภัยพิบัติต้องอยู่บนเครือข่ายฟิสิคัลเดียวกัน สามารถใช้เราเตอร์และเกตเวย์ระหว่างสองไซต์ได้ แทนสายเคเบิลดิสก์ที่ยาวมาก มีการใช้เครือข่าย TCP/IP และไดรเวอร์ อุปกรณ์ Remote Physical Volume (RPV) สำหรับการเข้าถึงรีโมตดิสก์

ผู้ใช้กำหนดคอนฟิกดิสก์ทางภูมิศาสตร์ระยะไกลเป็นรีโมตฟิสิคัลวอลุ่ม จากนั้นรวมรีโมตฟิสิคัลวอลุ่มดังกล่าวเข้ากับโลคัลฟิสิคัลวอลุ่ม เพื่อประกอบขึ้นเป็นกลุ่มวอลุ่มที่ทำสำเนาทางภูมิศาสตร์ กลุ่มวอลุ่มเหล่านี้มีการจัดการ โดย Logical Volume Manager (LVM) และทำงานคล้ายกับกลุ่มวอลุ่มปกติ GLVM สนับสนุนทั้งการมีร์เรอร์รีโมตแบบซิงโครนัสและแบบอะซิงโครนัส

กลุ่มวอลุ่มที่ไม่ใช่องค์ประกอบ:

Logical Volume Manager (LVM) ยกเลิกการเรียกใช้กลุ่มวอลุ่มโดยอัตโนมัติเมื่อขาดองค์ประกอบของ Volume Group Descriptor Areas (VGDA) หรือ Volume Group Status Areas (VGSA) อย่างไรก็ตาม คุณสามารถเลือกอ็อปชันที่อนุญาตให้กลุ่มยังคงออนไลน์ตราบเท่าที่มีหนึ่งคู่ VGDA/VGSA ในสภาพปกติ อ็อปชันนี้จัดทำ *กลุ่มวอลุ่มที่ไม่ใช่องค์ประกอบ*

LVM ต้องการสิทธิเข้าถึงในดิสก์ทั้งหมดในกลุ่มวอลุ่มที่ไม่ใช่องค์ประกอบ ก่อนจะอนุญาตการเรียกใช้ใหม่ ลักษณะนี้ทำให้มั่นใจว่า VGDA และ VGSA ทันทสมัย

คุณอาจต้องการจัดทำกลุ่มวอลุ่มที่ไม่ใช่องค์ประกอบในระบบซึ่งทุกโลจิคัลวอลุ่ม มีสำเนาอย่างน้อยสองชุด หากความล้มเหลวของดิสก์เกิดขึ้น กลุ่มวอลุ่ม ยังคงใช้งานอยู่ตราบเท่าที่มีดิสก์ที่ใช้งานอยู่หนึ่งดิสก์

หมายเหตุ: ทั้งกลุ่มวอลุ่มที่ผู้ใช้กำหนดและ `rootvg` สามารถดำเนินงานในสถานะที่ไม่ใช่ องค์ประกอบ แต่เมธอดที่ใช้กำหนดคอนฟิกกลุ่มวอลุ่มที่ผู้ใช้กำหนดและกลุ่มวอลุ่ม `rootvg` เป็นไม่ใช่องค์ประกอบและสำหรับการกู้คืนหลังจากฮาร์ดแวร์ล้มเหลวแตกต่างกัน ตรวจสอบให้แน่ใจว่าคุณใช้เมธอดที่ถูกต้องสำหรับกลุ่มวอลุ่มที่เหมาะสม

แม้ว่าคุณกำลังใช้กลุ่มวอลุ่มที่ไม่ใช่องค์ประกอบ อาจ เป็นไปได้ที่จะสูญเสียองค์ประกอบและเห็นข้อความต่อไปนี้ในเอาต์พุตคำสั่ง `errpt` :

```
QUORUM LOST, VOLUME GROUP CLOSING LVM.
```

ข้อความนี้ เกิดขึ้นเมื่อฟิลต์วอลุ่มทั้งหมดอยู่ในสถานะ `missing` และ LVM varies off กลุ่มวอลุ่มโดยอัตโนมัติ

ข้อความแจ้งว่า `QUORUM LOST` เนื่องจากการปิดใช้งาน องค์ประกอบบนกลุ่มวอลุ่มลดความต้องการองค์ประกอบเป็น 1 คุณสามารถใช้คำสั่ง `lsvg vgrname` เพื่อแสดงค่าองค์ประกอบ ซึ่งอยู่ในฟิลต์ `QUORUM`: ในกรณีที่ฟิลต์วอลุ่มทั้งหมดเป็น `missing` แม้ว่าคุณต้องการองค์ประกอบต่ำสุด นี้ถูกละเมิด จะส่งผลให้เกิดข้อความ `lost quorum` และ `vary off` อัตโนมัติของกลุ่มวอลุ่ม

ข้อมูลที่เกี่ยวข้อง:

 Logical Volume Manager จาก A ถึง Z: บทนำและแนวคิด

การแปลงกลุ่มวอลุ่มเป็นสถานะไม่ใช่องค์ประกอบ:

คุณสามารถเปลี่ยนกลุ่มวอลุ่มเป็นสถานะไม่ใช่องค์ประกอบเพื่อให้ข้อมูล มีอยู่อย่างต่อเนื่องแม้ว่าไม่มีองค์ประกอบ

โพรซีเดรน์นี้มักใช้สำหรับระบบที่มีคอนฟิกูเรชันต่อไปนี้:

- กลุ่มวอลุ่มสองดิสก์ซึ่งมีการมีร์เรอร์โลจิคัลวอลุ่ม
- กลุ่มวอลุ่มสามดิสก์ซึ่งมีการมีร์เรอร์โลจิคัลวอลุ่ม หนึ่งหรือสองครั้ง

เมื่อกลุ่มวอลุ่มภายใต้สถานการณ์เหล่านี้สามารถดำเนินงานในสถานะไม่ใช่องค์ประกอบได้แม้ว่าเกิดความล้มเหลวดิสก์ กลุ่มวอลุ่มยังคงใช้งานอยู่ ตราบเท่าที่มีอย่างน้อยหนึ่งดิสก์ในกลุ่มวอลุ่มใช้งานอยู่

เมื่อต้องการกู้คืนกลุ่มที่ไม่ใช่องค์ประกอบที่เป็นไปได้ให้ตรวจสอบจนแน่ใจดังต่อไปนี้:

- หากระบบของคุณใช้ระบบไฟล์ JFS หรือ JFS2 ให้มีร์เรอร์โลจิคัลวอลุ่มบันทึก JFS

- วางสำเนาที่มิร์เรอร์บนดิสก์แยกต่างหาก หากคุณไม่แน่ใจเกี่ยวกับคอนฟิกูเรชัน ให้พิมพ์คำสั่งต่อไปนี้เพื่อตรวจสอบที่ตั้งทางกายภาพ (PV1, PV2, และ PV3) ของแต่ละพาร์ติชันโลจิคัล (เพื่อวางสำเนาบนดิสก์แยกต่างหาก คอลัมน์ PV1, PV2, และ PV3 ต้องมีหมายเลข hdisk ที่แตกต่างกัน)

```
lslv -m LVName
```

หาก โลจิคัลวอลุ่มมีเฉพาะสำเนาของตนเองที่ตั้งอยู่บนดิสก์เดียวกัน และดิสก์นั้น ไม่พร้อมใช้งาน วอลุ่มจะไม่มีอยู่สำหรับผู้ใช้โดยไม่คำนึงถึง สถานะองค์ประกอบหรือไม่ใช่ขององค์ประกอบของกลุ่มวอลุ่ม

ทั้งกลุ่มวอลุ่มที่ผู้ใช้กำหนดและ rootvg สามารถดำเนินงานในสถานะที่ไม่ใช่ องค์ประกอบ แต่คอนฟิกูเรชันและเมธอดการกู้คืนแตกต่างกัน

เพื่อเรียกใช้กลุ่มวอลุ่มที่ไม่ใช่ขององค์ประกอบซึ่งผู้ใช้กำหนด ฟิสิคัลวอลุ่มของกลุ่ม วอลุ่มทั้งหมดต้องสามารถเข้าถึงได้ มิฉะนั้น การเรียกใช้จะล้มเหลว เนื่องจากกลุ่มวอลุ่มที่ไม่ใช่ องค์ประกอบยังคงออนไลน์จนกว่าดิสก์ล่าสุดไม่สามารถเข้าถึงได้ ดังนั้นจึงจำเป็น ที่แต่ละดิสก์ต้องสามารถเข้าถึงได้ในเวลาที่เรียกใช้

ข้อควรระวัง: เมื่อดิสก์ที่เชื่อมโยงกับกลุ่มวอลุ่ม rootvg ขาดหายไป ให้หลีกเลี่ยงการเปิดระบบยกเว้นว่าไม่สามารถซ่อมแซมดิสก์ที่ขาดหายไปได้ Logical Volume Manager (LVM) ใช้แฟล็ก `-f` เสมอ เพื่อบังคับการเรียกใช้ (vary on) rootvg ที่ไม่ใช่ขององค์ประกอบ; การดำเนินงานนี้มีความเสี่ยง LVM must force the activation because the operating system cannot be started unless rootvg is activated. กล่าวอีกอย่างคือ LVM ทำความพยายามครั้งสุดท้าย เพื่อเรียกใช้ (vary on) rootvg ที่ไม่ใช่ขององค์ประกอบเฉพาะถ้าสามารถเข้าถึงได้เพียงดิสก์เดียวเท่านั้น

หลักการที่เกี่ยวข้อง:

“การมีอยู่สูงในกรณีของความล้มเหลวของดิสก์” ในหน้า 440

เมธอดหลักที่ใช้เพื่อป้องกันความล้มเหลวของดิสก์เกี่ยวข้องกับ ค่าติดตั้งคอนฟิกูเรชันโลจิคัลวอลุ่ม เช่น การมิร์เรอร์

“การมีอยู่สูงในกรณีของความล้มเหลวอะแดปเตอร์หรือแหล่งจ่ายไฟ” ในหน้า 441

เมื่อต้องการป้องกันความล้มเหลวอะแดปเตอร์หรือแหล่งจ่ายไฟ ขึ้นอยู่กับ ความต้องการของคุณ ให้ทำหนึ่งหรือหลายอย่างต่อไปนี้

“การนำนโยบายกลุ่มวอลุ่มไปใช้” ในหน้า 453

หลังจากคุณตัดสินใจเลือกนโยบายกลุ่มวอลุ่มซึ่งคุณต้องการใช้แล้ว ให้วิเคราะห์คอนฟิกูเรชันปัจจุบันโดยพิมพ์คำสั่ง `lspv` บนบรรทัดคำสั่ง

“องค์ประกอบ” ในหน้า 385

องค์ประกอบเป็นกลไกหนึ่งที่ LVM ใช้เพื่อให้อุ่นใจว่า กลุ่มวอลุ่มพร้อมใช้งานและมีข้อมูลล่าสุด

การกำหนดคอนฟิก Logical Volume Manager

Logical Volume Manager (LVM) มีการติดตั้งพร้อมกับระบบปฏิบัติการ พื้นฐานและต้องการคอนฟิกูเรชันเพิ่มเติม อย่างไรก็ตาม ต้องกำหนดคอนฟิกดิสก์ และกำหนดเป็นฟิสิคัลวอลุ่มก่อน LVM สามารถใช้ดิสก์นั้นได้

งานที่เกี่ยวข้อง:

“การกำหนดโลจิคัลวอลุ่มแบบดิบสำหรับแอ็พพลิเคชัน” ในหน้า 433

โลจิคัลวอลุ่มแบบดิบ เป็นพื้นที่ของพื้นที่ว่างดิสก์ฟิสิคัลและ โลจิคัลซึ่งอยู่ภายใต้การควบคุมโดยตรงของแอ็พพลิเคชัน เช่น ฐานข้อมูลหรือพาร์ติชัน แทนภายใต้การควบคุมโดยตรงของ ระบบปฏิบัติการหรือระบบไฟล์

คำสั่งการบำรุงรักษา LVM และพาด่วน:

ภารกิจง่ายที่สุดซึ่งคุณอาจต้องการเมื่อบำรุงรักษาเอนทิตีที่ LVM ควบคุม (ฟิลิคัลและโลจิคัลวอลุ่ม กลุ่มวอลุ่ม และระบบไฟล์) คือการจัดกลุ่มภายในตารางต่อไปนี้

ตารางที่ 62. การจัดการโลจิคัลวอลุ่มและภารกิจหน่วยเก็บ

ภารกิจ	พาด่วน SMIT	คำสั่งหรือไฟล์
เรียกใช้กลุ่มวอลุ่ม	smit varyonvg	
เพิ่มดิสก์คงที่ซึ่งไม่มีข้อมูลลงในกลุ่มวอลุ่มที่มีอยู่	smit extendvg	
เพิ่มดิสก์คงที่ซึ่งไม่มีข้อมูลลงในกลุ่มวอลุ่มใหม่	smit mkvg	
เพิ่มโลจิคัลวอลุ่ม ^{Note 1}	smit mklv	
เพิ่มกลุ่มวอลุ่ม	smit mkvg	
เพิ่มและเรียกใช้กลุ่มวอลุ่มใหม่	smit mkvg	
เปลี่ยนโลจิคัลวอลุ่มเพื่อใช้การจัดสรรข้อมูล	smit chlv 1	
เปลี่ยนชื่อของกลุ่มวอลุ่ม ^{Note 2}	1. smit varyoffvg 2. smit exportvg 3. smit importvg 4. smit mountfs	1. varyoffvg OldVGName 2. exportvg OldVGName 3. importvg NewVGName 4. mount all
เปลี่ยนกลุ่มวอลุ่มเพื่อใช้การเรียกใช้อัตโนมัติ	smit chvg	
เปลี่ยนหรือตั้งค่านโยบายโลจิคัลวอลุ่ม	smit chlv 1	
คัดลอกโลจิคัลวอลุ่มไปยังโลจิคัลวอลุ่มใหม่ ^{Note 3}	smit cplv	
คัดลอกโลจิคัลวอลุ่มไปยังโลจิคัลวอลุ่มที่มีอยู่ซึ่งมีขนาดเดียวกัน ^{Attm 1}	smit cplv	
คัดลอกโลจิคัลวอลุ่มไปยังโลจิคัลวอลุ่มที่มีอยู่ซึ่งมีขนาดเล็กกว่า ^{Attm 1 Note 3}	อย่าใช้ SMIT ^{Attm 2}	1. สร้างโลจิคัลวอลุ่ม ตัวอย่างเช่น: mklv -y hdiskN vg00 4 2. สร้างระบบไฟล์ใหม่บนโลจิคัลวอลุ่มใหม่ ตัวอย่างเช่น: crfs -v jfs -d hdiskN -m /doc -A yes 3. ติดตั้งระบบไฟล์ ตัวอย่างเช่น: mount /doc 4. สร้างไดเรกทอรีที่จุดต่อใหม่ ตัวอย่างเช่น: mkdir /doc/options 5. โอนย้ายระบบไฟล์จากโลจิคัลวอลุ่มต้นทางไปยังปลายทาง ตัวอย่าง เช่น: cp -R /usr/adam/oldoptions/* \ /doc/options
คัดลอกโลจิคัลวอลุ่มไปยังโลจิคัลวอลุ่มที่มีอยู่ซึ่งมีขนาดใหญ่กว่า ^{Attm 1}	smit cplv	
ยกเลิกการเรียกใช้กลุ่มวอลุ่ม	smit varyoffvg	
เปิดใช้งานการตรวจสอบการเขียนและเปลี่ยนนโยบายการกำหนดตารางเวลา	smit chlv 1	

ตารางที่ 62. การจัดการโลจิกัลวอลุ่มและภารกิจหน่วยเก็บ (ต่อ)

ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
เพิ่มขนาดสูงสุดของโลจิกัลวอลุ่ม	smit chlv1	
เพิ่มขนาดของโลจิกัลวอลุ่ม	smit extendlv	
แสดงรายการโลจิกัลวอลุ่มทั้งหมดโดยเรียงตามกลุ่มวอลุ่ม	smit lslv2	
แสดงรายการฟิลิควอลุ่มทั้งหมดในระบบ	smit lspv2	
แสดงรายการกลุ่มวอลุ่มทั้งหมด	smit lsvg2	
แสดงรายการสถานะ โลจิกัลวอลุ่ม หรือพาร์ติชันของฟิลิควอลุ่ม	smit lspv	
แสดงรายการเนื้อหาของกลุ่มวอลุ่ม	smit lsvg1	
แสดงรายการสถานะของโลจิกัลวอลุ่มหรือการแม็พ	smit lslv	
มีร์เรอร์โลจิกัลวอลุ่มที่มีหรือไม่มีการจัดสรรข้อมูล	smit mklvcopy	
ปิดดิสก์ที่ถอดออกได้	smit offdsk	มีอยู่สำหรับคุณลักษณะ hot-removability เท่านั้น
เปิดดิสก์ที่ถอดออกได้	smit ondsk	มีอยู่สำหรับคุณลักษณะ hot-removability เท่านั้น
ลบการมีร์เรอร์ออกจากกลุ่มวอลุ่ม	smit unmirrorvg	
ลบกลุ่มวอลุ่ม	smit reducevg2	
จัดระเบียบกลุ่มวอลุ่มอีกครั้ง	smit reorgvg	
ยกเลิกกำหนดคอนฟิกและปิดดิสก์	smit rmvdsk 1 หรือ smit rmvdsk จากนั้น smit opendoor	

ข้อควรสนใจ:

1. การใช้ไพรซีเดอร์นี้เพื่อคัดลอกไปยังโลจิกัลวอลุ่มที่มีอยู่จะเขียนทับข้อมูลใดๆ บนวอลุ่มนั้นโดยไม่ต้องร้องขอการยืนยันจากผู้ใช้
2. อย่าใช้ไพรซีเดอร์ SMIT หรือคำสั่ง cplv เพื่อ คัดลอกโลจิกัลวอลุ่มที่ใหญ่กว่าไปยังโลจิกัลวอลุ่มที่เล็กกว่า การทำเช่นนั้นจะทำให้ระบบไฟล์เสียหายเนื่องจากข้อมูลบางส่วน (รวมถึงซูเปอร์บล็อก) ไม่ถูกคัดลอก ไปยังโลจิกัลวอลุ่มที่เล็กกว่า

หมายเหตุ:

1. หลังจากคุณสร้างโลจิกัลวอลุ่ม สถานะจะถูกปิดเนื่องจากไม่มีโครงสร้าง LVM ใดกำลังใช้โลจิกัลวอลุ่มนั้น สถานะจะยังคงถูกปิดจนกว่าติดตั้ง ระบบไฟล์บนโลจิกัลวอลุ่มแล้ว หรือเปิดโลจิกัลวอลุ่ม สำหรับ I/O ดิบ
2. คุณไม่สามารถเปลี่ยนชื่อของ อิมพอร์ต หรือเอ็กซ์พอร์ต rootvg
3. คุณต้องมีหน่วยเก็บการเข้าถึงโดยตรงมากเพียงพอเพื่อทำซ้ำโลจิกัลวอลุ่ม เฉพาะ

งานที่เกี่ยวข้อง:

“การกำหนดโลจิกัลวอลุ่มแบบดิบสำหรับแอ็พพลิเคชัน” ในหน้า 433

โลจิกัลวอลุ่มแบบดิบ เป็นพื้นที่ของพื้นที่ว่างดิสก์ฟิลิควอลุ่มและ โลจิกัลวอลุ่มซึ่งอยู่ภายใต้การควบคุมโดยตรงของแอ็พพลิเคชัน เช่นฐานข้อมูลหรือพาร์ติชัน แทนภายใต้การควบคุมโดยตรงของ ระบบปฏิบัติการหรือระบบไฟล์

การเพิ่มดิสก์ในขณะที่ระบบยังคงเปิดอยู่:

โพรซีเดอร์ท่อนี้ใช้อธิบายวิธีการเปิดและกำหนดคอนฟิก ดิสก์โดยใช้คุณลักษณะ hot-removability ซึ่งอนุญาตให้คุณเพิ่มดิสก์ได้โดยไม่ต้อง ปิดระบบ

คุณสามารถเพิ่มดิสก์สำหรับหน่วยเก็บเพิ่มเติมหรือเพื่อแก้ไขความล้มเหลว ดิสก์ คุณลักษณะนี้มีอยู่บนบางระบบเท่านั้น

1. ติดตั้งดิสก์ในสล็อตที่ว่างของตู้ สำหรับข้อมูลรายละเอียด เกี่ยวกับโพรซีเดอร์ทดติดตั้ง ให้ดูที่คู่มือเซอร์วิสของเครื่องของคุณ
2. เปิดดิสก์ใหม่โดยพิมพ์พารามิเตอร์ต่อไปนี้บน บรรทัดคำสั่ง:

```
smit ondisk
```

ณ จุดนี้ ดิสก์ถูกเพิ่มลงในระบบแล้วแต่ยังไม่สามารถ ใช้ได้ สิ่งที่คุณจะทำถัดไปขึ้นอยู่กับว่าดิสก์ใหม่มีข้อมูลหรือไม่

- ถาดิสก์ไม่มีข้อมูล ให้เพิ่มดิสก์เป็นฟิสิคัลวอลุ่มลงในกลุ่มวอลุ่ม โดยใช้วิธีใดอย่างหนึ่งต่อไปนี้:

- เมื่อต้องการเพิ่มดิสก์ลงในกลุ่มวอลุ่มที่มีอยู่ ให้พิมพ์พารามิเตอร์ต่อไปนี้บนบรรทัดคำสั่ง:

```
smit extendvg
```

- เมื่อต้องการเพิ่มดิสก์ลงในกลุ่มวอลุ่มใหม่ ให้พิมพ์พารามิเตอร์ต่อไปนี้บนบรรทัดคำสั่ง:

```
smit mkvg
```

- หากดิสก์มีข้อมูล ให้อิมพอร์ตข้อมูล

หลักการที่เกี่ยวข้อง:

“การนำนโยบายกลุ่มวอลุ่มไปใช้” ในหน้า 453

หลังจากคุณตัดสินใจเลือกนโยบายกลุ่มวอลุ่มซึ่งคุณต้องการใช้แล้ว ให้วิเคราะห์คอนฟิกูเรชันปัจจุบันโดยพิมพ์คำสั่ง `lspv` บนบรรทัดคำสั่ง

งานที่เกี่ยวข้อง:

“การอิมพอร์ตหรือการเอ็กซ์พอร์ตกลุ่มวอลุ่ม” ในหน้า 395

ตารางต่อไปนี้อธิบายวิธีการใช้การอิมพอร์ตและการเอ็กซ์พอร์ตเพื่อย้าย กลุ่มวอลุ่มที่ผู้ใช้กำหนดจากระบบหนึ่งไปยังอีกระบบหนึ่ง (ไม่สามารถอิมพอร์ตหรือเอ็กซ์พอร์ต กลุ่มวอลุ่ม `rootvg`)

“การเอาดิสก์ที่มีข้อมูลออก” ในหน้า 435

ใช้โพรซีเดอร์ท่อนี้เพื่อเอาดิสก์ที่มีข้อมูลออก โดยไม่ต้องปิดระบบ

“การเอาดิสก์ที่ไม่มีข้อมูลออก” ในหน้า 436

โพรซีเดอร์ท่อนี้ใช้อธิบายวิธีการเอาดิสก์ที่ไม่มีข้อมูล หรือไม่มีข้อมูลซึ่งคุณต้องการเก็บไว้

การเปลี่ยนชื่อของโลจิคัลวอลุ่ม:

โพรซีเดอร์ท่อนี้ใช้อธิบายวิธีการเปลี่ยนชื่อโลจิคัลวอลุ่ม โดยไม่สูญเสียข้อมูลบนโลจิคัลวอลุ่มนั้น

ในตัวอย่างต่อไปนี้ ชื่อโลจิคัลวอลุ่มถูกเปลี่ยนจาก `lv00` เป็น `lv33`

1. ถอนติดตั้งระบบไฟล์ทั้งหมดที่เชื่อมโยงกับโลจิคัล วอลุ่ม โดยพิมพ์:

```
umount /FSname
```

โดยที่ `FSname` คือชื่อเต็มของระบบไฟล์

หมายเหตุ:

- a. คำสั่ง **umount** จะล้มเหลวถ้าระบบไฟล์ซึ่งคุณกำลังพยายาม ถอนติดตั้งถูกใช้งานอยู่ในปัจจุบัน คำสั่ง **umount** ดำเนินการเฉพาะถ้าไม่มีไฟล์ของระบบไฟล์เปิดอยู่และไม่มีไดเรกทอรีปัจจุบันของผู้ใช้ใดอยู่บนอุปกรณ์นั้น
- b. อีกชื่อหนึ่งสำหรับคำสั่ง **umount** คือ **umount** ชื่อทั้งสองนี้สามารถใช้ทดแทนกันได้

2. เปลี่ยนชื่อโลจิคัลวอลุ่ม โดยพิมพ์:

```
chlv -n NewLVname OldLVname
```

โดยที่แฟล็ก **-n** ระบุชื่อโลจิคัลวอลุ่มใหม่ (**NewLVname**) และ **OldLVname** เป็น ชื่อที่คุณต้องการเปลี่ยน ตัวอย่างเช่น:

```
chlv -n lv33 lv00
```

หมายเหตุ: หาก คุณเปลี่ยนชื่อบันทึก JFS หรือ JFS2 ระบบจะพร้อมทำให้คุณรันคำสั่ง **chfs** บนระบบไฟล์ทั้งหมดที่ใช้ อุปกรณ์บันทึกซึ่งเปลี่ยนชื่อ

3. ติดตั้งระบบไฟล์ซึ่งคุณถอนติดตั้งในขั้นตอน 1 ในหน้า 391 ใหม่โดยพิมพ์:

```
mount /test1
```

ณ จุดนี้ โลจิคัลวอลุ่มถูกเปลี่ยนชื่อแล้วและพร้อม ใช้งานได้

การคัดลอกโลจิคัลวอลุ่มไปยังฟิสิคัลวอลุ่มอื่น:

ขึ้นอยู่กับความต้องการของคุณ มีหลายวิธีในการคัดลอกโลจิคัลวอลุ่ม ไปยังฟิสิคัลวอลุ่มอื่นในขณะที่คงบูรณภาพระบบไฟล์ไว้

มีหลายเมธอดสำหรับการคัดลอกโลจิคัลวอลุ่มหรือ JFS ไปยัง ฟิสิคัลวอลุ่มอื่น เลือกเมธอดที่เหมาะสมกับวัตถุประสงค์ของคุณมากที่สุด

การคัดลอกโลจิคัลวอลุ่ม:

เมธอดที่ง่ายที่สุดคือการใช้คำสั่ง **cplv** เพื่อคัดลอกโลจิคัลวอลุ่มดั้งเดิมและสร้างโลจิคัลวอลุ่มใหม่บน ฟิสิคัลวอลุ่มปลายทาง

1. หยุดการใช้โลจิคัลวอลุ่ม ถอนติดตั้งระบบไฟล์ ถ้าใช้ได้ และหยุดแอ็พพลิเคชันใดๆ ที่เข้าถึงโลจิคัลวอลุ่ม
2. เลือกฟิสิคัลวอลุ่มที่มีความสามารถรองรับข้อมูลทั้งหมดในโลจิคัลวอลุ่มดั้งเดิมได้

ข้อควรสนใจ: หากคุณคัดลอกจากโลจิคัลวอลุ่มที่ใหญ่กว่าและมีข้อมูล ไปยังโลจิคัลวอลุ่มที่เล็กกว่า คุณอาจทำให้ระบบไฟล์เสียหายได้เนื่องจากข้อมูลบางส่วน (รวมถึงซูเปอร์บล็อก) อาจสูญหายไป

3. คัดลอกโลจิคัลวอลุ่มดั้งเดิม (ในตัวอย่างนี้ มีชื่อว่า **lv00**) และสร้างโลจิคัลวอลุ่มใหม่ โดยใช้คำสั่งต่อไปนี้:

หมายเหตุ: คำสั่ง **cplv** ต่อไปนี้จะล้มเหลวถ้าคำสั่งสร้าง โลจิคัลวอลุ่มใหม่และกลุ่มวอลุ่มถูก **varied on** ในโหมดการเกิดขึ้นพร้อมกัน

```
cplv lv00
```

4. ติดตั้งระบบไฟล์ ถ้าใช้ได้ และรีสตาร์ทแอ็พพลิเคชัน เพื่อเริ่มต้นการใช้โลจิคัลวอลุ่ม

ณ จุดนี้ สามารถใช้สำเนาของโลจิคัลวอลุ่มได้แล้ว

การคัดลอกโลจิคัลวอลุ่มในขณะที่โลจิคัลวอลุ่มดั้งเดิมยังคงใช้ได้:

หากสภาวะแวดล้อมของคุณต้องการใช้โลจิคัลวอลุ่มดั้งเดิมต่อไป คุณสามารถใช้คำสั่ง **splitvcopy** เพื่อคัดลอกเนื้อหา ดังแสดงในตัวอย่างต่อไปนี้

1. มีร์เรอร์โลจิคัลวอลุ่ม โดยใช้พาร์ต่วน SMIT ต่อไปนี้:

smit mklvcopy

- หยุดการใช้โลจิคัลวอลุ่ม ถอนติดตั้งระบบไฟล์ ถ้าใช้ได้ และหยุดหรือวางแอฟพลิเคชันใดๆ ที่เข้าถึงโลจิคัลวอลุ่มไว้ในโหมดหยุด

ข้อควรสนใจ: ขั้นตอนถัดไปใช้คำสั่ง `splitlvcopy` ปิดโลจิคัลวอลุ่มเสมอ ก่อนแบ่งและถอนติดตั้งระบบไฟล์ที่มีใดๆ ก่อนการใช้คำสั่งนี้ การแบ่งโลจิคัลวอลุ่มเปิดอาจทำให้ระบบไฟล์ของคุณ เสียหายและส่งผลให้สูญเสียความสอดคล้องกันระหว่างโลจิคัลวอลุ่มดั้งเดิม และสำเนา ถ้ามีการเข้าถึงโลจิคัลวอลุ่มพร้อมกัน โดยหลายโปรเซส

- ด้วยสิทธิการใช้งานราก คัดลอกโลจิคัลวอลุ่มดั้งเดิม (`oldlv`) ไปยังโลจิคัลวอลุ่มใหม่ (`newlv`) โดยใช้คำสั่งต่อไปนี้:

```
splitlvcopy -y newlv oldlv
```

แฟล็ก `-y` กำหนด ชื่อโลจิคัลวอลุ่มใหม่ หากวอลุ่ม `oldlv` ไม่มี บล็อกควบคุมโลจิคัลวอลุ่ม คำสั่ง `splitlvcopy` จะสำเร็จ แต่สร้างข้อความว่า มีการสร้างวอลุ่ม `newlv` ขึ้นแล้ว โดยไม่มีบล็อกควบคุมโลจิคัลวอลุ่ม

- ติดตั้งระบบไฟล์ ถ้าใช้ได้ และรีสตาร์ทแอฟพลิเคชัน เพื่อเริ่มต้นการใช้โลจิคัลวอลุ่ม

ณ จุดนี้ สามารถใช้สำเนาของโลจิคัลวอลุ่มได้แล้ว

การคัดลอกโลจิคัลวอลุ่มแบบดิบไปยังฟิสิคัลวอลุ่มอื่น:

เมื่อต้องการคัดลอกโลจิคัลวอลุ่มแบบดิบไปยังฟิสิคัลวอลุ่มอื่น ให้ปฏิบัติตาม ขั้นตอนต่อไปนี้

- สร้างสำเนาที่มึร์เรอร์ของโลจิคัลวอลุ่มบนฟิสิคัลวอลุ่มใหม่ ในกลุ่มวอลุ่มโดยใช้คำสั่งต่อไปนี้:

```
mklvcopy LogVol_name 2 new_PhysVol_name
```

- ซิงโครไนซ์พาร์ติชันในสำเนามึร์เรอร์ใหม่โดยใช้คำสั่ง ต่อไปนี้:

```
syncvg -l LogVol_name
```

- ลบสำเนาของโลจิคัลวอลุ่มออกจากฟิสิคัลวอลุ่ม โดยใช้คำสั่งต่อไปนี้:

```
rmlvcopy LogVol_name 1 old_PhysVol_name
```

ณ จุดนี้ สามารถใช้สำเนาของโลจิคัลวอลุ่มแบบดิบได้แล้ว

การสร้างบันทึกระบบไฟล์บนดิสก์เฉพาะงานสำหรับกลุ่มวอลุ่มที่ผู้ใช้กำหนด:

บันทึกระบบไฟล์ JFS หรือ JFS2 คือรายการที่จัดรูปแบบของเร็กคอร์ด อูทกรรรมระบบไฟล์ ล็อกช่วยให้อุ่นใจในบุรณภาพของระบบไฟล์ (แต่ไม่จำเป็นสำหรับบุรณภาพของข้อมูล) ในกรณีที่ระบบ ตาวนก่อนที่อูทกรรรมจะเสร็จสมบูรณ์

ดิสก์เฉพาะงานมีการสร้างขึ้นบน `hd8` สำหรับ `rootvg` เมื่อติดตั้ง ระบบ โพรซีเดอร์ทต่อไปนี้ช่วยคุณสร้างล็อก JFS บนดิสก์แยกต่างหาก สำหรับกลุ่มวอลุ่มอื่น เมื่อคุณสร้างล็อก JFS2 โพรซีเดอร์ทต้องการ การเปลี่ยนแปลงต่อไปนี้:

- ชนิดอุปกรณ์บันทึกเป็น `jfs2log`
- คำสั่ง `logform` ต้องการอ็อพชัน `-V jfs2` เพื่อระบุอุปกรณ์ล็อก JFS2
- คำสั่ง `crfs` ต้องระบุ `jfs2` แทน `jfs`

หมายเหตุ: ไม่มีข้อกำหนดสำหรับล็อก JFS2 ที่จะอยู่บนดิสก์แยกต่างหากเป็นระบบไฟล์ มีข้อกำหนดเพียง อุปกรณ์ล็อกต้องอยู่บนกลุ่มวอลุ่มเดียวกันกับ ระบบไฟล์ ในโพรซีเดอร์ทนี้ ล็อก JFS2 ต้องอยู่บนดิสก์แยกต่างหาก เพื่อการปรับปรุงประสิทธิภาพ

การสร้างไฟล์ล็อกของระบบไฟล์สำหรับกลุ่มวอลุ่มแบบผู้ใช้กำหนดสามารถเพิ่มประสิทธิภาพได้ภายใต้เงื่อนไขเฉพาะ เช่น ถ้าคุณมีเซิร์ฟเวอร์ NFS และคุณต้องการ ธุรกรรมสำหรับเซิร์ฟเวอร์นี้ถูกประมวลผลโดยไม่มีโปรเซสอื่น มาแย่ง

คุณสามารถใช้โปรแกรมต่อไปนี้ ซึ่งสร้าง หนึ่งกลุ่มวอลุ่ม (fsvg1) ที่มีสองพิลิคัลวอลุ่ม (hdisk1 และ hdisk2) ระบบไฟล์อยู่บน hdisk2 (ระบบไฟล์ 256 MB ที่เมาท์อยู่ที่ /u/myfs) และล็อกอยู่บน hdisk1 โดยค่าดีฟอลต์ขนาดบันทึก JFS คือ 4 MB คุณสามารถวางโปรแกรมที่ใช้งานน้อย เช่น /blv ไว้บนวอลุ่มพิลิคัลเดียวกัน กับล็อกโดยไม่มีผลกระทบต่อประสิทธิภาพการใช้งาน

เมื่อต้องการสร้างล็อก JFS สำหรับกลุ่มวอลุ่มแบบผู้ใช้กำหนดโดยใช้ SMIT และอินเตอร์เฟซ บรรทัดรับคำสั่ง ให้ทำตามขั้นตอนต่อไปนี้:

1. เพิ่มกลุ่มวอลุ่มใหม่ (ในตัวอย่างนี้คือ fsvg1) โดยใช้ พาด่วน SMIT:

```
smit mkvg
```

2. เพิ่มวอลุ่มโลจิคัลใหม่ลงในกลุ่มวอลุ่มนี้โดยใช้พาด่วน SMIT:

```
smit mklv
```

3. บนหน้าจอ **เพิ่มโลจิคัลวอลุ่ม** ให้เพิ่มข้อมูลของคุณลงในฟิลด์ต่อไปนี้ ตัวอย่างเช่น:

Logical Volumes NAME	fsvg1log
Number of LOGICAL PARTITIONS	1
PHYSICAL VOLUME names	hdisk1
Logical volume TYPE	jfslog
POSITION on Physical Volume	center

4. หลังจากคุณตั้งค่าฟิลด์แล้ว กด Enter เพื่อยอมรับการเปลี่ยนแปลงของคุณและ ออกจาก SMIT

5. พิมพ์คำสั่งต่อไปนี้บนบรรทัดรับคำสั่ง:

```
/usr/sbin/logform /dev/fsvg1log
```

6. เมื่อคุณได้รับพร้อมต์ต่อไปนี้ให้พิมพ์ y และกด Enter:

```
Destroy /dev/fsvg1log
```

ถึงแม้มีค่าอยู่ในพร้อมต์นี้ แต่ไม่มีสิ่งใดถูกทำลาย เมื่อคุณ ตอบ y ในพร้อมต์นี้ ระบบจะจัดรูปแบบโลจิคัลวอลุ่ม สำหรับบันทึก JFS เพื่อให้โลจิคัลวอลุ่มสามารถบันทึกธุรกรรมระบบไฟล์ได้

7. เพิ่มวอลุ่มโลจิคัลอื่นโดยใช้ พาด่วน SMIT ต่อไปนี้:

```
smit mklv
```

8. พิมพ์ชื่อของกลุ่มวอลุ่มเดียวกันกับที่คุณใช้ในขั้นตอนที่ 2 (fsvg1 ในตัวอย่างนี้) ในหน้าจอ **โลจิคัลวอลุ่ม** ให้เพิ่มข้อมูลของคุณลงในฟิลด์ต่อไปนี้ โปรดจำว่าต้องกำหนดพิลิคัลวอลุ่มซึ่งแตกต่างจากที่คุณทำในขั้นตอนที่ 3 สำหรับโลจิคัลวอลุ่มนี้ ตัวอย่างเช่น:

Logical Volumes NAME	fs1v1
Number of LOGICAL PARTITIONS	64
PHYSICAL VOLUME names	hdisk2
Logical volume TYPE	jfs

หลังจากคุณตั้งค่าไฟล์แล้ว กด Enter เพื่อยอมรับการเปลี่ยนแปลงของคุณและ ออกจาก SMIT

9. เพิ่มระบบไฟล์ลงในวอลุ่มโลจิคัลใหม่ กำหนดบล็อก และเมทา์ระบบไฟล์ใหม่ โดยใช้ลำดับคำสั่งต่อไปนี้:

```
crfs -v jfs -d LogVolName -m FileSysName -a logname=FSLogPath
```

```
mount FileSysName
```

โดยที่ *LogVolName* คือ ชื่อของโลจิคัลวอลุ่มซึ่งคุณสร้างขึ้นในขั้นตอนที่ 2; *FileSysName* คือ ชื่อของระบบไฟล์ซึ่งคุณต้องการติดตั้งบนโลจิคัลวอลุ่มนี้; และ *FSLogPath* คือ ชื่อของโลจิคัลวอลุ่มซึ่งคุณสร้างขึ้นในขั้นตอนที่ 2 ตัวอย่างเช่น:

```
crfs -v jfs -d fslv1 -m /u/myfs -a logname=/dev/fsvg1log
mount /u/myfs
```

10. เมื่อต้องการตรวจสอบว่าคุณตั้งค่าระบบไฟล์และบล็อกถูกต้องหรือไม่ พิมพ์ คำสั่งต่อไปนี้ (แทนที่ด้วยชื่อกลุ่มวอลุ่มของคุณ):

```
lsvg -l fsvg1
```

เอาต์พุต แสดงทั้งวอลุ่มโลจิคัลที่คุณสร้างขึ้น พร้อมชนิดของระบบไฟล์ ตามตัวอย่างต่อไปนี้:

LV NAME	TYPE	...
/dev/fsvg1log	jfslog	...
fslv1	jfs	...

คุณสร้างกลุ่มวอลุ่มที่มีวอลุ่มโลจิคัลอย่างน้อยสองตัว บนวอลุ่มฟิสิคัลที่แยกกัน และหนึ่งในวอลุ่มโลจิคัลเหล่านั้น มีบล็อกของระบบไฟล์

หมายเหตุ: เมื่อต้องการจัดเตรียม การทำซ้ำ คุณสามารถมีมिरเรอร์บนระดับวอลุ่มโลจิคัล สำหรับอุปกรณ์บล็อก JFS2 อย่างไรก็ตาม การจัดเตรียมมिरเรอร์ไม่ใช่แนวทางปฏิบัติ ทั่วไปและไม่จำเป็น

การอิมพอร์ตหรือการเอ็กซ์พอร์ตกลุ่มวอลุ่ม:

ตารางต่อไปนี้อธิบายวิธีการใช้การอิมพอร์ตและการเอ็กซ์พอร์ตเพื่อย้าย กลุ่มวอลุ่มที่ผู้ใช้กำหนดจากระบบหนึ่งไปยังอีกระบบหนึ่ง (ไม่สามารถอิมพอร์ตหรือเอ็กซ์พอร์ต กลุ่มวอลุ่ม rootvg)

โพรซีเจอร์เอ็กซ์พอร์ตลบนโยบายของกลุ่มวอลุ่ม ออกจากระบบ โพรซีเจอร์อิมพอร์ตใช้เพื่อแนะนำกลุ่มวอลุ่มใน ระบบใหม่

คุณยังสามารถใช้โพรซีเจอร์อิมพอร์ตเพื่อแนะนำ กลุ่มวอลุ่มในระบบอีกครั้งเมื่อเคยเชื่อมโยงกลุ่มวอลุ่มนั้นกับระบบแล้วแต่ ถูกเอ็กซ์พอร์ต คุณยังสามารถใช้อิมพอร์ตและเอ็กซ์พอร์ตเพื่อเพิ่มฟิสิคัลวอลุ่มที่มี ข้อมูลลงในกลุ่มวอลุ่ม โดยวางดิสก์ที่จะเพิ่มไว้ในกลุ่มวอลุ่มของ ตนเอง

ข้อควรสนใจ: คำสั่ง `importvg` เปลี่ยนชื่อของโลจิคัลวอลุ่มที่อิมพอร์ตถ้าโลจิคัลวอลุ่มชื่อนั้นมีอยู่แล้ว บนระบบใหม่ หากคำสั่ง `importvg` ต้องเปลี่ยนชื่อ โลจิคัลวอลุ่ม คำสั่งจะพิมพ์ข้อความแสดงข้อผิดพลาดไปยังข้อผิดพลาดมาตรฐาน เมื่อไม่มีข้อขัดแย้ง คำสั่ง `importvg` ยังสร้าง จุดต่อไฟล์และรายการในไฟล์ `/etc/filesystems` ด้วย

ภารกิจอิมพอร์ตและเอ็กซ์พอร์ต กลุ่มวอลุ่ม		
ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
อิมพอร์ตกลุ่มวอลุ่ม	smit importvg	
เอ็กซ์พอร์ตกลุ่มวอลุ่ม	1. ถอนติดตั้งระบบไฟล์บนโลจิคัลวอลุ่มในกลุ่มวอลุ่ม: smit umntdsk 2. Vary off กลุ่มวอลุ่ม: smit varyoffvg 3. เอ็กซ์พอร์ตกลุ่มวอลุ่ม: smit exportvg	

ข้อควรสนใจ: กลุ่มวอลุ่มที่มีวอลุ่มพื้นที่การสลับหน้าอยู่จะไม่สามารถถูกเอ็กซ์พอร์ต ขณะที่พื้นที่การสลับหน้าใช้งานอยู่ ก่อนการเอ็กซ์พอร์ต กลุ่มวอลุ่มที่มีพื้นที่การเลื่อนหน้าที่ใช้งานอยู่ ตรวจสอบให้แน่ใจว่าพื้นที่การเลื่อน หน้าไม่ได้ใช้งานอยู่โดยอัตโนมัติที่การเริ่มต้นระบบ โดยพิมพ์ คำสั่งต่อไปนี้:

```
chps -a n paging_space name
```

จากนั้น รีบูตระบบเพื่อให้พื้นที่การเลื่อนหน้าไม่ได้ใช้งานอยู่

งานที่เกี่ยวข้อง:

“การเพิ่มดิสก์ในขณะที่ระบบยังคงเปิดอยู่” ในหน้า 391

โพรซีเจอร์ต่อไปนี้อธิบายวิธีการเปิดและกำหนดคอนฟิก ดิสก์โดยใช้คุณลักษณะ hot-removability ซึ่งอนุญาตให้คุณเพิ่มดิสก์ได้โดยไม่ต้อง ปิดระบบ

“การเอาดิสก์ที่มีข้อมูลออก” ในหน้า 435

ใช้โพรซีเจอร์นี้เพื่อเอาดิสก์ที่มีข้อมูลออก โดยไม่ต้องปิดระบบ

การย้ายเนื้อหาของฟิสิคัลวอลุ่ม:

เมื่อต้องการย้ายพาร์ติชันฟิสิคัลที่เป็นของโลจิคัลวอลุ่มที่ระบุตั้งแต่หนึ่ง รายการขึ้นไปจากฟิสิคัลวอลุ่มหนึ่งไปยังฟิสิคัลวอลุ่มอื่นหนึ่งหรือหลายวอลุ่ม ในกลุ่มวอลุ่ม ให้ใช้คำแนะนำต่อไปนี้ คุณยังสามารถใช้โพรซีเจอร์นี้ เพื่อย้ายข้อมูลจากดิสก์ที่ล้มเหลว ก่อนเปลี่ยนหรือซ่อมแซม ดิสก์ที่ล้มเหลว โพรซีเจอร์นี้สามารถใช้บนฟิสิคัลวอลุ่มใน กลุ่มวอลุ่มรากหรือกลุ่มวอลุ่มที่ผู้ใช้ กำหนด ได้อย่างใดอย่างหนึ่ง

ข้อควรสนใจ: เมื่อ บุตโลจิคัลวอลุ่มถูกย้ายจากฟิสิคัลวอลุ่ม ต้องล้างข้อมูลบูตเรกคอร์ด บนซอร์ส มิฉะนั้น อาจทำให้ระบบ hang ได้ เมื่อคุณดำเนินการ คำสั่ง **bosboot** คุณยังต้องดำเนินการคำสั่ง **chpv -c** ซึ่งอธิบายในขั้นตอนที่ 4 ของ โพรซีเจอร์ต่อไปนี้ด้วย

1. ถ้าคุณต้องการย้ายข้อมูลไปยังดิสก์ใหม่ ให้ทำขั้นตอน ต่อไปนี้ มิฉะนั้น ทำขั้นตอนที่ 2 ต่อไป

a. ตรวจสอบว่าดิสก์เป็นที่รู้จักของระบบและมีอยู่โดยพิมพ์:

```
lsdev -Cc disk
```

เอาต์พุต คล้ายกับตัวอย่างต่อไปนี้:

```
hdisk0 Available 10-60-00-8,0 16 Bit LVD SCSI Disk Drive
hdisk1 Available 10-60-00-9,0 16 Bit LVD SCSI Disk Drive
hdisk2 Available 10-60-00-11,0 16 Bit LVD SCSI Disk Drive
```

b. หากดิสก์แสดงรายการขึ้นและอยู่ในสถานะพร้อมใช้งาน ให้ตรวจสอบว่าดิสก์ไม่ได้ เป็นของกลุ่มวอลุ่มอื่น โดยพิมพ์:

```
lspv
```

เอาต์พุต มีลักษณะคล้ายตัวอย่างต่อไปนี้:

```
hdisk0      0004234583aa7879      rootvg      active
hdisk1      00042345e05603c1      none        active
hdisk2      00083772caa7896e      imagesvg    active
```

ในตัวอย่างสามารถใช้ `hdisk1` เป็นดิสก์ปลายทางเนื่องจากฟิลด์ที่สามแสดงว่า `hdisk1` ไม่ได้ถูกใช้โดยกลุ่มวอลุ่ม หากดิสก์ใหม่ไม่แสดงขึ้นหรือไม่พร้อมใช้งาน คุณต้องกำหนดคอนฟิกดิสก์ หรือหน่วยเก็บโลจิคัลวอลุ่ม

- c. เพิ่มดิสก์ใหม่ลงในกลุ่มวอลุ่มโดยพิมพ์:

```
extendvg VGName diskname
```

โดยที่ `VGName` คือชื่อกลุ่มวอลุ่มของคุณ และ `diskname` คือชื่อของดิสก์ใหม่ ในตัวอย่างที่แสดงในขั้นตอนก่อนหน้านี้ `diskname` จะถูกแทนที่โดย `hdisk1`

2. ฟิสิคัลวอลุ่มต้นทางและปลายทาง ต้องอยู่ในกลุ่มวอลุ่มเดียวกัน เมื่อต้องการกำหนดว่าทั้งสองฟิสิคัลวอลุ่ม อยู่ในกลุ่มวอลุ่มหรือไม่ให้พิมพ์:

```
lsvg -p VGname
```

โดยที่ `VGname` คือชื่อกลุ่มวอลุ่มของคุณ เอาต์พุตสำหรับกลุ่มวอลุ่มรากมีลักษณะคล้ายกับตัวอย่างต่อไปนี้:

```
rootvg:
PV_NAME      PV STATE      TOTAL PPs   FREE PPs   FREE DISTRIBUTION
hdisk0       active        542         85         00..00..00..26..59
hdisk1       active        542         306        00..00..00..00..06
```

จดบันทึก หมายเลขของ FREE PPs

3. ตรวจสอบว่าคุณมีพื้นที่ว่างเพียงพอบนดิสก์เป้าหมายสำหรับดิสก์ต้นทาง ซึ่งคุณต้องการย้าย:

- a. กำหนดจำนวนของพาร์ติชันฟิสิคัลบนดิสก์ต้นทางโดยพิมพ์:

```
lspv SourceDiskName | grep "USED PPs"
```

โดยที่ `SourceDiskName` คือชื่อของดิสก์ต้นทาง ตัวอย่าง เช่น `hdisk0` เอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```
USED PPs:      159 (636 megabytes)
```

ในตัวอย่างนี้ คุณต้องการ 159 FREE PPs บนดิสก์ปลายทางเพื่อทำการย้ายให้สำเร็จ

- b. เปรียบเทียบจำนวนของ USED PPs จากดิสก์ต้นทางกับจำนวนของ FREE PPs บนดิสก์ปลายทาง (ขั้นตอนที่ 2) ถ้าจำนวนของ FREE PPs มากกว่าจำนวนของ USED PPs แสดงว่าคุณมีพื้นที่ว่างเพียงพอสำหรับการย้าย
4. ปฏิบัติตามขั้นตอนนี้เฉพาะถ้าคุณกำลังย้ายข้อมูลจากดิสก์ในกลุ่มวอลุ่ม `rootvg` เท่านั้น หากคุณกำลังย้ายข้อมูลจากดิสก์ในกลุ่มวอลุ่มที่ผู้ใช้กำหนดให้ทำขั้นตอนที่ 5 ต่อไป

ตรวจสอบเพื่อดูว่าบูตโลจิคัลวอลุ่ม (`hd5`) อยู่บนดิสก์ต้นทางหรือไม่โดยพิมพ์:

```
lspv -l SourceDiskNumber | grep hd5
```

ถ้าคุณไม่ได้รับเอาต์พุต จะไม่มีการระบุตำแหน่งบูตโลจิคัลวอลุ่มบนดิสก์ต้นทาง ทำขั้นตอนที่ 5 ต่อไป

ถ้าคุณได้รับเอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```
hd5          2  2  02..00..00..00..00  /blv
```

ให้รันคำสั่งต่อไปนี้:

```
migratepv -l hd5 SourceDiskName DestinationDiskName
```

คุณจะได้รับข้อความเตือนให้คุณใช้คำสั่ง **bosboot** บนดิสก์ปลายทาง คุณยังต้องใช้คำสั่ง **mkboot -c** เพื่อล้าง บูตเร็กคอร์ดบนดิสก์ต้นทางด้วย พิมพ์ลำดับของคำสั่งต่อไปนี้:

```
bosboot -a -d /dev/DestinationDiskName  
bootlist -m normal DestinationDiskName  
mkboot -c -d /dev/SourceDiskName
```

5. ย้ายข้อมูลของคุณโดยพิมพ์พาราด่วน SMIT ต่อไปนี้:

```
smit migratepv
```

6. แสดงรายการฟิลิคัลวอลุ่ม และเลือกฟิลิคัลวอลุ่มต้นทาง ซึ่งคุณตรวจสอบก่อนหน้านี้
7. ไปยังฟิลิคัลวอลุ่ม **DESTINATION** หากคุณยอมรับค่าดีฟอลต์ ฟิลิคัลวอลุ่มทั้งหมดในกลุ่มวอลุ่มจะ พร้อมสำหรับการโอนย้าย มิฉะนั้น เลือกดิสก์ตั้งแต่หนึ่งดิสก์ขึ้นไปที่มีพื้นที่ว่างเพียงพอ สำหรับพาร์ติชันซึ่งคุณกำลังย้าย (จากขั้นตอนที่ 4)
8. ถ้าคุณต้องการ ให้ไปยังข้อมูลย้ายอย่างเดี่ยวที่เป็นสมาชิกของฟิลิคัลวอลุ่ม **LOGICAL VOLUME** นี้ และแสดงรายการและเลือกโลจิคัลวอลุ่ม คุณย้ายเฉพาะ พาร์ติชันฟิลิคัลที่จัดสรรให้กับโลจิคัลวอลุ่มที่ระบุซึ่งอยู่บน ฟิลิคัลวอลุ่มที่เลือกเป็นฟิลิคัลวอลุ่มต้นทางเท่านั้น
9. กด Enter เพื่อย้ายพาร์ติชันฟิลิคัล

ณ จุดนี้ ข้อมูลอยู่บนดิสก์ (ปลายทาง) ใหม่ อย่างไรก็ตาม ดิสก์ดั้งเดิม (ต้นทาง) ยังคงอยู่ในกลุ่มวอลุ่ม หากดิสก์ยังคงเชื่อถือได้ คุณควรใช้ดิสก์นั้นต่อไปเป็นดิสก์ฮอตสเปร์ โดยเฉพาะ เมื่อดิสก์ล้มเหลว ขอแนะนำ ให้ทำขั้นตอนต่อไปนี้:

1. เมื่อต้องการลบดิสก์ต้นทางออกจากกลุ่มวอลุ่ม ให้พิมพ์:

```
reducevg VGName SourceDiskName
```

2. เมื่อต้องการเอาดิสก์ต้นทางออกจากระบบทางกายภาพ ให้พิมพ์:

```
rmdev -l SourceDiskName -d
```

หลักการที่เกี่ยวข้อง:

“หน่วยเก็บโลจิคัลวอลุ่ม” ในหน้า 421

โลจิคัลวอลุ่มคือกลุ่มของข้อมูลที่ตั้งอยู่บนฟิลิคัลวอลุ่ม

งานที่เกี่ยวข้อง:

“การกำหนดคอนฟิกดิสก์”

คุณสามารถกำหนดคอนฟิกดิสก์ใหม่ได้หลายเมธอด

“การแก้ไขปัญหาดิสก์ไดร์ฟ” ในหน้า 408

ข้อมูลนี้แนะนำวิธีการวินิจฉัยและแก้ไขปัญหาดิสก์ไดร์ฟ

การกำหนดคอนฟิกดิสก์:

คุณสามารถกำหนดคอนฟิกดิสก์ใหม่ได้หลายเมธอด

คุณสามารถกำหนดคอนฟิกดิสก์ใหม่ในวิธีใดๆ ต่อไปนี้

- หากคุณสามารถปิดระบบได้ ให้ใช้เมธอด 1 เมื่อใดก็ตามที่ทำได้ ควรปิดระบบต่างๆ เสมอ เมื่อคุณต่อพ่วงฟิลิคัลดิสก์เข้ากับระบบ

- หากคุณไม่สามารถปิดระบบของคุณและคุณทราบรายละเอียดเกี่ยวกับ ดิสก์ใหม่ เช่น คลาสย่อย ชนิด ชื่อพาร์ติชัน และ ตำแหน่งที่ เชื่อมต่อให้ใช้เมธอด 2
- หากคุณไม่สามารถปิดระบบและคุณทราบเฉพาะ ที่ตั้งของดิสก์ให้ใช้เมธอด 3

หลังจากกำหนดคอนฟิกดิสก์แล้ว แม้ว่าโดยทั่วไป ดิสก์ จะพร้อมใช้งานได้ แต่ Logical Volume Manager กำหนดให้มีการระบุเพิ่มเติมเป็นฟิลิควอลุ่ม

เมธอด 1

ใช้ เมธอดต่อไปนี้เมื่อคุณสามารถปิดระบบ ก่อนต่อพ่วงดิสก์:

1. เชื่อมต่อดิสก์ใหม่เข้ากับระบบทางกายภาพ จากนั้น เปิดดิสก์และระบบตามเอกสารคู่มือที่มาพร้อมกับ ระบบของคุณ
2. ในระหว่างบูตระบบ ปล่อยให้ Configuration Manager (cgmgr) กำหนดคอนฟิก ดิสก์โดยอัตโนมัติ
3. หลังจากบูตระบบ ด้วยสิทธิการใช้งานราก ให้พิมพ์คำสั่ง `lspv` ที่บรรทัดคำสั่ง เพื่อค้นหาชื่อของดิสก์ใหม่ ระบบส่งคืน รายการคล้ายกับอย่างใดอย่างหนึ่งต่อไปนี้:

```
hdisk1 none none
```

หรือ:

```
hdisk1 00005264d21adb2e none
```

ฟิลด์แรกระบุชื่อที่ระบบกำหนดของ ดิสก์ ฟิลด์ที่สองแสดง ID ฟิลิควอลุ่ม (PVID) ถ้ามี หากดิสก์ใหม่ไม่ปรากฏขึ้นใน เอาต์พุต `lspv` ให้อ้างอิง *Installation and migration*

ณ จุดนี้ ระบบสามารถใช้ดิสก์ได้ แต่ดิสก์ต้องการ PVID เพื่อให้ LVM สามารถใช้ดิสก์ได้ หากดิสก์ใหม่ไม่มี PVID ให้ดูที่ “การทำดิสก์ที่มีอยู่เป็นฟิลิควอลุ่ม” ในหน้า 400

เมธอด 2

ใช้เมธอดต่อไปนี้เมื่อคุณไม่สามารถปิดระบบ และคุณทราบข้อมูลต่อไปนี้เกี่ยวกับดิสก์ใหม่:

- วิธีการต่อพ่วงดิสก์ (คลาสย่อย)
- ชนิดของดิสก์ (ชนิด)
- ส่วนต่อพ่วงระบบซึ่งดิสก์เชื่อมต่ออยู่ (ชื่อพาร์ติชัน)
- โลจิคัลแอดเดรสของดิสก์ (ตำแหน่งที่เชื่อมต่อ)

ปฏิบัติดังต่อไปนี้:

1. เชื่อมต่อดิสก์ใหม่เข้ากับระบบทางกายภาพ จากนั้น เปิดดิสก์และระบบตามเอกสารคู่มือที่มาพร้อมกับ ระบบของคุณ
2. เมื่อต้องการกำหนดคอนฟิกดิสก์และทำให้แน่ใจว่าดิสก์มีอยู่เป็นฟิลิควอลุ่ม ให้ใช้คำสั่ง `mkdev` พร้อมด้วยแฟล็กที่แสดง ดังเช่นในตัวอย่างต่อไปนี้:

```
mkdev -c disk -s scsi -t 2200mb -p scsi3 \
-w 6,0 -a pv=yes
```

ตัวอย่างนี้เพิ่มดิสก์ 2.2 GB ที่มี SCSI ID เป็น 6 และหมายเลขโลจิคัลยูนิตเป็น 0 ลงในบัส `scsi3` SCSI แฟล็ก `-c` กำหนดคลาสของอุปกรณ์ แฟล็ก `-s` กำหนดคลาสย่อย แฟล็ก `-t` กำหนดชนิดของอุปกรณ์ แฟล็ก `-p` กำหนดชื่ออุปกรณ์พาร์ติชัน

ซึ่งคุณต้องการกำหนด แฟล็ก -w กำหนดที่ตั้งของดิสก์โดยใช้ SCSI ID และหมายเลขโลจิคัลยูนิต แฟล็ก -a ระบุค่าแอตทริบิวต์อุปกรณ์ pv=yes ซึ่งทำให้ดิสก์เป็นฟิสิคัลวอลุ่ม และเขียนเร็กคอร์ดบูตที่มีตัวระบุฟิสิคัลวอลุ่มเฉพาะบนดิสก์ (ถ้าไม่มีอยู่แล้ว)

ณ จุดนี้ ดิสก์ถูกกำหนดเป็นทั้งอุปกรณ์ที่มีอยู่ และเป็นฟิสิคัลวอลุ่ม คุณสามารถพิมพ์คำสั่ง lspv บนบรรทัดคำสั่งเพื่อแสดงรายการดิสก์ใหม่ หากดิสก์ใหม่ไม่ปรากฏขึ้นในเอาต์พุต lspv ให้อ้างอิง *Installation and migration*

เมธอด 3

ใช้เมธอดต่อไปนี้อย่างที่คุณไม่สามารถปิดระบบ และคุณทราบเฉพาะที่ตั้งของดิสก์เท่านั้น:

1. เชื่อมต่อดิสก์ใหม่เข้ากับระบบทางกายภาพ จากนั้น เปิดดิสก์และระบบตามเอกสารคู่มือที่มาพร้อมกับ ระบบของคุณ
2. เมื่อต้องการตรวจสอบว่าฟิสิคัลดิสก์ได้มีการกำหนดคอนฟิกแล้วบนระบบ ให้พิมพ์คำสั่ง lspv บนบรรทัดคำสั่ง สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง lspv ให้ดูที่หัวข้อ คำสั่ง lspv เอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```
hdisk0      000005265ac63976    rootvg
```

3. พิมพ์ `cfgmgr` บนบรรทัดคำสั่งเพื่อป้อน Configuration Manager Configuration Manager ตรวจสอบและกำหนดคอนฟิกอุปกรณ์ที่เชื่อมต่อใหม่ทั้งหมดบนระบบโดยอัตโนมัติ รวมถึง ดิสก์ใหม่ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `cfgmgr` ให้ดูที่ `cfgmgr`

4. เมื่อต้องการยืนยันว่าดิสก์ใหม่มีการกำหนดคอนฟิกแล้ว ให้พิมพ์คำสั่ง `lspv` อีกครั้ง เอาต์พุตมีลักษณะคล้ายกับอย่างใดอย่างหนึ่งต่อไปนี้:

```
hdisk1     none                none
```

หรือ

```
hdisk1     00005264d21adb2e    none
```

ฟิลด์แรกระบุชื่อที่ระบบกำหนดของ ดิสก์ ฟิลด์ที่สองแสดง ID ฟิสิคัลวอลุ่ม (PVID) ถ้ามี หากดิสก์ใหม่ไม่ปรากฏขึ้นในเอาต์พุต `lspv` ให้อ้างอิง *Installation and migration*

ณ จุดนี้ ระบบสามารถใช้ดิสก์ได้ แต่ดิสก์ต้องการ PVID เพื่อให้ LVM สามารถใช้ดิสก์ได้ หากดิสก์ใหม่ไม่มี PVID ให้ดูที่ “การทำดิสก์ที่มีอยู่เป็นฟิสิคัลวอลุ่ม”

งานที่เกี่ยวข้อง:

“การย้ายเนื้อหาของฟิสิคัลวอลุ่ม” ในหน้า 396

การทำให้ดิสก์ที่มีอยู่เป็นฟิสิคัลวอลุ่ม:

ต้องกำหนดคอนฟิกดิสก์เป็นฟิสิคัลวอลุ่ม ก่อนสามารถกำหนดดิสก์นั้น ให้กับกลุ่มวอลุ่มและใช้โดย LVM

ใช้คำแนะนำต่อไปนี้อย่างคุณกำหนดคอนฟิกฟิสิคัลวอลุ่ม:

1. ตรวจสอบให้แน่ใจว่าดิสก์เป็นที่รู้จักของระบบปฏิบัติการ มีอยู่ และไม่ได้ถูกใช้อยู่โดยระบบปฏิบัติการหรือแอปพลิเคชันใดๆ พิมพ์คำสั่ง `lspv` บนบรรทัดคำสั่ง เอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```
hdisk1     none                none
```

ตรวจสอบ เอาต์พุตในเรื่องต่อไปนี้:

- ถ้าชื่อของดิสก์ใหม่ไม่ปรากฏขึ้นในคำสั่งเอาต์พุต ให้อ้างอิง “การกำหนดคอนฟิกดิสก์” ในหน้า 398

- ถ้าฟิลด์ที่สองของเอาต์พุตแสดงตัวระบุฟิสิคัลวอลุ่ม (physical volume identifier หรือ PVID) ที่ระบบสร้างขึ้น (ตัวอย่างเช่น 00005264d21adb2e) แสดงว่าดิสก์ มีการกำหนดคอนฟิกเป็นฟิสิคัลวอลุ่มแล้ว และคุณไม่ต้องทำ โพรซีเดอร์นี้
- ถ้าฟิลด์ที่สามของเอาต์พุตแสดงชื่อกลุ่มวอลุ่ม (ตัวอย่างเช่น rootvg) แสดงว่าดิสก์ถูกใช้อยู่ในปัจจุบัน และไม่ใช่ตัวเลือกที่เหมาะสมสำหรับโพรซีเดอร์นี้

ถ้าดิสก์ใหม่ไม่มี PVID และไม่ใช่ถูกใช้ ให้ทำขั้นตอน ถัดไปต่อไป

2. เมื่อต้องการเปลี่ยนดิสก์ที่มีอยู่เป็นฟิสิคัลวอลุ่ม ให้พิมพ์คำสั่ง `chdev` บนบรรทัดคำสั่ง ตัวอย่าง:

```
chdev -l hdisk3 -a pv=yes
```

แฟล็ก `-l` ระบุชื่ออุปกรณ์ของดิสก์ แฟล็ก `-a` ระบุค่าแอตทริบิวต์อุปกรณ์ `pv=yes` ซึ่งทำให้ดิสก์เป็นฟิสิคัลวอลุ่ม และเขียนเร็กคอร์ดบูตที่มีตัวระบุฟิสิคัลวอลุ่มเฉพาะบน ดิสก์ (ถ้าไม่มีอยู่แล้ว)

ณ จุดนี้ ดิสก์ถูกกำหนดเป็นฟิสิคัลวอลุ่ม คุณสามารถ พิมพ์คำสั่ง `lspv` บนบรรทัดคำสั่ง เพื่อแสดงรายการ ดิสก์ใหม่

การเปลี่ยนแปลง PVID และ VGID ของ rootvg:

คุณสามารถเปลี่ยนแปลง physical volume identifier (PVID) และ volume group identifier (VGID) ของกลุ่มวอลุ่ม rootvg ระหว่าง เฟสการบูตระบบ

เมื่อต้องการเปลี่ยนแปลง PVID และ VGID ของ rootvg ให้ตั้งค่าแอตทริบิวต์ `sys0 dev ghostdev` ด้วยค่า 2 และรีบูต ระบบ แอตทริบิวต์ `sys0 device ghostdev` เป็นแฟล็ก bitwise

- เมื่อต้องการตั้งค่าแอตทริบิวต์ `sys0 device ghostdev` เพื่อเปลี่ยนแปลง PVID และ VGID ของกลุ่มวอลุ่ม rootvg ป้อนคำสั่งต่อไปนี้:

```
chdev -l sys0 -a ghostdev=2
```

หมายเหตุ: ค่า ของ 2 สำหรับแอตทริบิวต์ `sys0 device ghostdev` ถูกยกเลิกการตั้งค่า หลังจากคำสั่ง `ipl_varyon` เปลี่ยนแปลง PVID และ VGID ของดิสก์ทั้งหมดใน rootvg ถ้าคำสั่ง `chdev` สำหรับการเปลี่ยนแปลง PVID ของดิสก์ rootvg ใด ๆ ล้มเหลว คำสั่ง `ipl_varyon` ส่งข้อความเตือน และยังคงขึ้นอยู่กับ rootvg ถ้าคำสั่ง `chdev` สำหรับการเปลี่ยนแปลง PVID ของดิสก์ใด ๆ ใน rootvg ล้มเหลว และคุณต้องการ เปลี่ยนแปลง PVID และ VGID ระหว่างการรีบูตถัดไป ตั้งค่าแอตทริบิวต์ `sys0 device ghostdev` เป็น 2 อีกครั้ง

- เมื่อต้องการแสดงรายการค่าของแอตทริบิวต์ `ghostdev` ป้อน คำสั่งต่อไปนี้:

```
lsattr -E -l sys0 -a ghostdev
```

การเปลี่ยนฟิสิคัลวอลุ่มที่ล้มเหลวในกลุ่มวอลุ่มที่ทำสำเนา:

โพรซีเดอร์ต่อไปนี้เปลี่ยนฟิสิคัลวอลุ่ม (PV) ที่ล้มเหลวภายในกลุ่มวอลุ่มที่ทำสำเนา คำสั่ง `replacepv` นำเสนอ เมธอดการเปลี่ยน PV ที่ล้มเหลวในคอนฟิกูเรชันส่วนใหญ่ นอกจากนี้ ยังมีโพรซีเดอร์อื่นสำหรับคอนฟิกูเรชันที่ไม่สามารถใช้ คำสั่ง `replacepv` ได้ด้วย

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

ข้อกำหนดเบื้องต้น

- โลจิคัลวอลุ่มทั้งหมดที่ใช้ PV ที่ล้มเหลวมีสำเนาที่ถูกต้องบน PVs อื่นที่มีอยู่ (ด้วยข้อยกเว้นที่เป็นไปได้ของดัมพ์โลจิคัลวอลุ่ม เฉพาะงาน)

การเปลี่ยน PV ที่ล้มเหลวโดยใช้คำสั่ง replacepv

ข้อกำหนดเบื้องต้น

หากไม่สามารถปฏิบัติตามข้อกำหนดเบื้องต้นใดๆ ที่แสดงรายการข้างล่าง ให้ดูที่โพรซีเจอร์อื่น

- กลุ่มวอลุ่มซึ่งมี PV ที่ล้มเหลวไม่ใช่ rootvg
- PV ที่นำมาเปลี่ยนสามารถเพิ่มลงในกลุ่มวอลุ่มซึ่งมี PV ที่ล้มเหลว (ซึ่งอาจจะเป็นไปได้ ทั้งนี้ขึ้นอยู่กับขนาด PV และลักษณะกลุ่มวอลุ่ม เช่น MAX PPs per PV)
- PV ที่นำมาเปลี่ยนต้องสามารถถูกกำหนดคอนฟิกเข้าในระบบ ในเวลาเดียวกันกับ PV ที่ล้มเหลวได้
- ชื่อของ PV ที่นำมาเปลี่ยนสามารถแตกต่างจากชื่อของ PV ที่ล้มเหลว
- ขนาดของ PV ที่นำมาเปลี่ยนต้องเท่ากับขนาดของ PV ที่ล้มเหลวเป็นอย่างน้อย
- กลุ่มวอลุ่มซึ่งมี PV ที่ล้มเหลวต้องไม่เป็นกลุ่มวอลุ่มสแน็ปช็อต หรือมีกลุ่มวอลุ่มสแน็ปช็อต

ทำขั้นตอนต่อไปนีให้สมบูรณ์ โดยสมมติว่า PV ที่ล้มเหลว คือ hdisk2 และ PV ที่นำมาเปลี่ยนคือ hdisk10:

1. หากยังไม่ได้ติดตั้ง PV ที่นำมาเปลี่ยนบนระบบ ให้ปฏิบัติตาม ขั้นตอนที่จำเป็นเพื่อติดตั้ง เมื่อต้องการใช้ผู้จัดการคอนฟิกูเรชัน เพื่อกำหนด PV ใหม่ให้รันคำสั่งต่อไปนี้:

```
cfgmgr
```

ใช้ คำสั่ง `lspv` เพื่อกำหนดชื่อที่กำหนดให้กับ PV สำหรับตัวอย่างนี้ สมมติว่า PV ใหม่มีชื่อว่า hdisk10

2. เมื่อต้องการเปลี่ยน PV ที่ล้มเหลวด้วย PV ซึ่งกำหนดในขั้นตอนที่ 1 ให้รัน คำสั่งต่อไปนี้:

```
replacepv hdisk2 hdisk10
```

เมื่อ คำสั่งรัน hdisk2 ถูกแทนที่ด้วย hdisk10 และ hdisk2 ไม่มีการกำหนดให้กับกลุ่มวอลุ่มอีกต่อไป

3. เมื่อต้องการยกเลิกการกำหนด PV ที่ล้มเหลว ให้รันคำสั่งต่อไปนี้:

```
rmdev -d1 hdisk2
```

4. ถอดดิสก์ที่ล้มเหลวออกจากระบบทางกายภาพ

5. ตรวจสอบว่าโพรซีเจอร์เสร็จเรียบร้อยแล้วโดยทำขั้นตอนต่อไปนีให้สมบูรณ์:

- เมื่อต้องการตรวจสอบว่าโลจิคัลวอลุ่มทั้งหมดมีการมีร์เรอร์เป็น PV ใหม่ตาม ต้องการ ให้รันคำสั่งต่อไปนี้:

```
lslv lvname
```

ตรวจสอบ แอ็ททริบิวต์ COPIES ของแต่ละโลจิคัลวอลุ่มที่ได้รับผลกระทบจาก PV ที่ล้มเหลวเพื่อให้มั่นใจว่า ขณะนี้มีจำนวนสำเนาตามที่ต้องการ หากจำนวนสำเนาของโลจิคัลวอลุ่มต่ำกว่าจำนวนที่ต้องการ ให้ใช้คำสั่ง `mklvcopy` เพื่อสร้างสำเนา เพิ่มเติม

- เมื่อต้องการตรวจสอบว่าโลจิคัลวอลุ่มพาร์ติชันทั้งหมดมีการซิงโครไนซ์ และไม่มีพาร์ติชันเก่า ให้รันคำสั่งต่อไปนี้:

```
lspv hdisk10
```

ตรวจสอบ แอ็ททริบิวต์ STALE PARTITIONS ของ PV ที่เปลี่ยน เพื่อให้มั่นใจว่าจำนวนเป็นศูนย์ หากมีพาร์ติชันเก่า ให้ใช้คำสั่ง `syncvg` เพื่อซิงโครไนซ์พาร์ติชัน

ขั้นตอนที่ 5 ทำโพสดีเตอร์การเปลี่ยน PV ที่ล้มเหลว

การเปลี่ยน PV ที่ล้มเหลวเมื่อคอนฟิกรูเรชัน ไม่นอญูตให้ใช้คำสั่ง `replacev`

สมมติว่า ฟิสิคัลวอลุ่มที่ล้มเหลวคือ `hdisk0` และมีเรอร์ `hdisk1` เป็นส่วนประกอบหนึ่งของกลุ่มวอลุ่ม `yourvg`

1. เมื่อต้องการลบสำเนาเรอร์ออกจาก PV ที่ล้มเหลว ให้รันคำสั่ง ต่อไปนี้:

```
unmirrorvg yourvg hdisk0
```

2. หากความล้มเหลว PV เกิดขึ้นบน `rootvg` ให้ลบ `hdisk0` ออกจาก รายการบูตโดยรันคำสั่งต่อไปนี้:

หมายเหตุ: ถ้าคอนฟิกรูเรชันของคุณ ใช้อุปกรณ์บูตอื่นที่ไม่ใช่ `hdisk0` และ `hdisk1` ให้เพิ่มอุปกรณ์นั้นลงในไวยากรณ์คำสั่ง

```
bootlist -om normal hdisk1
```

ขั้นตอนนี้ กำหนดว่า `hdisk1` ยังคงเป็นอุปกรณ์ที่บูตได้ใน `rootvg` หลังจากทำขั้นตอนนี้เสร็จสมบูรณ์แล้ว ตรวจสอบให้แน่ใจว่า `hdisk0` ไม่ปรากฏขึ้นในเอาต์พุต

3. หากความล้มเหลว PV เกิดขึ้นบน `rootvg` ให้สร้างอุปกรณ์ดัมพ์เฉพาะงานขึ้นใหม่ จาก PV ที่ล้มเหลว

ถ้าคุณมีอุปกรณ์ดัมพ์เฉพาะงาน ที่อยู่บน PV ที่ล้มเหลว คุณสามารถใช้คำสั่ง `mklv` เพื่อสร้าง โลจิคัลวอลุ่มใหม่บน PV ที่มีอยู่ ใช้คำสั่ง `sysdumpdev` เพื่อตั้งค่าโลจิคัลวอลุ่มใหม่เป็นอุปกรณ์ดัมพ์หลัก

4. เมื่อต้องการยกเลิกการกำหนด PV ที่ล้มเหลว ให้รันคำสั่งต่อไปนี้:

หมายเหตุ: การลบ รายการอุปกรณ์ดิสก์จะลบฮาร์ดดิสก์ `/dev/ipldevice` ออกด้วย ถ้า PV ที่ล้มเหลวเป็น PV ที่ใช้ในการบูตระบบ

```
reducevg yourvg hdisk0  
rmdev -dl hdisk0
```

5. หาก PV ที่ล้มเหลวเป็นอุปกรณ์บูตที่ใช้งานล่าสุด ให้สร้างฮาร์ดดิสก์ `/dev/ipldevice` ที่ถูกลบออกในขั้นตอนที่ 4 ขึ้นใหม่ โดยรันคำสั่งต่อไปนี้:

```
ln /dev/rhdisk1 /dev/ipldevice
```

โปรดสังเกต `r` ที่เติมหน้าชื่อ PV

เมื่อต้องการตรวจสอบว่าฮาร์ดดิสก์ `/dev/ipldevice` มีการสร้างขึ้นใหม่แล้ว ให้รันคำสั่งต่อไปนี้:

```
ls /dev/ipldevice
```

6. เปลี่ยนดิสก์ที่ล้มเหลว

7. เมื่อต้องการกำหนด PV ใหม่ ให้รันคำสั่งต่อไปนี้:

```
cfgmgr
```

คำสั่ง `cfgmgr` กำหนดชื่อ PV ให้กับ PV ที่นำมาเปลี่ยน ชื่อ PV ที่กำหนดอาจเหมือนกับ ชื่อ PV ที่กำหนดก่อนหน้านี้ให้กับ PV ที่ล้มเหลว ในตัวอย่างนี้ สมมติว่าอุปกรณ์ `hdisk0` ถูก กำหนดให้กับ PV ที่นำมาเปลี่ยน

8. เมื่อต้องการเพิ่ม PV ใหม่ลงในกลุ่มวอลุ่ม ให้รันคำสั่งต่อไปนี้:

```
extendvg yourvg hdisk0
```

คุณ อาจพบข้อความแสดงข้อผิดพลาดต่อไปนี้:

```
0516-050 Not enough descriptor space left in this volume group.  
Either try adding a smaller PV or use another volume group.
```

ถ้า คุณพบข้อผิดพลาดนี้และไม่สามารถเพิ่ม PV ลงในกลุ่มวอลุ่มได้ คุณสามารถลองมีร์เรอร์โลจิคัลวอลุ่มเป็น PV อื่นที่มีอยู่แล้ว ในกลุ่มวอลุ่ม หรือเพิ่ม PV ที่เล็กลง หากข้อผิดพลาดทั้งสองอย่างเป็นไปไม่ได้ คุณสามารถลองบายพาสข้อจำกัดนี้ โดยการอัปเดตกลุ่มวอลุ่มเป็น กลุ่มวอลุ่ม Big-type หรือ Scalable-type โดยใช้คำสั่ง `chvg`

9. มีร์เรอร์กลุ่มวอลุ่ม

หมายเหตุ: คำสั่ง `mirrorvg` ไม่สามารถใช้ได้ถ้ามีเงื่อนไขทั้งหมดต่อไปนี้:

- ระบบเป้าหมายเป็นพาร์ติชันโลจิคัล (LPAR)
- สำเนาของบูตโลจิคัลวอลุ่ม (โดยค่าดีฟอลต์คือ `hd5`) ตั้งอยู่บน PV ที่ล้มเหลว
- อะแดปเตอร์ของ PV ที่นำมาเปลี่ยนถูกกำหนดคอนฟิกแบบไดนามิกเข้าไปใน LPAR ตั้งแต่ cold บูตครั้งล่าสุด

หากมีเงื่อนไขข้างบนครบทั้งหมด ให้ใช้คำสั่ง `mklvcopy` เพื่อสร้างสำเนา มีร์เรอร์สำหรับแต่ละโลจิคัลวอลุ่มขึ้นใหม่ดังนี้:

- สร้างสำเนาของบูตโลจิคัลวอลุ่มเพื่อให้แน่ใจว่า สำเนา ถูกจัดสรรให้กับชุดที่ต่อเนื่องของพาร์ติชันฟิสิคัล
- สร้างสำเนาของโลจิคัลวอลุ่มที่เหลือ และซิงโครไนซ์สำเนา โดยใช้คำสั่ง `syncvg`
- จัดทำติสก์ที่บูตได้โดยปิด LPAR และเรียกใช้ แทนการรีบูตโดยใช้คำสั่ง `shutdown` หรือ `reboot` การปิดนี้ไม่จำเป็นต้องทำในที่ที่ แต่จำเป็นต้องที่ระบบ ต้องบูตจาก PV ใหม่

มีฉะนั้น สร้างสำเนาใหม่ของโลจิคัลวอลุ่มใน กลุ่มวอลุ่มที่ใช้ PV ใหม่ด้วยคำสั่งต่อไปนี้:

หมายเหตุ: คำสั่ง `mirrorvg` ปิดใช้งานองค์ประกอบโดยค่าดีฟอลต์สำหรับ `rootvg` คุณจะต้องใช้อ็อปชัน `-m` เพื่อให้มั่นใจว่าสำเนาโลจิคัลวอลุ่มใหม่ถูกแม็พเข้ากับ `hdisk0` ใน วิธีเดียวกันกับติสก์ทำงาน

```
mirrorvg yourvg hdisk0
```

10. หากคอนฟิกูเรชันมีสำเนาของบางโลจิคัลวอลุ่มอยู่ คุณอาจต้องสร้างสำเนาเหล่านั้นขึ้นใหม่โดยใช้คำสั่ง ต่อไปนี้:

```
mklvcopy -k
```

11. หากความล้มเหลว PV เกิดขึ้นบน `rootvg` ให้เริ่มต้นบูตเร็กคอร์ด โดยรันคำสั่งต่อไปนี้:

```
bosboot -a
```

12. หากความล้มเหลว PV เกิดขึ้นบน `rootvg` ให้อัปเดตรายการบูต โดยรันคำสั่งต่อไปนี้:

หมายเหตุ: ถ้าคอนฟิกูเรชันของคุณ ใช้อุปกรณ์บูตอื่นที่ไม่ใช่ `hdisk0` และ `hdisk1` ให้เพิ่มอุปกรณ์นั้นลงในคำสั่ง

```
bootlist -om normal hdisk0 hdisk1
```

13. ตรวจสอบว่าโปรซีเดอร์เสร็จเรียบร้อยแล้ว

- เมื่อต้องการตรวจสอบว่าโลจิคัลวอลุ่มทั้งหมดมีการมีร์เรอร์เป็น PV ใหม่ ให้รันคำสั่งต่อไปนี้:

```
lslv lvname
```

ตรวจสอบ แอ็ททริบิวต์ COPIES ของแต่ละโลจิคัลวอลุ่มที่ได้รับผลกระทบจาก PV ที่ล้มเหลว เพื่อให้มั่นใจว่า ขณะนี้มีจำนวนสำเนาตามที่ต้องการ หากจำนวนสำเนาของโลจิคัลวอลุ่มต่ำกว่าจำนวนที่ต้องการ ให้ใช้คำสั่ง `mklvcopy` เพื่อสร้างสำเนา เพิ่มเติม

- เมื่อต้องการตรวจสอบว่าโลจิคัลวอลุ่มพาร์ติชันทั้งหมดมีการซิงโครไนซ์ และไม่มีพาร์ติชันเก่า ให้รันคำสั่งต่อไปนี้:

```
lspv hdisk0
```

ตรวจสอบ แอ็ททริบิวต์ STALE PARTITIONS ของ PV ที่เปลี่ยน เพื่อให้มั่นใจว่าจำนวนเป็นศูนย์ หากมีพาร์ติชันเก่า ให้ใช้ คำสั่ง syncvg เพื่อซิงโครไนซ์พาร์ติชัน

หากความล้มเหลว PV เกิดขึ้นบน rootvg ให้ใช้ขั้นตอนต่อไป นี้ เพื่อตรวจสอบลักษณะอื่นๆ ของโพรซีเดอร์นี้:

- เมื่อต้องการตรวจสอบรายการบูตให้รันคำสั่งต่อไปนี้:

```
bootlist -om normal
```

- เมื่อต้องการตรวจสอบอุปกรณ์ดัมพ์ให้รันคำสั่งต่อไปนี้:

```
sysdumpdev -l
```

- เมื่อต้องการตรวจสอบรายการของ PV ที่บูตได้ให้รันคำสั่งต่อไปนี้:

```
ipl_varyon -i
```

- เมื่อต้องการตรวจสอบ /dev/ipl_device ให้รันคำสั่งต่อไปนี้:

```
ls -i /dev/rhdisk1 /dev/ipldevice
```

ตรวจสอบว่า เอาต์พุตของคำสั่ง ls มีหมายเลขไอ-โหนด เหมือนกันสำหรับทั้งสองรายการ

ขั้นตอนนี้ทำให้โพรซีเดอร์เสร็จสมบูรณ์

ข้อมูลที่เกี่ยวข้อง:



Logical Volume Manager จาก A ถึง Z: บทนำและแนวคิด

การแจ้งผู้ดูแลระบบเมื่อฟิลิคัลวอลุ่มขาดไป:

แม้ว่า AIX บันทึก ข้อผิดพลาดเมื่อไม่สามารถเข้าถึงฟิลิคัลวอลุ่ม แต่มีสถานการณ์ ซึ่งไม่สามารถตรวจพบข้อผิดพลาดได้

ตัวอย่างเช่น เมื่อฟิลิคัลวอลุ่มเป็นส่วนหนึ่งของกลุ่มวอลุ่มที่ทำสำเนา ผู้ใช้ไม่สังเกตเห็นปัญหาเนื่องจากสำเนาที่ดีของข้อมูลยังคงสามารถ เข้าถึงได้ ในกรณีนั้น การแจ้งเตือนอัตโนมัติสามารถแจ้งเตือนผู้ดูแลระบบให้ทราบถึง ปัญหา ก่อนผู้ใช้สังเกตเห็น การขัดจังหวะใดๆ ในงานของตน

โพรซีเดอร์ ต่อไปนี้อธิบายวิธีการตั้งค่าการแจ้งเตือนอัตโนมัติเมื่อ ฟิลิคัลวอลุ่มขาดไป โดยการแก้ไขโพรซีเดอร์ต่อไปนี้ คุณสามารถติดตามข้อผิดพลาดอื่นที่มีความสำคัญสำหรับคุณ

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

1. ด้วยสิทธิการใช้งานราก ให้จัดทำสำเนาสำรองของไฟล์ /etc/objrepos/errnotify ODM คุณสามารถตั้งชื่อสำเนาสำรองได้ตามต้องการในตัวอย่าง ต่อไปนี้ สำเนาสำรองจะผนวกชื่อไฟล์ errnotify เข้ากับวันที่ปัจจุบัน:

```
cd /etc/objrepos
cp errnotify errnotifycurrent_date
```

2. ใช้โปรแกรมแก้ไขที่โปรดปรานของคุณเพื่อสร้าง ไฟล์ที่ชื่อ /tmp/pvmiss.add ซึ่งมี stanza ต่อไปนี้:

```
errnotify:
  en_pid = 0
  en_name = "LVM_SA_PVMISS"
  en_persistenceflg = 1
  en_label = "LVM_SA_PVMISS"
  en_crcid = 0
  en_type = "UNKN"
```

```

en_alertflg = ""
en_resource = "LVDD"
en_rtype = "NONE"
en_rclass = "NONE"
en_method = "/usr/lib/ras/pvmiss.notify $1 $2 $3 $4 $5 $6 $7 $8 $9"

```

After you complete all the steps in this topic, the error notification daemon will automatically expand the \$1 through \$9 in this script with detailed information from the error log entry within the notification message.

3. ใช้โปรแกรมแก้ไขที่โปรดปรานของคุณเพื่อสร้างไฟล์ที่ชื่อ /usr/lib/ras/pvmiss.notify ซึ่งมีเนื้อหาต่อไปนี้:

```

#!/bin/ksh
exec 3>/dev/console
print -u3 "?"
print -u3 - "-----"
print -u3 "ALERT! ALERT! ALERT! ALERT! ALERT! ALERT!"
print -u3 ""
print -u3 "Desc: PHYSICAL VOLUME IS MISSING. SEE ERRPT."
print -u3 ""
print -u3 "Error label: $9"
print -u3 "Sequence number: $1"
print -u3 "Error ID: $2"
print -u3 "Error class: $3"
print -u3 "Error type: $4"
print -u3 "Resource name: $6"
print -u3 "Resource type: $7"
print -u3 "Resource class: $8"
print -u3 - "-----"
print -u3 "?"
mail - "PHYSICAL VOLUME DECLARED MISSING" root <<-EOF
-----
ALERT! ALERT! ALERT! ALERT! ALERT! ALERT!
Desc: PHYSICAL VOLUME IS MISSING. SEE ERRPT.
Error label: $9
Sequence number: $1
Error ID: $2
Error class: $3
Error type: $4
Resource name: $6
Resource type: $7
Resource class: $8
-----
EOF

```

4. บันทึกไฟล์ของคุณและออกจากโปรแกรมแก้ไข
5. ตั้งค่าสิทธิ์ใช้งานที่เหมาะสมบนไฟล์ที่คุณสร้างขึ้น ตัวอย่างเช่น:

```
chmod 755 /usr/lib/ras/pvmiss.notify
```

6. พิมพ์คำสั่งต่อไปนี้เพื่อเพิ่มนิยาม LVM_SA_PVMISS ซึ่งคุณสร้างขึ้นในขั้นตอนที่ 2 ลงใน ODM:

```
odmadd /tmp/pvmiss.add
```

ในตอนนี้ ระบบรันสคริปต์ /usr/lib/ras/pvmiss.notify เมื่อเกิดข้อผิดพลาด LVM_SA_PVMISS ขึ้น สคริปต์นี้ส่งข้อความไปที่คอนโซลและยังส่งเมลถึงผู้ใช้อุปกรณ์ด้วย

หลักการที่เกี่ยวข้อง:

“หน่วยเก็บโลจิคัลวอลุ่ม” ในหน้า 421
โลจิคัลวอลุ่มคือกลุ่มของข้อมูลที่ตั้งอยู่บนฟิสิคัลวอลุ่ม
ข้อมูลที่เกี่ยวข้อง:
คำสั่ง `odmadd`

การแบ่งดิสก์ที่มีรีเรอร์จากกลุ่มวอลุ่ม:

Snapshot สนับสนุนให้คุณปกป้องความสอดคล้องกันของ กลุ่มวอลุ่มที่ทำมีรีเรอร์ไว้จากความล้มเหลวของดิสก์ที่อาจเกิดขึ้น

โดยใช้คุณลักษณะ snapshot คุณสามารถแบ่งดิสก์ที่มีรีเรอร์ เพื่อใช้เป็นสำเนาสำรอง (จากจุดยืนของข้อมูลเมตา LVM) point-in-time ที่เชื่อถือได้ของกลุ่มวอลุ่ม และเมื่อต้องการ รวมดิสก์ที่แบ่งเข้าในกลุ่ม วอลุ่มอีกครั้งอย่างเชื่อถือได้ ในโพรซีเตอร์ต่อไปนี้ อันดับแรก คุณแบ่ง ดิสก์ที่มีรีเรอร์จากกลุ่มวอลุ่ม จากนั้น คุณรวมดิสก์ที่แบ่ง เข้าในกลุ่มวอลุ่มดั้งเดิม เพื่อให้มั่นใจถึงความน่าเชื่อถือของ snapshot มากขึ้น ต้องถอนติดตั้งระบบไฟล์และแอปพลิเคชันที่ใช้โลจิคัลวอลุ่ม แบบดิบต้องอยู่ในสภาพที่รู้จัก (สภาพซึ่งสามารถกู้คืนแอปพลิเคชันได้ ถ้าคุณต้องการใช้สำเนาสำรอง)

กลุ่มวอลุ่มไม่สามารถแบ่งได้ถ้าเงื่อนไขอย่างใดอย่างหนึ่ง ต่อไปนี้เป็นจริง:

- ดิสก์ขาดไปแล้ว
- พาร์ติชันที่ไม่เก่าล่าสุดจะอยู่บนกลุ่มวอลุ่มที่แบ่ง
- มีพาร์ติชันเก่าอยู่ในกลุ่มวอลุ่ม ยกเว้นว่าคุณใช้แฟล็ก บังคับใช้ (`-f`) พร้อมกับคำสั่ง `splitvg`

ยิ่งไปกว่านั้น ยังไม่สามารถใช้คุณลักษณะ snapshot (โดยเฉพาะ คำสั่ง `splitvg`) ในโหมดพร้อมกันที่เพิ่มประสิทธิภาพหรือคลาสสิก กลุ่มวอลุ่มที่แบ่งไม่สามารถทำให้เป็นแบบเกิดขึ้นพร้อมกันหรือเกิดขึ้นพร้อมกัน ที่เพิ่มประสิทธิภาพ และมีข้อจำกัดเกี่ยวกับการเปลี่ยนแปลงที่ใช้ได้สำหรับทั้งกลุ่ม วอลุ่มที่แบ่งและกลุ่มวอลุ่มดั้งเดิม สำหรับรายละเอียด โปรดอ่านคำอธิบายคำสั่ง `chvg`

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

1. ตรวจสอบให้แน่ใจว่ากลุ่มวอลุ่มมีการมีรีเรอร์ทั้งหมด และมีรีเรอร์ อยู่บนดิสก์หรือชุดของดิสก์ที่มีเฉพาะชุดของมีรีเรอร์นี้เท่านั้น
2. เมื่อต้องการเปิดใช้งานการสนับสนุน snapshot ให้แบ่งกลุ่มวอลุ่มดั้งเดิม (`origVG`) เป็นดิสก์หรือชุดของดิสก์อื่น โดยใช้คำสั่งต่อไปนี้:

```
splitvg origVG
```

ณ จุดนี้ ขณะนี้คุณมีสำเนาสำรอง point-in-time ที่เชื่อถือได้ของกลุ่มวอลุ่ม ดั้งเดิม อย่างไรก็ตาม ระวังว่าคุณไม่สามารถเปลี่ยนการจัดสรรบนกลุ่มวอลุ่มที่แบ่ง

3. เรียกใช้ดิสก์ที่แบ่งอีกครั้งและรวมเข้าในกลุ่มวอลุ่มดั้งเดิม โดยใช้คำสั่งต่อไปนี้:

```
joinvg origVG
```

ณ จุดนี้ ขณะนี้กลุ่มวอลุ่มที่แบ่งถูกรวมเข้ากับ กลุ่มวอลุ่มดั้งเดิมอีกครั้ง

หลักการที่เกี่ยวข้อง:

“หน่วยเก็บโลจิคัลวอลุ่ม” ในหน้า 421
โลจิคัลวอลุ่มคือกลุ่มของข้อมูลที่ตั้งอยู่บนฟิสิคัลวอลุ่ม

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง chvg

คำสั่ง recreatevg

คำสั่ง splitvg

➡ Logical Volume Manager จาก A ถึง Z: บทนำและแนวคิด

การแก้ไขปัญหา LVM

มีปัญหาทั่วไปหลายชนิดเกี่ยวกับ LVM ซึ่ง คุณสามารถแก้ไขได้

การแก้ไขปัญหาดิสก์ไดร์ฟ:

ข้อมูลนี้แนะนำวิธีการวินิจฉัยและแก้ไขปัญหาดิสก์ไดร์ฟ

หากคุณสงสัยว่าดิสก์ไดร์ฟอาจจะมีปัญหาทางกลไกหรือล้มเหลวแล้ว ให้รับการวินิจฉัยบนดิสก์โดยใช้ไพรซีเดนต์ต่อไปนี้:

1. ด้วยสิทธิการใช้งานราก ให้พิมพ์พาด่วน SMIT ต่อไปในบรรทัด คำสั่ง:
smit diag
2. เลือก การวินิจฉัยเซลล์ปัจจุบัน เพื่อป้อน เครื่องมือการวินิจฉัย AIX
3. หลังจากคุณอ่านหน้าจอ คำแนะนำในการใช้การวินิจฉัย แล้ว ให้กด Enter
4. เลือก รูทีนการวินิจฉัย
5. เลือก การตรวจสอบระบบ
6. เลื่อนหน้าจอลงผ่านรายการเพื่อค้นหาและเลือกไดร์ฟซึ่งคุณต้องการ ทดสอบ
7. เลือก Commit

ขึ้นอยู่กับผลการวินิจฉัย คุณควรจะสามารถกำหนด สภาพของดิสก์ได้:

- หากคุณตรวจพบว่าดิสก์ไดร์ฟกำลังจะล้มเหลวหรือล้มเหลวแล้ว สิ่งสำคัญที่สุดคือ การกู้คืนข้อมูลจากดิสก์นั้น การย้ายนับเป็นวิธีที่เหมาะสมในการกู้คืน ข้อมูลจากดิสก์ที่ล้มเหลว ไพรซีเดนต์ต่อไปนี้อธิบายวิธีการกู้คืน หรือเรียกคืนข้อมูลในโลจิคัลวอลุ่มถ้าไม่สามารถทำการย้ายให้สำเร็จได้
- หากไดร์ฟของคุณกำลังจะล้มเหลวและคุณสามารถซ่อมไดร์ฟนั้นได้โดยไม่ต้องจัดรูปแบบใหม่ ข้อมูลจะไม่สูญหาย
- หากต้องจัดรูปแบบใหม่หรือเปลี่ยนดิสก์ไดร์ฟให้ทำสำเนาสำรอง ถ้าเป็นไปได้ และเอาดิสก์ไดร์ฟนั้นออกจากกลุ่มวอลุ่ม และคอนฟิกูเรชันระบบ ก่อนเปลี่ยนดิสก์ไดร์ฟ ข้อมูลบางส่วนจากระบบไฟล์สำเนาเดียวอาจสูญหายได้

หลักการที่เกี่ยวข้อง:

“พื้นที่ว่างดิสก์ไดร์ฟ” ในหน้า 409

หากคุณขาดพื้นที่ว่างบนดิสก์ไดร์ฟ สามารถแก้ไขปัญหานี้ได้ หลายวิธี คุณสามารถติดตามและลบไฟล์ที่ไม่ต้องการออกโดยอัตโนมัติ จำกัดไม่ให้ผู้ใช้เข้าถึงบางไดเร็กทอรี หรือติดตั้งพื้นที่ว่างจากดิสก์ไดร์ฟอื่น

“การกู้คืนดิสก์ไดร์ฟโดยไม่ต้องจัดรูปแบบใหม่” ในหน้า 410

หากคุณซ่อมแซมดิสก์ที่ไม่ดีและวางดิสก์นั้นกลับเข้าไปในระบบโดยไม่ต้องจัดรูปแบบใหม่ คุณสามารถปล่อยให้ระบบเรียกใช้ และซิงโครไนซ์พาร์ติชันฟิสิคัล เก้าอีกครั้งโดยอัตโนมัติบนไดร์ฟในเวลาชุด *พาร์ติชันฟิสิคัล เก้า* มีข้อมูลที่ระบบของคุณไม่สามารถใช้ได้

งานที่เกี่ยวข้อง:

“การย้ายเนื้อหาของฟิสิกส์วอลุ่ม” ในหน้า 396

“การกู้คืนโดยใช้ดิสก์ไดร์ฟที่ฟอร์แมตใหม่หรือเปลี่ยนดิสก์ไดร์ฟ” ในหน้า 411

คุณสามารถกู้คืนข้อมูลจากดิสก์ไดร์ฟที่ล้มเหลวโดยคุณต้องจัดรูปแบบใหม่ หรือเปลี่ยนดิสก์ที่ล้มเหลว

“การกู้คืนจากความล้มเหลวของดิสก์ในขณะที่ระบบยังคง พร้อมใช้งาน” ในหน้า 416

คุณสามารถกู้คืนจากความล้มเหลวของดิสก์โดยใช้คุณลักษณะ hot removability

พื้นที่ว่างดิสก์ไดร์ฟ:

หากคุณขาดพื้นที่ว่างบนดิสก์ไดร์ฟ สามารถแก้ไขปัญหานี้ได้ หลายวิธี คุณสามารถติดตามและลบไฟล์ที่ไม่ต้องการออกโดยอัตโนมัติ จำกัดไม่ให้ผู้ใช้เข้าถึงบางไดเร็กทอรี หรือติดตั้งพื้นที่ว่างจากดิสก์ไดร์ฟอื่น

คุณต้องมีสิทธิ์ผู้ใช้ราก กลุ่มระบบ หรือสิทธิกลุ่มการจัดการ เพื่อดำเนินการกิจเหล่านี้
งานที่เกี่ยวข้อง:

“การแก้ไขปัญหาดิสก์ไดร์ฟ” ในหน้า 408

ข้อมูลนี้แนะนำวิธีการวินิจฉัยและแก้ไขปัญหาดิสก์ไดร์ฟ

คำสั่งสำหรับการทำความสะอาดระบบไฟล์โดยอัตโนมัติ:

ใช้คำสั่ง skulker เพื่อทำความสะอาดระบบไฟล์ โดยการลบไฟล์ที่ไม่ต้องการออก

พิมพ์คำสั่งต่อไปนี้จากบรรทัดคำสั่ง:

```
skulker -p
```

คำสั่ง skulker ใช้เพื่อล้างไฟล์ที่เลิกใช้แล้วหรือไม่ต้องการออกจากระบบเป็นครั้งคราว ตัวเลือกรวมถึงไฟล์ในไดเร็กทอรี /tmp, ไฟล์ที่เก่ากว่า อายุที่ระบุ, ไฟล์ a.out, แแกนไฟล์, หรือไฟล์ ed.hup สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง skulker ให้ดูที่ skulker

โดยปกติ คำสั่ง skulker มีการรันรายวัน เป็นส่วนหนึ่งของโปรเซเดอร์การบัญชีที่รันโดยคำสั่ง cron ในระหว่างชั่วโมงที่ใช้งานไม่มากนัก

หลักการที่เกี่ยวข้อง:

“ดิสก์โอเวอร์โฟลว์” ในหน้า 489

ดิสก์โอเวอร์โฟลว์เกิดขึ้นเมื่อมีไฟล์อยู่มากเกินไปในพื้นที่ว่างซึ่ง จัดสรร กรณีนี้อาจเกิดขึ้นจากโปรเซส runaway ที่สร้างไฟล์ซึ่งไม่จำเป็นจำนวนมาก

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

การจำกัดผู้ใช้จากบางไดเร็กทอรี:

คุณสามารถรีเซ็ตพื้นที่ว่างดิสก์และอาจปล่อยให้อ่างไว้ โดยการจำกัด สิทธิเข้าถึงในไดเร็กทอรีและการมอนิเตอร์การใช้ดิสก์

1. จำกัดผู้ใช้จากบางไดเร็กทอรีโดยพิมพ์:

```
chmod 755 DirName
```

คำสั่งนี้ตั้งค่าสิทธิการอ่านและการเขียนสำหรับเจ้าของ (ราก) และตั้งค่าสิทธิอ่านอย่างเดียวสำหรับกลุ่มและบุคคลอื่น *DirName* คือ ชื่อพาธแบบเต็มของไดเรกทอรีซึ่งคุณต้องการจำกัด

2. มอนิเตอร์การใช้งานดิสก์ของผู้ใช้แต่ละรายโดยเพิ่ม บรรทัดต่อไปนี้ลงไฟล์ `/var/spool/cron/crontabs/adm:`

```
0 2 * * 4 /usr/sbin/acct/dodisk
```

บรรทัดนี้รันคำสั่ง `dodisk` ที่เวลา 2 (0 2) ของแต่ละวันพฤหัสบดี (4) คำสั่ง `dodisk` เริ่มต้นบัญชีการใช้ดิสก์ คำสั่งนี้มีกรันเป็นส่วนหนึ่งของโปรแกรมบัญชีที่รันโดยคำสั่ง `cron` ในระหว่างรอบเวลาที่ใช้งานไม่มากนัก

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

การติดตั้งพื้นที่ว่างจากดิสก์ไดรฟ์อื่น:

คุณสามารถได้รับพื้นที่ว่างมากขึ้นบนดิสก์ไดรฟ์ได้โดยการติดตั้งพื้นที่ว่าง จากไดรฟ์อื่น

คุณสามารถติดตั้งพื้นที่ว่างจากดิสก์ไดรฟ์หนึ่งไปยังอีกดิสก์ไดรฟ์หนึ่งในวิธีต่อไปนี้:

- ใช้พาธด่วน `smit mountfs`
- ใช้คำสั่ง `mount` ตัวอย่างเช่น:

```
mount -n nodeA -vnfs /usr/spool /usr/myspool
```

คำสั่ง `mount` ทำให้ระบบไฟล์พร้อมใช้งาน ที่ตั้งเฉพาะ

สิ่งอ้างอิงที่เกี่ยวข้อง:

“การบำรุงรักษาระบบไฟล์” ในหน้า 477

ภารกิจสุดท้ายที่สุดซึ่งคุณอาจต้องการเมื่อบำรุงรักษาระบบไฟล์ มีการจัดกลุ่มไว้ภายในตารางนี้

การกู้คืนดิสก์ไดรฟ์โดยไม่ต้องจัดรูปแบบใหม่:

หากคุณซ่อมแซมดิสก์ที่ไม่ดีและวางดิสก์นั้นกลับเข้าไปในระบบโดยไม่ต้อง จัดรูปแบบใหม่ คุณสามารถปล่อยให้ระบบเรียกใช้ และซิงโครไนซ์พาร์ติชันฟิสิคัล เก้าอีกครั้งโดยอัตโนมัติบนไดรฟ์ในเวลาบูต พาร์ติชันฟิสิคัล เก้า มีข้อมูลที่ระบบของคุณไม่สามารถใช้ได้

หากคุณสงสัยว่าพาร์ติชันฟิสิคัลเก้าให้พิมพ์ดังต่อไปนี้บน บรรทัดคำสั่ง:

```
lspv -M PhysVolName
```

โดยที่ *PhysVolName* คือชื่อฟิสิคัลวอลุ่มของคุณ เอาต์พุตคำสั่ง `lspv` จะแสดงรายการพาร์ติชันทั้งหมดบนฟิสิคัลวอลุ่มของคุณ ข้อมูลต่อไปนี้ เป็น บทคัดย่อจากเอาต์พุตตัวอย่าง:

hdisk16:112	lv01:4:2	stale
hdisk16:113	lv01:5:2	stale
hdisk16:114	lv01:6:2	stale
hdisk16:115	lv01:7:2	stale
hdisk16:116	lv01:8:2	stale
hdisk16:117	lv01:9:2	stale
hdisk16:118	lv01:10:2	stale

คอลัมน์แรกแสดงพาร์ติชันฟิสิคัลและคอลัมน์ที่สอง แสดงพาร์ติชันโลจิคัล พาร์ติชันฟิสิคัลที่เก้าใดๆ แสดงอยู่ในคอลัมน์ที่สาม

งานที่เกี่ยวข้อง:

“การแก้ไขปัญหาดิสก์ไดรฟ์” ในหน้า 408

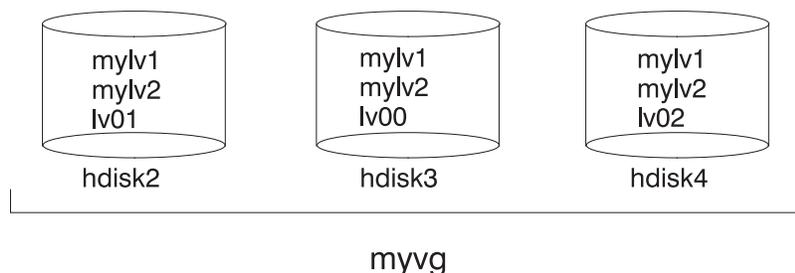
ข้อมูลนี้แนะนำวิธีการวินิจฉัยและแก้ไขปัญหาดิสก์ไดรฟ์

การกู้คืนโดยใช้ดิสก์ไดรฟ์ที่ฟอร์แมตใหม่หรือเปลี่ยนดิสก์ไดรฟ์:

คุณสามารถกู้คืนข้อมูลจากดิสก์ไดรฟ์ที่ล้มเหลวโดยคุณต้องจัดรูปแบบใหม่ หรือเปลี่ยนดิสก์ที่ล้มเหลว

ข้อควรสนใจ: ก่อนที่คุณจะฟอร์แมตดิสก์ไดรฟ์ใหม่หรือเปลี่ยนดิสก์ไดรฟ์ให้ลบการอ้างอิงไปยังระบบไฟล์แบบ non-mirrored จาก ดิสก์ที่ล้มเหลวหรือถอดดิสก์จากกลุ่มวอลุ่มและ การกำหนดคอนฟิกระบบ หากคุณไม่ได้ทำ นั่นหมายถึงคุณสร้างปัญหาขึ้นใน ODM (object data manager) และฐานข้อมูล คอนฟิกูเรชันระบบ คำแนะนำสำหรับขั้นตอนที่จำเป็นเหล่านี้มีการรวมไว้ในโพรซีเจอร์ ต่อไปนี้ ภายใต้อำนาจก่อน การเปลี่ยนหรือการจัดรูปแบบดิสก์ที่ล้มเหลวหรือกำลังจะล้มเหลวอีกครั้ง

โพรซีเจอร์ต่อไปนี้ ใช้สถานการณ์จำลองที่กลุ่มวอลุ่มชื่อ *myvg* มี ดิสก์ไดรฟ์สามตัวที่ชื่อ *hdisk2*, *hdisk3*, และ *hdisk4* ในสถานการณ์จำลองนี้ *hdisk3* เกิดเสียหาย วอลุ่มโลจิคัลที่ไม่ทำมิเรอร์ *lv01* และ สำเนาของวอลุ่มโลจิคัล *mylv* มีอยู่บน *hdisk2* มีการทำมิเรอร์โลจิคัลวอลุ่ม *mylv* และมีสำเนาสามชุด โดยแต่ละชุดใช้พาร์ติชันฟิลิคัลสองพาร์ติชันบนดิสก์ของตน *hdisk3* ที่ล้มเหลว มีสำเนาอื่นของ *mylv* และวอลุ่มโลจิคัล ที่ไม่ทำมิเรอร์ชื่อ *lv00* สุดท้ายนี้ *hdisk4* มี สำเนาที่สามของ *mylv* เช่นเดียวกับวอลุ่มโลจิคัลชื่อ *lv02* รูปภาพต่อไปนี้แสดง สถานการณ์จำลองนี้



โพรซีเจอร์นี้แบ่งออกเป็นเซ็กเมนต์ที่สำคัญดังต่อไปนี้:

- สิ่งที่คุณทำเพื่อป้องกันข้อมูลก่อนคุณเปลี่ยนหรือฟอร์แมต ดิสก์ที่ล้มเหลวใหม่
- โพรซีเจอร์ที่คุณปฏิบัติตามเพื่อฟอร์แมตหรือเปลี่ยนดิสก์ใหม่
- สิ่งที่คุณทำเพื่อกู้คืนข้อมูลหลังจากฟอร์แมตหรือเปลี่ยน ดิสก์แล้ว

ก่อนที่คุณจะเปลี่ยนหรือฟอร์แมต ดิสก์ที่ล้มเหลวหรือกำลังล้มเหลวของคุณใหม่:

1. ให้ล็อกอินด้วยสิทธิในการใช้งานราก
2. หาก你不คุ้นเคยกับโลจิคัลวอลุ่มที่อยู่บนดิสก์ที่จะล้มเหลว ให้ใช้ดิสก์ทำงานเพื่อดูเนื้อหาของดิสก์ที่จะล้มเหลว ตัวอย่างเช่น เมื่อต้องการใช้ *hdisk4* เพื่อ ดู *hdisk3* ให้พิมพ์ต่อไปนี้บน บรรทัดรับคำสั่ง:

```
lspv -M -n hdisk4 hdisk3
```

คำสั่ง *lspv* แสดงข้อมูลเกี่ยวกับฟิลิคัลวอลุ่มภายในกลุ่มวอลุ่ม เอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```
hdisk3:1      mylv:1
hdisk3:2      mylv:2
hdisk3:3      lv00:1
hdisk3:4-50
```

คอลัมน์แรกแสดงพาร์ติชันฟิสิกัล และคอลัมน์ที่สองแสดงพาร์ติชันโลจิคัล พาร์ติชัน 4 ถึง 50 เป็นพาร์ติชันว่าง

3. ทำสำเนาสำรองโลจิคัลวอลุ่มทั้งหมดที่มีสำเนาเพียงชุดเดียวบนอุปกรณ์ที่ล้มเหลว ถ้าเป็นไปได้ หากต้องการคำแนะนำให้ดูที่ “การทำสำเนาสำรองไฟล์ผู้ใช้หรือระบบไฟล์” ในหน้า 29
4. หากคุณมีระบบไฟล์ที่มีสำเนาเพียงชุดเดียวให้ถอนติดตั้งระบบไฟล์นั้นจากดิสก์ (คุณสามารถระบุระบบไฟล์ที่มีสำเนาเพียงชุดเดียวได้จากเอาต์พุตของคำสั่ง `lspv` ระบบไฟล์ที่มีสำเนาเพียงชุดเดียวมีจำนวนของพาร์ติชันโลจิคัลเท่ากับจำนวนของ พาร์ติชันฟิสิกัลบนเอาต์พุต) ไม่จำเป็นต้องถอนติดตั้งระบบไฟล์ที่มีเรอร์

ในสถานการณ์จำลองนี้ `lv00` บนดิสก์ที่ล้มเหลว `hdisk3` เป็นระบบไฟล์ที่มีสำเนาเพียงชุดเดียว เมื่อต้องการถอนติดตั้งระบบไฟล์นั้น ให้พิมพ์ดังต่อไปนี้:

```
umount /dev/lv00
```

ถ้าคุณไม่รู้ชื่อของระบบไฟล์ให้ถือว่าไฟล์ `/etc/filesystems` ไม่ได้อยู่บนดิสก์ที่ล้มเหลวเพียงตัวเดียว พิมพ์ `mount` บน บรรทัดรับคำสั่งเพื่อแสดงรายการไฟล์ที่เม้าท์ทั้งหมดและค้นหาชื่อที่เกี่ยวข้องกับวอลุ่มโลจิคัลของคุณ คุณยังสามารถใช้คำสั่ง `grep` บนไฟล์ `/etc/filesystems` เพื่อแสดงรายการเฉพาะชื่อระบบไฟล์เท่านั้น ถ้ามี ซึ่งเชื่อมโยงกับโลจิคัลวอลุ่มของคุณ ตัวอย่างเช่น:

```
grep lv00 /etc/filesystems
```

The output looks similar to the following example:

```
dev          = /dev/lv00
log          = /dev/loglv00
```

หมายเหตุ:

- a. คำสั่ง `umount` จะล้มเหลว หากระบบไฟล์ ที่คุณพยายามยกเลิกการเม้าท์ถูกใช้งานอยู่ คำสั่ง `umount` จะรันได้ก็ต่อเมื่อไม่มีไฟล์ใดๆ ของระบบไฟล์ เปิดอยู่หรือไม่มีใดเรียกทอริปัจจุบันของผู้ใช้อยู่บนอุปกรณ์
 - b. อีกชื่อหนึ่งสำหรับคำสั่ง `umount` คือ `umount` ชื่อทั้งสองนี้สามารถใช้ทดแทนกันได้
5. ลบระบบไฟล์ที่มีสำเนาเพียงชุดเดียวทั้งหมดออกจากฟิสิกัลวอลุ่มที่ล้มเหลว โดยพิมพ์คำสั่ง `rmfs`:

```
rmfs /FSname
```

6. ลบวอลุ่มโลจิคัลที่ทำเรอร์ไว้ทั้งหมดบนดิสก์ที่กำลังล้มเหลว

หมายเหตุ: คุณไม่สามารถใช้ `rmlvcopy` บนวอลุ่มโลจิคัล `hd5` และ `hd7` จากวอลุ่มฟิสิกัลในวอลุ่ม `rootvg` ได้ ระบบไม่อนุญาตให้คุณลบวอลุ่มเหล่านี้เนื่องจากมีสำเนาเพียงชุดเดียวเท่านั้น

คำสั่ง `rmlvcopy` ลบสำเนาออกจาก แต่ละพาร์ติชันโลจิคัล ตัวอย่างเช่น พิมพ์:

```
rmlvcopy mylv 2 hdisk3
```

โดยการ ลบสำเนาบน `hdisk3` คุณได้ลดจำนวนสำเนาของแต่ละพาร์ติชันที่เป็นของวอลุ่มโลจิคัล `mylv` จากสามเหลือสอง (หนึ่งบน `hdisk4` และ อีกหนึ่งบน `hdisk2`)

7. ถ้าดิสก์ที่กำลังล้มเหลวเป็นส่วนหนึ่งของกลุ่มวอลุ่มรูท และมีวอลุ่มโลจิคัล `hd7` ให้ลบอุปกรณ์ดัมพ์หลัก (`hd7`) ออกโดยพิมพ์ต่อไปนี้ บนบรรทัดรับคำสั่ง:

```
sysdumpdev -P -p /dev/sysdumpnul
```

คำสั่ง `sysdumpdev` เปลี่ยนที่ตั้งอุปกรณ์ดัมพ์หลักหรือรองสำหรับระบบที่กำลังรัน เมื่อคุณรีบูต อุปกรณ์ดัมพ์จะกลับไปยังตำแหน่งดั้งเดิม

หมายเหตุ: คุณสามารถเลือกดัมพ์ลงอุปกรณ์ DVD ได้สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีกำหนดคอนฟิก DVD ให้เป็นอุปกรณ์ดัมพ์โปรดดูที่ `sysdumpdev`

8. ลบพื้นที่การเลื่อนหน้าบนดิสก์โดยใช้คำสั่งต่อไปนี้:

```
rmrps PSname
```

โดยที่ `PSname` คือชื่อของพื้นที่การเลื่อนหน้าที่จะลบออก ซึ่งแท้จริงแล้ว เป็นชื่อของโลจิคัลวอลุ่มซึ่งพื้นที่การเลื่อนหน้าตั้งอยู่

ถ้าคำสั่ง `rmrps` ไม่สำเร็จ คุณต้องใช้พารามิเตอร์ `smit chps` เพื่อปิดการเรียกใช้พื้นที่การเลื่อนหน้าและรีบูตก่อนที่คุณจะดำเนินการโพธิ์เตอร์ต่อไป คำสั่ง `reducevg` ในขั้นตอนที่ 10 อาจล้มเหลวได้ถ้า มีพื้นที่การเลื่อนหน้าที่ใช้งานอยู่

9. ลบวอลุ่มโลจิคัลอื่นจากกลุ่มวอลุ่ม เช่น วอลุ่มโลจิคัลที่ไม่มีระบบไฟล์ โดยใช้คำสั่ง `rm1v` ตัวอย่าง เช่น พิมพ์:

```
rm1v -f lv00
```

10. ลบดิสก์ที่ล้มเหลวออกจาก กลุ่มวอลุ่มโดยใช้คำสั่ง `reducevg` ตัวอย่าง เช่น พิมพ์:

```
reducevg -df myvg hdisk3
```

ถ้าคุณไม่สามารถรันคำสั่ง `reducevg` หรือ ถ้าคำสั่งไม่สำเร็จ โพธิ์เตอร์ในขั้นตอนที่ 13 สามารถช่วย ล้างข้อมูล VGDA/ODM หลังจากที่คุณฟอร์แมตใหม่หรือเปลี่ยนไดรฟ์

การเปลี่ยนหรือการจัดรูปแบบดิสก์ที่ล้มเหลวหรือกำลัง จะล้มเหลวอีกครั้ง:

11. ขั้นตอนถัดไปขึ้นอยู่กับว่าคุณต้องการ จัดรูปแบบใหม่หรือเปลี่ยนดิสก์ และชนิดของฮาร์ดแวร์ซึ่งคุณ ใช้อยู่:

- ถ้าคุณต้องการจัดรูปแบบดิสก์ไดรฟ์อีกครั้ง ให้ใช้โพธิ์เตอร์ต่อไปนี้:

- a. ด้วยสิทธิการใช้งานราก ให้พิมพ์พารามิเตอร์ SMIT ต่อไปนี้บนบรรทัด คำสั่ง:

```
smit diag
```

- b. เลือก การวินิจฉัยเซลล์ปัจจุบัน เพื่อป้อนเครื่องมือการวินิจฉัย AIX
- c. หลังจากคุณอ่านหน้าจอ คำแนะนำในการใช้การวินิจฉัย ให้กด Enter
- d. เลือก การเลือกภารกิจ
- e. เลื่อนหน้าจอลงผ่านรายการภารกิจเพื่อค้นหาและเลือก จัดรูปแบบ สื่อบันทึก
- f. เลือกดิสก์ที่คุณต้องการฟอร์แมตใหม่ หลังจากคุณยืนยันว่าคุณต้องการ จัดรูปแบบดิสก์อีกครั้ง เนื้อหาทั้งหมดบนดิสก์จะถูกลบออก

หลังจากจัดรูปแบบดิสก์อีกครั้งแล้ว ให้ทำขั้นตอนที่ 12 ต่อไป

- หากระบบของคุณสนับสนุนดิสก์ hot swap ให้ใช้โพธิ์เตอร์ใน “การกู้คืนจากความล้มเหลวดิสก์ในขณะที่ระบบยังคงพร้อมใช้งาน” ในหน้า 416 จากนั้นทำขั้นตอนที่ 13 ต่อไป
- ถ้าระบบของคุณไม่สนับสนุนดิสก์ hot swap ให้ทำตามขั้นตอน ต่อไปนี้:
 - ปิดไดรฟ์เก่าโดยใช้พารามิเตอร์ SMIT `smit rmvdsk` เปลี่ยนนิยาม KEEP ในฟิลด์ฐานข้อมูลเป็น No
 - โปรดติดต่อการสนับสนุนระบบระดับถัดไปเพื่อเปลี่ยนดิสก์ไดรฟ์

หลังจากที่คุณเปลี่ยนหรือฟอร์แมตดิสก์ที่ล้มเหลวหรือกำลังล้มเหลวใหม่:

12. ปฏิบัติตามคำแนะนำใน “การกำหนดคอนฟิกดิสก์” ในหน้า 398 และ “การทำให้ดิสก์ที่มีอยู่เป็นฟิสิคัลวอลุ่ม” ในหน้า 400
13. ถ้าคุณไม่สามารถใช้คำสั่ง `reducevg` บนดิสก์จากกลุ่มวอลุ่มเก่าก่อนที่ดิสก์จะถูกฟอร์แมต (ขั้นตอนที่ 10) โพธิ์เตอร์ต่อไปนี้สามารถช่วยล้างข้อมูล VGDA/ODM ได้

- a. หากกลุ่มวอลุ่มที่มีเพียงดิสก์เดียวซึ่งถูกจัดรูปแบบใหม่ให้พิมพ์:

```
exportvg VGName
```

โดยที่ *VGName* คือ ชื่อกลุ่มวอลุ่มของคุณ

- b. ถ้ากลุ่มวอลุ่มมีมากกว่าหนึ่งดิสก์ให้พิมพ์ดังต่อไปนี้บน บรรทัดคำสั่ง:

```
varyonvg VGName
```

ระบบ แสดงข้อความเกี่ยวกับดิสก์ที่ขาดไปหรือไม่อยู่ และมีการแสดงรายการดิสก์ใหม่ (หรือที่จัดรูปแบบใหม่) จดบันทึก physical volume identifier (PVID) ของดิสก์ใหม่ ซึ่งแสดงรายการอยู่ในข้อความ varyonvg PVID เป็นสตริง 16-อักขระระหว่างชื่อของดิสก์ที่ขาดไปและ เลเบล PVNOTFND ตัวอย่างเช่น:

```
hdisk3 00083772caa7896e PVNOTFND
```

โดยพิมพ์:

```
varyonvg -f VGName
```

ขณะนี้ ดิสก์ที่ขาดไปแสดงขึ้นพร้อมกับเลเบล PVREMOVED ตัวอย่างเช่น:

```
hdisk3 00083772caa7896e PVREMOVED
```

จากนั้น พิมพ์คำสั่ง:

```
reducevg -df VGName PVID
```

โดยที่ PVID เป็นตัวระบุฟิสิกัลวอลุ่ม (ในสถานการณ์จำลองนี้คือ 00083772caa7896e)

14. เมื่อต้องการเพิ่มดิสก์ไดรฟ์ใหม่ลงในกลุ่มวอลุ่ม ให้ใช้คำสั่ง `extendvg` ตัวอย่าง เช่น พิมพ์:

```
extendvg myvg hdisk3
```

15. เมื่อต้องการสร้างโลจิคัลวอลุ่มที่มีสำเนาเพียงชุดเดียวขึ้นใหม่บนดิสก์ไดรฟ์ใหม่ (หรือที่จัดรูปแบบ) ให้ใช้คำสั่ง `mk1lv` ตัวอย่าง เช่น พิมพ์:

```
mk1lv -y lv00 myvg 1 hdisk3
```

ตัวอย่างนี้สร้างวอลุ่มโลจิคัล lv00 ขึ้นใหม่บนไดรฟ์ *hdisk3* 1 หมายความว่า โลจิคัลวอลุ่มนี้ไม่มีการมีร์เรอร์

16. เมื่อต้องการสร้างระบบไฟล์ขึ้นใหม่บนโลจิคัลวอลุ่ม ให้ใช้คำสั่ง `crfs` ตัวอย่าง เช่น พิมพ์:

```
crfs -v jfs -d lv00 -m /dev/lv00
```

17. เมื่อต้องการเรียกคืนข้อมูลระบบไฟล์ที่มีสำเนาเพียงชุดเดียวจากสื่อบันทึกสำเนาสำรอง ให้ดูที่ “การเรียกคืนไฟล์ผู้ใช้จากรูปภาพสำเนาสำรอง” ในหน้า 34

18. เมื่อต้องการสร้างสำเนาที่มีร์เรอร์ของโลจิคัลวอลุ่มขึ้นใหม่ ให้ใช้คำสั่ง `mk1lvcopy` ตัวอย่าง เช่น พิมพ์:

```
mk1lvcopy mylv 3 hdisk3
```

ตัวอย่างนี้สร้างพาร์ติชันที่สามารถทำมีร์เรอร์ไว้ของวอลุ่มโลจิคัล *mylv* บน *hdisk3*

19. เมื่อต้องการซิงโครไนซ์มีร์เรอร์ใหม่กับข้อมูลบนมีร์เรอร์อื่น (ในตัวอย่างนี้ *hdisk2* และ *hdisk4*), ใช้คำสั่ง `syncvg` ตัวอย่าง เช่น พิมพ์:

```
syncvg -p hdisk3
```

ตั้งผลลัพธ์ระบบไฟล์ที่ทำมีร์เรอร์ไว้ทั้งหมดต้องถูกเรียกคืน และทันสมัย ถ้าคุณสามารถสำรองข้อมูลระบบไฟล์แบบ single-copy ของคุณได้ ระบบไฟล์พร้อมใช้งานแล้ว คุณต้องสามารถดำเนินการต่อการใช้ระบบปกติ

งานที่เกี่ยวข้อง:

“การแก้ไขปัญหาดิสก์ไดรฟ์” ในหน้า 408

ข้อมูลนี้แนะนำวิธีการวินิจฉัยและแก้ไขปัญหาดิสก์ไดรฟ์

ตัวอย่างของการกู้คืนจากดิสก์ไดรฟ์ที่ล้มเหลว:

เมื่อต้องการกู้คืนจากดิสก์ไดรฟ์ที่ล้มเหลว ให้ย้อนกลับวิธีที่คุณเข้ามา นั่นคือ แสดงรายการขั้นตอนซึ่งคุณผ่านมาในการสร้างกลุ่มวอลุ่ม จากนั้น เดินย้อนกลับ

ตัวอย่างต่อไปนี้เป็น การสาธิตเทคนิคนี้ ตัวอย่างแสดง วิธีการสร้างโลจิคัลวอลุ่มที่มีร์เรอร์ ตามด้วยวิธีการเปลี่ยน ย้อนกลับ ที่ละหนึ่งขั้นตอน เมื่อดิสก์ล้มเหลว

หมายเหตุ: ตัวอย่างต่อไปนี้แสดง อินสแตนซ์เฉพาะ และไม่ได้ใช้เป็นต้นแบบทั่วไปซึ่งใช้ในการสร้าง โปรซีเจอร์การกู้คืนทั่วไปใดๆ

1. ผู้จัดการระบบที่ชื่อ Jane ได้สร้างกลุ่มวอลุ่มที่เรียกว่า workvg บน hdisk1 โดยพิมพ์:

```
mkvg -y workvg hdisk1
```

2. จากนั้น เธอได้สร้างดิสก์เพิ่มเติมขึ้นสองดิสก์สำหรับกลุ่มวอลุ่มนี้ โดยพิมพ์:

```
extendvg workvg hdisk2
```

```
extendvg workvg hdisk3
```

3. Jane สร้างโลจิคัลวอลุ่มขนาด 40 MB ที่มีสำเนาสามชุด สำเนาแต่ละชุดอยู่บนดิสก์ใดดิสก์หนึ่งจากสามดิสก์ซึ่งประกอบเป็นกลุ่มวอลุ่ม workvg เธอใช้คำสั่งต่อไปนี้:

```
mk1lv -y test1v workvg 10
```

```
mk1lvcopy test1v 3
```

หลังจาก Jane สร้างกลุ่มวอลุ่ม workvg ที่มีร์เรอร์แล้ว hdisk2 ล้มเหลว เธอจึงใช้ขั้นตอนต่อไปนี้เพื่อกู้คืน:

1. เธอลบสำเนาโลจิคัลวอลุ่มออกจาก hdisk2 โดยการพิมพ์:

```
rmlvcopy test1v 2 hdisk2
```

2. เธอเอา hdisk2 ออกจากระบบเพื่อให้มีการอัปเดต ODM และ VGDA โดยพิมพ์:

```
reducevg workvg hdisk2
```

3. เธอลบ hdisk2 ออกจากคอนฟิกูเรชันระบบเพื่อจัดเตรียมสำหรับการแทนที่โดยพิมพ์:

```
rmdev -l hdisk2 -d
```

4. เธอเลือกที่จะปิดระบบ โดยพิมพ์:

```
shutdown -F
```

5. เธอเปลี่ยนดิสก์ ดิสก์ใหม่ไม่ได้มี SCSI ID เหมือนกับ hdisk2 ก่อนหน้านี้

6. เธอรีบูตระบบ

เนื่องจากคุณมีดิสก์ใหม่ (ระบบเห็นว่า มี PVID ใหม่บนดิสก์นี้) ระบบจึงเลือกชื่อ hdisk เปิด ลำดับแรก เนื่องจากใช้แฟล็ก -d ในขั้นตอนที่ 3 ชื่อ hdisk2 จึงถูกรีเส็ต ดังนั้นระบบจึงเลือก hdisk2 เป็นชื่อของ ดิสก์ใหม่ ถ้าไม่ได้ใช้แฟล็ก -d จะมีการเลือก hdisk4 เป็นชื่อใหม่

7. Jane เพิ่มดิสก์นี้ลงในกลุ่มวอลุ่ม workvg โดยพิมพ์:

```
extendvg workvg hdisk2
```

8. เธอสร้างสำเนาที่มีเรอร์ของโลจิคัลวอลุ่มจำนวนสองชุด โดยพิมพ์:

```
mkivcopy testlv 3
```

Logical Volume Manager วางสำเนาโลจิคัลวอลุ่มชุดที่สาม บน hdisk2 ใหม่โดยอัตโนมัติ

การกู้คืนจากความล้มเหลวดิสก์ในขณะที่ระบบยังคง พร้อมใช้งาน:

คุณสามารถกู้คืนจากความล้มเหลวดิสก์โดยใช้คุณลักษณะ hot removability

โพรซีเจอร์ในการกู้คืนจากความล้มเหลวดิสก์โดยใช้คุณลักษณะ hot removability โดยเนื้อหาส่วนใหญ่แล้วเหมือนกับที่อธิบายไว้ใน “การกู้คืนดิสก์ไดรฟ์โดยไม่ต้องจัดรูปแบบใหม่” ในหน้า 410 แต่มีข้อยกเว้นต่อไปนี้:

1. เมื่อต้องการถอนติดตั้งระบบไฟล์บนดิสก์ให้ใช้โพรซีเจอร์ติดตั้ง JFS หรือ JFS2
2. เมื่อต้องการเอาดิสก์ออกจากกลุ่มวอลุ่มและออกจากระบบปฏิบัติการให้ใช้โพรซีเจอร์ “การเอาดิสก์ที่ไม่มีข้อมูลออก” ในหน้า 436
3. เมื่อต้องการเปลี่ยนดิสก์ที่ล้มเหลวด้วยดิสก์ใหม่ คุณไม่ต้อง ปิดระบบ ใช้ลำดับของโพรซีเจอร์ต่อไปนี้:
 - a. “หน่วยเก็บโลจิคัลวอลุ่ม” ในหน้า 421
 - b. “การกำหนดคอนฟิกดิสก์” ในหน้า 398
 - c. ทำขั้นตอนที่ 13 ของ “การกู้คืนโดยใช้ดิสก์ไดรฟ์ที่ฟอร์แมตใหม่หรือเปลี่ยนดิสก์ไดรฟ์” ในหน้า 411 ต่อไป

งานที่เกี่ยวข้อง:

“การแก้ไขปัญหาดิสก์ไดรฟ์” ในหน้า 408

ข้อมูลนี้แนะนำวิธีการวินิจฉัยและแก้ไขปัญหาดิสก์ไดรฟ์

การเปลี่ยนดิสก์เมื่อกลุ่มวอลุ่มมีหนึ่งดิสก์:

ใช้โพรซีเจอร์อย่างใดอย่างหนึ่งต่อไปนี้ถ้าคุณสามารถเข้าถึงดิสก์ที่กำลังจะไม่ดีและเป็นส่วนหนึ่งของกลุ่มวอลุ่ม

- “การย้ายเนื้อหาของฟิสิคัลวอลุ่ม” ในหน้า 396

หากดิสก์ไม่ดีและไม่สามารถเข้าถึงได้ให้ปฏิบัติตามขั้นตอนเหล่านี้:

1. เอ็กซ์พอร์ตกลุ่มวอลุ่ม
2. เปลี่ยนไดรฟ์
3. สร้างข้อมูลขึ้นใหม่จากสื่อบันทึกสำเนาสำรองที่มีอยู่

ข้อผิดพลาดฟิสิคัลและโลจิคัลวอลุ่ม:

มีข้อผิดพลาดทั่วไปหลายอย่างเกี่ยวกับฟิสิคัลและโลจิคัลวอลุ่มซึ่ง คุณสามารถแก้ไขได้

ปัญหา hot spot:

หากคุณสังเกตเห็นประสิทธิภาพที่ลดลงเมื่อเข้าถึงโลจิคัลวอลุ่ม แสดงว่าคุณอาจมี hot spots ในโลจิคัลวอลุ่มที่มีดีสก์ I/O มากเกินไป

สำหรับข้อมูลเพิ่มเติม โปรดดูที่ “การจัดการ hot spot ในโลจิคัลวอลุ่ม” ในหน้า 451

คำเตือน LVCB:

จะมีคำเตือนถ้า LVCB มีข้อมูลที่ไม่ถูกต้อง

บล็อกควบคุมวอลุ่มโลจิคัล (LVCB) คือบล็อกแรก ของวอลุ่มโลจิคัล ขนาดของ LVCB คือขนาดบล็อก ของวอลุ่มฟิสิคัลภายในกลุ่มวอลุ่ม พื้นที่นี้จัดเก็บข้อมูลที่สำคัญ เช่น วันที่สร้าง โลจิคัลวอลุ่ม ข้อมูลเกี่ยวกับสำเนาที่มิเรอร์ และจุดต่อที่เป็นไปได้ ใน JFS ต้องใช้คำสั่ง LVM บางรายการเพื่ออัปเดต LVCB เป็นส่วนหนึ่งของขั้นตอนวิธีใน LVM LVCB เก่าถูกอ่านและวิเคราะห์เพื่อดูว่า ถูกต้องหรือไม่ หากข้อมูลเป็นข้อมูล LVCB ที่ถูกต้อง จะมีการอัปเดต LVCB หากข้อมูลไม่ถูกต้อง จะไม่มีการอัปเดต LVCB และคุณอาจได้รับข้อความต่อไปนี้:

```
Warning, cannot write lv control block data.
```

ในเวลาส่วนใหญ่ ข้อความนี้เกิดขึ้นเมื่อโปรแกรมฐานข้อมูลบายพาส JFS และเข้าถึงโลจิคัลวอลุ่มแบบดิบเป็นสื่อบันทึกหน่วยเก็บ เมื่อเกิดกรณีเช่นนี้ ข้อมูล ของฐานข้อมูลจะถูกเขียนทับตามตัวอักษรบน LVCB สำหรับโลจิคัลวอลุ่มแบบดิบ นี้ไม่ใช่กรณีร้ายแรง หลังจาก LVCB ถูกเขียนทับแล้ว ผู้ใช้ยังคงสามารถ:

- ขยายโลจิคัลวอลุ่ม
- สร้างสำเนาที่มิเรอร์ของโลจิคัลวอลุ่ม
- ลบโลจิคัลวอลุ่ม
- สร้างระบบไฟล์ที่เจอร์นัลเพื่อติดตั้งโลจิคัลวอลุ่ม

มีข้อจำกัดในการลบ LVCBs วอลุ่มโลจิคัลพร้อม LVCB ที่ถูกลบอาจไม่สามารถอิมพอร์ตไปยังระบบอื่นได้สมบูรณ์ ในระหว่างอิมพอร์ต คำสั่ง LVM `importvg` จะสแกน LVCBs ของโลจิคัลวอลุ่มที่กำหนด ไว้ทั้งหมดในกลุ่มวอลุ่มเพื่อหาข้อมูลเกี่ยวกับโลจิคัลวอลุ่ม หากไม่มี LVCB กลุ่มวอลุ่มที่อิมพอร์ตยังคงกำหนดโลจิคัลวอลุ่ม ให้กับระบบใหม่ที่กำลังเข้าถึงกลุ่มวอลุ่มนี้ และผู้ใช้ยังคงสามารถ เข้าถึงโลจิคัลวอลุ่มแบบดิบได้ อย่างไรก็ตาม โดยปกติแล้ว จะเกิด ดังต่อไปนี้:

- ข้อมูล JFS สูญหายไปและไม่อิมพอร์ตจุดต่อที่เชื่อมโยง เข้าในระบบใหม่ ในกรณีนี้ คุณต้องสร้างจุดต่อใหม่ และไม่มีารรับประกันถึงการมีอยู่ ของข้อมูลก่อนหน้านี้ที่จัดเก็บไว้ในระบบไฟล์
- ไม่พบข้อมูลที่ไม่เกี่ยวข้องกับ JFS บางส่วนซึ่งเกี่ยวข้องกับโลจิคัลวอลุ่ม เมื่อเกิดกรณีเช่นนี้ ระบบจะใช้ข้อมูลดีฟอลต์โลจิคัลวอลุ่มในการแสดง ข้อมูล ODM ดังนั้น เอาต์พุตบางรายการจากคำสั่ง `lsiv` จึงอาจไม่สอดคล้องกับโลจิคัลวอลุ่มจริง หากยังคงมีสำเนาโลจิคัลวอลุ่มใดๆ อยู่บนดีสก์ดั้งเดิม จะไม่มีการสะท้อนถึงข้อมูลอย่างถูกต้องใน ฐานข้อมูล ODM ใช้คำสั่ง `rmlvcopy` และ `mklvcopy` เพื่อสร้างสำเนาโลจิคัลวอลุ่มขึ้นใหม่และซิงโครไนซ์ ODM

ขีดจำกัดพาร์ติชันฟิสิคัล:

ในการออกแบบของ Logical Volume Manager (LVM) แต่ละพาร์ติชัน โลจิคัลจะแม็ปเข้ากับหนึ่งพาร์ติชันฟิสิคัล (PP) และแต่ละพาร์ติชันฟิสิคัลแม็ปเข้ากับ ดิสก์เซกเตอร์จำนวนหนึ่ง การออกแบบของ LVM จำกัดจำนวนของพาร์ติชันฟิสิคัล ที่ LVM สามารถติดตามต่อดิสก์เป็น 1016 ในกรณีส่วนใหญ่ ดิสก์ไม่ได้ใช้ พาร์ติชันการติดตามครบทั้ง 1016 พาร์ติชัน

เมื่อใช้งานเกินกว่าขีดจำกัดนี้ คุณจะเห็นข้อความคล้ายกับ ต่อไปนี้:

```
0516-1162 extndvlg: Warning, The Physical Partition Size of PPSize requires the
creation of TotalPPS partitions for PVname. The limitation for volume group
VGname is LIMIT physical partitions per physical volume. Use chvg command
with -t option to attempt to change the maximum Physical Partitions per
Physical volume for this volume group.
```

โดยที่:

PPsize เป็น 1 MB ถึง 1 GB ในกำลัง 2

Total PPs

เป็นจำนวนพาร์ติชันฟิสิกส์ทั้งหมดบนดิสก์นี้ ในหน่วย *PPsize*

PVname

เป็นชื่อของฟิสิกส์ลวลุ่ม ตัวอย่างเช่น `hdisk3`

VGname

เป็นชื่อของกลุ่มลวลุ่ม

LIMIT เป็น 1016 หรือผลคูณของ 1016

ขีดจำกัดนี้มีการบังคับใช้ในอินสแตนซ์ต่อไปนี้:

1. เมื่อสร้างกลุ่มลวลุ่มโดยใช้คำสั่ง `mkvg` คุณระบุจำนวนของพาร์ติชันฟิสิกส์บนดิสก์ในกลุ่มลวลุ่ม ที่เกินกว่า 1016 เพื่อหลีกเลี่ยงข้อจำกัดนี้ คุณสามารถเลือกจากช่วงขนาดพาร์ติชันฟิสิกส์ คือ 1, 2, 4 (ดีฟอลต์), 8, 16, 32, 64, 128, 256, 512 หรือ 1024 MB และใช้คำสั่ง `mkvg -s` เพื่อสร้าง กลุ่มลวลุ่ม หรือคุณสามารถใช้ตัวประกอบที่เหมาะสมที่อนุญาต ผลคูณของ 1016 พาร์ติชันต่อดิสก์ และใช้คำสั่ง `mkvg -t` เพื่อสร้างกลุ่มลวลุ่ม
2. เมื่อเพิ่มดิสก์ลงในกลุ่มลวลุ่มที่มีอยู่ก่อนแล้วโดยใช้คำสั่ง `extndvlg` ดิสก์ใหม่ทำให้เกิดการละเมิดขีดจำกัด 1016 เพื่อแก้ไขสถานการณ์นี้ ให้แปลงกลุ่มลวลุ่มที่มีอยู่เพื่อจำกัดเก็บผลคูณของ 1016 พาร์ติชันต่อ ดิสก์โดยใช้คำสั่ง `chvg -t` หรือ คุณสามารถสร้างกลุ่มลวลุ่มขึ้นใหม่โดยมีขนาดพาร์ติชันใหญ่ขึ้นซึ่งรองรับ ดิสก์ใหม่ได้ หรือคุณสามารถสร้างกลุ่มลวลุ่มแบบสแตนด์อโลนที่มี ขนาดทางกายภาพใหญ่ขึ้นสำหรับดิสก์ใหม่

ขีดจำกัดพาร์ติชันและ rootvg

หากไค้ดการติดตั้ง ตรวจสอบว่า `rootvg` ไดรฟ์ใหญ่กว่า 4 GB ไค้ดการติดตั้งจะเปลี่ยนค่า `mkvg-s` จนกว่าสามารถแก้พความจุติสก์ทั้งหมดเข้ากับ 1016 แทร็กที่มีอยู่ การเปลี่ยนแปลงการติดตั้งนี้ยังหมายความว่ ดิสก์อื่นทั้งหมดที่เพิ่มลงใน `rootvg` ไม่ว่าจะมืขนาดเท่าใดก็ตาม ยังมีการกำหนดที่ขนาดพาร์ติชันฟิสิกส์นั้นด้วย

ขีดจำกัดพาร์ติชันและระบบ RAID

สำหรับระบบ ที่ใช้ redundant array of identical disks (RAID) ชื่อ `/dev/hdiskX` ที่ใช้โดย LVM อาจประกอบด้วยดิสก์ที่ไม่ใช่ 4 GB จำนวนมาก ในกรณีนี้ ข้อกำหนด 1016 ยังคงมีอยู่ LVM ไม่รับรู้ขนาดของแต่ละดิสก์ที่ประกอบขึ้นเป็น `/dev/hdiskX` โดยแท้จริง LVM ใช้ขีดจำกัด 1016 ตามขนาดที่รับรู้ของ `/dev/hdiskX` ไม่ใช่ฟิสิกส์ดิสก์จริง ที่ประกอบขึ้นเป็น `/dev/hdiskX`

การซิงโครไนซ์ฐานข้อมูลคอนฟิกูเรชันอุปกรณ์:

ความบกพร่องของระบบอาจส่งผลให้ฐานข้อมูลคอนฟิกูเรชันอุปกรณ์ ไม่สอดคล้องกับ LVM คุณสามารถซิงโครไนซ์ฐานข้อมูลคอนฟิกูเรชันอุปกรณ์ กับข้อมูล LVM ได้

เมื่อฐานข้อมูลคอนฟิกูเรชันอุปกรณ์ไม่สอดคล้องกับ LVM คำสั่ง `losetup` จะสร้างข้อความแสดงข้อผิดพลาดเช่น:

```
0516-322 The Device Configuration Database is inconsistent ...
```

หรือ

```
0516-306 Unable to find logical volume LVname in the Device Configuration Database.
```

(โดยที่ `losetup` ชื่อที่เรียกว่า `LVname` มีอยู่ตามปกติ)

ข้อควรสนใจ: อย่าลบรายการ `/dev` ของกลุ่มวอลุ่มหรือโลจิคัลวอลุ่ม อย่าเปลี่ยนรายการฐานข้อมูลของ กลุ่มวอลุ่มหรือโลจิคัลวอลุ่มโดยใช้ Object Data Manager

เมื่อต้องการซิงโครไนซ์ฐานข้อมูลคอนฟิกูเรชันอุปกรณ์กับข้อมูล LVM ด้วยสิทธิการใช้งานราก ให้พิมพ์ดังต่อไปนี้บนบรรทัดคำสั่ง:

```
synclvodm -v VGName
```

โดยที่ `VGName` คือชื่อของกลุ่มวอลุ่มซึ่งคุณต้องการซิงโครไนซ์

ข้อมูลที่เกี่ยวข้อง:

“การแก้ไขข้อผิดพลาดกลุ่มวอลุ่ม”

ใช้เมธอดเหล่านี้เพื่อแก้ไขข้อผิดพลาดกลุ่มวอลุ่ม

การแก้ไขข้อผิดพลาดกลุ่มวอลุ่ม:

ใช้เมธอดเหล่านี้เพื่อแก้ไขข้อผิดพลาดกลุ่มวอลุ่ม

ถ้าคำสั่ง `importvg` ทำงานไม่ถูกต้อง ให้ลอง รีเฟรชฐานข้อมูลคอนฟิกูเรชันอุปกรณ์

การยกเลิกความล้มเหลวแปรผัน

ข้อควรสนใจ: การยกเลิก ความล้มเหลวแปรผันเป็นการดำเนินงานพิเศษ ซึ่งตรวจสอบหาต้นเหตุของปัญหาอื่น ทั้งหมดที่อาจเป็นไปได้ เช่น ฮาร์ดแวร์ สายเคเบิล อะแดปเตอร์ และแหล่งจ่ายไฟ ก่อนทำต่อไป การยกเลิกความล้มเหลวองค์ประกอบในระหว่างโปรเซสแปรผันใช้เฉพาะในกรณีฉุกเฉิน และใช้เป็นทางเลือกสุดท้ายเท่านั้น (ตัวอย่างเช่น การทำลายข้อมูลจากดิสก์ที่ล้มเหลว) ไม่สามารถรับประกันถึงบูรณภาพของข้อมูลสำหรับข้อมูลการจัดการที่มีอยู่ในสำเนา ที่เลือกของ VGDA และ VGSA เมื่อยกเลิกความล้มเหลวองค์ประกอบ

เมื่อคุณ เลือกที่จะบังคับใช้แปรผันกลุ่มวอลุ่มโดยการยกเลิกการขาดองค์ประกอบ PV STATE ของฟิสิคัลวอลุ่มทั้งหมดที่ขาดไปในระหว่างโปรเซสแปรผันนี้ จะถูกเปลี่ยนเป็นลบออก ซึ่งหมายความว่าสำเนา VGDA และ VGSA ทั้งหมดจะถูกลบออกจากฟิสิคัลวอลุ่มเหล่านี้ หลังจากทำเช่นนี้แล้ว ฟิสิคัลวอลุ่ม เหล่านี้จะไม่มีส่วนในการตรวจสอบองค์ประกอบอีกต่อไป และไม่ได้รับอนุญาตให้ใช้งานภายในกลุ่มวอลุ่มจนกว่าคุณส่งคืนไปยัง กลุ่มวอลุ่ม แพลลิก `varyonvg -f` (ที่ใช้เพื่อยกเลิกองค์ประกอบที่หายไป) จะถูกละเว้น ถ้ากลุ่มวอลุ่มไม่มีองค์ประกอบที่หายไป

ภายใต้เงื่อนไขหนึ่งหรือหลายอย่างต่อไปนี้ คุณอาจต้องการยกเลิกความล้มเหลวแปรผันเพื่อให้ข้อมูล บนดิสก์ที่มีอยู่ในกลุ่มวอลุ่มสามารถเข้าถึงได้:

- ฟิสิคัลวอลุ่มที่ไม่มีอยู่เสียหายอย่างถาวร

- คุณสามารถยืนยันว่าอย่างน้อยหนึ่งฟิสิคัลวอลุ่มที่เข้าถึงได้ในปัจจุบัน (ซึ่งต้องมีสำเนา VGDA และ VGSA ที่ดีด้วย) ออนไลน์เมื่อ varied on กลุ่มวอลุ่มครั้งล่าสุด ยกเลิกกำหนดคอนฟิกและปิดฟิสิคัลวอลุ่มที่หายไป จนกว่าสามารถวินิจฉัยและซ่อมแซมฟิสิคัลวอลุ่มนั้นได้

ใช้ไพรซีเตอร์ต่อไปนี้เพื่อหลีกเลี่ยงองค์ประกอบที่หายไป เมื่อดิสก์หนึ่ง หายไปหรืออาจล้มเหลวในไม่ช้าและต้องซ่อมแซม:

1. เมื่อต้องการลบวอลุ่มออกจากกลุ่มวอลุ่มเป็นการชั่วคราว ให้พิมพ์:

```
chpv -vr PVname
```

เมื่อคำสั่งนี้เสร็จแล้ว ฟิสิคัลวอลุ่ม *PVname* จะไม่มีส่วนในการตรวจสอบองค์ประกอบอีกต่อไป อย่างไรก็ตาม ในกลุ่มวอลุ่มสองดิสก์ คำสั่งนี้จะล้มเหลวหากคุณพยายามใช้คำสั่ง **chpv** บนดิสก์ที่มีสอง VGDA/VGSAs คำสั่งไม่อนุญาตให้คุณทำให้องค์ประกอบ หายไป

2. หากคุณต้องการเอาดิสก์ออกเพื่อซ่อมแซม ให้ปิดระบบ และถอด ดิสก์ออก (สำหรับคำแนะนำ ให้ดูที่ “การแก้ไขปัญหาหาคิสก์ไดรฟ์” ในหน้า 408) หลังจากแก้ไขดิสก์และติดตั้งดิสก์กลับเข้าในระบบแล้ว ให้ทำขั้นตอน ถัดไป
3. เมื่อต้องการทำให้ดิสก์พร้อมใช้งานในกลุ่มวอลุ่มอีกครั้งสำหรับการตรวจสอบ องค์ประกอบ ให้พิมพ์:

```
chpv -v a PVname
```

หมายเหตุ: คำสั่ง **chpv** ใช้สำหรับการเปลี่ยนการตรวจสอบองค์ประกอบเท่านั้น ข้อมูลที่อยู่บนดิสก์ ยังคงอยู่ที่นั่น และต้องย้ายหรือคัดลอกไปยังดิสก์อื่นถ้าจะไม่ติดตั้งดิสก์กลับ เข้าไปในระบบ

คำเตือน VGDA

ในบางอินสแตนซ์ ผู้ใช้พบปัญหา ในการเพิ่มดิสก์ใหม่ลงในกลุ่มวอลุ่มที่มีอยู่ หรือในการสร้าง กลุ่มวอลุ่มใหม่ ข้อความที่แสดงขึ้นโดย LVM คือ:

```
0516-1163 extendvg: VGname already has maximum physical volumes. With the maximum number of physical partitions per physical volume being LIMIT, the maximum number of physical volumes for volume group VGname is MaxDisks.
```

โดยที่:

VGname

เป็นชื่อของกลุ่มวอลุ่ม

LIMIT เป็น 1016 หรือผลคูณของ 1016

MaxDisks

เป็นจำนวนดิสก์สูงสุดในกลุ่มวอลุ่ม ตัวอย่างเช่น ถ้ามี 1016 พาร์ติชันฟิสิคัล (PPs) ต่อดิสก์ *MaxDisk* จะเป็น 32; ถ้าเป็น 2032 *MaxDisk* จะเป็น 16

คุณสามารถแก้ไขไฟล์ `image.data` จากนั้นใช้การติดตั้งดิสก์อื่น หรือเรียกคืนระบบโดยใช้คำสั่ง **mksysb** เพื่อสร้างกลุ่มวอลุ่มขึ้นใหม่เป็นชนิดกลุ่มวอลุ่มขนาดใหญ่ สำหรับข้อมูลเพิ่มเติม ให้ดูที่ *Installation and migration*

บน AIX เวอร์ชันเก่า เมื่อขีดจำกัดน้อยกว่า 32 ดิสก์ ข้อยกเว้นสำหรับคำอธิบาย VGDA สูงสุดนี้คือ `rootvg` เพื่อให้ผู้ใช้มีพื้นที่ว่างดิสก์ มากขึ้น เมื่อสร้าง `rootvg` คำสั่ง **mkvg -d** ใช้จำนวนดิสก์ที่เลือกไว้ในเมนูการติดตั้งเป็นตัวเลข อ้างอิง ตัวเลข **-d** นี้คือ 7 สำหรับหนึ่งดิสก์และเพิ่มขึ้นอีกหนึ่ง สำหรับแต่ละดิสก์เพิ่มเติมที่เลือก ตัวอย่างเช่น ถ้าเลือกสองดิสก์ ตัวเลขจะเป็น 8 และถ้าเลือกสามดิสก์ ตัวเลขจะเป็น 9 และต่อไป

หลักการที่เกี่ยวข้อง:

“การชิงโครโนพื้นฐานข้อมูลคอนฟิกูเรชันอุปกรณ์” ในหน้า 418

ความบกพร่องของระบบอาจส่งผลให้ฐานข้อมูลคอนฟิกูเรชันอุปกรณ์ไม่สอดคล้องกับ LVM คุณสามารถชิงโครโนพื้นฐานข้อมูลคอนฟิกูเรชันอุปกรณ์กับข้อมูล LVM ได้

หน่วยเก็บโลจิคัลวอลุ่ม

โลจิคัลวอลุ่มคือกลุ่มของข้อมูลที่ตั้งอยู่บนฟิสิคัลวอลุ่ม

มีการใช้ลำดับชั้นของโครงสร้างเพื่อจัดการกับหน่วยเก็บดิสก์ แต่ละดิสก์ไดร์ฟ ที่เรียกว่า *ฟิสิคัลวอลุ่ม* (PV) มีชื่อ เช่น /dev/hdisk0 ทุกฟิสิคัลวอลุ่มที่ใช้จะเป็นสมาชิกของ *กลุ่มวอลุ่ม* (VG) ฟิสิคัลวอลุ่ม ทั้งหมดในกลุ่มวอลุ่มมีการแบ่งออกเป็น *พาร์ติชันฟิสิคัล* (PPs) ที่มีขนาดเดียวกัน เพื่อวัตถุประสงค์ในการจัดสรรที่ว่าง แต่ละฟิสิคัลวอลุ่มถูกแบ่งออกเป็นห้าพื้นที่ (outer_edge, inner_edge, outer_middle, inner_middle และ center) จำนวนของพาร์ติชันฟิสิคัลในแต่ละพื้นที่แตกต่างกันไปขึ้นอยู่กับ ความจุทั้งหมดของดิสก์ไดร์ฟ

ภายในแต่ละกลุ่มวอลุ่ม มีการกำหนด *โลจิคัลวอลุ่ม* (LVs) ตั้งแต่หนึ่งรายการขึ้นไป ข้อมูลบนโลจิคัลวอลุ่มปรากฏขึ้นต่อกันสำหรับผู้ใช้แต่ อาจไม่ต่อกันบนฟิสิคัลวอลุ่ม ลักษณะนี้ช่วยให้ไฟล์ระบบ พื้นที่เพจ และโลจิคัลวอลุ่มอื่นๆ สามารถปรับขนาดหรือจัดสรรใหม่ได้ เพื่อขยายหลายฟิสิคัลวอลุ่ม และเพื่อให้ได้เนื้อหาที่ถูกรักษาเพื่อให้ความยืดหยุ่นและพร้อมใช้มากยิ่งขึ้นในการจัดเก็บข้อมูล

แต่ละโลจิคัลวอลุ่มประกอบด้วย *พาร์ติชันโลจิคัล* (LPs) ตั้งแต่หนึ่งรายการขึ้นไป แต่ละพาร์ติชันโลจิคัลสอดคล้องกับพาร์ติชันฟิสิคัลอย่างน้อยหนึ่งรายการ หาก มีการระบุการมีร์เรอร์สำหรับโลจิคัลวอลุ่ม จะมีการจัดสรรพาร์ติชันฟิสิคัลเพิ่มเติมเพื่อจัดเก็บสำเนาเพิ่มเติมของแต่ละพาร์ติชันโลจิคัล แม้ว่า พาร์ติชันโลจิคัลได้รับการกำหนดหมายเลขต่อกัน แต่พาร์ติชันฟิสิคัลที่ใช้ไม่จำเป็นต้องต่อเนื่องหรือต่อกัน

โลจิคัลวอลุ่มสามารถทำหน้าที่สำหรับการทำงานของระบบได้หลายอย่าง เช่น การเลื่อนหน้า แต่แต่ละโลจิคัลวอลุ่มจะทำเพียงหน้าที่เดียวเท่านั้น โลจิคัลวอลุ่มจำนวนมาก มีระบบไฟล์ที่เจอร์นัลเดียว (JFS หรือ JFS2) แต่ละ JFS ประกอบด้วย พูลของบล็อกขนาดหน้า (4 KB) เมื่อเขียนข้อมูลลงในไฟล์ จะมีการจัดสรรบล็อกเพิ่มเติม ตั้งแต่หนึ่งบล็อกขึ้นไปให้กับไฟล์นั้น บล็อกเหล่านี้อาจไม่ ต่อกันหรือไม่ต่อกับบล็อกอื่นซึ่งจัดสรรก่อนหน้านี้ให้กับไฟล์ ระบบไฟล์ที่กำหนดสามารถถูกกำหนดให้มีขนาดแฟรกเมนต์ที่ เล็กกว่า 4 KB (512 ไบต์, 1 KB, 2 KB)

หลังจากการติดตั้ง ระบบมีหนึ่งกลุ่มวอลุ่ม (กลุ่มวอลุ่ม rootvg) ที่ประกอบด้วยชุดพื้นฐานของโลจิคัลวอลุ่มซึ่งต้องใช้เพื่อเริ่มต้นระบบ และโลจิคัลวอลุ่มอื่นซึ่งคุณระบุในสคริปต์การติดตั้ง ฟิสิคัลวอลุ่ม อื่นใดที่คุณเชื่อมต่อกับระบบสามารถมีการเพิ่มลงใน กลุ่มวอลุ่มได้ (โดยใช้คำสั่ง `extendvg`) คุณสามารถเพิ่ม ฟิสิคัลวอลุ่มลงในกลุ่มวอลุ่ม rootvg หรือลงในกลุ่มวอลุ่มอื่น (ที่กำหนดโดยใช้คำสั่ง `mkvg`) โลจิคัลวอลุ่มสามารถปรับแต่งได้โดยใช้คำสั่ง อินเตอร์เฟซ System Management Interface Tool (SMIT) ที่ควบคุมด้วยเมนู

งานที่เกี่ยวข้อง:

“การย้ายเนื้อหาของฟิสิคัลวอลุ่ม” ในหน้า 396

“การแจ้งผู้ดูแลระบบเมื่อฟิสิคัลวอลุ่มขาดไป” ในหน้า 405

แม้ว่า AIX บันทึก ข้อผิดพลาดเมื่อไม่สามารถเข้าถึงฟิสิคัลวอลุ่ม แต่มีสถานการณ์ ซึ่งไม่สามารถตรวจพบข้อผิดพลาดได้

“การแบ่งดิสก์ที่มีร์เรอร์จากกลุ่มวอลุ่ม” ในหน้า 407

Snapshot สนับสนุนให้คุณปกป้องความสอดคล้องกันของ กลุ่มวอลุ่มที่ทำมีร์เรอร์ไว้จากความล้มเหลวของดิสก์ที่อาจเกิดขึ้น

“การลดขนาดของระบบไฟล์ในกลุ่มวอลุ่มรากของคุณ” ในหน้า 482

วิธีที่ง่ายที่สุดในการลดระบบไฟล์ทั้งหมด เป็นขนาดเล็กที่สุดคือ การตั้งค่าอ็อปชัน SHRINK เป็น ใช่ เมื่อ เรียกคืนระบบปฏิบัติการ การพื้นฐานจากสำเนาสำรอง

การจัดเตรียมเพื่อติดตั้งอุปกรณ์

การติดตั้งอุปกรณ์บนระบบของคุณประกอบด้วยการระบุตำแหน่งที่จะต่อพ่วงอุปกรณ์ การเชื่อมต่ออุปกรณ์ทางกายภาพ และการกำหนดคอนฟิก อุปกรณ์ด้วย Configuration Manager หรือ SMIT

หมายเหตุ: โพรซีเจอร์ต่อไปนี้เป็นต้องปิดระบบของคุณเพื่อ ติดตั้งอุปกรณ์ อย่างไรก็ตาม การติดตั้งบางอุปกรณ์ไม่จำเป็นต้องปิด ระบบของคุณ โปรดอ้างอิงเอกสารคู่มือที่จัดส่งมาพร้อมกับอุปกรณ์เฉพาะ

หัวข้อนี้อธิบายภารกิจการติดตั้งที่พบได้ทั่วไปในทุก อุปกรณ์ เนื่องจากมีอุปกรณ์จำนวนมากซึ่งคุณสามารถติดตั้งบนระบบของคุณได้ จึงมีการแสดงโพรซีเจอร์ทั่วไปเท่านั้น หากต้องการข้อมูลเพิ่มเติมให้ดูที่คำแนะนำการติดตั้งซึ่งจัดส่งมาพร้อมกับอุปกรณ์เฉพาะ

1. หยุดแอ็พพลิเคชันทั้งหมดที่กำลังรันบนยูนิตรระบบ และปิด ยูนิตรระบบโดยใช้คำสั่ง shutdown
2. ปิดยูนิตรระบบและอุปกรณ์ที่ต่อพ่วงทั้งหมด
3. ถอดปลั๊กยูนิตรระบบและอุปกรณ์ที่ต่อพ่วงทั้งหมด
4. เชื่อมต่ออุปกรณ์ใหม่เข้ากับระบบโดยใช้โพรซีเจอร์ที่อธิบายไว้ในคู่มือเซ็ทอัพและผู้ใช้งานของอุปกรณ์
5. เสียบปลั๊กยูนิตรระบบและอุปกรณ์ที่ต่อพ่วงทั้งหมด
6. เปิดอุปกรณ์ที่ต่อพ่วงทั้งหมด โดยปล่อยให้ยูนิตรระบบปิด อยู่
7. เปิดยูนิตรระบบเมื่ออุปกรณ์ทั้งหมดทดสอบการเปิดด้วยตนเอง (POST) เสร็จแล้ว

Configuration Manager สแกนอุปกรณ์ที่ต่อพ่วงโดยอัตโนมัติ และกำหนดคอนฟิกอุปกรณ์ใหม่ใดๆ ที่ตรวจพบ อุปกรณ์ใหม่ มีการกำหนดคอนฟิกด้วยดีฟอลต์แอดทริบิวต์และบันทึกไว้ในฐานข้อมูล คอนฟิกูเรชันที่กำหนดเอง โดยวางอุปกรณ์ในสถานะ มีอยู่

คุณสามารถกำหนดคอนฟิกอุปกรณ์โดยใช้พารามิเตอร์ SMIT, `smit dev` ถ้าคุณต้องการกำหนดแอดทริบิวต์อุปกรณ์เอง หรือถ้าไม่สามารถ กำหนดคอนฟิกอุปกรณ์โดยอัตโนมัติได้ ให้ดูที่เอกสารคู่มืออุปกรณ์ซึ่งจัดส่งมาพร้อมกับอุปกรณ์สำหรับข้อกำหนด คอนฟิกูเรชันเฉพาะ

หลักการที่เกี่ยวข้อง:

“ฐานข้อมูลคอนฟิกูเรชันอุปกรณ์และการจัดการอุปกรณ์” ในหน้า 579

ข้อมูลอุปกรณ์มีอยู่ในฐานข้อมูลที่กำหนดไว้ล่วงหน้า หรือฐานข้อมูล แบบกำหนดเองซึ่งประกอบเป็นฐานข้อมูลคอนฟิกูเรชัน อุปกรณ์

การกำหนดคอนฟิกอ็อปติคัลไดรฟ์การอ่าน/เขียน

การกำหนดคอนฟิกอ็อปติคัลไดรฟ์การอ่าน/เขียนมีอยู่สองเมธอด

อ็อปติคัลไดรฟ์การอ่าน/เขียนต้องมีการเชื่อมต่อเข้ากับระบบและเปิดอยู่

เมธอด 1

เมธอดหนึ่งเร็วกว่า เมธอดสอง เมธอดนี้กำหนดคอนฟิกอ็อปติคัลไดรฟ์การอ่าน/เขียนที่ระบุเท่านั้น เมื่อต้องการใช้เมธอดนี้ คุณต้องระบุข้อมูลต่อไปนี้:

ไอเท็ม	คำอธิบาย
คลาสย่อย	กำหนดวิธีการต่อพวงไดรฟ์
ชนิด	ระบุชนิดของอ็อปติคัลไดรฟ์การอ่าน/เขียน
ชื่อพารามิเตอร์	ระบุส่วนต่อพวงระบบซึ่งไดรฟ์เชื่อมต่อ
ตำแหน่งที่เชื่อมต่อ	ระบุโลจิคัลแอดเดรสของไดรฟ์

บ้นคำสั่งต่อไปนี้เพื่อกำหนดคอนฟิกอ็อปติคัลไดรฟ์การอ่าน/เขียน:

```
mkdev -c rwoptical -s Subclass -t Type -p ParentName -w WhereConnected
```

ข้อมูลต่อไปนี้ เป็นตัวอย่างของอ็อปติคัลไดรฟ์การอ่าน/เขียนที่มี SCSI ID เป็น 6, หมายเลขโลจิคัลยูนิตเป็นศูนย์ และเชื่อมต่อกับบัส (scsi3) SCSI ที่สาม:

```
mkdev -c rwoptical -s scsi -t osomd -p scsi3 -w 6,0 -a pv=yes
```

เมธอด 2

เมธอดสองใช้ Configuration Manager เพื่อค้นหาคอนฟิกูเรชันปัจจุบัน ตรวจสอบอุปกรณ์ใหม่ใดๆ และกำหนดคอนฟิกอุปกรณ์โดยอัตโนมัติ เมธอดนี้ใช้เมื่อทราบข้อมูลเกี่ยวกับอ็อปติคัลไดรฟ์การอ่าน/เขียนเพียงเล็กน้อยเท่านั้น

1. ใช้ configuration manager เพื่อกำหนดคอนฟิกอุปกรณ์ที่ตรวจสอบใหม่ ทั้งหมดบนระบบ (รวมถึงอ็อปติคัลไดรฟ์การอ่าน/เขียน) โดยพิมพ์:

```
cfgmgr
```

2. พิมพ์คำสั่งต่อไปนี้เพื่อแสดงรายการชื่อ โค้ดที่ตั้ง และ ชนิดของอ็อปติคัลไดรฟ์การอ่าน/เขียนที่กำหนดคอนฟิกในปัจจุบันทั้งหมด:

```
lsdev -C -c rwoptical
```

3. กำหนดชื่อของอ็อปติคัลไดรฟ์การอ่าน/เขียนที่กำหนดคอนฟิกใหม่ โดยใช้โค้ดที่ตั้งซึ่งตรงกับที่ตั้งของไดรฟ์ที่กำลังเพิ่ม

คอนฟิกูเรชันของอุปกรณ์จำนวนมาก

อุปกรณ์รวมถึงฮาร์ดแวร์คอมพิวเตอร์ เช่น เครื่องพิมพ์ ไดรฟ์ อะแดปเตอร์ บัส และส่วนแนบ ตลอดจนอุปกรณ์เทียม เช่น ไฟล์พิเศษ ข้อผิดพลาดและไฟล์พิเศษ null ไดรเวอร์อุปกรณ์ตั้งอยู่ในไดเรกทอรี /usr/lib/drivers

จำนวนอุปกรณ์ที่ AIX สามารถ สนับสนุนอาจแตกต่างกันไปในแต่ละระบบ ทั้งนี้ขึ้นอยู่กับปัจจัยที่สำคัญหลายอย่าง ปัจจัยต่อไปนี้ มีผลกระทบต่อระบบไฟล์ซึ่งสนับสนุน อุปกรณ์:

- การกำหนดคอนฟิกอุปกรณ์จำนวนมากต้องการหน่วยเก็บข้อมูลเพิ่มเติม ในฐานข้อมูลคอนฟิกูเรชันอุปกรณ์ ODM นอกจากนี้ยังต้องการไฟล์พิเศษอุปกรณ์ มากขึ้นด้วย ผลที่ตามมาคือ ต้องการพื้นที่ว่างและ i-nodes ของระบบไฟล์ มากขึ้น
- บางอุปกรณ์ต้องการพื้นที่ว่างมากกว่าอุปกรณ์อื่นในฐานข้อมูลคอนฟิกูเรชัน อุปกรณ์ ODM นอกจากนี้จำนวนของไฟล์พิเศษหรือ i-nodes ยังแตกต่างกันไปใน แต่ละอุปกรณ์ด้วย ส่งผลให้จำนวนของพื้นที่ว่างและ i-nodes ที่ต้องการของระบบไฟล์ ขึ้นอยู่กับชนิดของอุปกรณ์บนระบบ
- อุปกรณ์ multipath I/O (MPIO) ต้องการพื้นที่ว่างมากกว่าอุปกรณ์ที่ไม่ใช่ MPIO เนื่องจากข้อมูลมีการจัดเก็บไว้ใน ODM ของตัวอุปกรณ์เองและ ของแต่ละพาร์ตไปยังอุปกรณ์ สำหรับแนวทางคร่าวๆ มีการสมมติว่าแต่ละพาร์ต ใช้พื้นที่ว่างจำนวนเศษหนึ่งส่วนห้าของอุปกรณ์ ตัวอย่างเช่น อุปกรณ์ MPIO ที่มีห้าพาร์ตจะมีพื้นที่ว่างเทียบเท่ากับอุปกรณ์ที่ไม่ใช่ MPIO สองรายการ

- AIX รวมทั้งอุปกรณ์โลจิคัล และอุปกรณ์ฟิสิกัลไว้ในฐานข้อมูลคอนฟิกูเรชันอุปกรณ์ ODM อุปกรณ์โลจิคัลได้แก่ กลุ่มวอลุ่ม โลจิคัลวอลุ่ม อินเตอร์เฟซเครือข่าย และอื่นๆ ในบางกรณี ความสัมพันธ์ระหว่างอุปกรณ์โลจิคัลและฟิสิกัลอาจ ส่งผลกระทบอย่างมากต่อจำนวนทั้งหมดของอุปกรณ์ที่สนับสนุน ตัวอย่างเช่น ถ้าคุณ กำหนดกลุ่มวอลุ่มที่มีสองโลจิคัลวอลุ่ม สำหรับแต่ละฟิสิกัลดิสก์ซึ่งต่อพ่วง เข้ากับระบบ จะส่งผลให้มีอุปกรณ์ AIX สี่รายการ สำหรับดิสก์แต่ละตัว ในทางตรงกันข้าม ถ้าคุณกำหนดกลุ่มวอลุ่มที่มีหกโลจิคัลวอลุ่ม สำหรับแต่ละฟิสิกัลดิสก์ จะมีอุปกรณ์ AIX แปดรายการ สำหรับดิสก์แต่ละตัว ดังนั้น จึงสามารถต่อพ่วงดิสก์ได้เพียงครึ่งเดียวเท่านั้น
- การเปลี่ยนแอตทริบิวต์อุปกรณ์จากค่าติดตั้งดีฟอลต์ส่งผลให้ฐานข้อมูล คอนฟิกูเรชันอุปกรณ์ ODM มีขนาดใหญ่ขึ้น และอาจทำให้สามารถสนับสนุน อุปกรณ์ได้น้อยลง
- อุปกรณ์มากขึ้นต้องการหน่วยความจำแบบจริงมากขึ้น

AIX ใช้สองระบบไฟล์เพื่อ สนับสนุนอุปกรณ์:

- ใช้ระบบไฟล์ RAM ในระหว่างบูตในสถานะแวดล้อมที่ไม่มีพื้นที่การเลื่อนหน้า และไม่มีระบบไฟล์ดิสก์ที่ติดตั้ง ขนาดของระบบไฟล์ RAM คือ 25% ของขนาดหน่วยความจำระบบจนถึงสูงสุด 128 MB มีการจัดสรรหนึ่ง i-nodes สำหรับทุก KB ในระบบไฟล์ RAM ข้อกำหนดหน่วยความจำระบบขั้นต่ำสำหรับระบบปฏิบัติการ AIX คือ 256 MB ซึ่งแปลงเป็นขนาดของระบบไฟล์ RAM ขั้นต่ำ 64 MB ที่มี 65536 i-nodes หากขนาดหน่วยความจำระบบเป็น 512 MB หรือมากกว่า ระบบไฟล์ RAM จะอยู่ที่ขนาดสูงสุดจำนวน 128 MB ที่มี 131072 i-nodes ถ้าจำนวน ของพื้นที่ว่างระบบไฟล์ RAM หรือจำนวนไอ-โหนดที่ต้องการเพื่อสนับสนุน อุปกรณ์ที่ต่อพ่วงเกินกว่าจำนวนที่จัดสรรไปยังดิสก์ RAM ระบบอาจจะไม่บูต หากเป็นกรณีนี้ คุณต้องเอาบางอุปกรณ์ออก
- พื้นที่ว่างและ i-nodes ของระบบไฟล์รูท (rootvg) บนดิสก์สามารถ เพิ่มขึ้นได้ทราบเท่าที่มีพาร์ติชันซึ่งไม่ได้จัดสรรอยู่ใน rootvg ด้วยขนาดของระบบไฟล์ RAM สูงสุด ซึ่งสามารถกำหนดคอนฟิกูเรชัน AIX ได้สูงถึง 25,000 อุปกรณ์จำนวนเหล่านี้รวมทั้งอุปกรณ์ฟิสิกัลและโลจิคัล ขึ้นอยู่กับ ปัจจัยต่างๆ ที่ระบุในส่วนนี้ ระบบของคุณอาจจะสามารถ กำหนดคอนฟิกูเรชันได้มากกว่าหรือน้อยกว่าจำนวนนี้

หมายเหตุ: ด้วยอุปกรณ์จำนวนมากในระบบ เวลาคอนฟิกูเรชันที่นานขึ้น ส่งผลให้เวลาบูตนานขึ้น

หลักการที่เกี่ยวข้อง:

“บทนำเกี่ยวกับ AIX for BSD system managers” ในหน้า 353

ข้อมูลต่อไปนี้เป็นเคล็ดลับที่ช่วยผู้จัดการระบบการกระจายซอฟต์แวร์ของ Berkeley (BSD) เริ่มต้นจัดการ AIX

การเพิ่มไดรฟ์สื่อบันทึกที่ถอดออกได้

คุณสามารถเพิ่มไดรฟ์สื่อบันทึกที่ถอดออกได้

โปรซีเดเจอร์ต่อไปนีใช้ SMIT เพื่อเพิ่ม ซีตริอมไดรฟ์ลงในระบบของคุณ ไดรฟ์สื่อบันทึกที่ถอดออกได้ชนิดอื่นมีการเพิ่ม โดยใช้พารามิเตอร์ที่แตกต่างออกไป แต่ทุกชนิดใช้โปรซีเดเจอร์ทั่วไปเดียวกัน คุณยังสามารถเพิ่มไดรฟ์สื่อบันทึกที่ถอดออกได้โดยใช้ Configuration Manager หรือคำสั่ง **mkdev**

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

1. เมื่อต้องการเพิ่มซีตริอมไดรฟ์ลงในระบบของคุณ ให้ติดตั้งฮาร์ดแวร์ตาม เอกสารคู่มือซึ่งมาพร้อมกับระบบของคุณ
2. ด้วยสิทธิการใช้งานรากให้พิมพ์พารามิเตอร์ SMIT ต่อไปนี้:


```
smit makcdr
```
3. ในหน้าจอถัดไป เลือกชนิดไดรฟ์จากรายการที่มีอยู่ของ ไดรฟ์ที่สนับสนุน

4. ในหน้าจอลัดไปเลือกพาเรนตอะแด็ปเตอร์จากรายการที่มีอยู่
5. ในหน้าจอลัดไปเลือกแอดเดรสการเชื่อมต่อจากรายการที่มีอยู่ เป็นอย่างน้อย คุณยังสามารถใช้หน้าจอนี้เพื่อเลือกอีพชั้นอื่นได้ด้วย เมื่อคุณทำเสร็จแล้ว ให้กด Enter จากนั้น SMIT เพิ่มซีดีรอมไดรฟ์ใหม่

ณ จุดนี้ระบบของคุณรับรู้ถึงซีดีรอมไดรฟ์ใหม่ เมื่อต้องการเพิ่มอ็อปติคัลไดรฟ์การอ่าน/เขียน ให้ใช้พาทว่น `smit makomd` เมื่อต้องการเพิ่มเทปไดรฟ์ ให้ใช้พาทว่น `smit maktpe`

ให้ดูที่ คำอธิบายคำสั่ง `mkdev` ใน *Commands Reference, Volume 3* สำหรับ ข้อมูลเพิ่มเติม

การสนับสนุนการเรียกคืนพื้นที่สำหรับ Logical Volume Storage

ใน AIX 7.2 with the 7200-01 Technology Level หรือ ใหม่กว่า Logical Volume Manager (LVM) สนับสนุนการเรียกคืนพื้นที่สำหรับฟิสิคัลวอลุ่มที่มีความสามารถเรียกคืนพื้นที่

LVM จะแจ้งดิสก์ไดรเวอร์ ซึ่งจากนั้นจะแจ้งระบบย่อยหน่วยเก็บข้อมูลว่าพื้นที่พาร์ติชัน ไม่ได้ใช้งานแล้ว และระบบย่อยหน่วยเก็บข้อมูลสามารถเรียกคืนพื้นที่ที่จัดสรรไว้ได้ ดิสก์ไดรเวอร์จะช่วย LVM ตรวจสอบความจุการเรียกคืนพื้นที่ของฟิสิคัลวอลุ่ม คำสั่งการกำหนดคอนฟิก LVM และระบบไฟล์ เช่น คำสั่ง `rm1v`, คำสั่ง `rm1vcopy` และคำสั่ง `chfs (shrink fs)` จะเริ่มต้นการเรียกคืนพื้นที่สำหรับพาร์ติชันหลังจาก พาร์ติชันถูกทำให้ว่าง LVM จะตรวจสอบความจุการเรียกคืนพื้นที่ของวอลุ่ม เมื่อเปิดวอลุ่มระหว่างการเรียกทำงานคำสั่ง `varyonvg` หรือ `extendvg` LVM ยังพยายามตรวจสอบหาขนาดที่กลุ่มวอลุ่มออนไลน์ด้วย หากการตรวจสอบการเปลี่ยนแปลงสถานะต้องการ ให้เปิดฟิสิคัลวอลุ่มอีกครั้ง ผู้ดูแลระบบต้องรันคำสั่ง `varyoffvg` และ จากนั้นรันคำสั่ง `varyonvg` สำหรับกลุ่มวอลุ่ม

กลุ่มวอลุ่มที่สร้างขึ้นก่อน AIX 7.2 Technology Level 1 อาจมีพื้นที่พาร์ติชันว่าง ซึ่งไม่สามารถเหมาะสมสำหรับการเรียกคืน โดยอัตโนมัติ ผู้ดูแลระบบสามารถสร้างและลบโลจิคัลวอลุ่มเต็มมีบนพาร์ติชันว่างเหล่านั้นเพื่อเรียกคืน พื้นที่นี้ แต่ที่ว่างจะถูกเรียกคืนโดยอัตโนมัติสำหรับพาร์ติชัน ซึ่งถูกทำให้ว่างหลังจาก การติดตั้ง AIX 7.2 Technology Level 1

กระบวนการ LVM เพื่อเรียกคืนที่ว่างจะรันอยู่เบื้องหลังหลังจากคำสั่ง เช่น `rm1v` ดำเนินการเสร็จสิ้น หากกระบวนการหยุดทำงาน ก่อนที่กระบวนการ LVM จะเสร็จสิ้น กระบวนการการเรียกคืนสำหรับพาร์ติชันทั้งหมด ดังนั้นพาร์ติชันจะถูกทำให้ว่างแต่ที่ว่างจะไม่ถูกเรียกคืนสำหรับพาร์ติชันที่ค้างอยู่ หากสถานการณ์นี้เกิดขึ้น คุณสามารถสร้างและลบ โลจิคัลวอลุ่มเต็มมีเพื่อเรียกคืนที่ว่างจากพาร์ติชันที่เหลืออยู่

กระบวนการ LVM จะไม่หน่วงเวลาการประมวลผลคำสั่ง `varyoffvg` หรือคำสั่ง `reducevg` แม้ว่ากระบวนการการเรียกคืนพื้นที่ที่จะค้างอยู่ กระบวนการการเรียกคืนพื้นที่ จะถูกทิ้งแทนที่จะรอให้กระบวนการเสร็จสิ้น

หมายเหตุ: คำสั่งจะรอคำร้องขอการเรียกคืนพื้นที่ที่ค้างอยู่ถูกส่งไปยัง ดิสก์ไดรเวอร์แล้วเท่านั้น

ฟังก์ชันการเรียกคืนพื้นที่พร้อมใช้งานจากระบบย่อยหน่วยเก็บข้อมูลเพื่อเรียกคืนพื้นที่ที่ถูกทำให้ว่างจากฟิสิคัลวอลุ่ม แต่ละระบบย่อยหน่วยเก็บข้อมูลต้องการให้คำร้องขอการเรียกคืนพื้นที่ตรงกับ จำนวนเฉพาะของฟิสิคัลบล็อก และจะแตกต่างกันตามระบบย่อยหน่วยเก็บข้อมูล ดังนั้น บางครั้ง การเรียกคืนบล็อก (ทั้งหมดหรือบางบล็อก) จากพาร์ติชันไม่สามารถทำได้ เนื่องจากขนาดการเรียกคืนไม่ตรงกับ ฟิสิคัลบล็อกของพาร์ติชัน ระบบย่อยหน่วยเก็บข้อมูลบางระบบสนับสนุนการเรียกคืนขนาดบล็อก ที่มากกว่าขนาดพาร์ติชัน LVM และการเรียกคืนบล็อกบางส่วนไม่สามารถทำได้ ในสถานการณ์นี้ LVM อาจไม่สามารถสะสมพาร์ติชันว่างที่ต่อเนื่องกันได้เพียงพอเพื่อสร้าง คำร้องขอการเรียกคืนได้ ดังนั้น เมื่อคุณลบหลายพาร์ติชัน LVM คุณอาจไม่เรียกคืน จำนวนของที่ว่างที่เทียบเท่ากันในระบบย่อยหน่วยเก็บข้อมูล คุณสามารถใช้คำสั่ง `lvmstat` พร้อมกับอีพชั้น `-r` เพื่อดูข้อมูลเกี่ยวกับคำร้องขอการเรียกคืนพื้นที่ ที่สร้างโดย LVM

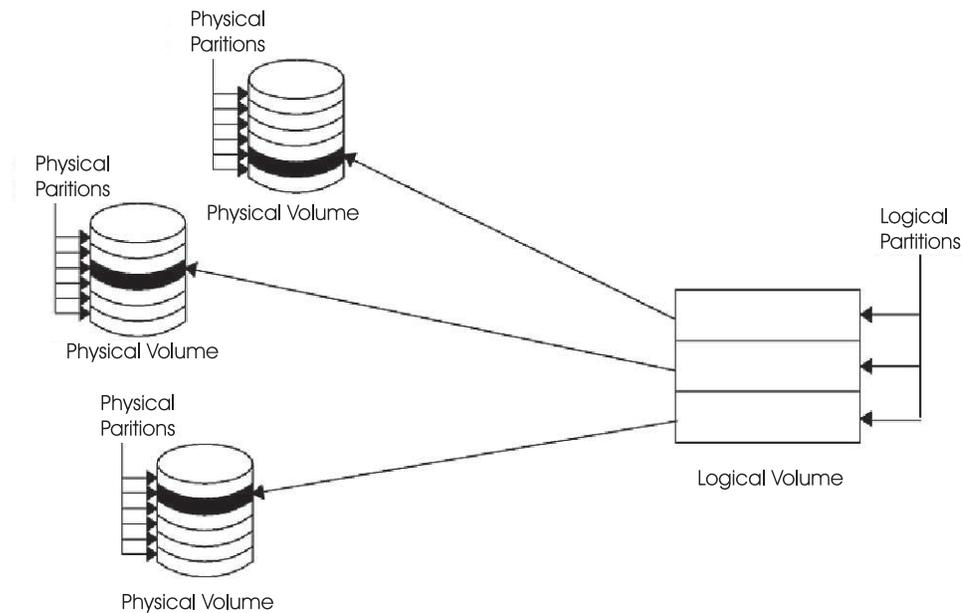
ข้อมูลที่เกี่ยวข้อง:

I คำสั่ง varyoffvg

แนวคิดหน่วยเก็บโลจิคัลวอลุ่ม

โลจิคัลวอลุ่ม (ซึ่งสามารถถอดข้ามฟิสิคัลวอลุ่ม) ประกอบด้วย พาร์ติชันโลจิคัลที่จัดสรรบนพาร์ติชันฟิสิคัล

รูปภาพต่อไปนี้แสดงความสัมพันธ์ระหว่าง แนวคิดหน่วยเก็บโลจิคัลพื้นฐาน



รูปที่ 1. กลุ่มวอลุ่ม. ภาพสาธิตนี้ แสดงกลุ่มวอลุ่มหนึ่งๆที่ประกอบด้วยสามฟิสิคัลวอลุ่มซึ่งมี ช่วงสูงสุดที่ระบุไว้ โลจิคัลวอลุ่ม (ซึ่งสามารถถอดข้ามฟิสิคัลวอลุ่ม) ประกอบด้วย พาร์ติชันโลจิคัลที่จัดสรรบนพาร์ติชันฟิสิคัล

ฟิสิคัลวอลุ่ม:

ต้องกำหนดดิสก์เป็นฟิสิคัลวอลุ่มและวางไว้ใน สภาพพร้อมใช้งาน ก่อนสามารถกำหนดดิสก์นั้นให้กับกลุ่มวอลุ่มได้

ฟิสิคัลวอลุ่มมีข้อมูลคอนฟิกูเรชันและการระบุบางอย่าง ที่เขียนบนนั้น ข้อมูลนี้รวมถึงตัวระบุฟิสิคัลวอลุ่มที่ไม่ซ้ำกันในระบบ

LVM สามารถใช้พื้นที่ว่างเพิ่มเติมที่ redundant array of identical disks (RAID) สามารถเพิ่มหมายเลขยูนิทโลจิคัล (LUN), โดยการเพิ่มพาร์ติชันฟิสิคัลลงในวอลุ่มฟิสิคัลที่เกี่ยวข้องกับ LUN

กลุ่มวอลุ่ม:

กลุ่มวอลุ่ม คือคอลเล็กชันของ 1 ถึง 32 ฟิสิคัลวอลุ่ม ที่มีขนาดและชนิดแตกต่างกัน

กลุ่มวอลุ่มขนาดใหญ่สามารถมีตั้งแต่ 1 ถึง 128 ฟิสิคัลวอลุ่ม กลุ่มวอลุ่มแบบ ปรับสเกลได้สามารถมีได้สูงสุดถึง 1024 ฟิสิคัลวอลุ่ม ฟิสิคัลวอลุ่มหนึ่งสามารถเป็นสมาชิก ของหนึ่งกลุ่มวอลุ่มเท่านั้นต่อระบบ และมีกลุ่มวอลุ่มที่ใช้งานอยู่ได้สูงสุด 255 กลุ่ม

เมื่อฟิสิคัลวอลุ่มถูกกำหนดให้กับกลุ่มวอลุ่ม ฟิสิคัลบล็อก ของสื่อบันทึกหน่วยเก็บบนนั้นจะมีการจัดระเบียบเป็นพาร์ติชันฟิสิคัลในขนาดที่คุณ ระบุเมื่อคุณสร้างกลุ่มวอลุ่ม

เมื่อคุณติดตั้งระบบ หนึ่งกลุ่มวอลุ่ม (กลุ่มวอลุ่มรากที่เรียกว่า rootvg) จะถูกสร้างขึ้นโดยอัตโนมัติ โดยกลุ่มนี้ประกอบด้วยชุดพื้นฐานของโลจิคัล วอลุ่มซึ่งต้องใช้เพื่อเริ่มต้นระบบ และโลจิคัลวอลุ่มอื่นซึ่งคุณระบุ ในสคริปต์การติดตั้ง Rootvg มีพื้นที่การเลื่อนหน้า บันทึกเจอร์นัล ข้อมูลบูต และหน่วยเก็บดัมพ์ โดยแต่ละรายการอยู่ในโลจิคัลวอลุ่มแยกต่างหากของตนเอง Rootvg มีแอ็ททริบิวต์ที่แตกต่างจากกลุ่มวอลุ่มที่ผู้ใช้กำหนด ตัวอย่าง เช่น ไม่สามารถอิมพอร์ตหรือเอ็กซ์พอร์ต rootvg เมื่อดำเนินการคำสั่ง หรือโพธิ์เตอร์บน rootvg คุณต้องคุ้นเคยกับลักษณะเฉพาะของ rootvg

คุณสร้างกลุ่มวอลุ่มโดยใช้คำสั่ง `mkvg` คุณ เพิ่มฟิสิคัลวอลุ่มลงในกลุ่มวอลุ่มโดยใช้คำสั่ง `extendvg` ใช้ขนาดที่เปลี่ยนของฟิสิคัลวอลุ่มโดยใช้คำสั่ง `chvg` และ ลบฟิสิคัลวอลุ่มออกจากกลุ่มวอลุ่มโดยใช้คำสั่ง `reducevg` คำสั่งอื่นบางคำสั่งซึ่งคุณใช้บนกลุ่มวอลุ่มได้แก่: แสดงรายการ (`lsvg`), ลบ (`exportvg`), ติดตั้ง (`importvg`), จัดระเบียบใหม่ (`reorgvg`), ซิงโครไนซ์ (`syncvg`), ทำให้พร้อมใช้งาน (`varyonvg`), และทำให้ไม่พร้อม ใช้งาน (`varyoffvg`)

ระบบขนาดเล็กอาจต้องการเพียงหนึ่งกลุ่มวอลุ่มเพื่อจัดเก็บฟิสิคัลวอลุ่มทั้งหมด ที่ต่อพ่วงเข้ากับระบบ อย่างไรก็ตาม คุณอาจต้องการสร้างกลุ่มวอลุ่มแยกต่างหาก เพื่อเหตุผลด้านความปลอดภัย เนื่องจากแต่ละกลุ่มวอลุ่มสามารถมีสิทธิการรักษา ความปลอดภัยของตัวเอง กลุ่มวอลุ่มแยกต่างหากยังทำให้การบำรุงรักษาง่ายขึ้นด้วย เนื่องจากกลุ่มอื่นที่ไม่ใช่กลุ่มที่กำลังเซอร์วิสยังคงใช้งานอยู่ได้ เนื่องจาก rootvg ต้องออนไลน์เสมอ จึงมีฟิสิคัลวอลุ่มจำนวนต่ำสุด ที่จำเป็นสำหรับการดำเนินงานระบบเท่านั้น

คุณสามารถย้ายข้อมูลจากฟิสิคัลวอลุ่มหนึ่งไปยังฟิสิคัลวอลุ่มอื่น ใน *กลุ่มวอลุ่มเดียวกัน* ได้โดยใช้คำสั่ง `migratepv` คำสั่งนี้ช่วยให้คุณสามารถทำให้ฟิสิคัลวอลุ่มว่างเพื่อให้สามารถถูกย้ายออกจาก กลุ่มวอลุ่มได้ ตัวอย่างเช่น คุณอาจย้ายข้อมูลจากฟิสิคัลวอลุ่มที่จะ ถูกแทนที่

กลุ่มวอลุ่มที่สร้างขึ้นด้วยขีดจำกัดฟิสิคัลและโลจิคัลวอลุ่มขนาดเล็ก สามารถถูกแปลงเป็นรูปแบบซึ่งสามารถจัดเก็บฟิสิคัลวอลุ่มและโลจิคัลวอลุ่ม ได้มากขึ้น การดำเนินงานนี้ต้องมีพาร์ติชันว่างจำนวนมากเพียงพอ บนทุกฟิสิคัลวอลุ่มในกลุ่มวอลุ่ม สำหรับการขยาย volume group descriptor area (VGDA) จำนวนของพาร์ติชันว่างที่ต้องการขึ้นอยู่กับ ขนาดของ VGDA ปัจจุบันและขนาดของพาร์ติชันฟิสิคัล เนื่องจาก VGDA อยู่ในบริเวณขอบของดิสก์และต้องใช้พื้นที่ต่อเนื่อง จึงจำเป็นต้องใช้พาร์ติชันว่างบนขอบของดิสก์ หากพาร์ติชันเหล่านี้ได้รับการ จัดสรรสำหรับการใช้งานของผู้ใช้ พาร์ติชันจะถูกโอนย้ายไปยังพาร์ติชันว่างอื่น บนดิสก์เดียวกัน ส่วนที่เหลือของพาร์ติชันฟิสิคัลจะถูกเรียงลำดับหมายเลขใหม่ เพื่อสะท้อน การสูญเสียพาร์ติชันสำหรับการใช้งาน VGDA การเรียงลำดับหมายเลขใหม่นี้เปลี่ยนการแม็พ ของพาร์ติชันโลจิคัลและฟิสิคัลในฟิสิคัลวอลุ่มทั้งหมดของ กลุ่มวอลุ่มนี้ หากคุณได้บันทึกการแม็พของโลจิคัลวอลุ่มสำหรับการดำเนินงาน กู้คืนสำคัญ คุณควรสร้างแม็พอีกครั้ง หลังจากการดำเนินงานแปลง เสร็จสมบูรณ์แล้ว นอกจากนี้ ถ้าทำสำเนาสำรองของกลุ่มวอลุ่มพร้อมกับอ็อพชัน แม็พ และคุณวางแผนจะเรียกคืนโดยใช้แม็พเหล่านั้น การดำเนินงานเรียกคืน อาจล้มเหลวเนื่องจากหมายเลขพาร์ติชันอาจไม่มีอยู่อีกต่อไป (เนื่องจากการลดจำนวน) ขอแนะนำให้ทำสำเนาสำรองก่อนการแปลงและทันทีหลังจากทำการแปลงเสร็จ ถ้าใช้อ็อพชันแม็พ เนื่องจากพื้นที่ VGDA มีการเพิ่มขึ้นอย่างมาก ทุกการดำเนินงานอัปเดต VGDA (การสร้างโลจิคัลวอลุ่ม การเปลี่ยนโลจิคัลวอลุ่ม การเพิ่มฟิสิคัลวอลุ่ม และอื่นๆ) อาจใช้เวลายาวนานขึ้น

หลักการที่เกี่ยวข้อง:

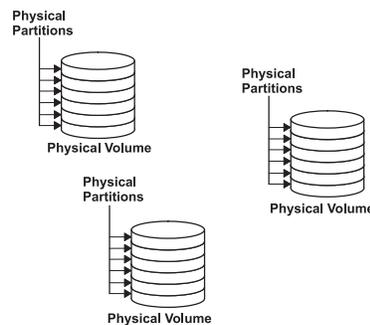
“พาร์ติชันฟิสิคัล”

เมื่อคุณเพิ่มฟิสิคัลวอลุ่มลงในกลุ่มวอลุ่ม จะมีการแบ่งพาร์ติชันฟิสิคัลวอลุ่ม ออกเป็นหน่วยของพื้นที่ว่างที่ต่อกันและมีขนาดเท่ากันซึ่งเรียกว่า พาร์ติชัน ฟิสิคัล พาร์ติชันฟิสิคัลเป็นหน่วยที่เล็กที่สุดของการจัดสรรพื้นที่ว่างหน่วยเก็บ และเป็นพื้นที่ติดกันบนฟิสิคัลวอลุ่ม

พาร์ติชันฟิสิคัล:

เมื่อคุณเพิ่มฟิสิคัลวอลุ่มลงในกลุ่มวอลุ่ม จะมีการแบ่งพาร์ติชันฟิสิคัลวอลุ่ม ออกเป็นหน่วยของพื้นที่ว่างที่ต่อกันและมีขนาดเท่ากันซึ่งเรียกว่า พาร์ติชัน ฟิสิคัล พาร์ติชันฟิสิคัลเป็นหน่วยที่เล็กที่สุดของการจัดสรรพื้นที่ว่างหน่วยเก็บ และเป็นพื้นที่ติดกันบนฟิสิคัลวอลุ่ม

ฟิสิคัลวอลุ่มรับสืบทอดขนาดพาร์ติชันฟิสิคัลของกลุ่มวอลุ่ม ซึ่งคุณสามารถ ตั้งค่าเมื่อคุณสร้างกลุ่มวอลุ่มเท่านั้น (ตัวอย่างเช่น โดยใช้คำสั่ง `mkvg -s`) ภาพสาดิตต่อไปนี้จะแสดงความสัมพันธ์ระหว่างพาร์ติชันฟิสิคัล บนฟิสิคัลวอลุ่มและกลุ่มวอลุ่ม



รูปที่ 2. กลุ่มวอลุ่มที่มีสามฟิสิคัลวอลุ่ม. ภาพสาดิต นี้แสดงสามฟิสิคัลวอลุ่ม โดยแต่ละวอลุ่มมีหกพาร์ติชันฟิสิคัล ภายในหนึ่งกลุ่มวอลุ่ม

หลักการที่เกี่ยวข้อง:

“กลุ่มวอลุ่ม” ในหน้า 426

กลุ่มวอลุ่ม คือคอลเล็กชันของ 1 ถึง 32 ฟิสิคัลวอลุ่ม ที่มีขนาดและชนิดแตกต่างกัน

โลจิคัลวอลุ่ม:

หลังจากคุณสร้างกลุ่มวอลุ่มแล้ว คุณสามารถสร้างโลจิคัลวอลุ่ม ภายในกลุ่มวอลุ่มนั้นได้

แม้ว่า โลจิคัลวอลุ่ม สามารถตั้งอยู่บนพาร์ติชันฟิสิคัลที่ไม่ต่อกัน หรือแม้แต่บนมากกว่าหนึ่งฟิสิคัลวอลุ่มได้ แต่โลจิคัลวอลุ่มปรากฏขึ้นสำหรับผู้ใช้และ แอปพลิเคชันเป็นดิสก์วอลุ่มเดี่ยวที่ต่อกันและสามารถขยายได้ คุณสามารถสร้าง โลจิคัลวอลุ่มเพิ่มเติมได้โดยใช้คำสั่ง `mklv` คำสั่งนี้ช่วยให้คุณสามารถระบุชื่อของโลจิคัลวอลุ่มและกำหนดลักษณะ รวมถึงจำนวนและที่ตั้งของพาร์ติชันโลจิคัลที่จะจัดสรร ให้กับโลจิคัลวอลุ่มนั้น

หลังจากคุณสร้างโลจิคัลวอลุ่มแล้ว คุณสามารถเปลี่ยนชื่อและลักษณะของโลจิคัล วอลุ่มนั้นได้โดยใช้คำสั่ง `chlv` และคุณสามารถเพิ่มจำนวนของ พาร์ติชันโลจิคัลที่จัดสรรให้กับโลจิคัลวอลุ่มนั้นได้โดยใช้คำสั่ง `extendlv` ขนาดสูงสุดที่พอลดสำหรับโลจิคัลวอลุ่มเมื่อสร้างคือ 512 พาร์ติชันโลจิคัล ยกเว้นว่ามีระบุให้ใหญ่ขึ้น คำสั่ง `chlv` ใช้เพื่อ ยกเลิกขีดจำกัดนี้

หมายเหตุ: หลังจากคุณสร้างโลจิคัลวอลุ่มแล้ว ลักษณะ LV STATE ซึ่ง สามารถเห็นได้โดยใช้คำสั่ง `lslv` ถูกปิด และจะเปิดเมื่อ ตัวอย่างเช่น ระบบไฟล์ถูกสร้างขึ้นแล้วในโลจิคัลวอลุ่มและติดตั้งโลจิคัลวอลุ่ม

โลจิคัลวอลุ่มยังสามารถคัดลอกได้โดยใช้คำสั่ง `cplv` ที่แสดงพร้อมกับคำสั่ง `lslv` สามารถลบได้โดยใช้คำสั่ง `rmlv` และสามารถเพิ่มหรือลดจำนวนสำเนาที่เก็บรักษาได้โดยใช้คำสั่ง `mklvcopy` และ คำสั่ง `rmlvcopy` ตามลำดับ นอกจากนี้ยังสามารถเปลี่ยนตำแหน่งโลจิคัลวอลุ่มเมื่อจัดระเบียบกลุ่มวอลุ่มอีกครั้งได้ด้วย

ระบบอนุญาตให้คุณกำหนดได้สูงสุด 255 โลจิคัลวอลุ่มต่อกลุ่มวอลุ่มมาตรฐาน (511 สำหรับกลุ่มวอลุ่มขนาดใหญ่และ 4095 สำหรับกลุ่มวอลุ่มที่ปรับสเกลได้) แต่จำนวนจริงซึ่งคุณสามารถกำหนดได้ขึ้นอยู่กับจำนวนทั้งหมดของหน่วยเก็บ ฟิสิคัลสำหรับกลุ่มวอลุ่มนั้นและขนาดของโลจิคัลวอลุ่มที่คุณกำหนด

พาร์ติชันโลจิคัล:

เมื่อคุณสร้างโลจิคัลวอลุ่ม คุณระบุจำนวนของ *พาร์ติชัน โลจิคัล* สำหรับโลจิคัลวอลุ่ม

พาร์ติชันโลจิคัลเป็นหนึ่งใน สอง หรือสามพาร์ติชันฟิสิคัล ขึ้นอยู่กับจำนวนอินสแตนซ์ของข้อมูลซึ่งคุณต้องการรักษาไว้ การระบุหนึ่ง อินสแตนซ์หมายความว่าไม่มีเพียงหนึ่งสำเนาของโลจิคัลวอลุ่ม (ค่าดีฟอลต์) ในกรณีนี้ มีการแมปโดยตรงของหนึ่งพาร์ติชันโลจิคัลกับหนึ่ง พาร์ติชันฟิสิคัล แต่ละอินสแตนซ์ รวมถึงรายการแรก เรียกว่าเป็น *สำเนา* ตำแหน่งของ พาร์ติชันฟิสิคัล (นั่นคือ แต่ละพาร์ติชันใกล้เคียงกันทางกายภาพมากเพียงใด) ถูกกำหนดโดยอ็อปชันซึ่งคุณระบุเมื่อคุณสร้างโลจิคัลวอลุ่ม

ระบบไฟล์:

โลจิคัลวอลุ่มกำหนดการจัดสรรพื้นที่ว่างดิสก์ลงไปยัง ระดับพาร์ติชันฟิสิคัล การจัดการข้อมูลในระดับที่ละเอียดมากขึ้นสำเร็จได้โดย คอมโพเนนต์ซอฟต์แวร์ระดับสูงเช่น Virtual Memory Manager หรือ ระบบไฟล์ ด้วยเหตุนี้ ขั้นตอนสุดท้ายในวิวัฒนาการของดิสก์คือ การสร้าง *ระบบไฟล์*

คุณสามารถสร้างหนึ่งระบบไฟล์ต่อโลจิคัลวอลุ่ม เมื่อต้องการสร้างระบบไฟล์ ให้ใช้คำสั่ง `crfs`

หลักการที่เกี่ยวข้อง:

“ระบบไฟล์” ในหน้า 464

ระบบไฟล์ คือโครงสร้างตามลำดับชั้น (แผนผังไฟล์) ของ ไฟล์และไดเรกทอรี

ขีดจำกัดสำหรับการจัดการหน่วยเก็บโลจิคัล:

ตารางต่อไปนี้จะแสดงขีดจำกัดสำหรับการจัดการ หน่วยเก็บโลจิคัล

แม้ว่าจำนวนสูงสุดดีฟอลต์ของฟิสิคัลวอลุ่มต่อกลุ่มวอลุ่ม คือ 32 (128 สำหรับกลุ่มวอลุ่มขนาดใหญ่, 1024 สำหรับกลุ่มวอลุ่มที่ปรับสเกลได้) คุณสามารถตั้งค่าสูงสุดสำหรับกลุ่มผู้ใช้ที่ผู้ใช้กำหนดได้เมื่อคุณ ใช้คำสั่ง `mkvg` อย่างไรก็ตาม สำหรับ `rootvg` ตัวแปรนี้มีการตั้งค่าเป็นค่าสูงสุดโดยอัตโนมัติโดยระบบ ในระหว่างการติดตั้ง

ขีดจำกัดสำหรับการจัดการหน่วยเก็บโลจิคัล

หมวดหมู่	ขีดจำกัด
กลุ่มวอลุ่ม	<ul style="list-style-type: none">255 กลุ่มวอลุ่มสำหรับเคอร์เนล 32 บิต4096 กลุ่มวอลุ่มสำหรับเคอร์เนล 64 บิต <p>หมายเหตุ: ตารางอุปกรณ์บนเคอร์เนล 64 บิตจำกัดจำนวน ของจำนวนหลักที่ใช้งานอยู่เป็น 1024 ในเวลาต่อมา จำนวนของกลุ่มวอลุ่มที่ใช้งานอยู่ ถูกจำกัดเป็นน้อยกว่า 1024 กลุ่มวอลุ่ม</p>

ขีดจำกัดสำหรับการจัดการหน่วยเก็บโลจิคัล

หมวดหมู่	ขีดจำกัด
ฟิลิคัลวอลุ่ม	(MAXPVS/ปัจจัยกลุ่มวอลุ่ม) ต่อกลุ่มวอลุ่ม MAXPVS เป็น 32 สำหรับกลุ่มวอลุ่มมาตรฐาน, 128 สำหรับกลุ่มวอลุ่มขนาดใหญ่, และ 1024 สำหรับกลุ่มวอลุ่มที่ปรับสเกลได้
พาร์ติชันฟิลิคัล	กลุ่มวอลุ่มปกติและขนาดใหญ่: (1016 x ปัจจัยกลุ่มวอลุ่ม) ต่อฟิลิคัลวอลุ่มในขนาดสูงสุดถึง 1024 MB สำหรับแต่ละรายการ กลุ่มวอลุ่มที่ปรับสเกลได้: 2097152 พาร์ติชันในขนาดสูงสุด 128 GB ไม่มีปัจจัยกลุ่มวอลุ่ม สำหรับกลุ่มวอลุ่มที่ปรับสเกลได้
โลจิคัลวอลุ่ม	MAXLVS สำหรับแต่ละกลุ่มวอลุ่ม ซึ่งเป็น 255 สำหรับกลุ่มวอลุ่มมาตรฐาน, 511 สำหรับกลุ่มวอลุ่มขนาดใหญ่, และ 4095 สำหรับกลุ่มวอลุ่มที่ปรับสเกลได้

หากคุณสร้างกลุ่มวอลุ่มขึ้นก่อนหน้าการบังคับใช้ข้อจำกัด 1016 พาร์ติชันฟิลิคัล ต่อฟิลิคัลวอลุ่ม พาร์ติชันเก่า (ไม่มีข้อมูลปัจจุบันอีกต่อไป) ในกลุ่มวอลุ่มจะไม่มี การติดตาม อย่างถูกต้อง ยกเว้นว่าคุณแปลงกลุ่มวอลุ่มเป็นสถานะที่สนับสนุน คุณ สามารถแปลงกลุ่มวอลุ่มโดยใช้คำสั่ง `chvg -t` มีการเลือกค่าปัจจัยที่เหมาะสมโดยค่าดีฟอลต์เพื่อสนับสนุนดิสก์ที่ใหญ่ที่สุดในกลุ่มวอลุ่ม

ตัวอย่างเช่น ถ้าคุณสร้างกลุ่มวอลุ่มที่มีดิสก์ 9 GB และขนาดพาร์ติชัน 4 MB กลุ่มวอลุ่มนี้จะมีพาร์ติชันประมาณ 2250 โดยใช้ปัจจัยการแปลงเป็น 3 ($1016 * 3 = 3048$) ทำให้สามารถ ติดตาม 2250 พาร์ติชันทั้งหมดได้อย่างถูกต้อง การแปลงกลุ่มวอลุ่มมาตรฐาน หรือขนาดใหญ่ด้วยปัจจัยที่สูงขึ้นสามารถรวมดิสก์ที่ใหญ่ขึ้น ของพาร์ติชันได้สูงสุดถึง $1016 * \text{ปัจจัย}$ คุณยังสามารถระบุ ปัจจัยที่สูงขึ้นเมื่อคุณสร้างกลุ่มวอลุ่ม เพื่อสนับสนุน ดิสก์ที่ใหญ่ขึ้นด้วยพาร์ติชันขนาดเล็ก

การดำเนินงานเหล่านี้ลดจำนวนดิสก์ทั้งหมดซึ่งคุณสามารถเพิ่ม ลงในกลุ่มวอลุ่ม จำนวนดิสก์สูงสุดใหม่ซึ่งคุณสามารถเพิ่ม จะเป็น MAXPVS/ปัจจัย ตัวอย่างเช่น สำหรับกลุ่มวอลุ่มปกติ ปัจจัย 2 จะลดจำนวนดิสก์สูงสุดในกลุ่มวอลุ่ม เป็น 16 ($32/2$) สำหรับกลุ่มวอลุ่มขนาดใหญ่ ปัจจัย 2 จะลด จำนวนดิสก์สูงสุดในกลุ่มวอลุ่มเป็น 64 ($128/2$)

ขีดจำกัดขนาดอุปกรณ์ LVM

ขีดจำกัดต่อไปนี้เป็นขีดจำกัดทางสถาปัตยกรรม LVM หากต้องการ LVM Bad Block Relocation ขนาด PV ต้องไม่ใหญ่กว่า 128 GB สำหรับขีดจำกัดขนาด ของอุปกรณ์หน่วยเก็บเฉพาะให้อ้างอิงเอกสารคู่มืออุปกรณ์หน่วยเก็บ

ขีดจำกัด ขนาดต่อไปนี้เป็นใช้สำหรับเคอร์เนล 64-บิต:

VG ดั้งเดิม

ขีดจำกัด PV: $1\text{GB (PP)} * 16256 \text{ (PPs/PV, ปัจจัย=16)} = 15.9 \text{ TB}$

ขีดจำกัด LV: $1\text{GB (PP)} * 32512 \text{ (PPs/VG)} = 31.8 \text{ TB}$

VG ขนาดใหญ่

ขีดจำกัด PV: $1\text{GB (PP)} * 65024 \text{ (PPs/PV, ปัจจัย=64)} = 63.5 \text{ TB}$

ขีดจำกัด LV: $1\text{GB (PP)} * 130048 \text{ (PPs/VG)} = 127 \text{ TB}$

SVG ขีดจำกัด PV & LV: $128\text{GB (PP)} * 2048\text{K (PPs/PV)} = 256 \text{ PB}$

ขีดจำกัดขนาดต่อไปนี้เป็นใช้สำหรับเคอร์เนล 32-บิต:

VG ทุกชนิด

ขีดจำกัด PV: < 1 TB

ขีดจำกัด LV: < 1 TB

การกำหนดคอนฟิก Logical Volume Storage

ด้วย Logical Volume Storage คุณสามารถมีร์เรอร์กลุ่มวอลุ่ม กำหนดโลจิคัลวอลุ่ม และลบดิสก์ด้วยระบบที่กำลังรัน

การมีร์เรอร์กลุ่มวอลุ่ม:

สถานการณ์จำลองเหล่านี้อธิบายวิธีการมีร์เรอร์กลุ่มวอลุ่มปกติ

คำแนะนำต่อไปนี้จะแสดงวิธีการมีร์เรอร์ กลุ่มวอลุ่มรากโดยใช้ System Management Interface Tool (SMIT)

(เลือก กลุ่มวอลุ่มในคอนเทนเนอร์ วอลุ่ม แล้วเลือก มีร์เรอร์ จาก เมนู ที่เลือกไว้) ผู้ดูแลระบบที่มีประสบการณ์สามารถใช้คำสั่ง `mirrorvg`

1. ด้วยสิทธิการใช้งานราก ให้เพิ่มดิสก์ลงในกลุ่มวอลุ่มโดยใช้พารามิเตอร์ SMIT ต่อไปนี้:

```
smit extendvg
```

2. มีร์เรอร์กลุ่มวอลุ่มบนดิสก์ใหม่โดยพิมพ์พารามิเตอร์ SMIT ต่อไปนี้:

```
smit mirrorvg
```

3. ในพาเนลแรก ให้เลือกกลุ่มวอลุ่มสำหรับการมีร์เรอร์

4. ในพาเนลที่สอง คุณสามารถกำหนดอ็อปชันการมีร์เรอร์หรือยอมรับ ค่าดีฟอลต์ วิธีใช้ทางออนไลน์มีอยู่ถ้าคุณต้องการ

หมายเหตุ: เมื่อคุณทำพาเนล SMIT เสร็จแล้ว และคลิก ตกลง หรือออก คำสั่งที่ดำเนินการอาจใช้เวลานานพอสมควรกว่าจะเสร็จสมบูรณ์ เวลาที่ใช้ขึ้นอยู่กับ การตรวจสอบข้อผิดพลาด ขนาดและจำนวนของโลจิคัลวอลุ่ม ในกลุ่มวอลุ่ม และเวลาที่ใช้ในการซิงโครไนซ์โลจิคัลวอลุ่ม ที่มีร์เรอร์ใหม่

ณ จุดนี้ การเปลี่ยนแปลงทั้งหมดในโลจิคัลวอลุ่ม จะถูกมีร์เรอร์ตามที่คุณระบุไว้ในพาเนล SMIT

งานที่เกี่ยวข้อง:

“การมีร์เรอร์กลุ่มวอลุ่มราก”

สถานการณ์จำลองต่อไปนี้อธิบายวิธีการมีร์เรอร์กลุ่มวอลุ่มราก (rootvg)

การมีร์เรอร์กลุ่มวอลุ่มราก:

สถานการณ์จำลองต่อไปนี้อธิบายวิธีการมีร์เรอร์กลุ่มวอลุ่มราก (rootvg)

หมายเหตุ: การมีร์เรอร์กลุ่มวอลุ่มรากต้องการการจัดการระบบ ขั้นสูง หากทำไม่ถูกต้อง คุณอาจทำให้ระบบของคุณบูตไม่ได้

ในสถานการณ์จำลองต่อไปนี้อยู่บน `hdisk01` และทำมีร์เรอร์ไปยังดิสก์ชื่อ `hdisk11`:

1. ตรวจสอบว่า `hdisk11` ได้รับการสนับสนุนโดย AIX เป็น อุปกรณ์บูต:

```
bootinfo -B hdisk11
```

หากคำสั่งนี้ส่งคืนค่าเป็น 1 ดิสก์ที่เลือกสามารถบูตได้โดย AIX ค่าอื่นใดบ่งชี้ว่า `hdisk11` ไม่ได้เป็นตัวเลือกสำหรับการมีร์เรอร์ `rootvg`

2. ขยาย rootvg เพื่อรวม hdisk11 โดยใช้คำสั่งต่อไปนี้:

```
extendvg rootvg hdisk11
```

หากคุณได้รับข้อความแสดงข้อผิดพลาดต่อไปนี้:

```
0516-050 Not enough descriptor space left in this volume group, Either try adding a smaller PV or use another volume group.
```

หรือข้อความคล้ายกับ:

```
0516-1162 extendvg: Warning, The Physical Partition size of 16 requires the creation of 1084 partitions for hdisk11. The limitation for volume group rootvg is 1016 physical partitions per physical volume. Use chvg command with the -t option to attempt to change the maximum physical partitions per Physical Volume for this volume group.
```

คุณมีอีกพชั่นต่อไปนี้:

- มิร์เรอร์ rootvg ไปยังดิสก์ที่ว่างซึ่งเป็นของ rootvg อยู่แล้ว
- ใช้ดิสก์ที่เล็กลง
- เปลี่ยนจำนวนพาร์ติชันสูงสุดที่สนับสนุนโดย rootvg โดยใช้ โพรซีเจอร์ต่อไปนี้:
 - a. ตรวจสอบข้อความเพื่อดูจำนวนของพาร์ติชันฟิสิคัลที่ต้องการสำหรับ ดิสก์ปลายทาง และจำนวนสูงสุดที่ได้รับการสนับสนุนจาก rootvg ในปัจจุบัน
 - b. ใช้คำสั่ง `chvg -t` เพื่อคุณจำนวนพาร์ติชันสูงสุด ที่ใช้ได้ในปัจจุบันใน rootvg (ในตัวอย่างข้างบนคือ 1016) กับจำนวนที่มากกว่าพาร์ติชันฟิสิคัลที่ต้องการสำหรับดิสก์ ปลายทาง (ในตัวอย่างข้างบนคือ 1084) ตัวอย่างเช่น:

```
chvg -t 2 rootvg
```
 - c. ออกใช้คำสั่ง `extendvg` อีกครั้งที่ตอนต้นของขั้นตอนที่ 2

3. มิร์เรอร์ rootvg โดยใช้พชั่นการแม็ปที่แน่นอน ดังแสดงใน คำสั่งต่อไปนี้:

```
mirrorvg -m rootvg hdisk11
```

คำสั่งนี้จะปิดค้ประกอบเมื่อกลุ่มวอลุ่มเป็น rootvg หากคุณไม่ได้ใช้พชั่นการแม็ปที่แน่นอน คุณต้องตรวจสอบว่าสำเนาใหม่ ของบูตโลจิคัลวอลุ่มซึ่งเป็น hd5 ประกอบด้วยพาร์ติชันที่ต่อกัน

4. เริ่มต้นบูตเร็กคอร์ดและอุปกรณ์ทั้งหมด โดยใช้คำสั่งต่อไปนี้:

```
bosboot -a
```

5. เริ่มต้นรายการบูตด้วยคำสั่งต่อไปนี้:

```
bootlist -m normal hdisk01 hdisk11
```

หมายเหตุ:

- a. แม้ว่าคำสั่ง `bootlist` ระบุ hdisk11 เป็นบูตดิสก์อื่น แต่ไม่สามารถรับประกันได้ว่าระบบจะใช้ hdisk11 เป็น อุปกรณ์บูต ถ้า hdisk01 ล้มเหลว ในกรณีดังกล่าว คุณอาจ ต้องบูตจากสื่อบันทึกผลิตภัณฑ์ เลือก การบำรุงรักษา และออกใช้คำสั่ง `bootlist` อีกครั้งโดยไม่ต้องระบุชื่อดิสก์ที่ ล้มเหลว
- b. หากฮาร์ดแวร์โมเดลของคุณไม่สนับสนุนคำสั่ง `bootlist` คุณยังคงสามารถมิร์เรอร์ rootvg ได้ แต่คุณต้องเลือกบูตดิสก์อื่น เมื่อดิสก์ดั้งเดิมไม่พร้อมใช้งาน

งานที่เกี่ยวข้อง:

“การมีเรออร์กุ่มวอลุ่ม” ในหน้า 431

สถานการณ์จำลองเหล่านี้อธิบายวิธีการมีเรออร์กุ่มวอลุ่มปกติ

การกำหนดโลจิคัลวอลุ่มแบบดิบสำหรับแอฟลิเคชัน:

โลจิคัลวอลุ่มแบบดิบ เป็นพื้นที่ของพื้นที่ว่างดิสก์ฟิสิคัลและ โลจิคัลซึ่งอยู่ภายใต้การควบคุมโดยตรงของแอฟลิเคชัน เช่น ฐานข้อมูลหรือพาร์ติชัน แทนภายใต้การควบคุมโดยตรงของ ระบบปฏิบัติการหรือระบบไฟล์

การบายพาสระบบไฟล์สามารถให้ ประสิทธิภาพที่ดีขึ้นจากการควบคุมแอฟลิเคชัน โดยเฉพาะจาก แอฟลิเคชันฐานข้อมูล อย่างไรก็ตาม ระดับการปรับปรุงขึ้นอยู่กับ ปัจจัยต่างๆ เช่น ขนาดของฐานข้อมูลหรือไดเรกทอรีของแอฟลิเคชัน

หมายเหตุ: คุณจะต้องระบุแอฟลิเคชันที่มีไฟล์อุปกรณ์พิเศษ อักขระหรือบล็อกสำหรับโลจิคัลวอลุ่มแบบดิบใหม่ ตาม ความเหมาะสม แอฟลิเคชันจะลิงก์กับไฟล์อุปกรณ์นี้เมื่อ แอฟลิเคชันพยายามเปิด อ่าน เขียน และอื่นๆ

ข้อควรสนใจ: แต่ละโลจิคัลวอลุ่มมีบล็อกควบคุมของโลจิคัลวอลุ่ม (LVCB) อยู่ในบล็อกแรก ขนาดของ LVCB คือขนาด บล็อกของวอลุ่มฟิสิคัลภายใน กลุ่มวอลุ่ม ข้อมูลเริ่มที่บล็อกที่สองของวอลุ่มฟิสิคัล ในโลจิคัลวอลุ่มแบบดิบ ไม่มีการป้องกัน LVCB หากแอฟลิเคชัน เขียนทับ LVCB คำสั่งที่โดยปกติอัปเดต LVCB จะล้มเหลว และสร้างข้อความ แม้ว่าโลจิคัลวอลุ่มจะ ดำเนินงานได้ถูกต้อง ต่อไปและการเขียนทับสามารถเป็นเหตุการณ์ที่ได้รับอนุญาตเท่านั้น ไม่แนะนำให้เขียนทับ LVCB

คำแนะนำต่อไปนี้นำใช้ SMIT และอินเตอร์เฟซบรรทัดคำสั่งเพื่อกำหนดโลจิคัลวอลุ่ม แบบดิบ

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

1. ด้วยสิทธิการใช้งานราก ให้ค้นหาพาร์ติชันฟิสิคัลว่าง ซึ่งคุณสามารถสร้างโลจิคัลวอลุ่มแบบดิบโดยพิมพ์พาด่วน SMIT ต่อไปนี้:

```
smit lspv
```

2. เลือกดิสก์

3. ยอมรับค่าดีฟอลต์ในไดอะล็อกที่สอง (สถานะ) และคลิก ตกลง

4. คุณค่าในฟิลด์ FREE PPs ด้วยค่าในฟิลด์ PP SIZE เพื่อให้ได้จำนวนเมกะไบต์ทั้งหมดที่มีอยู่สำหรับโลจิคัลวอลุ่มแบบดิบ บน ดิสก์ที่เลือก หากจำนวนพื้นที่ว่างไม่เพียงพอ ให้เลือกดิสก์อื่นจนกว่าคุณพบดิสก์ที่มีพื้นที่ว่าง เพียงพอ

5. ออกจาก SMIT

6. ใช้คำสั่ง `mklv` เพื่อสร้างโลจิคัลวอลุ่ม แบบดิบ คำสั่งต่อไปนี้สร้างโลจิคัลวอลุ่มแบบดิบ ที่ชื่อ `lvdb2003` ในกลุ่มวอลุ่ม `db2vg` โดยใช้พาร์ติชันฟิสิคัล `38 4-MB`:

```
mklv -y lvdb2003 db2vg 38
```

ใช้แฟล็ก `-y` เพื่อระบุชื่อสำหรับโลจิคัลวอลุ่มแทน การใช้ชื่อที่ระบบสร้างขึ้น

ณ จุดนี้ โลจิคัลวอลุ่มแบบดิบถูกสร้างขึ้นแล้ว หากคุณ แสดงรายการเนื้อหาของกลุ่มวอลุ่ม โลจิคัลวอลุ่มแบบดิบแสดงขึ้น ด้วย ชนิดดีฟอลต์ ซึ่งเป็น `jfs` รายการชนิดนี้ของโลจิคัลวอลุ่ม เป็นเพียงเลเบล และไม่ไดบ่งชี้ว่าติดตั้งระบบไฟล์ สำหรับโลจิคัลวอลุ่ม แบบดิบของคุณ

โปรดศึกษาคำแนะนำของแอฟลิเคชัน เพื่อดูวิธีการเปิด `/dev/rawLVname` และ วิธีการใช้พื้นที่ว่างดิบนี้

หลักการที่เกี่ยวข้อง:

“การกำหนดคอนฟิก Logical Volume Manager” ในหน้า 388

Logical Volume Manager (LVM) มีการติดตั้งพร้อมกับระบบปฏิบัติการ พื้นฐานและต้องการคอนฟิกเรชันเพิ่มเติม อย่างไรก็ตาม ต้องกำหนดคอนฟิกดิสก์ และกำหนดเป็นฟิลิคัลลวอลุ่มก่อน LVM สามารถใช้ดิสก์นั้นได้

สิ่งอ้างอิงที่เกี่ยวข้อง:

“คำสั่งการบำรุงรักษา LVM และพาร์ต่วน” ในหน้า 389

ภารกิจง่ายที่สุดซึ่งคุณอาจต้องการเมื่อบำรุงรักษาเอนทิตีที่ LVM ควบคุม (ฟิลิคัลลวอลุ่ม กลุ่มวอลุ่ม และระบบไฟล์) คือการจัดกลุ่มภายในตารางต่อไปนี้

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `mklv`

 Logical Volume Manager จาก A ถึง Z: บทนำและแนวคิด

การยกเลิกมีร์เรอร์กลุ่มวอลุ่ม:

คุณสามารถยกเลิกมีร์เรอร์กลุ่มวอลุ่มได้

ข้อควรสนใจ: การยกเลิกมีร์เรอร์กลุ่มวอลุ่มต้องการ ประสบการณ์ในการจัดการระบบขั้นสูง เพราะหากทำไม่ถูกต้อง ระบบ อาจไม่สามารถบูตได้

ในสถานการณ์จำลองต่อไปนี้ กลุ่มวอลุ่มรากอยู่บน `hdisk01` และมีการมีร์เรอร์บน `hdisk11` ตัวอย่างนี้ลบมีร์เรอร์ บน `hdisk11` โพรซีเจอร์เหมือนกัน ไม่ว่าดิสก์ล่าสุดที่คุณบูตเป็นดิสก์ใดก็ตาม

1. ใช้คำสั่งต่อไปนี้เพื่อยกเลิกมีร์เรอร์กลุ่มวอลุ่มรากบน `hdisk11`:

```
unmirrorvg rootvg hdisk11
```

คำสั่ง `unmirrorvg` เปิดองค์ประกอบขึ้นมาใหม่ สำหรับกลุ่มวอลุ่มราก

2. ใช้คำสั่งต่อไปนี้เพื่อลดดิสก์ออกจากกลุ่มวอลุ่ม ราก:

```
reducevg rootvg hdisk11
```

3. ใช้คำสั่งต่อไปนี้เพื่อเริ่มต้นบูตเรียกคอร์ตของดิสก์ที่เหลืออยู่ อีกครั้ง:

```
bosboot -a -d /dev/hdisk01
```

4. ใช้คำสั่งต่อไปนี้เพื่อแก้ไขรายการบูตเพื่อลบ ดิสก์ที่ยกเลิกมีร์เรอร์ออกจากรายการ:

```
bootlist -m normal hdisk01
```

ดิสก์ถูกยกเลิกมีร์เรอร์

การเอาดิสก์ออกในขณะที่ระบบยังคงพร้อมใช้งาน:

โพรซีเจอร์ต่อไปนี้อธิบายวิธีการเอาดิสก์ออก โดยใช้คุณลักษณะ `hot-removability` ซึ่งอนุญาตให้คุณเอาดิสก์ออกได้โดยไม่ต้อง ปิดระบบ คุณลักษณะนี้มีอยู่บนบางระบบเท่านั้น

Hot removability มีประโยชน์เมื่อคุณต้องการ:

- เอาดิสก์ที่มีข้อมูลในกลุ่มวอลุ่มที่ไม่ใช่ `rootvg` แยกต่างหากออก สำหรับวัตถุประสงค์ด้านความปลอดภัยหรือการบำรุงรักษา

- เอาดิสก์ออกจากกลุ่มวอลุ่มอย่างถาวร
- แก้ไขความล้มเหลวดิสก์

การเอาดิสก์ที่มีข้อมูลออก:

ใช้ไพรซีเดอร์นี้เพื่อเอาดิสก์ที่มีข้อมูลออก โดยไม่ต้องปิดระบบ

ดิสก์ที่คุณกำลังเอาออกต้องอยู่ในกลุ่มวอลุ่มที่ไม่ใช่ rootvg แยกต่างหาก ใช้ไพรซีเดอร์นี้เมื่อคุณต้องการย้ายดิสก์ไปยังระบบอื่น

1. เมื่อต้องการแสดงรายการกลุ่มวอลุ่มที่เชื่อมโยงกับดิสก์ซึ่งคุณต้องการ เอาออก ให้พิมพ์:

```
smit lspv
```

เอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```
PHYSICAL VOLUME:   hdisk2                VOLUME GROUP:   imagesvg
PV IDENTIFIER:    00083772caa7896e VG IDENTIFIER    0004234500004c00000000e9b5cac262
PV STATE:        active
STALE PARTITIONS: 0                ALLOCATABLE:    yes
PP SIZE:         16 megabyte(s)          LOGICAL VOLUMES: 5
TOTAL PPs:       542 (8672 megabytes)    VG DESCRIPTORS: 2
FREE PPs:        19 (304 megabytes)      HOT SPARE:      no
USED PPs:        523 (8368 megabytes)
FREE DISTRIBUTION: 00..00..00..00..19
USED DISTRIBUTION: 109..108..108..108..90
```

ชื่อของกลุ่มวอลุ่มแสดงขึ้นในฟิลด์ VOLUME GROUP ในตัวอย่างนี้ กลุ่มวอลุ่มคือ imagesvg

2. เมื่อต้องการตรวจสอบว่าดิสก์อยู่ในกลุ่มวอลุ่มที่ไม่ใช่ rootvg แยกต่างหาก ให้พิมพ์:

```
smit lsvg
```

จากนั้นเลือกกลุ่มวอลุ่มที่เชื่อมโยงกับดิสก์ (ในตัวอย่างนี้ imagesvg) เอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

```
VOLUME GROUP:   imagesvg                VG IDENTIFIER:  0004234500004c00000000e9b5cac262
VG STATE:       active                   PP SIZE:        16 megabyte(s)
VG PERMISSION:  read/write                TOTAL PPs:      542 (8672 megabytes)
MAX LVs:        256                     FREE PPs:       19 (304 megabytes)
LVs:            5                       USED PPs:       523 (8368 megabytes)
OPEN LVs:       4                       QUORUM:         2
TOTAL PVs:      1                       VG DESCRIPTORS: 2
STALE PVs:      0                       STALE PPs:      0
ACTIVE PVs:     1                       AUTO ON:        yes
MAX PPs per PV: 1016                    MAX PVs:        32
LTG size:       128 kilobyte(s)         AUTO SYNC:      no
HOT SPARE:      no
```

ในตัวอย่างนี้ ฟิลด์ TOTAL PVs บ่งชี้ว่ามีเพียง หนึ่งฟิลต์ลวอลุ่มเท่านั้นที่เชื่อมโยงกับ imagesvg เนื่องจากข้อมูลทั้งหมดในกลุ่มวอลุ่มนี้มีอยู่บน hdisk2 ดังนั้นจึงสามารถเอา hdisk2 ออกได้โดยใช้ไพรซีเดอร์นี้

3. เมื่อต้องการถอนติดตั้งระบบไฟล์ใดๆ บนโลจิคัลวอลุ่มบน ดิสก์ ให้พิมพ์:

```
smit umountfs
```

4. เมื่อต้องการหยุดการเรียกทำงานกลุ่มวอลุ่ม ให้พิมพ์:

smit varyoffvg

5. เมื่อต้องการเอ็กซ์พอร์ตกลุ่มวอลุ่ม ให้พิมพ์:

smit exportvg

6. เมื่อต้องการถอนดิสก์ให้พิมพ์:

smit rmvdsk

7. ดูที่จอแสดงผล LED สำหรับดิสก์ซึ่งคุณต้องการเอาออก ต้องแน่ใจว่า LED สีเหลืองปิด (ไม่ส่องสว่าง)

8. ถอดดิสก์ออกทางกายภาพ สำหรับข้อมูลเพิ่มเติม เกี่ยวกับโปรซีเดอ์การเอาออก ให้ดูที่คู่มือเซอร์วิสของเครื่องของคุณ

ณ จุดนี้ ดิสก์ถูกเอาออกจากระบบทั้งทางกายภาพและ ทางตรรกะ หากคุณกำลังเอาดิสก์นี้ออกอย่างถาวร โปรซีเดอ์นี้ เสร็จสมบูรณ์แล้ว

งานที่เกี่ยวข้อง:

“การอิมพอร์ตหรือการเอ็กซ์พอร์ตกลุ่มวอลุ่ม” ในหน้า 395

ตารางต่อไปนี้อธิบายวิธีการใช้การอิมพอร์ตและการเอ็กซ์พอร์ตเพื่อย้ายกลุ่มวอลุ่มที่ผู้ใช้กำหนดจากระบบหนึ่งไปยังอีกระบบหนึ่ง (ไม่สามารถอิมพอร์ตหรือเอ็กซ์พอร์ต กลุ่มวอลุ่ม rootvg)

“การเพิ่มดิสก์ในขณะที่ระบบยังคงเปิดอยู่” ในหน้า 391

โปรซีเดอ์ต่อไปนี้อธิบายวิธีการเปิดและกำหนดคอนฟิก ดิสก์โดยใช้คุณลักษณะ hot-removability ซึ่งอนุญาตให้คุณเพิ่มดิสก์ได้โดยไม่ต้อง ปิดระบบ

การเอาดิสก์ที่ไม่มีข้อมูลออก:

โปรซีเดอ์ต่อไปนี้อธิบายวิธีการเอาดิสก์ที่ไม่มีข้อมูล หรือไม่มีข้อมูลซึ่งคุณต้องการเก็บไว้

ข้อควรสนใจ: โปรซีเดอ์ต่อไปนี้จะลบข้อมูลใดๆ ที่อยู่บนดิสก์

1. เมื่อต้องการถอนติดตั้งระบบไฟล์ใดๆ บนโลจิคัลวอลุ่มบน ดิสก์ให้พิมพ์:

smit umountfs

2. เมื่อต้องการหยุดการเรียกทำงานกลุ่มวอลุ่มให้พิมพ์:

smit varyoffvg

3. เมื่อต้องการเอ็กซ์พอร์ตกลุ่มวอลุ่ม ให้พิมพ์:

smit exportvg

4. เมื่อต้องการถอนดิสก์ให้พิมพ์:

smit rmvdsk

5. ดูที่จอแสดงผล LED สำหรับดิสก์ซึ่งคุณต้องการเอาออก ต้องแน่ใจว่า LED สีเหลืองปิด (ไม่ส่องสว่าง)

6. ถอดดิสก์ออกทางกายภาพ สำหรับข้อมูลเพิ่มเติม เกี่ยวกับโปรซีเดอ์การเอาออก ให้ดูที่คู่มือเซอร์วิสของเครื่องของคุณ

ณ จุดนี้ ดิสก์ถูกเอาออกจากระบบทั้งทางกายภาพและ ทางตรรกะ หากคุณกำลังเอาดิสก์นี้ออกอย่างถาวร โปรซีเดอ์นี้ เสร็จสมบูรณ์แล้ว

งานที่เกี่ยวข้อง:

“การเพิ่มดิสก์ในขณะที่ระบบยังคงเปิดอยู่” ในหน้า 391

โปรซีเดอ์ต่อไปนี้อธิบายวิธีการเปิดและกำหนดคอนฟิก ดิสก์โดยใช้คุณลักษณะ hot-removability ซึ่งอนุญาตให้คุณเพิ่มดิสก์ได้โดยไม่ต้อง ปิดระบบ

การลบโลจิคัลวอลุ่มโดยการลบระบบไฟล์:

โพรซีเจอร์ต่อไปนี้อธิบายวิธีการลบระบบไฟล์ JFS หรือ JFS2, โลจิคัลวอลุ่มที่เชื่อมโยง, stanza ที่เชื่อมโยงในไฟล์ /etc/filesystems และสามารถเลือกที่จะลบจุดต่อ (ไดเร็กทอรี) ซึ่งติดตั้งระบบไฟล์

ข้อควรสนใจ: เมื่อคุณลบระบบไฟล์ คุณทำลาย ข้อมูลทั้งหมดในระบบไฟล์ที่ระบุและโลจิคัลวอลุ่ม

ถ้าคุณต้องการ ลบโลจิคัลวอลุ่มที่มีระบบไฟล์ชนิดอื่นติดตั้งอยู่บนนั้น หรือโลจิคัลวอลุ่มที่ไม่มีระบบไฟล์ คุณสามารถลบ โลจิคัลวอลุ่มได้อย่างเดียว

เมื่อต้องการลบระบบไฟล์ที่เจอร์นัลโดยใช้ SMIT ให้ใช้โพรซีเจอร์ ต่อไปนี้:

1. ถอนติดตั้งระบบไฟล์ที่ตั้งอยู่บนโลจิคัลวอลุ่มโดยใช้คำสั่ง คล้ายกับตัวอย่างต่อไปนี้:

```
umount /adam/usr/local
```

หมายเหตุ: คุณไม่สามารถใช้คำสั่ง `umount` บนอุปกรณ์ที่ใช้อยู่ อุปกรณ์ถูกใช้อยู่ถ้ามีไฟล์เปิดอยู่ด้วยเหตุผลใดๆ หรือ ถ้าไดเร็กทอรีปัจจุบันของผู้ใช้ อยู่บนอุปกรณ์นั้น

2. เมื่อต้องการลบระบบไฟล์ ให้พิมพ์พาด่วนต่อไปนี้:

```
smit rmfs
```

3.

1. เลือกชื่อของระบบไฟล์ซึ่งคุณต้องการลบออก
2. ไปที่ฟิลด์ ลบจุดต่อ และสลับไปยังการกำหนดค่าตามความชอบ ถ้าคุณเลือก ใช้ คำสั่งที่ใช้ยังจะ ลบจุดต่อ (ไดเร็กทอรี) ซึ่งติดตั้งระบบไฟล์ด้วย (ถ้า ไดเร็กทอรีว่าง)
3. กด Enter เพื่อลบระบบไฟล์ SMIT พร้อมทำให้คุณยืนยันว่า คุณต้องการลบระบบไฟล์หรือไม่
4. ยืนยันว่าคุณต้องการลบระบบไฟล์ SMIT แสดงข้อความเมื่อ ระบบไฟล์ถูกลบออกเสร็จเรียบร้อยแล้ว

ณ จุดนี้ ระบบไฟล์ ข้อมูล และโลจิคัลวอลุ่มที่เชื่อมโยง ของระบบไฟล์นั้นถูกลบออกจากระบบโดยสมบูรณ์แล้ว งานที่เกี่ยวข้อง:

“การลบโลจิคัลวอลุ่มอย่างเดียว”

ใช้โพรซีเจอร์นี้เพื่อลบโลจิคัลวอลุ่มที่มีระบบไฟล์ ชนิดอื่นติดตั้งอยู่บนนั้น หรือโลจิคัลวอลุ่มที่ไม่มี ระบบไฟล์

การลบโลจิคัลวอลุ่มอย่างเดียว:

ใช้โพรซีเจอร์นี้เพื่อลบโลจิคัลวอลุ่มที่มีระบบไฟล์ ชนิดอื่นติดตั้งอยู่บนนั้น หรือโลจิคัลวอลุ่มที่ไม่มี ระบบไฟล์

ข้อควรสนใจ: การลบโลจิคัลวอลุ่มทำลาย ข้อมูลทั้งหมดในระบบไฟล์ที่ระบุและโลจิคัลวอลุ่ม

โพรซีเจอร์ต่อไปนี้อธิบายวิธีการลบโลจิคัลวอลุ่มและระบบไฟล์ที่ เชื่อมโยง คุณสามารถใช้โพรซีเจอร์นี้เพื่อลบระบบไฟล์ที่ไม่ใช่ JFS หรือโลจิคัลวอลุ่มที่ไม่มีระบบไฟล์ หลังจากโพรซีเจอร์ต่อไปนี้อธิบายวิธีการลบโลจิคัลวอลุ่มแล้ว โพรซีเจอร์ยังอธิบายวิธีการลบ stanza ของระบบ ไฟล์ที่ไม่ใช่ JFS ในไฟล์ /etc/filesystems

เมื่อต้องการลบโลจิคัลวอลุ่มโดยใช้ SMIT ให้ใช้โพรซีเจอร์ต่อไปนี้:

1. ถ้าโลจิคัลวอลุ่มไม่มีระบบไฟล์ ให้ข้ามไปยังขั้นตอนที่ 4
2. ถอนติดตั้งระบบไฟล์ทั้งหมดที่เชื่อมโยงกับโลจิคัลวอลุ่ม โดยพิมพ์:

umount /FSname

โดยที่ /FSname คือ ชื่อพาทแบบเต็มของระบบไฟล์

หมายเหตุ:

- a. คำสั่ง **umount** จะล้มเหลวถ้าระบบไฟล์ซึ่งคุณกำลังพยายาม ถอนติดตั้ง ถูกใช้งานอยู่ในปัจจุบัน คำสั่ง **umount** ดำเนินการ เฉพาะถ้าไม่มีไฟล์ของระบบไฟล์เปิดอยู่และไม่มีไต่เรียกทอริปัจจุบันของผู้ใช้ใด อยู่บนอุปกรณ์นั้น
 - b. อีกชื่อหนึ่งสำหรับคำสั่ง **umount** คือ **umount** ชื่อทั้งสองนี้สามารถใช้ทดแทนกันได้
3. เมื่อต้องการแสดงข้อมูลซึ่งคุณต้องการทราบเกี่ยวกับระบบไฟล์ ให้พิมพ์ พาทด่วนต่อไปนี้:

```
smit lsfs
```

ข้อมูลต่อไปนี้เป็นรายการ บางส่วน:

Name	Nodename	Mount Pt	...
/dev/hd3	--	/tmp	...
/dev/locallv	--	/adam/usr/local	...

4. โดยสมมติระเบียบการตั้งชื่อ มาตรฐานสำหรับไอเท็มที่แสดงรายการลำดับที่สอง ระบบไฟล์มีชื่อว่า /adam/usr/local และ โลจิคัลวอลุ่มคือ locallv เมื่อต้องการตรวจสอบข้อมูลนี้ ให้พิมพ์พาทด่วน ต่อไปนี้:

```
smit ls1v2
```

ข้อมูลต่อไปนี้เป็นรายการ บางส่วน:

```
imagesvg:  
LV NAME          TYPE      LPs  PPs  PVs  LV STATE    MOUNT POINT  
hd3              jfs      4    4    1    open/syncd  /tmp  
locallv         mine     4    4    1    closed/syncd /adam/usr/local
```

5. เมื่อต้องการลบโลจิคัลวอลุ่ม ให้พิมพ์พาทด่วนต่อไปนี้ บนบรรทัดคำสั่ง:

```
smit rmlv
```

6. เลือกชื่อของโลจิคัลวอลุ่มซึ่งคุณต้องการลบออก
7. ไปที่ฟิลด์ **ลบจุดต่อ** และสลับไปยัง การกำหนดค่าตามความชอบ ถ้าคุณเลือก ใช้ คำสั่งที่ใช้ยังจะ **ลบจุดต่อ** (ไต่เรียกทอริ) ซึ่งติดตั้งระบบไฟล์ด้วย (ถ้ามี และถ้าไต่เรียกทอรินั้นว่าง)
8. กด Enter เพื่อลบโลจิคัลวอลุ่ม SMIT พร้อมทั้งให้คุณยืนยันว่าคุณต้องการลบโลจิคัลวอลุ่มหรือไม่
9. ยืนยันว่าคุณต้องการลบโลจิคัลวอลุ่ม SMIT แสดงข้อความเมื่อ โลจิคัลวอลุ่มถูกลบออกเสร็จเรียบร้อยแล้ว
10. หากโลจิคัลวอลุ่มมีระบบไฟล์ที่ไม่ใช่ JFS ติดตั้งอยู่บนนั้น ให้ลบระบบไฟล์และ stanza ที่เชื่อมโยงในไฟล์ /etc/filesystems ดังแสดงในตัวอย่างต่อไปนี้:

```
rmfs /adam/usr/local
```

หรือ คุณสามารถใช้ชื่อระบบไฟล์ดังนี้:

```
rmfs /dev/locallv
```

ณ จุดนี้ โลจิคัลวอลุ่มถูกลบออกแล้ว หากโลจิคัลวอลุ่มมี ระบบไฟล์ที่ไม่ใช่ JFS stanza ของระบบนั้นถูกลบออกจาก ไฟล์ /etc/filesystems แล้วด้วย

งานที่เกี่ยวข้อง:

“การลบโลจิกัลวอลุ่มโดยการลบระบบไฟล์” ในหน้า 437

โพรซีเจอร์ต่อไปนี้อธิบายวิธีการลบระบบไฟล์ JFS หรือ JFS2, โลจิกัลวอลุ่มที่เชื่อมโยง, stanza ที่เชื่อมโยงในไฟล์ /etc/filesystems และสามารถเลือกที่จะลบจุดต่อ (ไดเร็กทอรี) ซึ่งติดตั้งระบบไฟล์

การปรับขนาดกลุ่มวอลุ่ม RAID:

บนระบบที่ใช้ redundant array of independent disks (RAID), อ็อพชันคำสั่ง **chvg** และ **chpv** มีความสามารถในการเพิ่มดิสก์ไปยังกลุ่ม RAID และขยาย ขนาดของวอลุ่มฟิสิคัลที่ LVM ใช้โดยไม่ขัดจังหวะการใช้งาน หรือสภาพพร้อมใช้ของระบบ

หมายเหตุ:

1. คุณลักษณะนี้ไม่มีอยู่ในขณะที่กลุ่มวอลุ่มมีการเรียกใช้งานในโหมดคลาสสิก หรือในโหมดเกิดขึ้นพร้อมกันที่เพิ่มประสิทธิภาพ
2. ไม่สามารถปรับขนาดกลุ่มวอลุ่ม rootvg โดยใช้โพรซีเจอร์ต่อไปนี
3. ไม่สามารถปรับขนาดกลุ่มวอลุ่มที่มีพื้นที่การเลื่อนหน้าซึ่งใช้งานอยู่โดยใช้โพรซีเจอร์ต่อไปนี

ขนาดของดิสก์ทั้งหมดในกลุ่มวอลุ่มถูกตรวจสอบโดย อัตโนมัตเมื่อเรียกใช้กลุ่มวอลุ่ม (varyon) หากตรวจพบว่าขนาดเพิ่มขึ้น ระบบจะสร้างข้อความแจ้งขอมูล

โพรซีเจอร์ต่อไปนี้อธิบายวิธีการเพิ่มเติมดิสก์ในสภาวะแวดล้อม RAID:

1. เมื่อต้องการตรวจสอบการเติบโตของดิสก์และจำเป็นต้องปรับขนาดหรือไม่ให้ พิมพ์คำสั่งต่อไปนี้:

```
chvg -g VGname
```

โดยที่ *VGname* คือชื่อกลุ่มวอลุ่มของคุณ คำสั่งนี้ตรวจสอบดิสก์ทั้งหมดในกลุ่มวอลุ่ม หากดิสก์ใดๆ มีขนาดใหญ่ขึ้น คำสั่งจะพยายามเพิ่มพาร์ติชันฟิสิคัลลงในฟิสิคัลวอลุ่ม ถ้าจำเป็น คำสั่งจะกำหนดตัวคูณขีดจำกัด 1016 ที่เหมาะสมและแปลงกลุ่มวอลุ่ม เป็นกลุ่มวอลุ่มขนาดใหญ่

2. เมื่อต้องการปิดการเปลี่ยนตำแหน่งบล็อก LVM ที่ไม่ดีสำหรับกลุ่มวอลุ่ม ให้ พิมพ์คำสั่งต่อไปนี้:

```
chvg -b ny VGname
```

โดยที่ *VGname* คือชื่อกลุ่มวอลุ่มของคุณ

กลยุทธ์กลุ่มวอลุ่ม

ความล้มเหลวดิสก์เป็นความล้มเหลวฮาร์ดแวร์ที่พบบ่อยที่สุดในระบบ หน่วยเก็บ ตามด้วยความล้มเหลวของอะแดปเตอร์และแหล่งจ่ายไฟ การป้องกันความล้มเหลว ดิสก์โดยหลักแล้ว เกี่ยวข้องกับคอนฟิกูเรชันของโลจิกัลวอลุ่ม

เพื่อป้องกันความล้มเหลวของอะแดปเตอร์และแหล่งจ่ายไฟ ให้พิจารณา การตั้งค่าฮาร์ดแวร์พิเศษสำหรับกลุ่มวอลุ่มเฉพาะใดๆ คอนฟิกูเรชันดังกล่าว มีสองอะแดปเตอร์และอย่างน้อยหนึ่งดิสก์ต่ออะแดปเตอร์ พร้อมการมีเรอร์บน อะแดปเตอร์ และคอนฟิกูเรชันกลุ่มวอลุ่มที่ไม่ใช่ของค์ประกอบ ค่าใช้จ่ายเพิ่มเติม ของคอนฟิกูเรชันนี้ไม่เหมาะสมสำหรับเซิร์ฟเวอร์หรือระบบทั้งหมด แนะนำให้ใช้ เฉพาะสำหรับระบบที่ต้องการการมีอยู่สูง (จนถึงวินาทีล่าสุด) เท่านั้น ขึ้นอยู่กับคอนฟิกูเรชัน การมีอยู่สูงสามารถครอบคลุมความล้มเหลวฮาร์ดแวร์ ที่เกิดขึ้นระหว่างสำเนาสำรองล่าสุดและรายการข้อมูลปัจจุบัน การมีอยู่สูง ไม่ได้ใช้กับไฟล์ที่ลบออกโดยอุบัติเหตุ

หลักการที่เกี่ยวข้อง:

“กลยุทธ์โลจิกัลวอลุ่ม” ในหน้า 441

นโยบายที่อธิบายที่นี้ช่วยคุณกำหนดกลยุทธ์สำหรับการใช้งาน โลจิกัลวอลุ่มซึ่งมุ่งเน้นการผสมผสานระหว่างการมีอยู่

ประสิทธิภาพการทำงาน และต้นทุนที่เหมาะสมสำหรับไซต์ของคุณ

ควรรสร้างกลุ่มวอลุ่มแยกต่างหากเมื่อไร:

มีหลายเหตุผลซึ่งคุณอาจต้องการจัดระเบียบฟิสิคัลวอลุ่ม เข้าเป็นกลุ่มวอลุ่มที่แยกต่างหากจาก rootvg

- เพื่อให้บำรุงรักษาได้ปลอดภัยขึ้นและง่ายขึ้น
 - อัปเดตระบบปฏิบัติการ การติดตั้งใหม่ และการกู้คืนความเสียหายมีความปลอดภัย มากขึ้นเนื่องจากคุณสามารถแยก ระบบไฟล์ผู้ใช้ออกจากระบบปฏิบัติการ เพื่อให้ไฟล์ผู้ใช้ไม่ได้รับอันตรายในระหว่างการดำเนินงานเหล่านี้
 - การบำรุงรักษาง่ายขึ้นเนื่องจากคุณสามารถอัปเดตหรือติดตั้งระบบปฏิบัติการอีก ครั้งได้โดยไม่ต้องเรียกคืนข้อมูลผู้ใช้ ตัวอย่างเช่น ก่อนอัปเดต คุณสามารถลบกลุ่มวอลุ่มที่ผู้ใช้กำหนดออกจากระบบโดยการถอนติดตั้ง ระบบไฟล์ของกลุ่ม วอลุ่มนั้น ยกเลิกการเรียกใช้กลุ่มนั้นโดยใช้คำสั่ง `varyoffvg` จากนั้นเอ็กซ์พอร์ตกลุ่มโดยใช้คำสั่ง `exportvg` หลังจาก อัปเดตซอฟต์แวร์ระบบแล้ว คุณสามารถแนะนำกลุ่มวอลุ่มที่ผู้ใช้กำหนดอีกครั้ง โดยใช้คำสั่ง `importvg` จากนั้น ติดตั้ง ระบบไฟล์ของกลุ่มนั้น อีกครั้ง
- สำหรับขนาดของพาร์ติชันฟิสิคัลที่แตกต่างกัน ฟิสิคัลวอลุ่มทั้งหมดภายใน กลุ่มวอลุ่มเดียวกันต้องมีขนาดพาร์ติชันฟิสิคัล เดียวกัน เมื่อต้องการฟิสิคัล วอลุ่มที่มีขนาดพาร์ติชันฟิสิคัลแตกต่างกัน ให้วางแต่ละขนาดในกลุ่มวอลุ่ม แยกต่างหาก
- เมื่อต้องการลักษณะองค์ประกอบที่แตกต่างกัน หากคุณมีระบบไฟล์ ซึ่งคุณต้องการสร้างกลุ่มวอลุ่มที่ไม่ใช่องค์ประกอบ ให้ใช้ กลุ่มวอลุ่มแยกต่างหากสำหรับข้อมูลนั้น ระบบไฟล์อื่นทั้งหมดควรยังคง อยู่ในกลุ่มวอลุ่มที่ดำเนินงานภายใต้องค์ ประกอบ
- เพื่อความปลอดภัย ตัวอย่างเช่น คุณอาจต้องการลบกลุ่มวอลุ่มตอน กลางคืน
- เพื่อสลับฟิสิคัลวอลุ่มระหว่างระบบ ถ้าคุณสร้างกลุ่มวอลุ่มแยกต่างหาก สำหรับแต่ละระบบบนอะแด็ปเตอร์ที่สามารถเข้า ถึงได้จากมากกว่า หนึ่งระบบ คุณสามารถสลับฟิสิคัลวอลุ่มระหว่างระบบที่สามารถเข้าถึงได้บนอะแด็ปเตอร์นั้นโดยไม่ต้อง ชัดจังหวะการดำเนินงานปกติ ของอย่างใดอย่างหนึ่ง (ใหญ่ที่คำสั่ง `varyoffvg`, `exportvg`, `importvg`, และ `varyonvg`)

การมีอยู่สูงในกรณีของความล้มเหลววิสก:

เมธอดหลักที่ใช้เพื่อป้องกันความล้มเหลววิสกเกี่ยวข้องกับ ค่าติดตั้งคอนฟิกูเรชันโลจิคัลวอลุ่ม เช่น การมีร์เรอร์

ในขณะที่กลุ่มวอลุ่มเป็นข้อควรพิจารณาลำดับที่สอง แต่กลุ่มวอลุ่มสื่อความทาง เศรษฐกิจที่สำคัญเนื่องจากเกี่ยวข้องกับ จำนวนของฟิสิคัลวอลุ่ม ต่อกลุ่มวอลุ่ม:

- คอนฟิกูเรชันองค์ประกอบซึ่งเป็นค่าดีฟอลต์ รักษากลุ่มวอลุ่มให้ใช้งานอยู่ (varied on) トラบเท่าที่เมืองค์ประกอบ (51%) ของดิสก์อยู่ในกรณีส่วนใหญ่ คุณต้องการสามดิสก์เป็นอย่างน้อยพร้อมด้วยสำเนาที่มีร์เรอร์ในกลุ่มวอลุ่ม เพื่อป้องกัน ความล้มเหลววิสก
- คอนฟิกูเรชันที่ไม่ใช่เมืองค์ประกอบรักษากลุ่มวอลุ่มให้ใช้งานอยู่ (varied on) トラบเท่าที่มีหนึ่ง VGDA อยู่บนดิสก์ ด้วย คอนฟิกูเรชันนี้ คุณต้องการเพียงสองดิสก์พร้อมด้วยสำเนาที่มีร์เรอร์ในกลุ่มวอลุ่ม เพื่อป้องกันความล้มเหลววิสก

เมื่อตัดสินใจเลือกจำนวนดิสก์ในแต่ละกลุ่มวอลุ่ม คุณยังต้อง วางแผนสำหรับพื้นที่ว่างเพื่อมีร์เรอร์ข้อมูลด้วย โปรดจำไว้ว่า คุณสามารถมีร์เรอร์และย้ายข้อมูล ระหว่างดิสก์ที่อยู่ในกลุ่มวอลุ่มเดียวกันเท่านั้น หากไซต์ใช้ระบบไฟล์ขนาดใหญ่ การค้นหา พื้นที่ว่างดิสก์ซึ่งจะมีร์เรอร์อาจกลายเป็นปัญหา ในเวลาต่อมาได้ ระวังการสื่อความถึงการมีอยู่ของค่าติดตั้งระหว่างดิสก์ สำหรับสำเนาโลจิคัลวอลุ่ม และการจัดสรร intra-disk สำหรับ โลจิคัลวอลุ่ม

หลักการที่เกี่ยวข้อง:

“โปรเซสแปรผัน” ในหน้า 384

โปรเซสแปรผันเป็นกลไกหนึ่งที่ LVM ใช้ เพื่อให้มั่นใจว่ากลุ่มวอลุ่มพร้อมใช้งานและมีข้อมูล ล่าสุด

“การแปลงกลุ่มวอลุ่มเป็นสถานะไม่ใช่อุปกรณ์ประกอบ” ในหน้า 387

คุณสามารถเปลี่ยนกลุ่มวอลุ่มเป็นสถานะไม่ใช่อุปกรณ์ประกอบเพื่อให้ข้อมูล มีอยู่อย่างต่อเนื่องแม้ว่าไม่มีอุปกรณ์ประกอบ

“คำติดตั้งดิสก์อินเทอร์สำหรับสำเนาโลจิคัลวอลุ่ม” ในหน้า 447

การจัดสรรสำเนาเดี่ยวของโลจิคัลวอลุ่มบนดิสก์คอนข้าง ตรงไปตรงมา

“นโยบายการจัดสรร intra-disk สำหรับแต่ละโลจิคัลวอลุ่ม” ในหน้า 448

ตัวเลือกนโยบายการจัดสรร intra-disk ขึ้นอยู่กับพื้นที่ทำสวน ของดิสก์ซึ่งสามารถระบุตำแหน่งพาร์ติชันฟิสิคัลได้

การมีอยู่สูงในกรณีของความล้มเหลวอะแด็ปเตอร์หรือแหล่งจ่ายไฟ:

เมื่อต้องการป้องกันความล้มเหลวอะแด็ปเตอร์หรือแหล่งจ่ายไฟ ขึ้นอยู่กับ ความต้องการของคุณ ให้ทำหนึ่งหรือหลายอย่างต่อไปนี้

- ใช้สองอะแด็ปเตอร์ที่ตั้งอยู่ในแชสซีเดียวกันหรือต่างกัน การระบุตำแหน่ง อะแด็ปเตอร์ในแชสซีต่างกันป้องกันไม่ให้สูญเสียอะแด็ปเตอร์ทั้งสองตัว ถ้าเกิด ความล้มเหลวแหล่งจ่ายไฟในแชสซีหนึ่ง
- ใช้อะแด็ปเตอร์สองตัว โดยต่อพ่วงอย่างน้อยหนึ่งดิสก์เข้ากับแต่ละอะแด็ปเตอร์ ลักษณะนี้ป้องกัน ความล้มเหลวที่อะแด็ปเตอร์ตัวใดตัวหนึ่ง (หรือแหล่งจ่ายไฟถ้าอะแด็ปเตอร์อยู่ในตู้ แยกต่างหาก) โดยยังคงรักษาองค์ประกอบไว้ในกลุ่มวอลุ่ม และสมมติ การมีเรอร์ไชว (สำเนา ของพาร์ติชันโลจิคัลไม่สามารถแบ่งใช้ฟิสิคัลวอลุ่มเดียวกัน) ระหว่าง โลจิคัลวอลุ่มบนดิสก์ A (อะแด็ปเตอร์ A) และโลจิคัลวอลุ่มบนดิสก์ B (อะแด็ปเตอร์ B) นี้หมายความว่าคุณคัดลอกโลจิคัลวอลุ่มที่ตั้งอยู่บนดิสก์ ซึ่งต่อพ่วงกับอะแด็ปเตอร์ A ไปยังดิสก์ที่ตั้งอยู่บนอะแด็ปเตอร์ B และคุณยัง คัดลอกโลจิคัลวอลุ่มที่ตั้งอยู่บนดิสก์ซึ่งต่อพ่วงกับอะแด็ปเตอร์ B ไปยังดิสก์ที่ตั้งอยู่บนอะแด็ปเตอร์ A ด้วย
- กำหนดคอนฟิกร์ดิสก์ทั้งหมดจากอะแด็ปเตอร์ทั้งสองตัวเข้าในกลุ่มวอลุ่มเดียวกัน ซึ่งทำให้แน่ใจว่า ยังคงมีสำเนาโลจิคัลวอลุ่มอย่างน้อยหนึ่งสำเนาสามารถใช้ได้ในกรณีที่อะแด็ปเตอร์ ล้มเหลว หรือถ้าตู้แยกต่างหาก ในกรณีที่แหล่งจ่ายไฟล้มเหลว
- ทำให้กลุ่มวอลุ่มเป็นกลุ่มวอลุ่มที่ไม่ใช่อุปกรณ์ประกอบ การทำเช่นนี้อ่อนุญาตให้กลุ่มวอลุ่ม ยังคงใช้งานอยู่ตราบเท่าที่หนึ่ง Volume Group Descriptor Area (VGDA) สามารถเข้าถึงได้บนดิสก์ใดๆ ในกลุ่มวอลุ่ม
- หากมีสองดิสก์ในกลุ่มวอลุ่ม ให้ใช้การมีเรอร์ไชว ระหว่างอะแด็ปเตอร์ หากมีดิสก์มากกว่าหนึ่งตัวบนแต่ละอะแด็ปเตอร์ ให้ใช้การมีเรอร์ไชวดับเบิล ในกรณีนั้น คุณสร้างสำเนาที่มีเรอร์ไชวบนดิสก์ ที่ใช้อะแด็ปเตอร์เดียวกันและสำเนาหนึ่งบนดิสก์ที่ใช้อะแด็ปเตอร์อื่น

หลักการที่เกี่ยวข้อง:

“การแปลงกลุ่มวอลุ่มเป็นสถานะไม่ใช่อุปกรณ์ประกอบ” ในหน้า 387

คุณสามารถเปลี่ยนกลุ่มวอลุ่มเป็นสถานะไม่ใช่อุปกรณ์ประกอบเพื่อให้ข้อมูล มีอยู่อย่างต่อเนื่องแม้ว่าไม่มีอุปกรณ์ประกอบ

กลยุทธ์โลจิคัลวอลุ่ม

นโยบายที่อธิบายที่นี่ช่วยให้คุณกำหนดกลยุทธ์สำหรับการใช้งาน โลจิคัลวอลุ่มซึ่งมุ่งเน้นการผสมผสานระหว่างการมีอยู่ ประสิทธิภาพการทำงาน และต้นทุนที่เหมาะสมสำหรับไซต์ของคุณ

การมีอยู่ คือความสามารถในการเข้าถึงข้อมูลแม้ว่าดิสก์ที่เชื่อมโยง ล้มเหลวหรือไม่สามารถเข้าถึงได้ ข้อมูลอาจยังสามารถเข้าถึงได้ผ่านทาง สำเนาของข้อมูลที่จัดทำและเก็บรักษาไว้บนดิสก์แยกต่างหากและอะแด็ปเตอร์ ในระหว่างการดำเนินงานระบบปกติ เทคนิค เช่น การมีเรอร์ไชว และการใช้ ดิสก์ฮอตสแปร์สามารถช่วยให้อุ่นใจถึงการมีอยู่ของข้อมูล

ประสิทธิภาพ คือความเร็วเฉลี่ยในการเข้าถึงข้อมูล นโยบาย เช่น การตรวจสอบการเขียนและการมีเรอร์ไชวช่วยเพิ่มการมีอยู่ แต่เพิ่มโหลดการ ประมวลผลของระบบ และส่งผลให้ประสิทธิภาพด้อยลง การมีเรอร์ไชวเพิ่มขนาดของโลจิคัลวอลุ่ม ขึ้นเป็นสองเท่าหรือสามเท่า โดยทั่วไป การเพิ่มการมีอยู่จะทำให้ประสิทธิภาพ ด้อยลง การกวาดดิสก์ช่วยปรับปรุงประสิทธิภาพ การก

วาดดิสก์อนุญาตให้ใช้กับการทำ mirror คุณสามารถตรวจสอบและแก้ไข ปัญหา hot-spot ที่เกิดขึ้นเมื่อพาร์ติชันโลจิคัลบางพาร์ติชันบนดิสก์ของคุณ มี I/O ดิสก์มากจนส่งผลกระทบต่อประสิทธิภาพของระบบ อย่างเห็นได้ชัด

โดยการควบคุมการจัดสรรข้อมูลบนดิสก์และระหว่างดิสก์ คุณสามารถ ปรับระบบหน่วยเก็บให้มีประสิทธิภาพการทำงานสูงสุดเท่าที่เป็นไปได้ โปรดดูที่ *Performance management* สำหรับข้อมูลรายละเอียด เกี่ยวกับวิธีการเพิ่มประสิทธิภาพระบบหน่วยเก็บให้สูงสุด

ใช้ส่วนต่อไปนี้เป็นเพื่อประเมินการแลกเปลี่ยนระหว่างประสิทธิภาพ การมีอยู่ และต้นทุน โปรดจำไว้ว่าการมีอยู่ที่สูงขึ้นมักลดประสิทธิภาพ และในทางกลับกัน การมีเรอร์อาจเพิ่มประสิทธิภาพ อย่างไรก็ตาม LVM เลือกสำเนาบนดิสก์ที่ยุ่่น้อยที่สุดสำหรับการอ่าน

หมายเหตุ: การมีเรอร์ไม่ได้ป้องกันการสูญเสียไฟล์แต่ละไฟล์ ที่ถูกลบออกโดยไม่ได้ตั้งใจ หรือการสูญหายเนื่องจากปัญหาซอฟต์แวร์ ไฟล์เหล่านี้สามารถเรียกคืนได้จากสำเนาสำรองบนเทปหรือดิสก์แบบเดิมเท่านั้น

หลักการที่เกี่ยวข้อง:

“กลยุทธ์กลุ่มวอลุ่ม” ในหน้า 439

ความล้มเหลวดิสก์เป็นความล้มเหลวฮาร์ดแวร์ที่พบบ่อยที่สุดในระบบ หน่วยเก็บ ตามด้วยความล้มเหลวของอะแดปเตอร์และแหล่งจ่ายไฟ การป้องกันความล้มเหลว ดิสก์โดยหลักแล้ว เกี่ยวข้องกับคอนฟิกรูเรชันของโลจิคัลวอลุ่ม

ข้อกำหนดสำหรับการมีเรอร์และการกวาด:

กำหนดว่าข้อมูลที่จัดเก็บไว้ในโลจิคัลวอลุ่มมีคุณค่ามากเพียงพอ ที่จะคุ้มกับต้นทุนในการประมวลผลและพื้นที่ว่างดิสก์ของการมีเรอร์หรือไม่ หากคุณมีระบบไฟล์ที่เข้าถึงตามลำดับขนาดใหญ่ซึ่งประสิทธิภาพอาจได้รับผลกระทบ ได้ง่าย คุณอาจต้องพิจารณาการกวาดดิสก์

ประสิทธิภาพและการมีเรอร์ไม่ได้ตรงข้ามกันเสมอไป ถ้าอินสแตนซ์ (สำเนา) ที่แตกต่างกัน ของพาร์ติชันโลจิคัลอยู่บนฟิสิคัลวอลุ่มที่แตกต่างกัน และควรต่อพ่วงกับ อะแดปเตอร์ที่แตกต่างกัน LVM สามารถปรับปรุงประสิทธิภาพการอ่านได้โดย การอ่านสำเนาบนดิสก์ที่ยุ่่น้อยที่สุด ประสิทธิภาพการเขียนมีต้นทุนเหมือนกันเสมอเนื่องจากคุณ ต้องอัปเดตสำเนาทั้งหมด ยกเว้นว่าต่อพ่วงดิสก์เข้าถึงอะแดปเตอร์ที่แตกต่างกัน จำเป็นต้องอ่านเพียงหนึ่งสำเนาเท่านั้นสำหรับการดำเนินงานอ่าน

AIX LVM สนับสนุนอ็อปชัน RAID ต่อไปนี้:

ตารางที่ 63. การสนับสนุน Logical Volume Manager สำหรับ RAID

ไอเท็ม	คำอธิบาย
RAID 0	การกวาด
RAID 1	การมีเรอร์
RAID 10 หรือ 0+1	การมีเรอร์และการกวาด

ในขณะที่การมีเรอร์ช่วยปรับปรุงการมีอยู่ของระบบหน่วยเก็บ แต่ไม่ได้มีไว้ เพื่อใช้ทดแทนการจัดเตรียมสำเนาสำรองบนเทปแบบที่ใช้กันมา

คุณสามารถมีเรอร์ rootvg แต่ถ้าคุณทำให้สร้างดัมพ์โลจิคัลวอลุ่มแยก ต่างหาก การดัมพ์ไปยังโลจิคัลวอลุ่มที่มีเรอร์อาจส่งผลให้ดัมพ์ไม่สอดคล้อง กัน นอกจากนี้ เนื่องจากอุปกรณ์ดัมพ์ดีฟอลต์เป็นโลจิคัลวอลุ่มการเลื่อนหน้าหลัก ให้สร้างดัมพ์โลจิคัลวอลุ่มแยกต่างหากถ้าคุณมีเรอร์โลจิคัลวอลุ่มการเลื่อนหน้า ของคุณ

โดยปกติแล้ว เมื่อใดก็ตามที่อัปเดตข้อมูลบนพาร์ติชันโลจิคัล พาร์ติชันฟิสิคัล ทั้งหมดที่มีพาร์ติชันโลจิคัลนั้นจะมีการอัปเดตให้โดยอัตโนมัติ อย่างไรก็ตาม พาร์ติชันฟิสิคัลอาจ *เก่า* (ไม่มีข้อมูลล่าสุดอีกต่อไป) เนื่องจากความบกพร่องของระบบหรือเนื่องจากฟิสิคัลวอลุ่มไม่พร้อมใช้งาน ในเวลาอัปเดต LVM สามารถรีเฟรชพาร์ติชันเก่า เป็นสภาพที่สอดคล้องกันโดยการคัดลอกข้อมูลปัจจุบันจากพาร์ติชันฟิสิคัลที่ทันสมัย ไปยังพาร์ติชันเก่า โพรเซสนี้เรียกว่า *การซิงโครไนซ์มีร์เรอร์* รีเฟรชสามารถเกิดขึ้นเมื่อรีสตาร์ทระบบ เมื่อฟิสิคัลวอลุ่มกลับมาออนไลน์ หรือเมื่อคุณออกใช้คำสั่ง `syncvg`

การเปลี่ยนแปลงใดๆ ที่กระทบต่อส่วนประกอบพาร์ติชันฟิสิคัลของบูตโลจิคัล วอลุ่ม กำหนดให้คุณต้องรันคำสั่ง `bosboot` หลังจากการเปลี่ยนแปลง นั้น ซึ่งหมายความว่าดำเนินการ เช่น การเปลี่ยนการมีร์เรอร์ของบูต โลจิคัลวอลุ่มต้องการ `bosboot`

นโยบายการกำหนดตารางเวลาสำหรับการเขียนที่มีร์เรอร์ในดิสก์:

สำหรับข้อมูลที่มีสำเนาทางกายภาพเพียงชุดเดียว ไดรเวอร์อุปกรณ์โลจิคัล วอลุ่ม (LVDD) จะแปลแอตเตรสคำร้องขอการอ่านหรือเขียนแบบโลจิคัลเป็นฟิสิคัล แอตเตรส และเรียกไดรเวอร์อุปกรณ์ฟิสิคัลที่เหมาะสมเพื่อให้บริการคำร้องขอนั้น นโยบายสำเนาเดียวหรือที่ไม่มีมีร์เรอร์นี้จัดการกับการเปลี่ยนตำแหน่งของบล็อกที่ไม่ดี สำหรับคำร้องขอการเขียน และส่งคืนข้อผิดพลาดการอ่านทั้งหมดไปยังโปรเซสที่เรียกใช้

หากคุณใช้โลจิคัลวอลุ่มที่มีร์เรอร์ สามารถตั้งค่านโยบายการกำหนดตารางเวลาต่อไป นี้สำหรับการเขียนลงในดิสก์สำหรับโลจิคัลวอลุ่มที่มีสำเนาหลายชุด:

นโยบายการกำหนดตารางเวลาตามลำดับ

ทำการเขียนในหลายสำเนาหรือมีร์เรอร์ตามลำดับ พาร์ติชันฟิสิคัลหลายรายการที่ แสดงถึงสำเนาที่มีร์เรอร์ของพาร์ติชันโลจิคัลเดียว มีการแบ่งออกเป็นระดับหลัก รอง และระดับที่สาม ในการจัดตารางเวลาตามลำดับ พาร์ติชันฟิสิคัล ถูกเขียนเป็นลำดับ ระบบรอให้การดำเนินงานเขียน ในพาร์ติชันฟิสิคัลหนึ่งเสร็จสมบูรณ์ก่อนเริ่มต้นการดำเนินงานเขียนในพาร์ติชันฟิสิคัลถัดไป เมื่อการดำเนินงานเขียนทั้งหมดสำหรับมีร์เรอร์ทั้งหมด เสร็จสมบูรณ์แล้ว ถือว่าการดำเนินงานเขียนเสร็จสมบูรณ์

นโยบายการกำหนดตารางเวลาแบบขนาน

เริ่มต้นการดำเนินงานเขียนสำหรับพาร์ติชันฟิสิคัลทั้งหมดในพาร์ติชันโลจิคัล พร้อมกัน เมื่อการดำเนินงานเขียนลงในพาร์ติชันฟิสิคัล ที่ใช้เวลานานที่สุดเสร็จสมบูรณ์แล้ว ถือว่าการดำเนินงานเขียนเสร็จสมบูรณ์ การระบุโลจิคัลวอลุ่มที่มีร์เรอร์พร้อมกันนโยบายการกำหนดตารางเวลาแบบขนานอาจ ปรับปรุงประสิทธิภาพการดำเนินงานอ่าน I/O เนื่องจากสำเนาหลายชุดช่วยให้ ระบบสามารถกำหนดทิศทางการดำเนินงานอ่านไปยังดิสก์ที่ยุ่งน้อยที่สุดของโลจิคัลวอลุ่มนี้ได้

การเขียนแบบขนานพร้อมกันนโยบายการกำหนดตารางเวลาการอ่านตามลำดับ

เริ่มต้นการดำเนินงานเขียนสำหรับพาร์ติชันฟิสิคัลทั้งหมดในพาร์ติชันโลจิคัล พร้อมกัน สำเนาหลักของการอ่านจะถูกอ่านก่อนเสมอ หากการดำเนินงานอ่านนั้นไม่สำเร็จ จะมีการอ่านสำเนาชุดถัดไป ในระหว่างการดำเนินงาน ลองอ่านซ้ำบนสำเนาถัดไป สำเนาหลักที่ล้มเหลวจะได้รับการแก้ไข โดย LVM ด้วยวิธีการเปลี่ยนตำแหน่งฮาร์ดแวร์ ซึ่งจะซ่อมแซมบล็อกที่ไม่ดีเพื่อให้สามารถเข้าถึงได้ในอนาคต

การเขียนแบบขนานพร้อมกันนโยบายการกำหนดตารางเวลาการอ่านแบบวนรอบ

เริ่มต้นการดำเนินงานเขียนสำหรับพาร์ติชันฟิสิคัลทั้งหมดในพาร์ติชันโลจิคัล พร้อมกัน การอ่านจะสลับไปมาระหว่างสำเนาที่มีร์เรอร์

นโยบายบล็อกที่ไม่ดี

บ่งชี้ว่ากลุ่มวอลุ่มมีการเปิดใช้งานสำหรับการเปลี่ยนตำแหน่งบล็อกที่ไม่ดีหรือไม่ ค่าดีฟอลต์คือ `ใช่` เมื่อมีการตั้งค่าเป็น `yes` สำหรับกลุ่มวอลุ่ม สามารถเปลี่ยนตำแหน่งบล็อกที่ไม่ดีได้ เมื่อมีการตั้งค่าเป็น `no` นโยบายจะยกเลิกค่าติด

ตั้ง โลจิคัลวอลุ่ม เมื่อค่าถูกเปลี่ยน โลจิคัลวอลุ่มทั้งหมดยังคงใช้ค่าที่ตั้ง ก่อนหน้านี้ต่อไป ค่าบางตัวต้องกำหนดทิศทาง I/O ที่ร้องขอ ไปยังบล็อกที่เปลี่ยนตำแหน่งหรือไม่ เมื่อมีการตั้งค่า เป็น yes กลุ่มวอลุ่มจะอนุญาตให้เปลี่ยนตำแหน่งบล็อกที่ไม่ดีได้ เมื่อมีการตั้งค่าเป็น no การจัดสรรบล็อกที่ไม่ดีจะไม่สมบูรณ์ LVM ทำการเปลี่ยนตำแหน่งซอฟต์แวร์ เฉพาะถ้าการเปลี่ยนตำแหน่งฮาร์ดแวร์ล้มเหลว มิฉะนั้น แฟล็กการเปลี่ยนตำแหน่งบล็อกที่ไม่ดี LVM (BBR) จะไม่มีผล

หมายเหตุ: การเปลี่ยนตำแหน่งบล็อกที่ไม่ดีถูกปิดใช้งาน ยกเว้นว่า ค่าที่ตั้งนโยบายบล็อกที่ไม่ดีสำหรับกลุ่มวอลุ่ม และโลจิคัลวอลุ่มมีการตั้งค่าเป็น yes ทั้งคู่

นโยบายความต้อกันในการเขียนสำเนาสำหรับโลจิคัลวอลุ่ม:

เมื่อเปิดความต้อกันในการเขียนสำเนา (Mirror Write Consistency หรือ MWC) จะมีการระบุพาร์ติชันโลจิคัลที่อาจไม่สอดคล้องกันถ้าระบบหรือ กลุ่มวอลุ่มไม่ได้เปิดอย่างถูกต้อง เมื่อกลุ่มวอลุ่มถูกเปลี่ยนกลับมาออนไลน์ ระบบจะใช้ข้อมูลนี้ เพื่อให้พาร์ติชันโลจิคัลสอดคล้องกัน ลักษณะนี้เรียกว่า เป็น *active MWC*

เมื่อโลจิคัลวอลุ่มใช้ active MWC อยู่ คำร้องขอสำหรับโลจิคัลวอลุ่มนี้ จะถูกพักไว้ภายในเลย์เออร์การจัดตารางเวลาจนกว่าสามารถอัปเดต MWC แคช บล็อกบนฟิสิคัลวอลุ่มเป้าหมายได้ เมื่ออัปเดต MWC แคชบล็อกแล้ว คำร้องขอจะดำเนินการเขียนข้อมูลฟิสิคัลต่อไป เฉพาะดิสก์ที่ข้อมูลตั้งอยู่จริงเท่านั้นต้องถูกเขียน MWC แคชบล็อกเหล่านี้ ก่อนสามารถทำการเขียนต่อไปได้

เมื่อใช้ active MWC อยู่ ประสิทธิภาพของระบบอาจได้รับผลกระทบ ผลกระทบอาจเกิดขึ้นจากส่วนเพิ่มของการบันทึกหรือการเจอร์นัลคำร้องขอการเขียน ซึ่ง Logical Track Group (LTG) ใช้งานอยู่ขนาด LTG ที่ใช้ได้ สำหรับกลุ่มวอลุ่มคือ 128 K, 256 K, 512 K, 1024 K, 2 MB, 4 MB, 8 MB, และ 16 MB

หมายเหตุ: เพื่อให้ขนาด LTG มากกว่า 128 K ดิสก์ที่มีอยู่ใน กลุ่มวอลุ่มต้องสนับสนุนคำร้องขอ I/O ขนาดนี้จากรูทีนกลยุทธ์ของ ดิสก์ LTG เป็นบล็อกต้อกันที่มีอยู่ภายในโลจิคัลวอลุ่ม และปรับแนวตามขนาดของ LTG การดำเนินการเพิ่มเติมนี้ใช้สำหรับ การเขียนที่มีร์เรอร์เท่านั้น

จำเป็นต้องรับประกันความสอดคล้องของข้อมูลระหว่างมีร์เรอร์เฉพาะถ้า ระบบหรือกลุ่มวอลุ่มเสียหายก่อนการเขียนลงในมีร์เรอร์ทั้งหมดเสร็จสมบูรณ์ โลจิคัลวอลุ่มทั้งหมดในกลุ่มวอลุ่มแบ่งใช้บันทึก MWC บันทึก MWC มีการเก็บรักษาไว้บนขอบด้านนอกของแต่ละดิสก์ ควรวางโลจิคัลวอลุ่มที่ใช้ Active MWC ไว้ที่ขอบด้านนอกของดิสก์เพื่อให้โลจิคัลวอลุ่มอยู่ใกล้กับบันทึก MWC บนดิสก์

เมื่อ MWC มีการตั้งค่าเป็น passive กลุ่มวอลุ่มจะบันทึกว่าโลจิคัลวอลุ่ม ถูกเปิดแล้ว หลังจากความเสียหายเมื่อกลุ่มวอลุ่มถูก varied on การบังคับใช้ซึ่งก้อตโนมิติของโลจิคัลวอลุ่มจะเริ่มต้นขึ้น ความสอดคล้องกันมีการรักษาไว้ในขณะที่ การบังคับใช้ซึ่งก้อตโนมิติอยู่โดยใช้สำเนาของนโยบายการกู้คืนการอ่าน ซึ่งจะกระจายบล็อกที่กำลังอ่านไปยังมีร์เรอร์อื่นในโลจิคัล วอลุ่ม นโยบายนี้ได้รับการสนับสนุนบนกลุ่มวอลุ่มชนิดขนาดใหญ่เท่านั้น

เมื่อปิด MWC มีร์เรอร์ของโลจิคัลวอลุ่มที่มีร์เรอร์อาจถูกปล่อยไว้ใน สภาพไม่สอดคล้องกันในกรณีที่ระบบหรือกลุ่มวอลุ่มเสียหาย ไม่มีการปกป้องความสอดคล้องของมีร์เรอร์โดยอ้อมมิติ การเขียนที่ค้างอยู่ในเวลาที่เกิดความเสียหายอาจทำให้มีร์เรอร์ซึ่งมีข้อมูลไม่สอดคล้องกันอยู่ใน ครั้งถัดไปที่กลุ่มวอลุ่มถูก varied on หลังจากความเสียหาย โลจิคัลวอลุ่มที่มีร์เรอร์ใดๆ ที่ปิด MWC ควรทำการบังคับใช้ซึ่งก้อตโนมิติก่อนจะใช้ข้อมูลภายใน โลจิคัลวอลุ่ม ตัวอย่างเช่น

```
syncvg -f -l LTVname
```

ข้อยกเว้นในการบังคับใช้ซึ่งก็คือ โลจิคัลวอลุ่มที่มีเนื้อหาซึ่งถูกต้อง ในขณะที่โลจิคัลวอลุ่มเปิดอยู่เท่านั้น เช่น พื้นที่การเลื่อนหน้า

โลจิคัลวอลุ่มที่มีรีเรอร์เหมือนกับโลจิคัลวอลุ่มที่ไม่มีรีเรอร์ซึ่งต้องดำเนินการเขียน เมื่อ LVM จัดการกับคำร้องขอการเขียนเสร็จแล้ว แสดงว่าข้อมูล ถูกเขียนลงในไดรฟ์ทั้งหมดที่ต่ำกว่า LVM แล้ว ผลของการเขียนจะยังไม่ทราบ จนกว่า LVM ออกใช้ **iodone** ในการเขียน หลังจากการดำเนินงานนี้เสร็จสมบูรณ์แล้ว ไม่จำเป็นต้องกู้คืนหลังจากความเสียหาย บล็อกใดๆ ที่ยังเขียนไม่เสร็จ (**iodone**) เมื่อเครื่องเสียหายควรจะถูกตรวจสอบและเขียนใหม่ โดยไม่คำนึงถึง ค่าติดตั้ง MWC หรือมีการมีรีเรอร์หรือไม่

เนื่องจากโลจิคัลวอลุ่มที่มีรีเรอร์เหมือนกับโลจิคัลวอลุ่มที่ไม่มีรีเรอร์ ดังนั้นจึงไม่มีสิ่งทีเรียกว่าเป็นข้อมูลล่าสุด แอ็พพลิเคชั่นทั้งหมดที่ให้ความสำคัญเกี่ยวกับ ความถูกต้องของข้อมูลจำเป็นต้องพิจารณาความถูกต้องของข้อมูลที่ค้างอยู่ หรือการเขียนระหว่างดำเนินการที่ยังไม่เสร็จก่อนกลุ่มวอลุ่มหรือระบบ เสียหาย ไม่ว่าโลจิคัลวอลุ่มมีการมีรีเรอร์หรือไม่ก็ตาม

Active และ passive MWC ทำให้มีรีเรอร์สอดคล้องกันเฉพาะถ้ากลุ่มวอลุ่ม ถูกนำกลับมาออนไลน์หลังจากความเสียหาย โดยการเลือกหนึ่งมีรีเรอร์และกระจาย ข้อมูลนั้นไปยังมีรีเรอร์อื่น นโยบาย MWC เหล่านี้ไม่ได้เก็บประวัติของ ข้อมูลล่าสุด Active MWC เก็บประวัติของ LTGs ที่ถูกเขียนอยู่ในปัจจุบันเท่านั้น ดังนั้น MWC จึงไม่ได้รับประกันว่าข้อมูลล่าสุดจะมีการกระจายไปยัง มีรีเรอร์ทั้งหมด Passive MWC ทำให้มีรีเรอร์สอดคล้องกันโดยเข้าไปในโหมด กระจายเมื่ออ่าน หลังจากความเสียหาย นี้ เป็นแอ็พพลิเคชั่นข้างบน LVM ที่ต้องกำหนด ความถูกต้องของข้อมูลหลังจากความเสียหาย จากมุมมอง LVM ถ้าแอ็พพลิเคชั่นออกใช้การเขียนที่ค้างอยู่ ทั้งหมดอีกครั้งเสมอหลังจากที่เกิดความเสียหาย มีรีเรอร์ที่อาจไม่สอดคล้องกันจะ สอดคล้องกันเมื่อการเขียนเหล่านี้เสร็จสิ้น (ตราบที่เขียนลงในบล็อกเดียวกันกับที่เขียนค้างอยู่เมื่อ เกิดความเสียหาย)

หมายเหตุ: โลจิคัลวอลุ่มที่มีรีเรอร์ซึ่งมีบันทึก JFS หรือระบบไฟล์ ต้องถูกซิงโครไนซ์หลังจากความเสียหาย โดยซิงก์ที่บังคับใช้ก่อนการใช้งาน หรือ โดยการเปิด MWC หรือโดยการเปิด passive MWC

นโยบายการจัดสรรของดิสก์อินเตอร์:

นโยบายการจัดสรรของดิสก์อินเตอร์ระบุจำนวนดิสก์ ซึ่งพาร์ติชันฟิสิคัลของโลจิคัลวอลุ่มตั้งอยู่

พาร์ติชันฟิสิคัลของโลจิคัลวอลุ่มอาจตั้งอยู่บนดิสก์เดียว หรือกระจายบนดิสก์ทั้งหมดในกลุ่มวอลุ่ม อีอพชันต่อไปนี้ใช้ พร้อมกับคำสั่ง **mklv** และ **chlv** เพื่อกำหนดนโยบายดิสก์อินเตอร์:

- อีอพชัน **Range** กำหนดจำนวนดิสก์ที่ใช้สำหรับ สำเนาฟิสิคัลหนึ่งของโลจิคัลวอลุ่ม
- อีอพชัน **Strict** กำหนดว่าการดำเนินงาน **mklv** สำเร็จ หรือไม่ถ้าสำเนาตั้งแต่สองชุดขึ้นไปต้องใช้ฟิสิคัลวอลุ่มเดียวกัน
- อีอพชัน **Super Strict** ระบุว่าพาร์ติชันที่จัดสรรสำหรับมีรีเรอร์หนึ่ง ไม่สามารถแบ่งใช้ฟิสิคัลวอลุ่มกับพาร์ติชันจากมีรีเรอร์อื่น
- **Striped** โลจิคัลวอลุ่มสามารถมีนโยบายดิสก์อินเตอร์ช่วงสูงสุดและ super strict เท่านั้น

หลักการที่เกี่ยวข้อง:

“การเปลี่ยนที่ตั้งและการลดพื้นที่การเลื่อนหน้า hd6” ในหน้า 459

คุณอาจต้องการลดหรือย้ายพื้นที่การเลื่อนหน้าดีฟอลต์เพื่อ เพิ่มประสิทธิภาพระบบหน่วยเก็บ โดยการบังคับใช้การเลื่อนหน้า และการสลับไปยัง ดิสก์อื่นในระบบที่ยิ่งน้อยกว่า การลดหรือการย้ายการเลื่อนหน้าดีฟอลต์ ยังช่วยประหยัดพื้นที่ว่างดิสก์บน **hdisk0** ด้วย

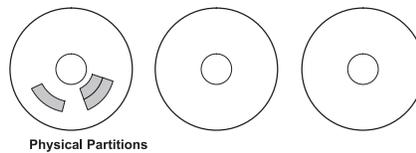
ค่าติดตั้งดิสก์อินเทอร์สำหรับสำเนาเดียวของโลจิคัลวอลุ่ม:

หากคุณเลือกค่าติดตั้งดิสก์อินเทอร์ต่ำสุด (Range = minimum) พาร์ติชันฟิสิคัลที่กำหนดให้กับโลจิคัลวอลุ่มจะอยู่บนดิสก์เดียว เพื่อเพิ่มประสิทธิภาพการมีอยู่ หากคุณเลือกค่าติดตั้งดิสก์อินเทอร์สูงสุด (Range = maximum) พาร์ติชันฟิสิคัลจะอยู่บนหลายดิสก์ เพื่อเพิ่มประสิทธิภาพการทำงาน

สำหรับโลจิคัลวอลุ่มที่ไม่มีเรอร์ให้ใช้ค่าติดตั้งต่ำสุด เพื่อให้ได้การมีอยู่สูงสุด (การเข้าถึงข้อมูลในกรณีที่ยาร์ดแวร์ล้มเหลว) ค่าติดตั้งต่ำสุดบ่งชี้ว่าหนึ่งฟิสิคัลวอลุ่มมีพาร์ติชันฟิสิคัลตั้งเดิมทั้งหมด ของโลจิคัลวอลุ่มนี้ ถ้าเป็นไปได้ หากโปรแกรมการจัดสรร ต้องใช้ฟิสิคัลวอลุ่มตั้งแต่สองตัวขึ้นไป โปรแกรมจะใช้จำนวนต่ำสุด ซึ่งยังคงสอดคล้องกับพารามิเตอร์อื่น

โดยใช้จำนวนต่ำสุดของฟิสิคัลวอลุ่ม คุณสามารถลดความเสี่ยงของ การสูญเสียข้อมูลเนื่องจากความล้มเหลวของดิสก์ ฟิสิคัลวอลุ่มเพิ่มเติมแต่ละรายการที่ใช้ สำหรับสำเนาฟิสิคัลหนึ่งเพิ่มความเสี่ยงนั้น โลจิคัลวอลุ่มที่ไม่มีเรอร์ ซึ่งกระจายบนสี่ฟิสิคัลวอลุ่มอาจมีโอกาสสูญเสียข้อมูลเนื่องจากความล้มเหลวของ ฟิสิคัลวอลุ่มหนึ่งมากถึงสี่เท่า เมื่อเปรียบเทียบกับโลจิคัลวอลุ่มที่มีบนฟิสิคัลวอลุ่มเดียว

รูปภาพต่อไปนี้แสดงนโยบายการจัดสรรของดิสก์อินเทอร์ต่ำสุด

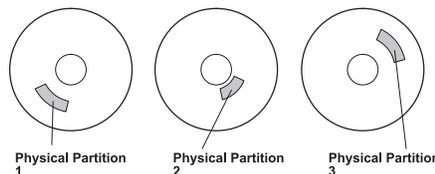


รูปที่ 3. นโยบายการจัดสรรของดิสก์อินเทอร์ต่ำสุด

ภาพสาธิตนี้แสดงสามดิสก์ ดิสก์หนึ่งมีสามพาร์ติชันฟิสิคัล ดิสก์อื่นไม่มีพาร์ติชันฟิสิคัล

ค่าติดตั้งสูงสุดกระจายพาร์ติชันฟิสิคัลของโลจิคัลวอลุ่มอย่างเท่าเทียมที่สุด เท่าที่เป็นไปได้บนฟิสิคัลวอลุ่มจำนวนมากที่สุดเท่าที่เป็นไปได้ โดยพิจารณา ถึงข้อจำกัดอื่น นี่เป็นอีกขั้นที่มุ่งเน้นประสิทธิภาพ เนื่องจากการกระจาย พาร์ติชันฟิสิคัลบนหลายดิสก์ช่วยลดเวลาการเข้าถึงเฉลี่ย ของโลจิคัลวอลุ่ม เพื่อปรับปรุงการมีอยู่ ค่าติดตั้งสูงสุดจึง ใช้กับโลจิคัลวอลุ่มที่มีเรอร์เท่านั้น

รูปภาพต่อไปนี้แสดงนโยบายการจัดสรรของดิสก์อินเทอร์สูงสุด



รูปที่ 4. นโยบายการจัดสรรของดิสก์อินเทอร์สูงสุด

ภาพสาธิตนี้แสดงสามดิสก์ แต่ละดิสก์มีหนึ่งพาร์ติชันฟิสิคัล

นิยามเหล่านี้ยังใช้ได้เมื่อขยายหรือตัดออกโลจิคัลวอลุ่ม ที่มีอยู่ การจัดสรรของพาร์ติชันฟิสิคัลใหม่มีการกำหนดโดย นโยบายการจัดสรรปัจจุบันของคุณและในตำแหน่งที่มีพาร์ติชันฟิสิคัล ที่ใช้

หลักการที่เกี่ยวข้อง:

“ค่าติดตั้งดิสก์อินเทอร์สำหรับสำเนาโลจิคัลวอลุ่ม”

การจัดสรรสำเนาเดียวของโลจิคัลวอลุ่มบนดิสก์คอนข้าง ตรงไปตรงมา

ค่าติดตั้งดิสก์อินเทอร์สำหรับสำเนาโลจิคัลวอลุ่ม:

การจัดสรรสำเนาเดียวของโลจิคัลวอลุ่มบนดิสก์คอนข้าง ตรงไปตรงมา

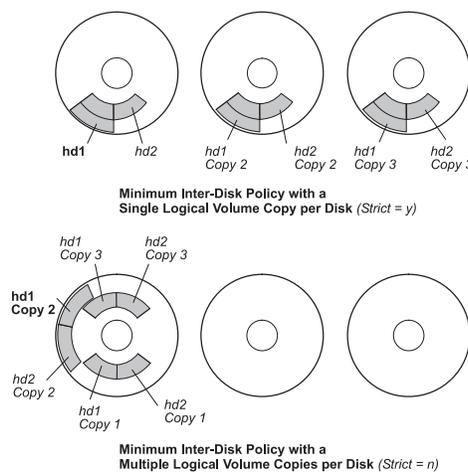
อย่างไรก็ตาม เมื่อคุณสร้างสำเนาที่มีเรอร์ การจัดสรรที่ได้มี ลักษณะซับซ้อน รูปภาพต่อมาแสดงค่าต่ำสุดและสูงสุดและค่าติดตั้งดิสก์อินเทอร์ (ช่วง) สำหรับอินสแตนซ์แรกของโลจิคัลวอลุ่ม ควบคู่ไปกับค่าติดตั้ง Strict ที่มีอยู่สำหรับสำเนาโลจิคัลวอลุ่มที่มีเรอร์

ตัวอย่างเช่น ถ้ามีสำเนาที่มีเรอร์ของโลจิคัลวอลุ่ม ค่าติดตั้งต่ำสุด ส่งผลให้พาร์ติชันฟิลิคัลที่มีอินสแตนซ์แรกของ โลจิคัลวอลุ่มถูกจัดสรรบนฟิลิคัลวอลุ่มเดียว ถ้าเป็นไปได้ จากนั้น ขึ้นอยู่กับค่าติดตั้งของอ็อปชัน Strict สำเนาเพิ่มเติม จะถูกจัดสรรบนฟิลิคัลวอลุ่มเดียวกันหรือที่แยกต่างหาก อธิบายอีกอย่างคือ ขั้นตอนวิธีจะใช้จำนวนต่ำสุดของฟิลิคัลวอลุ่มที่เป็นไปได้ภายในข้อจำกัดที่กำหนดโดยพารามิเตอร์อื่น เช่น อ็อปชัน Strict เพื่อ จัดเก็บพาร์ติชันฟิลิคัลทั้งหมด

ค่าติดตั้ง Strict = y หมายความว่าแต่ละสำเนาของโลจิคัล พาร์ติชันถูกวางไว้บนฟิลิคัลวอลุ่มที่แตกต่างกัน ค่าติดตั้ง Strict = n หมายความว่าไม่มีการจำกัดว่าสำเนาต้องอยู่บนฟิลิคัลวอลุ่มที่ แตกต่างกัน เมื่อเปรียบเทียบกัน อ็อปชัน Super Strict จะไม่อนุญาต ให้พาร์ติชันฟิลิคัลใดๆ จากมีเรอร์หนึ่งอยู่บนดิสก์เดียวกันกับพาร์ติชันฟิลิคัล จากมีเรอร์อื่นของโลจิคัลวอลุ่มเดียวกัน

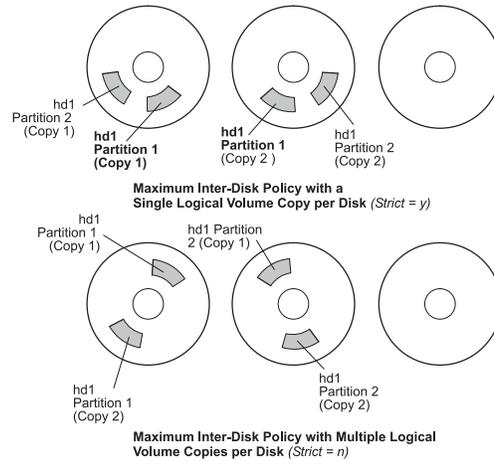
หมายเหตุ: หากมีจำนวนฟิลิคัลวอลุ่มในกลุ่มวอลุ่มน้อยกว่าจำนวน สำเนาต่อพาร์ติชันโลจิคัลซึ่งคุณเลือกไว้ให้ตั้งค่า Strict เป็น n ถ้า Strict มีการตั้งค่าเป็น y จะมีการส่งคืนข้อความแสดงข้อผิดพลาดเมื่อคุณ พยายามสร้างโลจิคัลวอลุ่ม

รูปภาพต่อไปนี้แสดงนโยบายการจัดสรรของดิสก์อินเทอร์ต่ำสุด พร้อมด้วยค่าติดตั้ง Strict ที่แตกต่างกัน:



รูปที่ 5. นโยบายดิสก์อินเทอร์ต่ำสุด/Strict. ภาพสามนี้ แสดงว่าถ้าอ็อปชัน Strict เป็น ใช่ แต่ละสำเนาของโลจิคัล พาร์ติชันจะอยู่บนฟิลิคัลวอลุ่มที่แตกต่างกัน ถ้า Strict เป็น ไม่ สำเนาทั้งหมด ของโลจิคัลวอลุ่มจะอยู่บนฟิลิคัลวอลุ่มเดียว

รูปภาพต่อไปนี้แสดงนโยบายการจัดสรรของดิสก์อินเทอร์สูงสุด พร้อมด้วยค่าติดตั้ง Strict ที่แตกต่างกัน:



รูปที่ 6. นโยบายดิสก์อินเตอร์สูงสุด/Strict. ภาพสาธิตนี้ แสดงว่าถ้าอ็อปชัน Strict เป็น ใช่ แต่ละสำเนาของ พาร์ติชันจะอยู่บนฟิสิคัลวอลุ่มที่แตกต่างกัน ถ้า Strict เป็น ไม่ สำเนาทั้งหมด จะอยู่บนฟิสิคัลวอลุ่มเดียว

หลักการที่เกี่ยวข้อง:

“การมีอยู่สูงในกรณีของความล้มเหลวดิสก์” ในหน้า 440

เมธอดหลักที่ใช้เพื่อป้องกันความล้มเหลวดิสก์เกี่ยวข้องกับ ค่าติดตั้งคอนฟิกูเรชันโลจิคัลวอลุ่ม เช่น การมีร์เรอร์

“ค่าติดตั้งดิสก์อินเตอร์สำหรับสำเนาเดี่ยวของโลจิคัลวอลุ่ม” ในหน้า 446

หากคุณเลือกค่าติดตั้งดิสก์อินเตอร์ต่ำสุด (Range = minimum) พาร์ติชันฟิสิคัลที่กำหนดให้กับโลจิคัลวอลุ่มจะอยู่บนดิสก์เดียว เพื่อเพิ่มประสิทธิภาพการมีอยู่ หากคุณเลือกค่าติดตั้งดิสก์อินเตอร์สูงสุด (Range = maximum) พาร์ติชันฟิสิคัลจะอยู่บนหลายดิสก์ เพื่อเพิ่มประสิทธิภาพการทำงาน

นโยบายการจัดสรร intra-disk สำหรับแต่ละโลจิคัลวอลุ่ม:

ตัวเลือกนโยบายการจัดสรร intra-disk ขึ้นอยู่กับพื้นที่ห้าส่วน ของดิสก์ซึ่งสามารถระบุตำแหน่งพาร์ติชันฟิสิคัลได้

ยิ่งใกล้กับพาร์ติชันฟิสิคัลที่กำหนดคือศูนย์กลางของฟิสิคัลวอลุ่ม เวลาการค้นหาเฉลี่ยยิ่งต่ำลงเนื่องจากศูนย์กลางมีระยะทางการค้นหาเฉลี่ยสั้นที่สุด จากส่วนอื่นใดของดิสก์

บันทึกระบบไฟล์เป็นตัวเลือกที่ดีที่สุดสำหรับการจัดสรรที่ศูนย์กลางของ ฟิสิคัลวอลุ่ม เนื่องจากการใช้บ่อยโดยระบบปฏิบัติการ ในทางตรงกันข้าม บุตโลจิคัลวอลุ่มมีการใช้ไม่บ่อยนัก ดังนั้นจึง ถูกจัดสรรไว้ที่ขอบหรือกึ่งกลางของฟิสิคัลวอลุ่ม

กฎทั่วไปคือ I/Os ยิ่งมาก ซึ่งอาจเป็นแบบสลับหรือในระหว่าง การรันแอ็พพลิเคชันที่สำคัญ ยิ่งต้องจัดสรรพาร์ติชันฟิสิคัลของโลจิคัลวอลุ่ม ใกล้กับศูนย์กลางของฟิสิคัลวอลุ่มมากขึ้น

กฎนี้มีข้อยกเว้นสำคัญหนึ่งข้อคือ: โลจิคัลวอลุ่มที่มีร์เรอร์ซึ่งมีการตั้งค่า ความต้อกันในการเขียนสำเนา (MWC) เป็น On อยู่ที่ขอบด้านนอก เนื่องจาก นี่เป็นตำแหน่งที่ระบบเขียนข้อมูล MWC หากการมีร์เรอร์ไม่มีผลบังคับใช้ MWC จะไม่ใช่และไม่ มีผลกระทบต่อประสิทธิภาพ

ห้าภูมิภาคซึ่งสามารถระบุตำแหน่งพาร์ติชันฟิสิคัลได้ มีดังนี้:

1. ขอบด้านนอก
2. ขอบด้านใน

3. กลางด้านนอก
4. กลางด้านใน
5. ศูนย์กลาง

พาร์ติชันขอบมีเวลาการค้นหาเฉลี่ยช้าที่สุด ซึ่งโดยทั่วไปส่งผลให้เวลาการตอบกลับนานขึ้นสำหรับแอปพลิเคชันใดๆ ที่ใช้พาร์ติชันนั้น พาร์ติชันศูนย์กลางมีเวลาการค้นหาเฉลี่ยเร็วที่สุด ซึ่งโดยทั่วไปส่งผลให้เวลาการตอบกลับดีที่สุดสำหรับแอปพลิเคชันใดๆ ที่ใช้พาร์ติชันนั้น อย่างไรก็ตาม พาร์ติชันบนพริสคัลวอลุ่มที่ศูนย์กลางมีจำนวนน้อยกว่าที่พื้นที่อื่น

หลักการที่เกี่ยวข้อง:

“การมีอยู่สูงในกรณีของความล้มเหลว disks” ในหน้า 440

เมธอดหลักที่ใช้เพื่อป้องกันความล้มเหลว disks เกี่ยวข้องกับ ค่าติดตั้งคอนฟิกูเรชันโลจิคัลวอลุ่ม เช่น การมีเรอร์

การรวมนโยบายการจัดสรร:

หากคุณเลือกนโยบาย inter-disk และ intra-disk ที่ไม่เข้ากัน คุณอาจได้รับผลลัพธ์ที่คาดการณไม่ได้อีก

ระบบกำหนดพาร์ติชันพริสคัลโดยการอนุญาตให้นโยบายหนึ่ง มีความสำคัญมากกว่าอีกนโยบายหนึ่ง ตัวอย่างเช่น ถ้าคุณเลือกนโยบาย intra-disk ของศูนย์กลางและนโยบาย inter-disk ของต่ำสุด นโยบาย inter-disk จะมีความสำคัญมากกว่า ระบบจะวางพาร์ติชันทั้งหมดของโลจิคัลวอลุ่ม ไว้บนดิสก์เดียวถ้าเป็นไปได้ แม้ว่าพาร์ติชันทั้งหมดไม่พอดีในพื้นที่ศูนย์กลาง โปรดตรวจสอบให้แน่ใจว่าคุณเข้าใจการโต้ตอบระหว่างนโยบายซึ่งคุณเลือกก่อนนำนโยบายนั้นไปใช้

การใช้ไฟล์แม่พิมพ์สำหรับการจัดสรรที่แน่นอน:

หากดีฟอลต์อ็อปชันที่นำเสนอโดยนโยบาย inter- และ intra-disk ไม่เพียงพอสำหรับความต้องการของคุณ ให้พิจารณาการสร้างไฟล์แม่พิมพ์เพื่อระบุ ลำดับและที่ตั้งที่แน่นอนของพาร์ติชันพริสคัลสำหรับโลจิคัลวอลุ่ม

คุณสามารถใช้ SMIT หรือคำสั่ง `mkiv -m` เพื่อสร้างไฟล์แม่พิมพ์

ตัวอย่าง เช่น เมื่อต้องการสร้างลิบบพาร์ติชันโลจิคัลวอลุ่มที่เรียกว่า lv06 ใน rootvg ในพาร์ติชัน 1 ถึง 3, 41 ถึง 45, และ 50 ถึง 60 ของ hdisk1 คุณสามารถใช้ไพรซีเดอร์ต่อไปนี้จากบรรทัดคำสั่ง

1. เมื่อต้องการตรวจสอบว่าพาร์ติชันพริสคัลซึ่งคุณวางแผนจะใช้ว่างสำหรับการจัดสรร ให้พิมพ์:

```
lspv -p hdisk1
```

2. สร้างไฟล์ เช่น /tmp/mymap1 ที่มี:

```
hdisk1:1-3
hdisk1:41-45
hdisk1:50-60
```

คำสั่ง `mkiv` จัดสรรพาร์ติชันพริสคัลในลำดับที่ปรากฏขึ้นในไฟล์ แม่พิมพ์ ตรวจสอบให้แน่ใจว่ามีพาร์ติชันพริสคัลที่เพียงพออยู่ในไฟล์แม่พิมพ์ เพื่อจัดสรรโลจิคัลวอลุ่มทั้งหมดซึ่งคุณระบุด้วยคำสั่ง `mkiv` (คุณสามารถแสดงรายการมากกว่าที่คุณต้องการได้)

3. พิมพ์คำสั่ง:

```
mkiv -t jfs -y lv06 -m /tmp/mymap1 rootvg 10
```

การพัฒนาคลุ้มลอจิคัลวอลุ่มแบบแถบ:

ลอจิคัลวอลุ่มแบบแถบใช้สำหรับระบบไฟล์ตามลำดับขนาดใหญ่ที่มีการเข้าถึงบ่อยและมีระดับประสิทธิภาพที่เปลี่ยนแปลงได้ง่าย แบบแถบ มีไว้เพื่อปรับปรุงประสิทธิภาพ

หมายเหตุ: พื้นที่ว่างดัมพ์หรือบูตลอจิคัลวอลุ่มไม่สามารถทำเป็นแบบแถบได้ บูตลอจิคัลวอลุ่มต้องเป็นพาร์ติชันฟิสิคัลที่ต่อเนื่องกัน

เมื่อต้องการสร้าง ลอจิคัลวอลุ่มแบบแถบ 12-พาร์ติชันที่เรียกว่า lv07 ใน VGName ที่มีขนาดแถบ (ขนาดแถบคูณด้วยจำนวนดิสก์ในอาร์เรย์จะเท่ากับขนาดแถบ) เป็น 16 KB บน hdisk1, hdisk2, และ hdisk3 ให้พิมพ์:

```
mk1v -y lv07 -S 16K VGName 12 hdisk1 hdisk2 hdisk3
```

เมื่อต้องการสร้าง ลอจิคัลวอลุ่มแบบแถบ 12-พาร์ติชันที่เรียกว่า lv08 ใน VGName ที่มีขนาดแถบเป็น 8 KB บนสามดิสก์ใดๆ ภายใน VGName ให้พิมพ์:

```
mk1v -y lv08 -S 8K -u 3 VGName 12
```

สำหรับข้อมูล เพิ่มเติมเกี่ยวกับวิธีการปรับปรุงประสิทธิภาพโดยใช้การกำหนดแถบดิสก์ ให้ดูที่ *Performance management*

นโยบายตรวจสอบการเขียน:

การใช้ฮ็อทซันตรวจสอบการเขียนทำให้การดำเนินงานเขียนทั้งหมด ถูกตรวจสอบโดยการดำเนินงานอ่านที่ตามมาในทันที เพื่อตรวจสอบความสำเร็จของการเขียน

หากการดำเนินงานเขียนไม่สำเร็จ คุณจะรับข้อความแสดงข้อผิดพลาด นโยบายนี้ ช่วยเพิ่มการมีอยู่แต่ลดประสิทธิภาพ เนื่องจากต้องใช้เวลาเพิ่มเติม สำหรับการอ่าน คุณสามารถระบุการใช้นโยบายตรวจสอบการเขียน บนลอจิคัลวอลุ่ม เมื่อคุณสร้างลอจิคัลวอลุ่มโดยใช้คำสั่ง `mk1v` หรือในภายหลังโดยการเปลี่ยนโดยใช้คำสั่ง `ch1v`

นโยบายดิสก์ฮอตสแปร์:

คุณสามารถกำหนดดิสก์ให้เป็นดิสก์ฮอตสแปร์สำหรับกลุ่มวอลุ่ม ที่มีวอลุ่มลอจิคัลที่ทำมีร์เรอร์ไว้

เมื่อคุณกำหนดดิสก์ที่จะใช้เป็นดิสก์ฮอตสแปร์ คุณสามารถระบุ นโยบายที่จะใช้ถัดิสก์เริ่มล้มเหลว และคุณสามารถระบุ ลักษณะซิงโครไนซ์ได้

ถ้าคุณเพิ่มฟิสิคัลวอลุ่มลงในกลุ่มวอลุ่ม (เพื่อทำเครื่องหมายเป็นดิสก์ฮอตสแปร์) ดิสก์ต้องมีความจุเหมือนกับดิสก์ที่เล็กที่สุด ซึ่งอยู่ในกลุ่มวอลุ่มอยู่แล้ว เป็นอย่างน้อย เมื่อนำคุณลักษณะนี้ไปใช้ ข้อมูลจะถูกย้าย ไปยังดิสก์ฮอตสแปร์เมื่อความล้มเหลวในการเขียนความตองกันในการเขียนสำเนา (MWC) ทำเครื่องหมายฟิสิคัลวอลุ่มที่ขาดไป

คำสั่งเพื่อเปิดใช้งานการสนับสนุนดิสก์ฮอตสแปร์ `chvg` และ `chpv`, จัดเตรียมหลายฮ็อทซันเกี่ยวกับวิธีการอิมพลีเมนต์คุณลักษณะในไซต์ของคุณ ดังที่แสดงไว้โดยไวยากรณ์ต่อไปนี้:

```
chvg -hhotsparepolicy -ssyncpolicy VolumeGroup
```

โดยที่ *hotsparepolicy* กำหนดนโยบายต่อไปนี้ ซึ่งคุณต้องการใช้เมื่อดิสก์ล้มเหลว:

y ย้ายพาร์ติชันจากดิสก์ที่ล้มเหลวหนึ่งไปยังดิสก์สแปร์หนึ่งโดยอัตโนมัติ จากพูลของดิสก์ฮอตสแปร์ ดิสก์ที่เล็กที่สุดซึ่งใหญ่เพียงพอจะทดแทน ดิสก์ที่ล้มเหลวจะถูกนำมาใช้

- Y** ย้ายพาร์ติชันจากดิสก์ที่ล้มเหลวโดยอัตโนมัติ แต่จะใช้พูลทั้งหมด ของดิสก์ฮอตสแปร์
- n** ไม่ย้ายโดยอัตโนมัติ (ดีฟอลต์)
- r** ลบดิสก์ทั้งหมดออกจากพูลของดิสก์ฮอตสแปร์สำหรับกลุ่มวอลุ่มนี้

อาร์กิวเมนต์ *syncpolicy* กำหนดว่าคุณต้องการซิงโครไนซ์พาร์ติชันเก่าใดๆ โดยอัตโนมัติหรือไม่:

- y** พยายามซิงโครไนซ์พาร์ติชันเก่าโดยอัตโนมัติ
- n** ไม่พยายามซิงโครไนซ์พาร์ติชันเก่าโดยอัตโนมัติ (อ็อปชันนี้เป็นค่าดีฟอลต์)

อาร์กิวเมนต์ *VolumeGroup* ระบุชื่อของกลุ่มวอลุ่มที่ทำสำเนาซึ่ง เชื่อมโยง

การจัดการ hot spot ในโลจิคัลวอลุ่ม:

คุณสามารถจำแนกปัญหาของ *hot spot* ด้วยโลจิคัลวอลุ่ม ของคุณ และแก้ไขปัญหาเหล่านั้นโดยไม่รบกวนการใช้งานของ ระบบของคุณ

ปัญหา hot-spot เกิดขึ้นเมื่อบางพาร์ติชันโลจิคัลบนดิสก์ของคุณ มีดิสก์ I/O มากจนส่งผลกระทบต่อประสิทธิภาพของระบบ อย่างเห็นได้ชัด

ขั้นตอนแรกในการแก้ไขปัญหาคือการระบุปัญหา โดยค่าดีฟอลต์ ระบบไม่ได้รวบรวมสถิติเกี่ยวกับการใช้โลจิคัลวอลุ่ม หลังจากคุณเปิดใช้งาน การรวบรวมสถิติเหล่านี้ในครั้งแรกที่คุณป้อนคำสั่ง *lvmstat* ระบบแสดงค่าตัวนับตั้งแต่วินาทีระบบครั้งก่อนหน้านั้น หลังจากนั้น ในแต่ละครั้งที่คุณป้อนคำสั่ง *lvmstat* ระบบแสดง ความแตกต่างตั้งแต่คำสั่ง *lvmstat* ก่อนหน้านั้น

โดยการตีความเอาต์พุตของคำสั่ง *lvmstat* คุณสามารถระบุพาร์ติชันโลจิคัลที่มีการจราจรหนาแน่นที่สุดได้ หาก你有หลายพาร์ติชันโลจิคัลที่มีการใช้งานมากบนหนึ่งฟิสิคัลดิสก์และต้องการ ปรับสมดุลระหว่างดิสก์ที่มีอยู่ คุณสามารถใช้คำสั่ง *migratep* เพื่อย้ายพาร์ติชันโลจิคัลเหล่านี้ไปยังฟิสิคัลดิสก์อื่น

ในตัวอย่างต่อไปนี้ การรวบรวมสถิติมีการเปิดใช้งานและใช้คำสั่ง *lvmstat* ซ้ำๆ เพื่อรวบรวมสถิติพื้นฐาน:

```
# lvmstat -v rootvg -e
# lvmstat -v rootvg -C
# lvmstat -v rootvg
```

เอาต์พุตคล้ายกับตัวอย่างต่อไปนี้:

Logical Volume	iocnt	Kb_read	Kb_wrtn	Kbps
hd8	4	0	16	0.00
paging01	0	0	0	0.00
lv01	0	0	0	0.00
hd1	0	0	0	0.00
hd3	0	0	0	0.00
hd9var	0	0	0	0.00
hd2	0	0	0	0.00
hd4	0	0	0	0.00
hd6	0	0	0	0.00
hd5	0	0	0	0.00

เอาต์พุตก่อนหน้านี้นี้แสดงว่าตัวนับทั้งหมดถูกรีเซ็ตเป็นศูนย์ ในตัวอย่าง ต่อไปนี้ อันดับแรก ข้อมูลถูกคัดลอกจากไต่เร็กทอรี */unix* ไปยังไต่เร็กทอรี */tmp* เอาต์พุตคำสั่ง *lvmstat* สะท้อนถึงกิจกรรมของ *rootvg*:

```
# cp -p /unix /tmp
# lvmstat -v rootvg
```

Logical Volume	iocnt	Kb_read	Kb_wrtn	Kbps
hd3	296	0	6916	0.04
hd8	47	0	188	0.00
hd4	29	0	128	0.00
hd2	16	0	72	0.00
paging01	0	0	0	0.00
lv01	0	0	0	0.00
hd1	0	0	0	0.00
hd9var	0	0	0	0.00
hd6	0	0	0	0.00
hd5	0	0	0	0.00

เอาต์พุตแสดงกิจกรรมบนโลจิคัลวอลุ่ม **hd3** ซึ่งติดตั้งใน ไดรฟ์ทอริ /tmp บน **hd8** ซึ่งเป็นโลจิคัลวอลุ่มบันทึก JFS บน **hd4** ซึ่งเป็น / (ราก) บน **hd2** ซึ่งเป็นไดเร็กทอรี /usr และบน **hd9var** ซึ่งเป็นไดเร็กทอรี /var เอาต์พุตต่อไปนี้แสดง รายละเอียดสำหรับ **hd3** และ **hd2**:

```
# lvmstat -l hd3
```

Log_part	mirror#	iocnt	Kb_read	Kb_wrtn	Kbps
1	1	299	0	6896	0.04
3	1	4	0	52	0.00
2	1	0	0	0	0.00
4	1	0	0	0	0.00

```
# lvmstat -l hd2
```

Log_part	mirror#	iocnt	Kb_read	Kb_wrtn	Kbps
2	1	9	0	52	0.00
3	1	9	0	36	0.00
7	1	9	0	36	0.00
4	1	4	0	16	0.00
9	1	1	0	4	0.00
14	1	1	0	4	0.00
1	1	0	0	0	0.00

เอาต์พุตสำหรับกลุ่มวอลุ่มแสดงข้อมูลสรุปสำหรับกิจกรรม I/O ทั้งหมดของโลจิคัลวอลุ่ม เอาต์พุตแบ่งออกเป็นจำนวนของ คำร้องขอ I/O (iocnt), กิโลไบต์ที่อ่านและที่เขียน (Kb_read และ Kb_wrtn ตามลำดับ) และข้อมูลที่โอนย้ายในหน่วย KB/s (Kbps) หาก คุณร้องขอข้อมูลโลจิคัลวอลุ่ม คุณจะได้รับข้อมูลเดียวกัน แต่สำหรับแต่ละพาร์ติชันโลจิคัลแยกต่างหากกัน หาก คุณมีโลจิคัลวอลุ่มที่มีเรอร์ คุณจะได้รับสถิติของแต่ละมีเรอร์วอลุ่ม ในเอาต์พุตตัวอย่างก่อนหน้านี้ หลายบรรทัดของพาร์ติชันโลจิคัลที่ไม่มีกิจกรรมใดๆ ถูกข้ามไป เอาต์พุตมีการเรียงลำดับในลำดับจากมากไปน้อยเสมอบนคอลัมน์ iocnt

คำสั่ง **migratep** ใช้ชื่อของโลจิคัลวอลุ่ม จำนวนของพาร์ติชัน โลจิคัล (ตามที่แสดงขึ้นในเอาต์พุต **lvmstat**) และจำนวนที่เลือก กำหนดได้ของสำเนา มีเรอร์เฉพาะ เป็น พารามิเตอร์ หากข้อมูลถูกข้ามไป จะใช้สำเนา มีเรอร์แรก คุณต้อง ระบุพิสิตัลวอลุ่ม เป้าหมายสำหรับการย้าย นอกจากนี้ คุณสามารถ ระบุหมายเลขพาร์ติชันพิสิตัลเป้าหมาย ถ้าสำเร็จ เอาต์พุตจะคล้ายกับ ตัวอย่างต่อไปนี้:

```
# migratep hd3/1 hdisk1/109
migratep: Mirror copy 1 of logical partition 1 of logical volume
hd3 migrated to physical partition 109 of hdisk1.
```

หลังจากเปิดใช้งานคุณลักษณะ hot spot สำหรับโลจิกัลวอลุ่มหรือกลุ่มวอลุ่มอย่างใดอย่างหนึ่ง คุณสามารถกำหนดการรายงานและสถิติ แสดงสถิติของคุณ เลือกพาร์ติชันฟิสิคัลที่จะย้าย ระบุพาร์ติชันฟิสิคัลปลายทาง และตรวจสอบข้อมูลก่อน committing การเปลี่ยนแปลงของคุณ

การนำนโยบายกลุ่มวอลุ่มไปใช้

หลังจากคุณตัดสินใจเลือกนโยบายกลุ่มวอลุ่มซึ่งคุณต้องการใช้แล้ว ให้วิเคราะห์คอนฟิกูเรชันปัจจุบันโดยพิมพ์คำสั่ง lspv บนบรรทัดคำสั่ง

คอนฟิกูเรชันมาตรฐานนำเสนอกลุ่มวอลุ่มเดียวที่มีหลายฟิสิคัลวอลุ่มต่อพ่วงอยู่กับอะแดปเตอร์ดิสก์ตัวเดียวกัน และฮาร์ดแวร์ สับสวิตช์อื่น ในคอนฟิกูเรชันมาตรฐาน ยังมีดิสก์เป็นองค์ประกอบของกลุ่มวอลุ่มมากขึ้น โอกาสขององค์ประกอบที่ยังคงอยู่เมื่อดิสก์ล้มเหลวยังมี มากขึ้น ในกลุ่มที่ไม่ใช่ของค์ประกอบ ต้องมีดิสก์อย่างน้อยสองตัวประกอบขึ้นเป็นกลุ่ม วอลุ่ม เมื่อต้องการนำการเปลี่ยนแปลงนโยบายกลุ่มวอลุ่มไปใช้ให้ทำต่อไปนี้:

1. ใช้เอาต์พุตคำสั่ง lspv เพื่อตรวจสอบฟิสิคัลวอลุ่มที่จัดสรร และที่ว่าง
2. ทำให้มั่นใจเกี่ยวกับองค์ประกอบโดยการเพิ่มหนึ่งหรือหลายฟิสิคัลวอลุ่ม
3. เปลี่ยนกลุ่มวอลุ่มเป็นสถานะไม่ใช่ของค์ประกอบ
4. กำหนดคอนฟิกฮาร์ดแวร์อีกครั้งเฉพาะถ้าจำเป็นต้องให้มั่นใจถึงการมีอยู่สูง สำหรับคำแนะนำ ให้ดูที่คู่มือเซอร์วิสสำหรับระบบของคุณ

หลักการที่เกี่ยวข้อง:

“การแปลงกลุ่มวอลุ่มเป็นสถานะไม่ใช่ของค์ประกอบ” ในหน้า 387

คุณสามารถเปลี่ยนกลุ่มวอลุ่มเป็นสถานะไม่ใช่ของค์ประกอบเพื่อให้ข้อมูล มีอยู่อย่างต่อเนื่องแม้ว่าไม่มีองค์ประกอบ

งานที่เกี่ยวข้อง:

“การเพิ่มดิสก์ในขณะที่ระบบยังคงเปิดอยู่” ในหน้า 391

โพสดีเตอร์ต่อไปนี้อธิบายวิธีการเปิดและกำหนดคอนฟิก ดิสก์โดยใช้คุณลักษณะ hot-removability ซึ่งอนุญาตให้คุณเพิ่มดิสก์ได้โดยไม่ต้อง ปิดระบบ

พื้นที่การเลื่อนหน้าและหน่วยความจำเสมือน

AIX ใช้หน่วยความจำ เสมือนเพื่อระบุหน่วยความจำมากกว่าที่มีอยู่ทางกายภาพในระบบ

การจัดการหน้าหน่วยความจำใน RAM หรือบนดิสก์มีการจัดการโดย Virtual Memory Manager (VMM) เซ็กเมนต์หน่วยความจำเสมือนมีการแบ่งพาร์ติชันเป็นหน่วยที่เรียกว่า หน้า พื้นที่การเลื่อนหน้า เป็นโลจิกัลวอลุ่มชนิดหนึ่งซึ่งมีพื้นที่ว่างดิสก์ที่จัดสรร ซึ่งจัดเก็บข้อมูลที่มีอยู่ในหน่วยความจำเสมือนแต่ไม่ได้กำลังเข้าถึงในปัจจุบัน โลจิกัลวอลุ่มนี้มีชนิดแอดทริบิวต์ เหมือนกับการเลื่อนหน้า และโดยปกติ มีการเรียกแบบง่ายว่า พื้นที่การเลื่อนหน้า หรือ พื้นที่สลับ เมื่อจำนวนของ RAM ที่ว่างในระบบต่ำ โปรแกรมหรือข้อมูลที่ไม่ได้ใช้งานเมื่อเร็ว ๆ นี้ จะถูกย้ายออกจากหน่วยความจำไปยังพื้นที่การเลื่อนหน้าเพื่อรีเสิสหน่วยความจำสำหรับกิจกรรมอื่น

แนวคิดพื้นที่การเลื่อนหน้า

พื้นที่การเพจ เป็นชนิดของโลจิกัลวอลุ่มที่มีพื้นที่ดิสก์ที่จัดสรรไว้ที่เก็บข้อมูล ซึ่งอยู่ในหน่วยความจำเสมือนแต่ไม่ถูกเข้าถึงในปัจจุบัน

โลจิกัลวอลุ่มนี้มีชนิดแอดทริบิวต์เหมือนกับการเลื่อนหน้า และโดยปกติมีการเรียกแบบง่ายว่า พื้นที่การเลื่อนหน้า หรือ *พื้นที่สลับ* เมื่อจำนวนของ RAM ที่ว่างในระบบต่ำ โปรแกรมหรือข้อมูลที่ไม่ได้ใช้งานเมื่อเร็วๆ นี้ จะถูกย้ายออกจากหน่วยความจำไปยังพื้นที่การเลื่อนหน้าเพื่อรีลีสหน่วยความจำสำหรับกิจกรรมอื่น

พื้นที่การเลื่อนหน้าอีกชนิดหนึ่งสามารถเข้าถึงได้ผ่านทางอุปกรณ์ ที่ใช้เซิร์ฟเวอร์ NFS เป็นหน่วยเก็บพื้นที่การเลื่อนหน้า เพื่อให้ไคลเอ็นต์ NFS เข้าถึง พื้นที่การเลื่อนหน้านี้ได้ เซิร์ฟเวอร์ NFS ต้องสร้างและเอ็กซ์พอร์ตไฟล์ ไปยังไคลเอ็นต์นั้น ขนาดไฟล์แสดงถึงขนาดพื้นที่การเลื่อนหน้าสำหรับไคลเอ็นต์

จำนวนของพื้นที่การเลื่อนหน้าที่ต้องการขึ้นอยู่กับชนิดของกิจกรรมที่ทำ บนระบบ หากพื้นที่การเลื่อนหน้ารับเข้า โปรแกรมอาจสูญหายได้ และถ้าขาดพื้นที่การเลื่อนหน้า ระบบอาจล้มเหลว เมื่อตรวจพบสภาพพื้นที่การเลื่อนหน้าต่ำ ให้กำหนดพื้นที่การเลื่อนหน้าเพิ่มเติม

พื้นที่การเลื่อนหน้าโลจิกัลวอลุ่มถูกกำหนดโดยการจัดทำโลจิกัลวอลุ่ม พื้นที่การเลื่อนหน้าใหม่ หรือโดยการเพิ่มขนาดของโลจิกัลวอลุ่มพื้นที่การเลื่อนหน้า ที่มีอยู่ในการเพิ่มขนาดของพื้นที่การเลื่อนหน้า NFS ไฟล์ที่อยู่บนเซิร์ฟเวอร์ ต้องถูกเพิ่มโดยการดำเนินการที่ถูกต้องบนเซิร์ฟเวอร์

พื้นที่ว่างทั้งหมดที่มีอยู่ในระบบสำหรับการเลื่อนหน้าเป็นผลรวมของขนาด ของโลจิกัลวอลุ่มพื้นที่การเลื่อนหน้าที่ใช้งานอยู่ทั้งหมด

หลักการที่เกี่ยวข้อง:

“การแก้ไขปัญหาพื้นที่การเลื่อนหน้า” ในหน้า 461

ปัญหาที่พบบ่อยที่สุดเกี่ยวกับพื้นที่การเลื่อนหน้าเกิดขึ้นจากการขาด พื้นที่ว่างที่จัดสรร

นโยบายการจัดสรรพื้นที่การเลื่อนหน้า:

ตัวแปรสถานะแวดล้อม *PSALLOC* จะกำหนดอัลกอริทึมการจัดสรรพื้นที่การเพจที่ใช้: *deferred* หรือ *early*

AIX ใช้สองโหมดสำหรับการจัดสรร พื้นที่การเลื่อนหน้า ค่าดีฟอลต์ คือ *deferred* คุณสามารถสลับไปยังโหมดการจัดสรรพื้นที่การเลื่อนหน้า *early* โดยการเปลี่ยนค่าของตัวแปรสถานะแวดล้อม *PSALLOC* แต่มีหลายปัจจัยที่ควรพิจารณาก่อนทำการเปลี่ยนแปลงดังกล่าว เมื่อใช้ขั้นตอนวิธีการจัดสรร *early* ในสถานการณ์จำลองที่เลวร้ายที่สุด อาจทำให้ ระบบเสียหายได้เนื่องจากการใช้พื้นที่การเลื่อนหน้าที่มีอยู่ทั้งหมด

การเปรียบเทียบการจัดสรรพื้นที่การเพจแบบ deferred และ early:

ระบบปฏิบัติการใช้ตัวแปรสถานะแวดล้อม *PSALLOC* เพื่อกำหนดกลไกที่ใช้สำหรับการจัดสรรหน่วยความจำและพื้นที่การเลื่อนหน้า

หากไม่ได้ตั้งค่าตัวแปรสถานะแวดล้อม *PSALLOC*, ตั้งค่าเป็น *null* หรือตั้งค่าเป็นค่าอื่นที่ไม่ใช่ *early* ระบบจะใช้ อัลกอริทึมการจัดสรร *deferred* ดีฟอลต์

อัลกอริทึมการจัดสรร *deferred* จะช่วยให้การใช้ดิสก์รีดได้อย่างมีประสิทธิภาพ และสนับสนุน แอ็พพลิเคชันที่ต้องการกลไกการจัดสรรแบบกระจายสำหรับการจัดการรีด อัลกอริทึมนี้ไม่ได้สงวนพื้นที่การเพจไว้เมื่อมีการร้องขอหน่วยความจำ การจัดสรรดิสก์บล็อก ของพื้นที่การเพจจะถูกหน่วงเวลาจนกว่าจะจำเป็นต้องเพจเพจที่ร้องขอออก บางโปรแกรมจัดสรรหน่วยความจำเสมือนจำนวนมาก จากนั้นใช้หน่วยความจำเพียงบางส่วนเท่านั้น ตัวอย่างของโปรแกรมดังกล่าว คือแอ็พพลิเคชันด้านเทคนิคที่ใช้เว็กเตอร์หรือเมทริก *sparse* เป็นโครงสร้างข้อมูล อัลกอริทึมการจัดสรรแบบ *deferred* ยังมีประสิทธิภาพมากกว่าสำหรับเคอร์เนลแบบเรียลไทม์ และเคอร์เนลที่เพจตามความต้องการ เช่น ในระบบปฏิบัติการ

พื้นที่การเพจนี้อาจไม่ถูกใช้เลย โดยเฉพาะอย่างยิ่งบนระบบที่มีหน่วยความจำจริงจำนวนมาก ที่มีการเพจน้อย อัลกอริทึมแบบ deferred จะหน่วงเวลาการจัดสรรพื้นที่การเพจจนกว่าจะจำเป็นต้องเพจเพจที่ร้องขอออก ซึ่งทำให้ไม่มีการจัดสรรพื้นที่การเพจที่เสียเปล่า การจัดสรรแบบ delayed นี้สามารถ ทำให้อัลกอริทึมแบบ deferred สามารถจองจัดสรรพื้นที่การเพจมากกว่าพื้นที่ที่มี บนระบบ สถานการณ์นี้เรียกว่า over-commitment ของพื้นที่การเพจ

ในสถานการณ์ over-commitment ที่พื้นที่การเพจไม่เพียงพอและพยายามจัดสรร ดิสก์บล็อกของพื้นที่การเพจเพื่อเพจเพจออก ทำให้เกิดความล้มเหลว ระบบปฏิบัติการ จะพยายามหลีกเลี่ยงความล้มเหลวของระบบโดยสมบูรณ์โดยการจำกัดกระบวนการที่ได้รับผลกระทบโดย พื้นที่การเพจที่ over-commitment มีการส่งสัญญาณ SIGDANGER เพื่อแจ้งโปรเซสให้ทราบว่าจำนวนของ พื้นที่การเลื่อนหน้าที่ว่างต่ำ หากสถานการณ์พื้นที่การเลื่อนหน้าเข้าขั้นวิกฤต มากขึ้น โปรเซสที่เลือกซึ่งไม่ได้รับสัญญาณ SIGDANGER จะได้รับสัญญาณ SIGKILL

คุณสามารถใช้ตัวแปรสถานะแวดล้อม PSALLOC เพื่อสลับไปยังขั้นตอนวิธีการจัดสรร early ซึ่งจะจัดสรร พื้นที่การเลื่อนหน้าสำหรับโปรเซสที่ดำเนินการในเวลาที่ยังขอหน่วยความจำ หากมีพื้นที่การเลื่อนหน้าไม่เพียงพอในเวลาที่ยังขอ กลไกการจัดสรรแบบ early จะไม่สามารถตอบสนองต่อคำร้องขอหน่วยความจำได้

หากมีการตั้งค่าตัวแปรสถานะแวดล้อม PSALLOC เป็น early ทุกโปรแกรมที่เริ่มต้นในสถานะแวดล้อมนั้นนับจากเวลาที่ตั้งค่า แต่ไม่รวม โปรเซสที่กำลังรันอยู่ในปัจจุบัน จะรันในสถานะแวดล้อมการจัดสรรแบบ early ในสถานะแวดล้อมการจัดสรรแบบ early อินเทอร์เฟซ เช่น รูทีนย่อย malloc และรูทีนย่อย brk จะล้มเหลวถ้าไม่สามารถสำรองพื้นที่การเลื่อนหน้า ที่เพียงพอเมื่อจัดทำคำร้องขอ

กระบวนการที่รันในโหมดสถานะแวดล้อมการจัดสรรก่อนหน้านี้อาจไม่ส่งสัญญาณ SIGKILL หากเกิดเงื่อนไขพื้นที่การเพจเหลือน้อย

มีหลายวิธีในการเปลี่ยนตัวแปรสถานะแวดล้อม PSALLOC เป็น early ขึ้นอยู่กับว่าคุณต้องการใช้การเปลี่ยนแปลงกว้างขวางมากเพียงใด

รูทีนย่อยอินเทอร์เฟซการจัดสรรหน่วยความจำต่อไปนี้ได้รับผลกระทบจากการสลับไปยังสถานะแวดล้อมการจัดสรรแบบ early:

- malloc
- free
- calloc
- realloc
- brk
- sbrk
- shmget
- shmctl

งานที่เกี่ยวข้อง:

“การกำหนดคอนฟิกตัวแปรสถานะแวดล้อม PSALLOC สำหรับโหมด การจัดสรร early” ในหน้า 458

ระบบปฏิบัติการใช้ตัวแปรสถานะแวดล้อม PSALLOC เพื่อกำหนดกลไกที่ใช้สำหรับการจัดสรรหน่วยความจำและพื้นที่การเลื่อนหน้า

โหมดการจัดสรร early:

ขั้นตอนวิธีการจัดสรร early ช่วยรับประกันถึงพื้นที่การเลื่อนหน้ามากที่สุดที่ร้องขอโดยคำร้องขอการจัดสรรหน่วยความจำ ดังนั้น การจัดสรรพื้นที่การเลื่อนหน้าที่ถูกต้อง บนดิสก์ระบบจึงมีความสำคัญสำหรับการดำเนินงานที่มีประสิทธิภาพ

เมื่อพื้นที่การเลื่อนหน้าที่มีอยู่ลดลงต่ำกว่า threshold โพรเซสใหม่ จะไม่สามารถเริ่มต้นได้และโพรเซสที่กำลังรันอยู่ในปัจจุบัน อาจไม่สามารถได้รับ หน่วยความจำเพิ่มเติม กระบวนการใดๆ ที่รันภายใต้โหมดการจัดสรรแบบ deferred ดีฟอลต์จะมีความเสี่ยงสูงจากกลไกการส่งสัญญาณ SIGKILL นอกจากนี้ เนื่องจากในบางครั้งเคอร์เนลระบบปฏิบัติการต้องการการจัดสรร หน่วยความจำ จึงอาจทำให้ระบบเสียหายได้เนื่องจากการใช้พื้นที่การเลื่อนหน้าที่มีอยู่ทั้งหมด

ก่อนคุณใช้โหมดการจัดสรร early ทั้งทั้งระบบ สิ่งสำคัญมากคือ การกำหนดจำนวนพื้นที่การเลื่อนหน้าที่เหมาะสมสำหรับระบบ พื้นที่การเพจที่ต้องการสำหรับโหมดการจัดสรรแบบ early มักมากกว่าพื้นที่การเพจที่ต้องการสำหรับโหมดการจัดสรรแบบ deferred ดีฟอลต์เสมอ จำนวนของพื้นที่การเลื่อนหน้าที่จะกำหนดขึ้นอยู่กับวิธีการใช้ระบบของคุณและโปรแกรมที่คุณรัน จุดเริ่มต้นที่ดีสำหรับการกำหนดส่วนผสมที่ถูกต้องสำหรับระบบของคุณคือ การกำหนดพื้นที่การเลื่อนหน้ามากกว่าจำนวนของหน่วยความจำฟิสิกส์สี่เท่า

บางแอปพลิเคชันสามารถใช้พื้นที่การเลื่อนหน้าเป็นจำนวนมากถ้ารัน ในโหมดการจัดสรร early ในปัจจุบัน เซิร์ฟเวอร์ AIX windows ต้องการพื้นที่การเลื่อนหน้ามากกว่า 250 MB เมื่อแอปพลิเคชันรันใน โหมดการจัดสรร early จำนวนพื้นที่การเลื่อนหน้าที่ต้องการสำหรับแอปพลิเคชันใดๆ ขึ้นอยู่กับ วิธีการเขียนแอปพลิเคชันและวิธีการรัน

คำสั่งและรูทีนย่อยทั้งหมดที่แสดงพื้นที่การเลื่อนหน้าและการใช้หน่วยความจำ ของโพรเซส รวมพื้นที่การเลื่อนหน้าที่จัดสรร ภายใต้โหมดการจัดสรร early คำสั่ง lpsps ใช้แฟล็ก -s เพื่อแสดงการจัดสรรพื้นที่การเลื่อนหน้าทั้งหมด รวมถึง พื้นที่การเลื่อนหน้าที่จัดสรรภายใต้โหมดการจัดสรร early

ขนาดดีฟอลต์ของพื้นที่การเลื่อนหน้า:

ขนาดดีฟอลต์ของพื้นที่การเลื่อนหน้ามีการกำหนดในระหว่างระยะการกำหนด ระบบเองของการติดตั้ง AIX ตาม มาตรฐานต่อไปนี้

- พื้นที่การเลื่อนหน้าสามารถใช้ไม่เกิน 16 MB ยกเว้นสำหรับ hd6 ที่ใช้ไม่เกิน 64 MB
- พื้นที่การเลื่อนหน้าไม่สามารถใช้มากกว่า 20% ของพื้นที่ว่างดิสก์ทั้งหมด
- หากหน่วยความจำแบบจริงน้อยกว่า 256 MB พื้นที่การเลื่อนหน้าจะเป็นสองเท่าของหน่วยความจำแบบจริง
- หากหน่วยความจำแบบจริงมากกว่าหรือเท่ากับ 256 MB พื้นที่การเลื่อนหน้าจะเป็น 512 MB

ไฟล์พื้นที่การเลื่อนหน้า คำสั่ง และอ็อปชัน:

ไฟล์ /etc/swapspace ระบุพื้นที่การเลื่อนหน้า และแอตทริบิวต์ของพื้นที่การเลื่อนหน้า

พื้นที่การเลื่อนหน้ามีการเพิ่มลงในไฟล์ /etc/swapspace เมื่อ พื้นที่การเลื่อนหน้าถูกสร้างขึ้นโดยคำสั่ง mkps และพื้นที่การเลื่อนหน้าถูกลบออกจากไฟล์ /etc/swapspace เมื่อถูกลบออก โดยใช้คำสั่ง rmpr แอตทริบิวต์พื้นที่การเลื่อนหน้าในไฟล์ ถูกแก้ไขโดยคำสั่ง chps -a หรือคำสั่ง chps -c ไฟล์ที่ใช้รูปแบบก่อนหน้านี้ (ซึ่งไม่มีแอตทริบิวต์สำหรับขนาด checksum และ swap-on อัตโนมัตินใน stanzas) ยังคงได้รับการสนับสนุนต่อไป หากขนาดพื้นที่การเลื่อนหน้าใหญ่เกินไป คุณสามารถลบพาร์ติชันโลจิคัลออกจากพื้นที่การเลื่อนหน้าได้โดยไม่ต้องรีบูต โดยใช้คำสั่ง chps -d

ใช้คำสั่งต่อไปนี้เพื่อจัดการพื้นที่การเลื่อนหน้า:

ไอเอ็ม	คำอธิบาย
chps	เปลี่ยนแอตทริบิวต์ของพื้นที่การเลื่อนหน้า
lsp	แสดงลักษณะของพื้นที่การเลื่อนหน้า
mkps	เพิ่มพื้นที่การเลื่อนหน้าเพิ่มเติม คำสั่ง mkps ใช้ คำสั่ง mklv ที่มีชุดเฉพาะของ อีพซันเมื่อสร้างโลจิคัลวอลุ่มพื้นที่การเลื่อนหน้า เพื่อสร้างพื้นที่การเลื่อนหน้า NFS คำสั่ง mkps ใช้คำสั่ง mkdev ที่มีชุดเฉพาะ ของอีพซัน สำหรับพื้นที่การเลื่อนหน้า NFS คำสั่ง mkps ต้องการ ชื่อโฮสต์ของเซิร์ฟเวอร์ NFS และชื่อพารของไฟล์ที่จะเอ็กซ์พอร์ตจากเซิร์ฟเวอร์
rmps	ลบพื้นที่การเลื่อนหน้าที่ไม่ได้ใช้งานอยู่
swapoff	ยกเลิกการเรียกใช้พื้นที่การเลื่อนหน้าตั้งแต่หนึ่งพื้นที่ขึ้นไปโดยไม่ต้องรีบูตระบบ ข้อมูลในพื้นที่การเลื่อนหน้าถูกย้ายไปยังพื้นที่การเลื่อนหน้าที่ใช้ฐานอยู่อื่น จากนั้น สามารถลบพื้นที่การเลื่อนหน้าที่ยกเลิกการเรียกใช้ได้โดยใช้คำสั่ง rmfs
swapon	เรียกใช้พื้นที่การเลื่อนหน้า คำสั่ง swapon มีการใช้ ในระหว่างการเริ่มต้นระบบในระยะแรกเพื่อเรียกใช้อุปกรณ์พื้นที่การเลื่อนหน้า แรกเริ่ม ในระหว่างระยะต่อมาของการเริ่มต้น เมื่ออุปกรณ์อื่นมีอยู่แล้ว จะใช้คำสั่ง swapon เพื่อเรียกใช้พื้นที่การเลื่อนหน้าเพิ่มเติม เพื่อให้กิจกรรมการเลื่อนหน้าเกิดขึ้นบนหลายอุปกรณ์

ต้องใช้อีพซัน ชนิดการเลื่อนหน้า สำหรับพื้นที่การเลื่อนหน้า โลจิคัลวอลุ่มทั้งหมด

ใช้อีพซันต่อไปนี้เพื่อเพิ่มประสิทธิภาพการเลื่อนหน้าให้สูงสุดด้วยโลจิคัล วอลุ่ม:

- จัดสรรในตำแหน่งกึ่งกลางของดิสก์เพื่อลดระยะการเดินทางบนดิสก์
- ใช้หลายพื้นที่การเลื่อนหน้า โดยจัดสรรแต่ละพื้นที่จากฟิสิคัลวอลุ่มแยกต่างหาก

การกำหนดคอนฟิกพื้นที่การเลื่อนหน้า

งานการกำหนดคอนฟิกจำนวนมาสามารถทำได้โดยใช้ SMIT พื้นที่การเลื่อนหน้าและการจัดสรรหน่วยความจำถูกควบคุมโดยตัวแปรสภาพแวดล้อม PSALLOCS

การเพิ่มและการเรียกใช้พื้นที่การเลื่อนหน้า:

เพื่อให้พื้นที่การเลื่อนหน้ามีอยู่ในระบบของคุณ คุณต้องเพิ่มและ เรียกใช้พื้นที่การเลื่อนหน้า

จำนวนทั้งหมดของพื้นที่การเลื่อนหน้ามักถูกกำหนดโดยการทดลองและข้อผิดพลาด แนวทางที่ใช้กันทั่วไปอย่างหนึ่งคือการเพิ่มขนาด RAM เป็นสองเท่าและใช้ตัวเลขนั้นเป็นเป้าหมายพื้นที่การเลื่อนหน้า

เมื่อใช้อินเตอร์เฟส SMIT ให้พิมพ์หนึ่งในพารตวนต่อไปนี้ลงในบรรทัดคำสั่ง:

- เมื่อต้องการแสดงรายการพื้นที่การเลื่อนหน้าปัจจุบันของคุณ ให้พิมพ์: smit lsp
- เมื่อต้องการเพิ่มพื้นที่การเลื่อนหน้า ให้พิมพ์: smit mkps
- เมื่อต้องการเรียกใช้พื้นที่การเลื่อนหน้า ให้พิมพ์: smit swapon

งานที่เกี่ยวข้อง:

“การย้ายพื้นที่การเลื่อนหน้า hd6 ภายในกลุ่มวอลุ่มเดียวกัน” ในหน้า 461

การย้ายพื้นที่การเลื่อนหน้าดีฟอลต์จาก hdisk0 ไปยังดิสก์อื่น ภายในกลุ่มวอลุ่มเดียวกันไม่จำเป็นต้องปิดและ รีบูตระบบ

การปรับปรุงประสิทธิภาพการเลื่อนหน้า:

เมื่อต้องการปรับปรุงประสิทธิภาพการเลื่อนหน้า ให้ใช้หลายพื้นที่การเลื่อนหน้า และระบุตำแหน่งพื้นที่เหล่านั้นบนฟิสิคัลวอลุ่มแยกต่างหากเมื่อสามารถทำได้

อย่างไรก็ตาม สามารถระบุตำแหน่งมากกว่าหนึ่งพื้นที่การเลื่อนหน้าบนฟิสิคัลวอลุ่มเดียวกันได้ แม้ว่า你可以ใช้หลายฟิสิคัลวอลุ่มได้ แต่แนวคิดที่ดีที่สุดคือควร เลือกเฉพาะดิสก์ภายในกลุ่มวอลุ่ม rootvg ยกเว้นว่าคุณคุ้นเคยกับระบบ ของคุณทั่วทั้งระบบ

การกำหนดคอนฟิกตัวแปรสถานะแวดล้อม PSALLOC สำหรับโหมด การจัดสรร early:

ระบบปฏิบัติการใช้ตัวแปรสถานะแวดล้อม PSALLOC เพื่อกำหนดกลไกที่ใช้สำหรับการจัดสรรหน่วยความจำและพื้นที่การเลื่อนหน้า

ค่าติดตั้งดีฟอลต์คือ late ตัวอย่างต่อไปนี้จะแสดงวิธีที่แตกต่างกันในการเปลี่ยนตัวแปรสถานะแวดล้อม PSALLOC เป็น early เมธอดที่คุณเลือกขึ้นอยู่กับความกว้าง ซึ่งคุณต้องการใช้การเปลี่ยนแปลง

- พิมพ์คำสั่งต่อไปนี้บนบรรทัดคำสั่งเชลล์:

```
PSALLOC=early;export PSALLOC
```

คำสั่งนี้ทำให้คำสั่งในลำดับต่อมาทั้งหมดที่รับจากเซสชันเชลล์นั้นรัน ในโหมดการจัดสรร early

- เพิ่มคำสั่งต่อไปนี้ในไฟล์เชลล์ริชอร์ส (.shrc หรือ .kshrc):

```
PSALLOC=early;export PSALLOC
```

รายการนี้ทำให้โปรเซสทั้งหมด ในล็อกอินเซสชันของคุณ ยกเว้นล็อกอินเชลล์ รันภายใต้ โหมดการจัดสรร early เมธอดนี้ยังป้องกันโปรเซสจาก กลไกสัญญาณ SIGKILL ด้วย

- แทรกรูทีนย่อย putenv ภายในโปรแกรมเพื่อตั้งค่าตัวแปร สถานะแวดล้อม PSALLOC เป็น early โดยใช้เมธอดนี้ ลักษณะการจัดสรร early จะมีผลในครั้งถัดไปที่เรียก รูทีนย่อย exec

หลักการที่เกี่ยวข้อง:

“การเปรียบเทียบการจัดสรรพื้นที่การเพจแบบ deferred และ early” ในหน้า 454

ระบบปฏิบัติการใช้ตัวแปรสถานะแวดล้อม PSALLOC เพื่อกำหนดกลไกที่ใช้สำหรับการจัดสรรหน่วยความจำและพื้นที่การเลื่อนหน้า

การเปลี่ยนหรือการลบพื้นที่การเลื่อนหน้า:

การเปลี่ยนพื้นที่การจัดเพจสามารถทำได้ง่ายโดยใช้ SMIT แต่การลบพื้นที่การจัดเพจมีความเสี่ยงมากกว่า

การเปลี่ยนลักษณะของพื้นที่การจัดเพจอาจทำได้โดยใช้พารามิเตอร์ SMIT ต่อไปนี้เป็นบรรทัดคำสั่ง: smit chps

โปรดระวัง การลบพื้นที่การเลื่อนหน้ามีความเสี่ยงมากกว่า โดยเฉพาะ ถ้าพื้นที่การเลื่อนหน้า ซึ่งคุณต้องการลบเป็นพื้นที่การเลื่อนหน้าดีฟอลต์ เช่น hd6 ต้องใช้โปรแกรมพิเศษสำหรับการลบพื้นที่การเลื่อนหน้าดีฟอลต์ เนื่องจากพื้นที่ดังกล่าว มีการเรียกใช้ในระหว่างเวลาบูตโดยเซลล์สคริปต์ที่กำหนดคอนฟิกระบบ เพื่อลบพื้นที่การเลื่อนหน้าดีฟอลต์อย่างใดอย่างหนึ่ง ต้องเปลี่ยนสคริปต์เหล่านี้ และต้องสร้างอิมเมจเปิดเครื่องใหม่ด้วย

ข้อควรสนใจ: การลบ พื้นที่การเลื่อนหน้าดีฟอลต์อย่างไม่ถูกต้องอาจทำให้ไม่สามารถรีสตาบ์ระบบได้ โปรดระวังต่อไปนี้จะใช้สำหรับผู้จัดการระบบที่มีประสบการณ์เท่านั้น

เมื่อต้องการ ลบพื้นที่การเลื่อนหน้าที่มีอยู่ให้ใช้โปรแกรมต่อไปนี้:

1. ด้วยสิทธิการใช้งานราก ให้ยกเลิกการเรียกใช้พื้นที่การเลื่อนหน้าโดยพิมพ์ พารามิเตอร์ SMIT ต่อไปนี้เป็นบรรทัดคำสั่ง:

```
smi swapoff
```

2. หากพื้นที่การเลื่อนหน้าซึ่งคุณกำลังจะลบเป็นอุปกรณ์ดัมพ์ดีฟอลต์ คุณต้องเปลี่ยนอุปกรณ์ดัมพ์ดีฟอลต์เป็นพื้นที่การเลื่อนหน้าหรือโลจิคัลวอลุ่มอื่น ก่อนการลบพื้นที่การเลื่อนหน้า เมื่อต้องการเปลี่ยนอุปกรณ์ดัมพ์ ดีฟอลต์ ให้พิมพ์คำสั่งต่อไปนี้:

```
sysdumpdev -P -p /dev/new_dump_device
```

3. ลบพื้นที่การเลื่อนหน้าโดยพิมพ์พาด่วนต่อไปนี้:

```
smit rmps
```

หลักการที่เกี่ยวข้อง:

“การแก้ไขปัญหาพื้นที่การเลื่อนหน้า” ในหน้า 461

ปัญหาที่พบบ่อยที่สุดเกี่ยวกับพื้นที่การเลื่อนหน้าเกิดขึ้นจากการขาด พื้นที่ว่างที่จัดสรร

การใช้อินเตอร์เฟซการเขียนโปรแกรมโหมดการจัดสรรพื้นที่การเลื่อนหน้า:

อินเตอร์เฟซการเขียนโปรแกรมที่ควบคุมโหมดการจัดสรรพื้นที่การเลื่อนหน้า ใช้ตัวแปรสถานะแวดล้อม *PSALLOC*

เพื่อให้มั่นใจว่าแอ็พพลิเคชันรันภายใต้โหมดที่ต้องการเสมอ (โดยมีหรือไม่มีการจัดสรรพื้นที่การเลื่อนหน้า early) ให้ทำตามต่อไปนี้:

1. ใช้ยูทิลิตี้ *getenv* เพื่อตรวจสอบสถานะปัจจุบัน ของตัวแปรสถานะแวดล้อม *PSALLOC*
2. ถ้าค่าของตัวแปรสถานะแวดล้อม *PSALLOC* ไม่ใช่ ค่าที่ต้องการโดยแอ็พพลิเคชัน ให้ใช้ยูทิลิตี้ *setenv* เพื่อเปลี่ยน ค่าของตัวแปรสถานะแวดล้อม เนื่องจากยูทิลิตี้ *execve* เป็นเพียงยูทิลิตี้เดียว ที่ตรวจสอบสถานะของตัวแปรสถานะแวดล้อม *PSALLOC* ให้เรียกยูทิลิตี้ *execve* พร้อมกับชุดเดียวกันของพารามิเตอร์ และสถานะแวดล้อมที่ได้รับโดยแอ็พพลิเคชัน เมื่อแอ็พพลิเคชันตรวจสอบสถานะของตัวแปรสถานะแวดล้อม *PSALLOC* อีกครั้งและ พบค่าที่ถูกต้อง แอ็พพลิเคชันจะทำต่อไปตามปกติ
3. ถ้ายูทิลิตี้ *getenv* แสดงว่าสถานะปัจจุบันของตัวแปร สถานะแวดล้อม *PSALLOC* ถูกต้อง ไม่จำเป็นต้องทำการแก้ไข แอ็พพลิเคชันทำตามปกติ

การเปลี่ยนที่ตั้งและการลดพื้นที่การเลื่อนหน้า hd6:

คุณอาจต้องการลดหรือย้ายพื้นที่การเลื่อนหน้าดีฟอลต์เพื่อ เพิ่มประสิทธิภาพระบบหน่วยเก็บ โดยการบังคับใช้การเลื่อนหน้า และการสลับไปยัง ดิสก์อื่นในระบบที่ยุ่่น้อยกว่า การลดหรือการย้ายการเลื่อนหน้าดีฟอลต์ ยังช่วยประหยัดพื้นที่ว่างดิสก์บน *hdisk0* ด้วย

ไม่ว่าจะเป็นการย้ายพื้นที่การเลื่อนหน้าหรือการลดขนาด วัตถุประสงค์ เหมือนกันคือ: การย้ายกิจกรรมพื้นที่การเลื่อนหน้าไปยังดิสก์ที่ยุ่่น้อยลง ค่าดีฟอลต์การติดตั้ง สร้างโลจิคัลวอลุ่มการเลื่อนหน้า (*hd6*) บนไดรฟ์ *hdisk0* ที่มี ระบบไฟล์ / (ราก) และ /usr ที่ยุ่่น บางส่วนหรือทั้งหมด ถ้าเลือกนโยบายการจัดสรรของดิสก์อินเทอร์ต่ำสุด นั่นหมายความว่า ทั้งหมดของ / และส่วน ใหญ่ของ /usr อยู่ บน *hdisk0* การย้ายพื้นที่การเลื่อนหน้าไปยังดิสก์ที่ยุ่่น้อยกว่าสามารถช่วยปรับปรุง ประสิทธิภาพได้เป็นอย่างมาก แม้ว่าใช้นโยบายการจัดสรรของดิสก์อินเทอร์สูงที่สุด และทั้ง / และ /usr มีการกระจาย บนหลายฟิสิคัลวอลุ่ม *hdisk2* (สมมติว่ามีสามดิสก์) อาจมี พาร์ติชันโลจิคัลจำนวนน้อยลงที่เป็นของระบบไฟล์ที่ยุ่่นที่สุด

โปรดดูต่อไปนี้อธิบายวิธีการทำให้พื้นที่การเลื่อนหน้า *hd6* เล็กลง และวิธีการย้ายพื้นที่การเลื่อนหน้า *hd6* ภายในกลุ่มวอลุ่มเดียวกัน

หลักการที่เกี่ยวข้อง:

“นโยบายการจัดสรรของดิสก์อินเทอร์” ในหน้า 445

นโยบายการจัดสรรของดิสก์อินเทอร์ระบุจำนวนดิสก์ ซึ่งพาร์ติชันฟิสิคัลของโลจิคัลวอลุ่มตั้งอยู่

“การแก้ไขปัญหาพื้นที่การเลื่อนหน้า” ในหน้า 461

ปัญหาที่พบบ่อยที่สุดเกี่ยวกับพื้นที่การเลื่อนหน้าเกิดขึ้นจากการขาด พื้นที่ว่างที่จัดสรร

การทำให้พื้นที่การเลื่อนหน้า *hd6* เล็กลง:

โพรซีเดอร์ทต่อไปนี้ใช้คำสั่ง **chps** เพื่อ ลดขนาดของพื้นที่การเลื่อนหน้าที่มีอยู่ รวมถึงพื้นที่การเลื่อนหน้าหลักและ อุปกรณ์ดัมพ์หลักและรอง

คำสั่ง **chps** เรียกสคริปต์ **shrinkps** ซึ่งจะลดขนาดพื้นที่การเลื่อนหน้าอย่าง ปลอดภัยโดยไม่ต้องออกจากระบบในสภาพที่บูตไม่ได้ โดยเฉพาะอย่างยิ่ง สคริปต์ทำดังต่อไปนี้:

1. สร้างพื้นที่การเลื่อนหน้าชั่วคราวในวอลุ่มเดียวกัน
2. ย้ายข้อมูลไปยังพื้นที่ว่างชั่วคราวนั้น
3. สร้างพื้นที่การเลื่อนหน้าใหม่ที่เล็กลงในวอลุ่มเดียวกัน
4. ลบพื้นที่การเลื่อนหน้าเก่า

เพื่อให้คำสั่ง **chps** เสร็จสมบูรณ์ ต้องมีพื้นที่ดิสก์ว่างอยู่อย่างเพียงพอ (พื้นที่ซึ่งไม่ได้จัดสรรให้กับโลจิกัลวอลุ่มใดๆ) เพื่อสร้างพื้นที่การเลื่อนหน้าชั่วคราว ขนาดของพื้นที่การเลื่อนหน้าชั่วคราว เท่ากับจำนวนของพื้นที่ซึ่งต้องใช้เพื่อจัดเก็บหน้าที่ **paged out** ทั้งหมดใน พื้นที่การเลื่อนหน้าเก่า ขนาดต่ำสุดสำหรับพื้นที่การเลื่อนหน้าหลักคือ 32 MB ขนาดต่ำสุดสำหรับพื้นที่การเลื่อนหน้าอื่นคือ 16 MB

หมายเหตุ: ถ้าโพรซีเดอร์ทต่อไปนี้พบข้อผิดพลาด I/O ระบบอาจต้องปิดทันที และรีบูต

1. ตรวจสอบโลจิกัลวอลุ่มและการกระจายระบบไฟล์บนฟิสิคัลวอลุ่ม โดยพิมพ์คำสั่งต่อไปนี้:

```
lspv -l hdiskX
```

โดยที่ *hdiskX* คือชื่อฟิสิคัลวอลุ่มของคุณ

2. เมื่อต้องการลดขนาดของพื้นที่การเลื่อนหน้า ให้พิมพ์ดังต่อไปนี้บนบรรทัด คำสั่ง:

```
smit chps
```

หมายเหตุ: พื้นที่การเลื่อนหน้าหลักมีการฮาร์ดโคัดอยู่ในบูตเรกคอร์ด ดังนั้น การสลับหน้าพื้นที่ หลักจะถูกเรียกทำงานเมื่อระบบรีสตาร์ทเสมอ คำสั่ง **chps** ไม่สามารถยกเลิกเรียกใช้พื้นที่การเลื่อนหน้าหลัก

มีการกำหนดลำดับความสำคัญ เพื่อรักษาคอนฟิกรูเรชันด้านการดำเนินงาน การตรวจสอบระบบอาจส่งผลให้เกิดการปฏิเสธการลดขนาดพื้นที่การเลื่อนหน้าในทันที มีข้อผิดพลาดเกิดขึ้นในขณะที่สร้างพื้นที่การเลื่อนหน้าชั่วคราวเนื่องจากโพรซีเดอร์ทออก และระบบจะแปลงเป็น ค่าติดตั้งดั้งเดิม ปัญหาอื่นอาจทำให้เกิดสถานการณ์ ซึ่งต้องได้รับการแก้ไขโดยผู้ดูแลระบบหรืออาจต้อง รีบูตในทันที ข้อผิดพลาดบางอย่างอาจทำให้ไม่สามารถลบพื้นที่การเลื่อนหน้าชั่วคราวออกได้ ซึ่งโดยปกติ ต้องได้รับการแก้ไขจากผู้ดูแลระบบในเวลาต่อมา

ข้อควรสนใจ: หากตรวจพบข้อผิดพลาด I/O บนระบบที่ backing หน้าหรือผู้ใช้ที่ backing หน้าโดยใช้คำสั่ง **swapoff** ภายในสคริปต์ **shrinkps** ขอแนะนำให้ปิดในทันทีเพื่อหลีกเลี่ยงความเสียหายต่อระบบที่อาจเกิดขึ้นได้ เมื่อรีบูต พื้นที่การเลื่อนหน้าชั่วคราวมีการใช้งานอยู่ และสามารถพยายามหยุดและ รีสตาร์ทแอฟพลิเคชันซึ่งพบข้อผิดพลาด I/O ได้ หากความพยายามนั้น สำเร็จและคำสั่ง **swapoff** สามารถทำการยกเลิกเรียกใช้ได้ โพรซีเดอร์ทลดขนาดสามารถทำให้เสร็จสมบูรณ์ได้ด้วยตนเองโดยใช้คำสั่ง **mkps**, **swapoff** และ **rmpps** เพื่อสร้างพื้นที่การเลื่อนหน้า ที่มีขนาดที่ต้องการ และเพื่อลบพื้นที่การเลื่อนหน้าชั่วคราว

อย่า พยายามลบ (โดยใช้ **rmpps**) หรือเรียกใช้ใหม่ (โดยใช้ **chps**) พื้นที่การเลื่อนหน้าซึ่งยกเลิกเรียกใช้ที่อยู่ในสถานะ I/O ERROR ก่อนระบบ รีสตาร์ท เนื่องจากมีความเสี่ยงที่พื้นที่ว่างดิสก์จะถูกนำมาใช้ใหม่และอาจส่งผลให้เกิด ปัญหาเพิ่มเติม

การย้ายพื้นที่การเลื่อนหน้า hd6 ภายในกลุ่มวอลุ่มเดียวกัน:

การย้ายพื้นที่การเลื่อนหน้าดีฟอลต์จาก hdisk0 ไปยังดิสก์อื่น ภายในกลุ่มวอลุ่มเดียวกันไม่จำเป็นต้องปิดและรีบูตระบบ

ด้วยสิทธิการใช้งานราก ให้พิมพ์คำสั่งต่อไปนี้เพื่อย้ายพื้นที่การเลื่อนหน้า ดีฟอลต์ (hd6) จาก hdisk0 ไปยัง hdisk2:

```
migratepv -l hd6 hdisk0 hdisk2
```

ข้อควรสนใจ: ไม่แนะนำให้ย้ายพื้นที่การเลื่อนหน้าที่ชื่อ hd6 จาก rootvg ไปยัง กลุ่มวอลุ่มอื่น เนื่องจากชื่อมีการฮาร์ดโค้ดในหลายตำแหน่ง รวมถึงระยะที่สองของโปรเซสบูตและโปรเซสที่เข้าถึง กลุ่มวอลุ่มรากเมื่อบูตจากสื่อบันทึกที่ถอดออกได้ เฉพาะพื้นที่การเลื่อนหน้า ใน rootvg เท่านั้นจะใช้งานอยู่ในระหว่างระยะที่สองของโปรเซสบูต และการไม่มีพื้นที่การเลื่อนหน้าใน rootvg อาจส่งผลกระทบต่อประสิทธิภาพการบูตของระบบอย่างรุนแรง ถ้าคุณต้องการให้พื้นที่การเลื่อนหน้าส่วนใหญ่อยู่บนกลุ่มวอลุ่มอื่น ควรทำให้ hd6 เล็กที่สุดเท่าที่เป็นไปได้ (ขนาดเดียวกับหน่วยความจำฟิสิคัล) จากนั้นสร้างพื้นที่การเลื่อนหน้าที่ใหญ่ขึ้นบนกลุ่มวอลุ่มอื่น

หลักการที่เกี่ยวข้อง:

“การเพิ่มและการเรียกใช้พื้นที่การเลื่อนหน้า” ในหน้า 457

เพื่อให้พื้นที่การเลื่อนหน้ามีอยู่ในระบบของคุณ คุณต้องเพิ่มและเรียกใช้พื้นที่การเลื่อนหน้า

การแก้ไขปัญหาพื้นที่การเลื่อนหน้า

ปัญหาที่พบบ่อยที่สุดเกี่ยวกับพื้นที่การเลื่อนหน้าเกิดขึ้นจากการขาด พื้นที่ว่างที่จัดสรร

จำนวนทั้งหมดของพื้นที่การเลื่อนหน้ามักถูกกำหนดโดยการทดลองและข้อผิดพลาด แนวทางที่ใช้กันทั่วไปอย่างหนึ่งคือการเพิ่มขนาด RAM เป็นสองเท่าและใช้ตัวเลขนั้นเป็นเป้าหมายพื้นที่การเลื่อนหน้า หากพื้นที่การเลื่อนหน้ารับเข้า โปรเซสอาจสูญหายได้ และถ้าขาดพื้นที่การเลื่อนหน้า ระบบอาจล้มเหลว ข้อมูลสัญญาณและข้อผิดพลาด ต่อไปนี้สามารถช่วยคุณมอนิเตอร์และแก้ไขหรือป้องกันปัญหา พื้นที่การเลื่อนหน้าได้

ระบบปฏิบัติการมอนิเตอร์จำนวนของบล็อกพื้นที่การเลื่อนหน้าที่ว่างและ ตรวจพบเมื่อมีการขาดแคลนพื้นที่การเลื่อนหน้า เมื่อจำนวนของบล็อกพื้นที่การเลื่อนหน้าที่ว่าง ลดลงต่ำกว่า threshold ที่เรียกว่า ระดับการเตือนพื้นที่การเลื่อนหน้า ระบบจะแจ้งโปรเซสทั้งหมด (ยกเว้น kprocs) ให้ทราบถึง สถานะนี้โดยการส่งสัญญาณ SIGDANGER หากการขาดแคลนยังคงมีอยู่ต่อไป และลดลงต่ำกว่า threshold ที่สองซึ่งเรียกว่า ระดับการทำลายพื้นที่การเลื่อนหน้า ระบบจะส่งสัญญาณ SIGKILL ไปยังโปรเซสที่เป็นผู้ใช้หลักของ พื้นที่การเลื่อนหน้าและที่ไม่มีตัวจัดการสัญญาณสำหรับสัญญาณ SIGDANGER (การดำเนินการดีฟอลต์สำหรับสัญญาณ SIGDANGER คือละเว้นสัญญาณ) ระบบทำการส่งสัญญาณ SIGKILL อย่างต่อเนื่องจนกระทั่งจำนวนของบล็อก พื้นที่การเลื่อนหน้าที่ว่างอยู่เหนือระดับการทำลายพื้นที่การเลื่อนหน้า

หมายเหตุ: ถ้าพารามิเตอร์ low_ps_handling มีการตั้งค่าเป็น 2 (ภายใต้คำสั่ง vmo) และไม่พบโปรเซสที่จะทำลาย (โดยไม่มีตัวจัดการสัญญาณ SIGDANGER) ระบบจะส่ง สัญญาณ SIGKILL ไปยังโปรเซสแรกสุดที่มีตัวจัดการสัญญาณ สำหรับสัญญาณ SIGDANGER

โปรเซสที่จัดสรรหน่วยความจำแบบไดนามิกสามารถทำให้มั่นใจได้ว่ามี พื้นที่การเลื่อนหน้าที่เพียงพอ โดยการมอนิเตอร์ระดับพื้นที่การเลื่อนหน้าด้วยรูทีนย่อย psdanger หรือโดยใช้รูทีนการจัดสรรพิเศษ คุณสามารถใช้รูทีนย่อย disclaim เพื่อป้องกันไม่ให้โปรเซสสิ้นสุดลงเมื่อถึงระดับการทำลายพื้นที่การเลื่อนหน้า เมื่อต้องการทำเช่นนี้ ให้กำหนดตัวจัดการสัญญาณสำหรับสัญญาณ SIGDANGER และรีลีส หน่วยความจำและรีซอร์สพื้นที่การเลื่อนหน้าที่จัดสรรในข้อมูลและพื้นที่ที่สแต็ก และในเช็กเมนต์หน่วยความจำแบบแบ่งใช้

หากคุณได้รับข้อความแสดงข้อผิดพลาดคล้ายกับดังต่อไปนี้ ให้เพิ่ม พื้นที่การเลื่อนหน้า:

INIT: Paging space is low!

หรือ

You are close to running out of paging space.
You may want to save your documents because
this program (and possibly the operating system)
could terminate without future warning when the
paging space fills up.

หลักการที่เกี่ยวข้อง:

“การเปลี่ยนที่ตั้งและการลดพื้นที่การเลื่อนหน้า hd6” ในหน้า 459
คุณอาจต้องการลดหรือย้ายพื้นที่การเลื่อนหน้าดีฟอลต์เพื่อเพิ่มประสิทธิภาพระบบหน่วยเก็บ โดยการบังคับใช้การเลื่อนหน้า
และการสลับไปยัง ดิสก์อื่นในระบบที่ยุ่งน้อยกว่า การลดหรือการย้ายการเลื่อนหน้าดีฟอลต์ยังช่วยประหยัดพื้นที่วางดิสก์บน
hdisk0 ด้วย

“แนวคิดพื้นที่การเลื่อนหน้า” ในหน้า 453

พื้นที่การเพจ เป็นชนิดของโลจิคัลคลุ่มที่มีพื้นที่ดิสก์ที่จัดสรรไว้ที่เก็บข้อมูล ซึ่งอยู่ในหน่วยความจำเสมือนแต่ไม่ถูกเข้าถึงใน
ปัจจุบัน

งานที่เกี่ยวข้อง:

“การเปลี่ยนหรือการลบพื้นที่การเลื่อนหน้า” ในหน้า 458

การเปลี่ยนพื้นที่การจัดเพจสามารถทำได้ง่ายโดยใช้ SMIT แต่การลบพื้นที่การจัดเพจมีความเสี่ยงมากกว่า

Virtual Memory Manager

Virtual Memory Manager (VMM) จัดการคำขอหน่วยความจำที่กระทำโดยระบบและแอปพลิเคชัน

เซ็กเมนต์หน่วยความจำเสมือนมีการแบ่งพาร์ติชันออกเป็นหน่วยที่เรียกว่า หน้า โดยแต่ละ หน้าอยู่ใน real physical memory
(RAM) หรือจัดเก็บไว้บนดิสก์จนกว่า ต้องการใช้ AIX ใช้หน่วยความจำ เสมือนเพื่อระบุหน่วยความจำมากกว่าที่มีอยู่ทาง
กายภาพในระบบ การจัดการหน้าหน่วยความจำใน RAM หรือบนดิสก์ มีการจัดการโดย VMM

การจัดการหน่วยความจำแบบจริงใน Virtual Memory Manager:

ใน AIX เซ็กเมนต์หน่วยความจำเสมือน มีการแบ่งพาร์ติชันเป็นหน่วยขนาด 4096-ไบต์ที่เรียกว่าหน้า หน่วยความจำแบบจริง
ถูกแบ่งออกเป็นกรอบหน้าขนาด 4096-ไบต์

VMM มีฟังก์ชันที่สำคัญสองอย่างคือ:

- จัดการกับการจัดสรรกรอบหน้า
- แก้ไขการอ้างอิงไปยังหน้าหน่วยความจำเสมือนซึ่งไม่มีอยู่ในปัจจุบันใน RAM (ที่จัดเก็บไว้ในพื้นที่การเลื่อนหน้า) หรือยัง
ไม่มีอยู่

เพื่อให้ฟังก์ชันเหล่านี้สำเร็จ VMM จะรักษา *รายการอิสระ* ของ กรอบหน้าที่พร้อมใช้งานได้ VMM ยังใช้ขั้นตอนวิธีการแทนที่
หน้าเพื่อกำหนด หน้าหน่วยความจำเสมือนใน RAM ในปัจจุบันซึ่งจะมีการกำหนด กรอบหน้าให้อีกครั้งในรายการอิสระ ชั้น
ตอนวิธีการแทนที่หน้าพิจารณาถึง การมีอยู่ของเซ็กเมนต์ถาวรและเซ็กเมนต์การทำงาน, repaging, และ VMM thresholds

รายการอิสระ Virtual Memory Manager:

VMM เก็บรักษารายการของกรอบหน้าอิสระ (ไม่ได้จัดสรร) ที่ VMM ใช้สำหรับความผิดพลาดของเพจ

AIX พยายามใช้ RAM ทั้งหมดตลอดเวลา ยกเว้นจำนวนเล็กน้อยซึ่งระบบเก็บรักษาไว้บนรายการอิสระ เพื่อรักษาหน้าที่ไม่ได้จัดสรรจำนวนเล็กน้อยนี้ VMM จะใช้ *page outs* และ *page steals* เพื่อปล่อยพื้นที่ว่างและกำหนด กรอบหน้าที่ดังกล่าวอีกครั้งให้กับรายการอิสระ หน้าหน่วยความจำเสมือนที่เป็นเจ้าของกรอบหน้าที่จะถูกกำหนดใหม่ ถูกเลือกโดยใช้ขั้นตอนวิธีการเปลี่ยนหน้าของ VMM

เช็คเมนต์หน่วยความจำแบบถาวรหรือการทำงานใน Virtual Memory Manager:

AIX แยกความแตกต่าง ระหว่างเช็คเมนต์หน่วยความจำชนิดต่างๆ ในการทำความเข้าใจกับ VMM สิ่งสำคัญคือ การเข้าใจถึงความแตกต่างระหว่างเช็คเมนต์การทำงานและแบบถาวร

เช็คเมนต์แบบถาวร มีที่ตั้งหน่วยเก็บแบบถาวรบนดิสก์ ไฟล์ที่มีข้อมูลหรือโปรแกรมปฏิบัติการจะถูกแม็พเข้ากับเช็คเมนต์แบบถาวร เมื่อเปิดและเข้าถึงไฟล์ JFS หรือ JFS2 จะมีการคัดลอกข้อมูลไฟล์เข้าไปใน RAM พารามิเตอร์ VMM ควบคุมเวลาที่กรอบหน่วยความจำฟิสิกส์ซึ่งจัดสรรให้กับ หน้าแบบถาวรควรจะถูกเขียนทับและใช้เพื่อจัดเก็บข้อมูลอื่น

เช็คเมนต์การทำงาน เป็นเช็คเมนต์ชั่วคราวและมีอยู่ในระหว่างการใช้งาน โดยโปรเซสเท่านั้น เช็คเมนต์การทำงานไม่มีที่ตั้งหน่วยเก็บดิสก์ถาวร โปรเซสสแต็ก และพื้นที่ข้อมูลจะแม็พเข้ากับเช็คเมนต์การทำงานและเช็คเมนต์ข้อความไลบรารี แบบแบ่งใช้ หน้าของเช็คเมนต์การทำงานยังต้องใช้ที่ตั้งหน่วยเก็บดิสก์ เมื่อไม่สามารถเก็บไว้ในหน่วยความจำแบบจริง พื้นที่การเลื่อนหน้าดิสก์ใช้สำหรับ วัตถุประสงค์นี้ เมื่อโปรแกรมออก หน้าการทำงานทั้งหมดของโปรแกรมจะถูกวางกลับ ไปในรายการอิสระโดยทันที

เช็คเมนต์การทำงานและพื้นที่การเลื่อนหน้าใน Virtual Memory Manager:

หน้าการทำงานใน RAM ที่สามารถแก้ไขและ *paged out* ได้รับการกำหนด สล็อตที่สอดคล้องกันในพื้นที่การเลื่อนหน้า

พื้นที่การเลื่อนหน้าที่จัดสรรไว้เฉพาะถ้าหน้าต้อง ถูก *paged out* อย่างไรก็ตาม หน้าจัดสรรในพื้นที่การเลื่อนหน้าไม่สามารถใช้โดยหน้าอื่น และยังคงสงวนไว้สำหรับหน้าเฉพาะตรงเท่าที่มีหน้านั้น อยู่ในหน่วยความจำเสมือน เนื่องจากหน้าแบบถาวรถูก *paged out* ไปยังที่ตั้งเดียวกันบนดิสก์ซึ่งหน้าถูกดึงมา จึงไม่จำเป็นต้องจัดสรรพื้นที่การเลื่อน หน้าสำหรับหน้าแบบถาวรที่อยู่ใน RAM

VMM มีสองโหมดสำหรับการจัดสรรพื้นที่การเลื่อนหน้า: *early* และ *late* นโยบายการจัดสรร *early* สงวนพื้นที่การเลื่อนหน้าใน ทุกเมื่อที่จัดทำคำร้องขอ หน่วยความจำสำหรับหน้าการทำงาน นโยบายการจัดสรร *late* กำหนดพื้นที่การเลื่อนหน้าเมื่อ หน้าการทำงานถูก *paged out* จริงจากหน่วยความจำเท่านั้น ซึ่งลด ความต้องการพื้นที่การเลื่อนหน้าของระบบเป็นอย่างมาก

สิ่งอำนวยความสะดวกในการควบคุมไหลหน่วยความจำ Virtual Memory Manager:

เมื่อโปรเซสอ้างอิงหน้าหน่วยความจำเสมือนที่อยู่บนดิสก์ เนื่องจาก *paged out* หรือไม่เคยอ่าน หน้าอ้างอิงต้อง ถูก *paged in* และนี้อาจทำให้หน้าตั้งแต่หนึ่งหน้าขึ้นไปถูก *paged out* ถ้าจำนวนของกรอบหน้าที่มีอยู่ (ว่าง) ต่ำ VMM พยายาม ขโมยกรอบหน้าที่ยังไม่ได้อ้างอิงเมื่อเร็ว ๆ นี้และ ไม่มีแนวโน้มที่จะถูกอ้างอิงในอนาคตอันใกล้ โดยใช้ขั้นตอนวิธี การแทนที่หน้า

การแทนที่หน้าที่สำเร็จจะเก็บหน้าหน่วยความจำของโปรเซสที่ใช้งานอยู่ ทั้งหมดในปัจจุบันไว้ใน RAM ในขณะที่หน้าหน่วยความจำของโปรเซสที่ไม่ได้ใช้งานอยู่ถูก *paged out* อย่างไรก็ตาม เมื่อ RAM over-committed การเลือกหน้าสำหรับ *page out* จะทำได้ยาก เนื่องจากอาจจะถูกอ้างอิงในอนาคตอันใกล้ โดยกระบวนการที่กำลังรันในปัจจุบัน ผลคือหน้านั้น ซึ่งอาจถูกอ้างอิงในไม่ช้าอาจยังคงถูก *paged out* แล้ว *paged in* อีกครั้งเมื่อมีการอ้างอิงจริง เมื่อ RAM over-committed อาจเกิดการ *paging in* และ *paging out* อย่างต่อเนื่องที่เรียกว่า *การเรซ* เมื่อระบบ เรซ ระบบใช้เวลาส่วนใหญ่ไปกับ *paging in* และ *paging out* แทน

การดำเนินการคำสั่งที่มีประโยชน์ และไม่มีโปรเซสที่ใช้งานอยู่ใด คีบหน้าอย่างเห็นได้ชัด VMM มีขั้นตอนวิธีการควบคุมโหลด หน่วยความจำ ที่ตรวจพบเมื่อระบบเธรช จากนั้นพยายามแก้ไขเงื่อนไข

ระบบไฟล์

ระบบไฟล์คือโครงสร้างตามลำดับชั้น (แผนผังไฟล์) ของ ไฟล์และไดเรกทอรี

โครงสร้างชนิดนี้คล้ายกับแผนผังผกผันซึ่งมีรากอยู่ที่ ด้านบนสุดและสาขาอยู่ที่ด้านล่าง แผนผังไฟล์นี้ใช้ไดเรกทอรีเพื่อจัดระเบียบ ข้อมูลและโปรแกรมเป็นกลุ่ม เพื่อให้สามารถจัดการกับหลายไดเรกทอรี และไฟล์ในครั้งเดียวได้

ระบบไฟล์ตั้งอยู่บนโลจิคัลวอลุ่มเดี่ยว ทุกไฟล์และไดเรกทอรี เป็นสมาชิกของระบบไฟล์ภายในโลจิคัลวอลุ่ม เนื่องจากโครงสร้าง การกิจบางส่วนจึงมีการทำอย่างมีประสิทธิภาพมากกว่าบนระบบไฟล์เมื่อเทียบกับ การทำบนแต่ละไดเรกทอรีภายในระบบไฟล์ ตัวอย่างเช่น คุณสามารถทำสำเนาสำรอง ย้าย หรือรักษาความปลอดภัยทั้งระบบไฟล์ได้ คุณสามารถจัดทำ อิมเมจแบบ point-in-time ของระบบไฟล์ JFS หรือระบบไฟล์ JFS2 ที่เรียกว่า *สแน็ปช็อต*

หมายเหตุ: จำนวนสูงสุดของพาร์ติชันโลจิคัลต่อโลจิคัลวอลุ่มคือ 32,512 สำหรับข้อมูลเพิ่มเติมเกี่ยวกับลักษณะโลจิคัลวอลุ่มของระบบไฟล์ ใหญ่ที่คำสั่ง `chlv`

คำสั่ง `mkfs` (make file system) หรือ System Management Interface Tool (คำสั่ง `smit`) สร้างระบบไฟล์ บนโลจิคัลวอลุ่ม

เพื่อให้เข้าถึงได้ ต้องติดตั้งระบบไฟล์บนจุดต่อ ไดเรกทอรี เมื่อติดตั้งหลายระบบไฟล์ โครงสร้างไดเรกทอรีจะถูกสร้างขึ้น เพื่อแสดงถึงรูปภาพของระบบไฟล์เดี่ยว นี่เป็นโครงสร้างตามลำดับชั้น ที่มีรากเดียว โครงสร้างนี้มีระบบไฟล์พื้นฐานและระบบไฟล์ใดๆ ซึ่งคุณสร้างขึ้น คุณสามารถเข้าถึงทั้งระบบไฟล์โลคัลและรีโมต โดยใช้คำสั่ง `mount` คำสั่งนี้ทำให้ระบบไฟล์ มีอยู่สำหรับสิทธิการอ่านและเขียนจากระบบของคุณ โดยปกติ การติดตั้งหรือการถอนติดตั้ง ระบบไฟล์จำเป็นต้องใช้ความเป็นสมาชิกของกลุ่มระบบ ระบบไฟล์สามารถมีการติดตั้ง โดยอัตโนมัติได้ ถ้ามีการกำหนดไว้ในไฟล์ `/etc/filesystems` คุณสามารถถอนติดตั้ง ระบบไฟล์โลคัลหรือรีโมตโดยใช้คำสั่ง `umount` ยกเว้นว่าผู้ใช้หรือโปรเซส กำลังเข้าถึงระบบไฟล์นั้น สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการติดตั้งระบบไฟล์ ใหญ่ที่ “การติดตั้ง” ในหน้า 493

ชนิดพื้นฐานของระบบไฟล์ที่ใช้โดย AIX เรียกว่า *journaled file system (JFS)* ระบบไฟล์นี้ใช้เทคนิคการเจอร์นัลฐานข้อมูล เพื่อรักษาความสอดคล้องกันของโครงสร้าง ซึ่งช่วยป้องกัน ไม่ให้ระบบไฟล์เสียหายเมื่อระบบหยุดอย่างผิดปกติ

ระบบปฏิบัติการ AIX สนับสนุน ระบบไฟล์หลายแบบรวมถึง journaled file system (JFS) และ enhanced journaled file system (JFS2) สำหรับข้อมูลเพิ่มเติมเกี่ยวกับชนิดของระบบไฟล์และลักษณะ ของแต่ละชนิด ใหญ่ที่ “ชนิดระบบไฟล์” ในหน้า 499

ภารกิจการจัดการระบบที่สำคัญที่สุดบางอย่างต้องทำกับระบบไฟล์ โดยเฉพาะอย่างยิ่ง:

- การจัดสรรพื้นที่ว่างสำหรับระบบไฟล์บนโลจิคัลวอลุ่ม
- การสร้างระบบไฟล์
- การทำให้พื้นที่ว่างระบบไฟล์มีอยู่สำหรับผู้ใช้ระบบ
- การมอนิเตอร์การใช้พื้นที่ว่างระบบไฟล์
- การทำสำเนาสำรองระบบไฟล์เพื่อป้องกันข้อมูลสูญหายถ้าระบบล้มเหลว
- การรักษาระบบไฟล์ให้อยู่ในสภาพที่สอดคล้องกัน

ภารกิจเหล่านี้ควรทำโดยผู้ดูแลระบบของคุณ

หลักการที่เกี่ยวข้อง:

“ระบบไฟล์” ในหน้า 429

โลจิสติกส์วอลุ่มกำหนดการจัดสรรพื้นที่ว่างดิสก์ไปยัง ระดับพาร์ติชันฟิสิคัล การจัดการข้อมูลในระดับที่ละเอียดมากขึ้นสำเร็จได้โดย คอมโพเนนต์ซอฟต์แวร์ระดับสูงเช่น Virtual Memory Manager หรือ ระบบไฟล์ ด้วยเหตุนี้ ขั้นตอนสุดท้ายในวิวัฒนาการของดิสก์คือ การสร้าง ระบบไฟล์

งานที่เกี่ยวข้อง:

“การจัดทำและการทำสำเนาสำรอง snapshot ของ JFS2” ในหน้า 39

คุณสามารถจัดทำ snapshot ของ JFS2 ที่ติดตั้งซึ่งสร้างรูปภาพระดับ บล็อกที่สอดคล้องกันของระบบไฟล์ ณ จุดเวลาหนึ่ง

“การจัดทำและการทำสำเนาสำรอง snapshot ภายนอกของ JFS2” ในหน้า 40

คุณสามารถจัดทำ snapshot ของ JFS2 ที่ติดตั้งซึ่งสร้างรูปภาพระดับ บล็อกที่สอดคล้องกันของระบบไฟล์ ณ จุดเวลาหนึ่ง

“การจัดทำและการทำสำเนาสำรอง snapshot ภายในของ JFS2” ในหน้า 41

คุณสามารถจัดทำ snapshot ของ JFS2 ที่ติดตั้งซึ่งสร้างรูปภาพระดับ บล็อกที่สอดคล้องกันของระบบไฟล์ ณ จุดเวลาหนึ่ง

แนวคิดระบบไฟล์

ก่อนคุณสามารถจัดการและกำหนดคอนฟิกูเรชันระบบไฟล์ของคุณได้ คุณต้อง เข้าใจการจัดระเบียบพื้นฐานและเนื้อหาของแผนผังไฟล์

การจัดระเบียบและเนื้อหาของแผนผังไฟล์:

แผนผังไฟล์จัดระเบียบไฟล์ออกเป็นไดเรกทอรีที่มีข้อมูล คล้ายกัน การจัดระเบียบนี้ช่วยสนับสนุนการติดตั้งไดเรกทอรีและไฟล์แบบรีโมต

ผู้ดูแลระบบสามารถใช้ไดเรกทอรีเหล่านี้เมื่อสร้างบล็อกเพื่อจัดสร้างแผนผัง ไฟล์เฉพาะสำหรับแต่ละไคลเอ็นต์ ซึ่งติดตั้งที่ละไดเรกทอรีจากหนึ่ง เซิร์ฟเวอร์ขึ้นไป การติดตั้งไฟล์และไดเรกทอรีแบบรีโมต แทนการเก็บข้อมูล ทั้งหมดแบบโลคัล มีข้อดีต่อไปนี้:

- สงวนพื้นที่ว่างดิสก์
- ช่วยให้สามารถจัดการระบบจากส่วนกลางได้ง่าย
- นำเสนอสถานะแวดล้อมที่ปลอดภัยมากขึ้น

แผนผังไฟล์มีลักษณะต่อไปนี้:

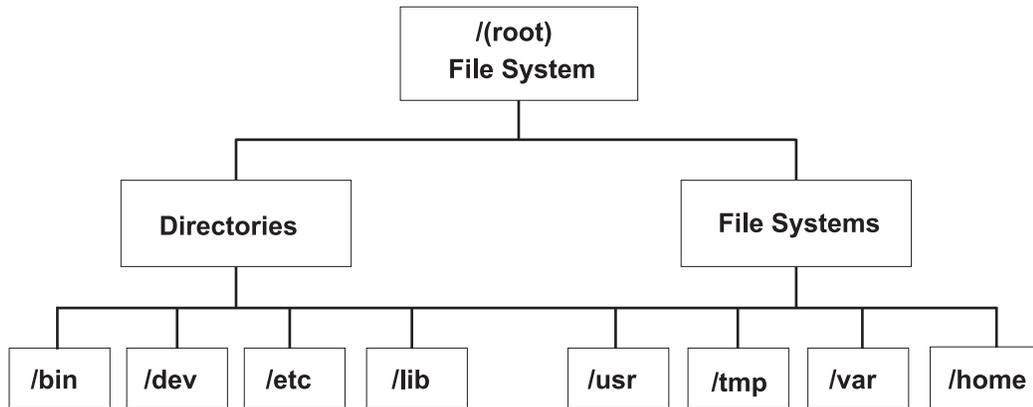
- ไฟล์ที่สามารถแบ่งใช้โดยเครื่องต่างๆ ที่มีสถาปัตยกรรมฮาร์ดแวร์เหมือนกัน ตั้งอยู่ในระบบไฟล์ /usr
- ไฟล์ตัวแปรสำหรับแต่ละไคลเอ็นต์ เช่น ไฟล์ spool และเมล ตั้งอยู่ในระบบไฟล์ /var
- ไฟล์ข้อความที่แบ่งใช้ได้และมีสถาปัตยกรรมอิสระ เช่น หน้าที่กำหนดเอง ตั้งอยู่ในไดเรกทอรี /usr/share
- ระบบไฟล์ / (ราก) มีไฟล์และไดเรกทอรีที่สำคัญมากสำหรับการดำเนินงานระบบ ตัวอย่างเช่น มีไดเรกทอรีอุปกรณ์ โปรแกรมที่ใช้สำหรับสตาร์ทอัพระบบ และจุดต่อซึ่งสามารถติดตั้งระบบไฟล์ บนระบบไฟล์รากได้
- ระบบไฟล์ /home คือจุดต่อสำหรับไดเรกทอรีโฮม ของผู้ใช้

โครงสร้างระบบไฟล์:

ข้อสำคัญคือการทำ ความเข้าใจกับความแตกต่างระหว่างระบบไฟล์ และไดเรกทอรี ระบบไฟล์คือส่วนของฮาร์ดดิสก์ที่ถูกจัดสรร เพื่อรองรับไฟล์ ฮาร์ดดิสก์ส่วนนี้มีการเข้าถึงโดยการติดตั้งระบบไฟล์ ไว้บนไดเรกทอรี หลังจากติดตั้งระบบไฟล์แล้ว ระบบไฟล์มีลักษณะคล้ายกับ ไดเรกทอรีอื่นๆ ของผู้ใช้ชั้นปลาย

อย่างไรก็ตาม เนื่องจากความแตกต่างทางโครงสร้างระหว่างระบบไฟล์ และไดเรกทอรี ข้อมูลภายในเอนทิตีเหล่านี้จึงสามารถจัดการแยกต่างหากกันได้

เมื่อติดตั้งระบบปฏิบัติการเป็นครั้งแรก ระบบปฏิบัติการจะถูกโหลด เข้าในโครงสร้างไดเรกทอรี ดังแสดงในภาพสาคิตต่อไปนี้



รูปที่ 7. แผนผังระบบไฟล์ / (ราก). แผนผังนี้แสดง โครงสร้างไดเรกทอรีที่มีระบบไฟล์ / (ราก) อยู่ที่ ด้านบนสุด และแตกสาขาย่อยลงมา เป็นไดเรกทอรีและระบบไฟล์ต่างๆ สาขาไดเรกทอรี เป็น /bin, /dev, /etc, และ /lib สาขา ระบบไฟล์เป็น /usr, /tmp, /var, และ /home

ไดเรกทอรีทางด้านขวา (/usr, /tmp, /var, และ /home) เป็นระบบไฟล์ทั้งหมด เพื่อให้มีส่วนแยกต่างหาก ของฮาร์ดดิสก์ที่จัดสรรสำหรับการใช้งาน ระบบไฟล์เหล่านี้มีการติดตั้ง โดยอัตโนมัติเมื่อระบบเริ่มต้นขึ้น ดังนั้นผู้ใช้ชั้นปลายจึงไม่เห็น ความแตกต่างระหว่างระบบไฟล์เหล่านี้กับไดเรกทอรีซึ่งแสดงรายการทางด้านซ้าย (/bin, /dev, /etc, และ /lib)

บนเครื่องแบบสแตนด์ออล ระบบไฟล์ต่อไปนี้ตั้งอยู่บน อุปกรณ์ที่เชื่อมโยงตามดีฟอลต์:

/Device	/File System
/dev/hd1	/home
/dev/hd2	/usr
/dev/hd3	/tmp
/dev/hd4	/(ราก)
/dev/hd9var	/var
/proc	/proc
/dev/hd10opt	/opt

แผนผังไฟล์มีลักษณะต่อไปนี้:

- ไฟล์ที่สามารถแบ่งใช้โดยเครื่องต่างๆ ที่มีสถาปัตยกรรมฮาร์ดแวร์เหมือนกัน ตั้งอยู่ในระบบไฟล์ /usr
- ไฟล์ตัวแปรสำหรับแต่ละโหนด ตัวอย่างเช่น ไฟล์ spool และเมล ตั้งอยู่ในระบบไฟล์ /var
- ระบบไฟล์ / (ราก) มีไฟล์และไดเรกทอรี ที่สำคัญมากสำหรับการดำเนินงานระบบ ตัวอย่างเช่น มี
 - ไดเรกทอรีอุปกรณ์ (/dev)
 - จุดต่อซึ่งสามารถติดตั้งระบบไฟล์บนระบบไฟล์ราก ตัวอย่างเช่น /mnt

- ระบบไฟล์ /home คือจุดต่อสำหรับไดเรกทอรีโฮมของผู้ใช้
- สำหรับเซิร์ฟเวอร์ไดเรกทอรี /export มีไฟล์พื้นที่การเลื่อนหน้า, ระบบไฟล์รากสำหรับแต่ละไคลเอ็นต์ (ไม่แบ่งใช้), ดัมพ์, โฮม, และไดเรกทอรี /usr/share สำหรับไคลเอ็นต์ที่ไม่มีดิสก์ และไดเรกทอรี /usr ที่เอ็กซ์พอร์ต
- ระบบไฟล์ /proc มีข้อมูลเกี่ยวกับสถานะของโปรเซสและเธรดในระบบ
- ระบบไฟล์ /opt มีซอฟต์แวร์ที่เป็นทางเลือก เช่น แอปพลิเคชัน

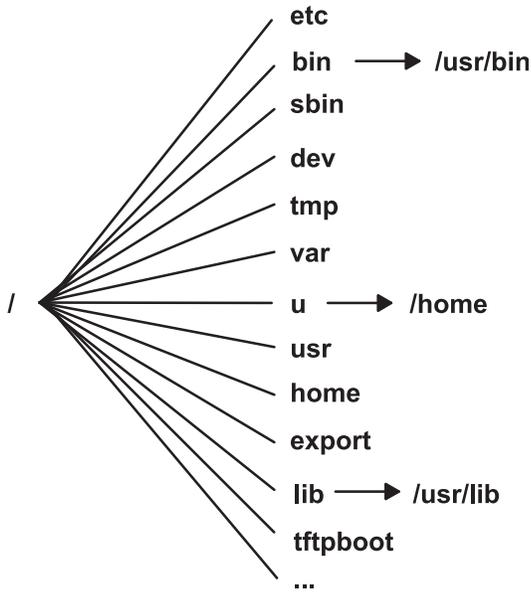
รายการต่อไปนี้จะแสดงข้อมูลเกี่ยวกับเนื้อหาของไดเรกทอรีย่อยบางส่วนของระบบไฟล์ / (ราก)

ไอเท็ม	คำอธิบาย
/bin	ลิงก์สัญลักษณ์ไปยังไดเรกทอรี /usr/bin
/dev	มีโหนดอุปกรณ์สำหรับไฟล์พิเศษของอุปกรณ์โลคัล ไดเรกทอรี /dev มีไฟล์พิเศษสำหรับเทปไดรฟ์ เครื่องพิมพ์ ดิสก์พาร์ติชัน และเทอร์มินัล
/etc	มีไฟล์คอนฟิกูเรชันที่แตกต่างกันไปในแต่ละเครื่อง ตัวอย่างรวมถึง: <ul style="list-style-type: none"> • /etc/hosts • /etc/passwd
/export	มีไดเรกทอรีและไฟล์บนเซิร์ฟเวอร์ที่ใช้สำหรับรีโมตไคลเอ็นต์
/home	ทำหน้าที่เป็นจุดต่อสำหรับระบบไฟล์ที่มีไดเรกทอรีโฮมของผู้ใช้ ระบบไฟล์ /home มีไฟล์และไดเรกทอรีสำหรับแต่ละผู้ใช้ ในเครื่องสแตนด์อะโลน ระบบไฟล์โลคัลแยกต่างหาก มีการติดตั้งบนไดเรกทอรี /home ในเครือข่าย เซิร์ฟเวอร์อาจมีไฟล์ผู้ใช้ที่ควรเข้าถึงได้จากหลายเครื่อง ในกรณีนี้ สำเนาของเซิร์ฟเวอร์ของไดเรกทอรี /home มีการติดตั้งบนรีโมตบนระบบไฟล์ /home โลคัล
/lib	ลิงก์สัญลักษณ์ไปยังไดเรกทอรี /usr/lib ซึ่งมีไลบรารีสถาปัตยกรรมอิสระที่มีชื่อในรูปแบบ lib*.a
/sbin	มีไฟล์ที่จำเป็นในการบูตเครื่องและติดตั้งระบบไฟล์ /usr คำสั่งส่วนใหญ่ที่ใช้ในระหว่างบูตมาจากระบบไฟล์ดิสก์ RAM ของ อิมเมจเปิดเครื่องใหม่ ดังนั้นจึงมีคำสั่งอยู่ในไดเรกทอรี /sbin น้อยมาก
/tmp	ทำหน้าที่เป็นจุดต่อสำหรับระบบไฟล์ที่มีไฟล์ชั่วคราวซึ่ง ระบบสร้างขึ้น
/u	ลิงก์สัญลักษณ์ไปยังไดเรกทอรี /home
/usr	ทำหน้าที่เป็นจุดต่อสำหรับระบบไฟล์ที่มีไฟล์ซึ่งไม่เปลี่ยนแปลงและสามารถแบ่งใช้โดยเครื่องต่างๆ (เช่น โปรแกรมปฏิบัติการ และเอกสารคู่มือ ASCII) เครื่องสแตนด์อะโลนติดตั้งระบบไฟล์โลคัล แยกต่างหากบนไดเรกทอรี /usr เครื่องที่ไม่มีดิสก์และที่ดิสก์ไม่ติดตั้งไดเรกทอรีจากรีโมตเซิร์ฟเวอร์บนระบบไฟล์ /usr
/var	ทำหน้าที่เป็นจุดต่อสำหรับไฟล์ที่แตกต่างกันในแต่ละเครื่อง ระบบไฟล์ /var มีการกำหนดคอนฟิกเป็นระบบไฟล์เนื่องจากไฟล์ที่มีในระบบนั้น มีแนวโน้มที่จะเติบโตขึ้น ตัวอย่างเช่น เป็นลิงก์สัญลักษณ์ไปยังไดเรกทอรี /usr/tmp ซึ่งมีไฟล์ทำงานชั่วคราว

ระบบไฟล์ราก:

ระบบไฟล์รากอยู่ที่ด้านบนสุดของแผนผังไฟล์ตามลำดับชั้น ระบบไฟล์รากมีไฟล์และไดเรกทอรีที่สำคัญมากต่อการดำเนินงานระบบ รวมถึง ไดเรกทอรีอุปกรณ์และโปรแกรมสำหรับการบูตระบบ ระบบไฟล์ราก ยังมีจุดต่อซึ่งสามารถติดตั้งระบบไฟล์เพื่อเชื่อมต่อกับ ลำดับชั้นระบบไฟล์รากได้

แผนภาพต่อไปนี้จะแสดงไดเรกทอรีย่อยจำนวนมากของระบบไฟล์ ราก



รูปที่ 8. ระบบไฟล์ราก. แผนภาพนี้แสดงระบบไฟล์รากและไดเรกทอรีย่อยของระบบไฟล์ราก ไดเรกทอรีย่อย /bin ซึ่งไปยังไดเรกทอรี /usr/bin ไดเรกทอรีย่อย /lib ซึ่งไปยังไดเรกทอรี /usr/lib ไดเรกทอรีย่อย /u ซึ่งไปยังไดเรกทอรี /home

รายการต่อไปนี้แสดงข้อมูลเกี่ยวกับเนื้อหาของไดเรกทอรีย่อยบางส่วนของระบบไฟล์ / (ราก)

ไอเท็ม	คำอธิบาย
/etc	มีไฟล์คอนฟิกูเรชันที่แตกต่างกันไปในแต่ละเครื่อง ตัวอย่างรวมถึง: <ul style="list-style-type: none"> • /etc/hosts • /etc/passwd <p>ไดเรกทอรี /etc มีไฟล์ที่โดยทั่วไปแล้วใช้ในการจัดการระบบ คำสั่งส่วนใหญ่ที่ก่อนหน้านี้ตั้งอยู่ใน ไดเรกทอรี /etc ขณะนี้ตั้งอยู่ใน ไดเรกทอรี /usr/sbin อย่างไรก็ตาม เพื่อให้เข้ากันได้ ไดเรกทอรี /usr/sbin จึงมีลิงก์สัญลักษณ์ไปยังที่ตั้งของไฟล์ปฏิบัติการบางไฟล์ ตัวอย่างรวมถึง:</p> <ul style="list-style-type: none"> • /etc/chown เป็นลิงก์สัญลักษณ์ไปยัง /usr/bin/chown • /etc/exportvg เป็นลิงก์สัญลักษณ์ไปยัง /usr/sbin/exportvg
/bin	ลิงก์สัญลักษณ์ไปยังไดเรกทอรี /usr/bin ในระบบไฟล์ UNIX ก่อนหน้านี้ ไดเรกทอรี /bin มีคำสั่งผู้ใช้ซึ่งในขณะนี้ตั้งอยู่ในไดเรกทอรี /usr/bin
/sbin	มีไฟล์ที่จำเป็นในการบูตเครื่องและติดตั้งระบบไฟล์ /usr คำสั่งส่วนใหญ่ที่ใช้ในระหว่างบูตมาจากระบบไฟล์ดิสก์ RAM ของ อิมเมจ เปิดเครื่องใหม่ ดังนั้นจึงมีคำสั่งอยู่ในไดเรกทอรี /sbin น้อยมาก
/dev	มีโหนดอุปกรณ์สำหรับไฟล์พิเศษของอุปกรณ์ไอซีแอล ไดเรกทอรี /dev มีไฟล์พิเศษสำหรับเทปไดรฟ์ เครื่องพิมพ์ ดิสก์พาร์ติชัน และเทอร์มินัล
/tmp	ทำหน้าที่เป็นจุดต่อสำหรับระบบไฟล์ที่มีไฟล์ชั่วคราวซึ่งระบบสร้างขึ้น ระบบไฟล์ /tmp เป็นไดเรกทอรีว่าง
/var	ทำหน้าที่เป็นจุดต่อสำหรับไฟล์ที่แตกต่างกันในแต่ละเครื่อง ระบบไฟล์ /var มีการกำหนดคอนฟิกเป็นระบบไฟล์เนื่องจากไฟล์ที่มีในระบบนั้น มีแนวโน้มที่จะเติบโตขึ้น โปรดดู “ระบบไฟล์ /var” ในหน้า 471 สำหรับข้อมูลเพิ่มเติม
/u	ลิงก์สัญลักษณ์ไปยังไดเรกทอรี /home
/usr	มีไฟล์ซึ่งไม่เปลี่ยนแปลงและสามารถแบ่งใช้โดยเครื่องต่างๆ เช่น โปรแกรมปฏิบัติการ และเอกสารคู่มือ ASCII

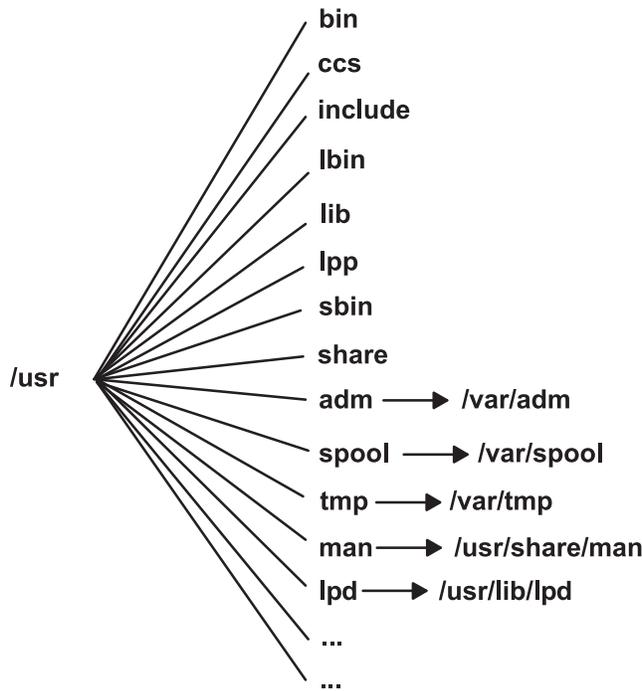
เครื่องสแตนด์ออลติดตั้ง รากของระบบไฟล์ไอซีแอลแยกต่างหากข้างบนไดเรกทอรี /usr เครื่องที่ไม่มีดิสก์และเครื่องที่มีดิสก์หรือส จำกัด ติดตั้งไดเรกทอรีจากรีโมตเซิร์ฟเวอร์บนระบบไฟล์ /usr ให้อูที่ “ระบบไฟล์ /usr” ในหน้า 469 สำหรับข้อมูลเพิ่มเติมเกี่ยวกับแผนผังไฟล์ ซึ่งติดตั้งบนไดเรกทอรี /usr

ไอเท็ม /home	คำอธิบาย ทำหน้าที่เป็นจุดต่อสำหรับระบบไฟล์ที่มีไดเรกทอรีโฮมของผู้ใช้ ระบบไฟล์ /home มีไฟล์และไดเรกทอรีสำหรับแต่ละผู้ใช้
	ในเครื่องสแตนด์อะโลน ไดเรกทอรี /home มีอยู่ใน ระบบไฟล์แยกต่างหากซึ่งมีรากที่ติดตั้งอยู่บนระบบไฟล์รากของไดเรกทอรี /home ในเครือข่าย เซิร์ฟเวอร์อาจมีไฟล์ผู้ใช้ที่ควรเข้าถึงได้จากหลายเครื่อง ในกรณีนี้ สำเนาเซิร์ฟเวอร์ของไดเรกทอรี /home มีการติดตั้งแบบรีโมตบนระบบไฟล์ /home โคลล์
/export	มีไดเรกทอรีและไฟล์บนเซิร์ฟเวอร์ที่ใช้สำหรับรีโมตโคลเอ็นต์
/lib /tftpboot	ให้ดูที่ “ไดเรกทอรี /export” ในหน้า 472 สำหรับข้อมูลเพิ่มเติม เกี่ยวกับแผนผังไฟล์ที่ตั้งอยู่ภายใต้ไดเรกทอรี /export ลิงก์สัญลักษณ์ไปยังไดเรกทอรี /usr/lib โปรดดู “ระบบไฟล์ /usr” สำหรับข้อมูลเพิ่มเติม มีอิมเมจเปิดเครื่องใหม่และข้อมูลบูตสำหรับโคลเอ็นต์ที่ไม่มีดิสก์

ระบบไฟล์ /usr:

ระบบไฟล์ /usr มีไฟล์ปฏิบัติการ ที่สามารถแบ่งใช้ระหว่างเครื่องต่างๆ

ไดเรกทอรีย่อยที่สำคัญของไดเรกทอรี /usr มีการแสดง ในแผนภาพต่อไปนี้



รูปที่ 9. ระบบไฟล์ /usr. แผนภาพนี้แสดงไดเรกทอรีย่อยที่สำคัญของไดเรกทอรี /usr ซึ่งรวมถึง: /bin, /ccs, /lib, /lpp, /adm และไดเรกทอรีย่อย /var/adm, และ /man และไดเรกทอรีย่อย /usr/share/man

บนเครื่องสแตนด์อะโลน ระบบไฟล์ /usr เป็น ระบบไฟล์แยกต่างหาก (ในโลจิคัลวอลุ่ม /dev/hd2) บนเครื่องที่ไม่มีดิสก์หรือเครื่องที่มีดิสก์รีซอร์สจำกัด ไดเรกทอรี จาการ์โมเตเซิร์ฟเวอร์มีการติดตั้งด้วยสิทธิอ่านอย่างเดียบบนระบบไฟล์ /usr โคลล์ระบบไฟล์ /usr มีคำสั่งอ่านอย่างเดียว โลบรารี และข้อมูล

ยกเว้นเนื้อหาของไดเรกทอรี /usr/share ไฟล์และไดเรกทอรีในระบบไฟล์ /usr สามารถแบ่งใช้โดยเครื่องทั้งหมดที่มีสถาปัตยกรรมฮาร์ดแวร์เหมือนกัน

ระบบไฟล์ /usr มีไดเรกทอรีต่อไปนี้:

ไอเท็ม	คำอธิบาย
/usr/bin	มีคำสั่งปกติและเชลล์สคริปต์ ตัวอย่างเช่น ไตเร็กทอรี /usr/bin มีคำสั่ง ls, cat, และ mkdir
/usr/ccs	มีไบนารีแพ็คเกจการพัฒนาที่ไม่ได้จัดกลุ่ม
/usr/include	มี include, หรือส่วนหัว, ไฟล์
/usr/libin	มีไฟล์ปฏิบัติการที่เป็น backends ของคำสั่ง
/usr/lib	มีไลบรารีสถาปัตยกรรมอิสระที่มีชื่อในรูปแบบ lib*.a ไตเร็กทอรี /lib ใน / (ราก) คือลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /usr/lib ดังนั้นไฟล์ทั้งหมดที่เคยอยู่ในไตเร็กทอรี /lib ขณะนี้อยู่ในไตเร็กทอรี /usr/lib ซึ่งมีไฟล์ที่ไม่ใช่ไลบรารีสองสามไฟล์เพื่อให้เข้ากันได้
/usr/lpp	มีผลิตภัณฑ์ที่ติดตั้งซึ่งเป็นทางเลือก
/usr/sbin	มียูทิลิตี้ที่ใช้ในการจัดการระบบ รวมถึงคำสั่ง System Management Interface Tool (SMIT) คำสั่งส่วนใหญ่ที่ก่อนหน้านี้ตั้งอยู่ใน ไตเร็กทอรี /etc ขณะนี้ตั้งอยู่ในไตเร็กทอรี /usr/sbin
/usr/share	มีไฟล์ที่สามารถแบ่งใช้ระหว่างเครื่องต่างๆ ซึ่งมีสถาปัตยกรรมต่างกัน โปรดดู “ไตเร็กทอรี /usr/share” สำหรับข้อมูลเพิ่มเติม

ข้อมูลต่อไปนี้เป็นลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /var:

ไอเท็ม	คำอธิบาย
/usr/adm	ลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /var/adm
/usr/mail	ลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /var/spool/mail
/usr/news	ลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /var/news
/usr/preserve	ลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /var/preserve
/usr/spool	ลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /var/spool
/usr/tmp	ลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /var/tmp เนื่องจาก ไตเร็กทอรี /usr มีโอกาสแบ่งใช้โดยโหนดจำนวนมาก และเป็นแบบอ่านอย่างเดียว

ข้อมูลต่อไปนี้เป็นลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /usr/share และ /usr/lib:

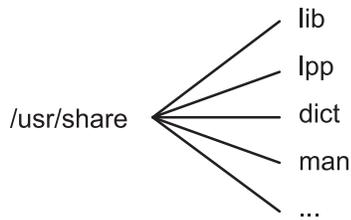
ไอเท็ม	คำอธิบาย
/usr/dict	ลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /usr/share/dict
/usr/man	ลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /usr/share/man
/usr/lpd	ลิงก์สัญลักษณ์ไปยังไตเร็กทอรี /usr/lib/lpd

ไตเร็กทอรี /usr/share:

ไตเร็กทอรี /usr/share มีไฟล์ข้อความสถาปัตยกรรมอิสระที่แบ่งใช้ได้ เนื้อหาของไตเร็กทอรีนี้สามารถแบ่งใช้ได้โดยเครื่องทุกเครื่อง โดยไม่คำนึงถึงสถาปัตยกรรมฮาร์ดแวร์

ในสภาวะแวดล้อมที่มีสถาปัตยกรรมหลายรูปแบบ โคลเอ็นต์ที่ไม่มีดิสก์ปกติจะ ติดตั้งไตเร็กทอรีเซิร์ฟเวอร์บนไตเร็กทอรี /usr ของตนเอง จากนั้น ติดตั้งไตเร็กทอรีอื่นบนไตเร็กทอรี /usr/share ไฟล์ข้างใต้ไตเร็กทอรี /usr/share มีอยู่ในแพ็คเกจติดตั้งได้แยกต่างหากตั้งแต่หนึ่งแพ็คเกจขึ้นไป ดังนั้น โหนดจึงอาจมีส่วนอื่น ของไตเร็กทอรี /usr ที่โหนดใช้เมื่อติดตั้งแบบโลคัล ในขณะที่ใช้เซิร์ฟเวอร์เพื่อจัดเตรียมไตเร็กทอรี /usr/share

บางไฟล์ในไตเร็กทอรี /usr/share มีไตเร็กทอรีและไฟล์แสดงในแผนภาพต่อไปนี้



รูปที่ 10. ไตเร็กทอรี /usr/share.

แผนภาพนี้แสดงหลายไตเร็กทอรีภายใต้ไตเร็กทอรี /usr/share รวมถึง /lib, /lpp, /dict, และ /man

ไตเร็กทอรี /usr/share มีดังต่อไปนี้:

ไอเท็ม	คำอธิบาย
/usr/share/man	มีหน้าคู่มือถ้ายังไม่ได้โหลด
/usr/share/dict	มีพจนานุกรมการสะกดและดัชนี
/usr/share/lib	มีไฟล์ข้อมูลสถาปัตยกรรมอิสระ รวมถึง terminfo, learn, tmac, me, และ macros
/usr/share/lpp	มีข้อมูลเกี่ยวกับผลิตภัณฑ์ที่สามารถเลือกติดตั้งได้ บนระบบ

ระบบไฟล์ /var:

ระบบไฟล์ /var มีแนวโน้มที่จะเติบโตขึ้นเนื่องจาก มีไตเร็กทอรีย่อยและไฟล์ข้อมูลที่ใช้โดยแอปพลิเคชันซึ่งมีงานมาก เช่น การบัญชีเมล และ spooler การพิมพ์

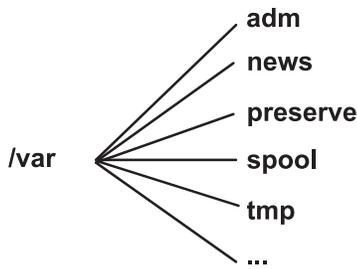
ข้อควรสนใจ: หากแอปพลิเคชันบนระบบของคุณใช้ระบบไฟล์ /var อย่างกว้างขวาง ให้รันคำสั่ง `skulker` เป็นประจำหรือเพิ่มขนาดของระบบไฟล์ให้มากกว่าค่าดีฟอลต์จำนวน 4MB ของ /var

ไฟล์ /var เฉพาะที่รับประกันการมอนิเตอร์เป็นครั้งคราว คือ /var/adm/wtmp และ /var/adm/ras/errlog

ไฟล์ /var อื่นที่จะมอนิเตอร์คือ:

ไอเท็ม	คำอธิบาย
/var/adm/ras/trcfile	ถ้าเปิดสิ่งอำนวยความสะดวกในการติดตาม
/var/tmp/snmpd.log	ถ้าคำสั่ง <code>snmpd</code> กำลังรันบนระบบของคุณ

แผนภาพไตเร็กทอรี /var แสดงบางไตเร็กทอรีในระบบไฟล์ /var



รูปที่ 11. ไดร็อกทอรี /var. แผนภาพนี้แสดงไดร็อกทอรีย่อยที่สำคัญของไดร็อกทอรี /var รวมถึง /adm, /news, /preserve, /spool, และ /tmp

ไอเท็ม	คำอธิบาย
/var/adm	มีไฟล์การบันทึกและการบัญชีของระบบ
/var/news	มีข่าวระบบ
/var/preserve	มีข้อมูลที่สงวนไว้จากเซสชันแก้ไขที่ถูกขัดจังหวะ ซึ่งคล้ายกับ ไดร็อกทอรี /usr/preserve ในรีลีสก่อนหน้า
/var/spool	มีไฟล์ที่กำลังประมวลผลโดยโปรแกรมต่างๆ เช่น อิเล็กทรอนิกส์เมล ซึ่งคล้าย กับไดร็อกทอรี /usr/spool ในรีลีสก่อนหน้า
/var/tmp	มีไฟล์ชั่วคราว ซึ่งคล้ายกับไดร็อกทอรี /usr/tmp ในรีลีสก่อนหน้า ขณะนี้ ไดร็อกทอรี /usr/tmp เป็น ลิงก์สัญลักษณ์ไปยัง /var/tmp

ไดร็อกทอรี /export:

ไดร็อกทอรี /export มีไฟล์เซิร์ฟเวอร์ ที่เอ็กซ์พอร์ตไปยังไคลเอ็นต์ เช่น เครื่องที่ไม่มีดิสก์ ไม่มีข้อมูล หรือดิสก์ไม่ดี

เซิร์ฟเวอร์สามารถเอ็กซ์พอร์ตพื้นที่ว่างดิสก์ได้หลายชนิด รวมถึงแฟกเกจของ โปรแกรมปฏิบัติการ พื้นที่การเลื่อนหน้าสำหรับ ไคลเอ็นต์ที่ไม่มีดิสก์ และระบบไฟล์ รากสำหรับไคลเอ็นต์ที่ไม่มีดิสก์หรือที่มีดิสก์รีซอร์สต่ำ ที่ตั้งมาตรฐาน สำหรับพื้นที่ว่างดิสก์ดังกล่าวในแผนผังไฟล์คือไดร็อกทอรี /export บางไดร็อกทอรีย่อยของไดร็อกทอรี /export มีการแสดง ในรายการต่อไปนี้:

- /exec มีไดร็อกทอรีที่ไคลเอ็นต์ที่ไม่มีดิสก์ติดตั้งบนระบบไฟล์ /usr
- /swap มีไฟล์สำหรับการเลื่อนหน้าแบบรีโมตของไคลเอ็นต์ที่ไม่มีดิสก์
- /share มีไดร็อกทอรีที่ไคลเอ็นต์ที่ไม่มีดิสก์ติดตั้งบนไดร็อกทอรี /usr/share
- /root มีไดร็อกทอรีที่ไคลเอ็นต์ที่ไม่มีดิสก์ติดตั้งบนระบบไฟล์ / (ราก)
- /home มีไดร็อกทอรีที่ไคลเอ็นต์ที่ไม่มีดิสก์ติดตั้งบนระบบไฟล์ / home

ไดร็อกทอรี /export คือที่ตั้งดีฟอลต์สำหรับ ไคลเอ็นต์รีซอร์สของคำสั่ง diskless ไดร็อกทอรี /export คือที่ตั้งเดียวของไคลเอ็นต์รีซอร์สบนเซิร์ฟเวอร์ เนื่องจากไคลเอ็นต์ติดตั้ง รีซอร์สเหล่านี้บนแผนผังไฟล์ของตนเอง รีซอร์สเหล่านี้จึงปรากฏขึ้นในไคลเอ็นต์ ที่ตำแหน่งปกติในแผนผังไฟล์ ไดร็อกทอรีย่อยที่สำคัญของไดร็อกทอรี /export และจุดติดตั้งที่สอดคล้องกันบนแผนผังไฟล์ไคลเอ็นต์ รวมถึง:

/export/root

ไดร็อกทอรีนี้ติดตั้งบนระบบไฟล์รากไคลเอ็นต์ (/) ไดร็อกทอรี รากไคลเอ็นต์ตั้งอยู่ในไดร็อกทอรี /export/root โดยค่าดีฟอลต์และมีชื่อเหมือนกับชื่อโฮสต์ของไคลเอ็นต์

/export/exec

เรียกอีกอย่างว่าไดเรกทอรี Shared Product Object Tree (SPOT) ไดเรกทอรีนี้ ตั้งอยู่บนระบบไฟล์ /usr SPOTs เป็นเวอร์ชันของระบบไฟล์ /usr ที่จัดเก็บไว้ในไดเรกทอรี /export/exec และมีชื่อที่สะท้อนระดับรีลีส โดยค่าดีฟอลต์ชื่อคือ RISCAIX

/export/share

ไดเรกทอรีนี้ติดตั้งบนไดเรกทอรีไคลเอ็นต์ /usr/share ไดเรกทอรีนี้มีข้อมูลที่สามารถแบ่งใช้โดยสถาปัตยกรรมจำนวนมาก ที่ตั้ง ดีฟอลต์คือ /export/share/AIX/usr/share

/export/home

ไดเรกทอรีนี้ตั้งอยู่บนระบบไฟล์ /home ไดเรกทอรีนี้มีไดเรกทอรีผู้ใช้ที่จัดกลุ่มตามชื่อโฮสต์ของไคลเอ็นต์ ที่ตั้ง ดีฟอลต์ สำหรับไดเรกทอรีโฮมไคลเอ็นต์คือ /export/home

/export/swap

เรียกอีกอย่างว่าไดเรกทอรีการเลื่อนหน้า ในระบบสแตนด์โอลนหรือที่ไม่มีข้อมูล การเลื่อนหน้า มีการนำเสนอโดยโลคัลดิสก์ สำหรับไคลเอ็นต์ที่ไม่มีดิสก์ เซอร์วิสนี้มีการนำเสนอ โดยไฟล์บนเซิร์ฟเวอร์ ไฟล์นี้มีชื่อเหมือนกับชื่อโฮสต์ของไคลเอ็นต์ และ อยู่ในไดเรกทอรี /export/swap โดยค่าดีฟอลต์

/export/dump

ระบบสแตนด์โอลนใช้โลคัลดิสก์เป็นอุปกรณ์ดัมพ์ ส่วนไคลเอ็นต์ที่ไม่มีดิสก์ ใช้ไฟล์บนเซิร์ฟเวอร์ ไฟล์ตั้งอยู่ในไดเรกทอรีที่มีชื่อเหมือนกับชื่อโฮสต์ของไคลเอ็นต์ และ อยู่ในไดเรกทอรี /export/dump โดยค่าดีฟอลต์

microcode

ไดเรกทอรีนี้มีไมโครโค้ดสำหรับอุปกรณ์ฟิสิกัล ที่ตั้ง ดีฟอลต์คือ /export/exec/RISCAIX/usr/lib/microcode

การเข้ารหัสระบบไฟล์ JFS2:

เริ่มต้นด้วย AIX Version 6.1 Encrypted File System (EFS) ได้รับการสนับสนุนบนระบบไฟล์ JFS2 EFS ช่วยให้คุณสามารถเข้ารหัสข้อมูลและควบคุมสิทธิเข้าถึงข้อมูลโดยใช้การป้องกันคีย์

คีย์มีการเชื่อมโยงกับผู้ใช้แต่ละราย และจัดเก็บไว้ในคลังคีย์ที่ป้องกันด้วย วิทยาการรหัสลับ ทันทีที่ล็อกอินสำเร็จ คีย์ของผู้ใช้จะถูกโหลดเข้าใน เคอร์เนลและเชื่อมโยงกับหลักฐานโปรเซส ในการเปิดไฟล์ที่มีการป้องกัน EFS หลักฐานโปรเซสจะถูกทดสอบ หากโปรเซสพบคีย์ที่ตรงกับ การป้องกันไฟล์ โปรเซสจะถอดรหัสคีย์ไฟล์และเนื้อหาไฟล์

โดยค่าดีฟอลต์ ระบบไฟล์ JFS2 ไม่ได้เปิดใช้งาน EFS ระบบไฟล์ JFS2 ต้องมีการ เปิดใช้งาน EFS ก่อนสามารถเข้ารหัสข้อมูลใดๆ ได้ สำหรับข้อมูลเกี่ยวกับวิธีการ เปิดใช้งาน EFS ให้ดูที่คำสั่ง `efsenable` ใน *Commands Reference, Volume 2*

การกำหนดคอนฟิกระบบไฟล์

เมื่อเพิ่มหรือกำหนดคอนฟิกระบบไฟล์ คุณสามารถเลือกตัวเลือกในคอนเทนเนอร์ File Systems ของพาด่วน SMIT

พาด่วน SMIT มีการนำเสนอในตารางต่อไปนี้:

ตารางที่ 64. การจัดการโลจิคัลวอลุ่มและภารกิจระบบไฟล์

ภารกิจ	พาด่วน SMIT
เพิ่ม JFS หรือ JFS2	smit crfs
เพิ่ม JFS2 ลงในโลจิคัลวอลุ่มที่มีอยู่	smit crjfs2lvstd
เพิ่ม JFS ลงในเมนูโลจิคัลวอลุ่มที่กำหนดไว้ก่อนหน้านี้	สร้างโลจิคัลวอลุ่ม แล้ว smit crjfslv
เปลี่ยนแอตทริบิวต์ของ JFS หรือ JFS2 ^{Note 1}	smit chfs
ตรวจสอบขนาดของระบบไฟล์	smit fs
เพิ่มขนาดของระบบไฟล์	JFS: smit chjfs JFS2: smit chjfs2
ลดขนาดของระบบไฟล์	JFS2: smit chjfs2

หมายเหตุ: พาด่วน SMIT สำหรับ ลดขนาดของระบบไฟล์ ใช้สำหรับ JFS2 เท่านั้น

การจัดการระบบไฟล์

ระบบไฟล์คือโครงสร้างไดเรกทอรีที่สมบูรณ์ และประกอบด้วยไดเรกทอรี ราก ไดเรกทอรีย่อย และไฟล์ที่อยู่ข้างใต้

ระบบไฟล์ถูกกำหนดเป็นโลจิคัลวอลุ่มเดียว ภารกิจการจัดการระบบที่สำคัญที่สุดบางส่วนเกี่ยวข้องกับระบบไฟล์โดยเฉพาะอย่างยิ่ง:

- การจัดสรรพื้นที่ว่างสำหรับระบบไฟล์บนโลจิคัลวอลุ่ม
- การสร้างระบบไฟล์
- การทำให้พื้นที่ว่างระบบไฟล์มีอยู่สำหรับผู้ใช้ระบบ
- การมอนิเตอร์การใช้พื้นที่ว่างระบบไฟล์
- การทำสำเนาสำรองระบบไฟล์เพื่อป้องกันข้อมูลสูญหายในกรณีที่ระบบ ล้มเหลว
- การทำ snapshot เพื่อดักจับรูปภาพระดับบล็อกที่สอดคล้องกันของระบบไฟล์ ณ จุดเวลาที่กำหนด
- การรักษาระบบไฟล์ให้อยู่ในสภาพที่สอดคล้องกัน

ข้อมูลต่อไปนี้เป็นรายการของคำสั่งการจัดการระบบซึ่งช่วยจัดการ ระบบไฟล์:

ไอเท็ม	คำอธิบาย
backup	ทำสำเนาสำรองทั้งหมดหรือส่วนเพิ่มของระบบไฟล์
chfs -a splitcopy	สร้างการสำรองแบบออนไลน์ของระบบไฟล์ JFS ที่ติดตั้ง
dd	คัดลอกข้อมูลโดยตรงจากอุปกรณ์หนึ่งไปยังอีกอุปกรณ์หนึ่งเพื่อทำสำเนาสำรองระบบไฟล์
df	รายงานจำนวนพื้นที่ว่างที่ใช้และที่ว่างอยู่บนระบบไฟล์
fsck	ตรวจสอบระบบไฟล์และซ่อมแซมความไม่สอดคล้องกัน
mkfs	จัดทำระบบไฟล์ตามขนาดที่ระบุบนโลจิคัลวอลุ่มที่ระบุ
mount	แนบระบบไฟล์เข้ากับโครงสร้างการตั้งชื่อทั่วทั้งระบบ เพื่อให้ สามารถเข้าถึงไฟล์และไดเรกทอรีในระบบไฟล์นั้นได้
restore	เรียกคืนไฟล์จากสำเนาสำรอง
snapshot	สร้าง snapshot ของระบบไฟล์ JFS2
umount	ลบระบบไฟล์ออกจากโครงสร้างการตั้งชื่อทั่วทั้งระบบ เพื่อให้ ไม่สามารถเข้าถึงไฟล์และไดเรกทอรีในระบบไฟล์นั้น

การแสดงผลพื้นที่ว่างที่มีอยู่บนระบบไฟล์ (คำสั่ง df):

ใช้คำสั่ง df เพื่อแสดงข้อมูลเกี่ยวกับ พื้นที่ว่างทั้งหมดและพื้นที่ว่างที่มีอยู่บนระบบไฟล์ พารามิเตอร์ FileSystem ระบุชื่อของอุปกรณ์ซึ่งมีระบบไฟล์อยู่ ไดรฟ์หรือชื่อระบบไฟล์ถูกติดตั้ง หรือ ชื่อพารามิเตอร์ของระบบไฟล์

หากคุณไม่ได้ระบุพารามิเตอร์ FileSystem คำสั่ง df จะแสดงข้อมูลสำหรับระบบไฟล์ทั้งหมดซึ่ง ติดตั้งในปัจจุบัน ถ้ามีการระบุไฟล์หรือไดเรกทอรี คำสั่ง df จะแสดงข้อมูลสำหรับระบบไฟล์ซึ่งไฟล์หรือไดเรกทอรีตั้งอยู่

โดยปกติแล้ว คำสั่ง df ใช้ตัวนับบิตที่มิอยู่ในซูเปอร์บล็อก ภายใต้เงื่อนไขที่มีข้อยกเว้นบางอย่าง จำนวนเหล่านี้อาจมีข้อผิดพลาด ตัวอย่างเช่น ถ้าระบบไฟล์ถูกแก้ไข ในขณะที่คำสั่ง df กำลังรันอยู่ จำนวนที่ว่าง อาจจะไม่ถูกต้อง

ให้ดูที่คำสั่ง df ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

หมายเหตุ: บนระบบไฟล์รีโมตบางระบบ เช่น ระบบไฟล์เน็ตเวิร์ก (NFS) คอลัมน์ที่แสดงถึงพื้นที่ว่างที่มีอยู่บนจอแสดงผล ถูกปล่อยว่างไว้ถ้าเซิร์ฟเวอร์ไม่แสดงข้อมูล

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง df:

- เมื่อต้องการแสดงข้อมูลเกี่ยวกับระบบไฟล์ทั้งหมดที่ติดตั้ง ให้พิมพ์ดังต่อไปนี้:

```
df
```

หาก ระบบของคุณมีการกำหนดคอนฟิกให้ไดเรกทอรี /, /usr, /site, และ /usr/venus ตั้งอยู่ในระบบไฟล์แยกต่างหาก เอาต์พุตจากคำสั่ง df จะคล้ายกับตัวอย่างต่อไปนี้:

```
Filesystem 512-blocks free %used Iused %Iused Mounted on
/dev/hd4 20480 13780 32% 805 13% /
/dev/hd2 385024 15772 95% 27715 28% /usr
/dev/hd9var 40960 38988 4% 115 1% /var
/dev/hd3 20480 18972 7% 81 1% /tmp
/dev/hd1 4096 3724 9% 44 4% /home
```

- เมื่อต้องการแสดงพื้นที่ว่างที่มีอยู่บนระบบไฟล์ซึ่งไดเรกทอรีปัจจุบัน ของคุณตั้งอยู่ ให้พิมพ์ดังต่อไปนี้:

```
df .
```

คำสั่งระบบไฟล์:

มีคำสั่งจำนวนมากที่ออกแบบมาเพื่อทำงานบนระบบไฟล์ ไม่ว่าจะเป็นระบบไฟล์ชนิดใดก็ตาม

ไฟล์ /etc/filesystems ควบคุมรายการของระบบไฟล์ที่คำสั่งต่อไปนี้สามารถจัดการได้:

ไอเท็ม	คำอธิบาย
chfs	เปลี่ยนลักษณะของระบบไฟล์
crfs	เพิ่มระบบไฟล์
lsfs	แสดงลักษณะของระบบไฟล์
rmfs	ลบระบบไฟล์
mount	ทำให้ระบบไฟล์พร้อมใช้งาน

มีคำสั่งที่ดำเนินงานบนชนิดระบบไฟล์เสมือน ไฟล์ /etc/vfs มีข้อมูลเกี่ยวกับชนิดระบบไฟล์ที่คำสั่งต่อไปนี้ สามารถจัดการได้:

ไอเอ็ม	คำอธิบาย
chvfs	เปลี่ยนลักษณะของชนิดระบบไฟล์
crvfs	เพิ่มชนิดระบบไฟล์ใหม่
lsvfs	แสดงรายการลักษณะของชนิดระบบไฟล์
rmvfs	ลบชนิดระบบไฟล์

การเปรียบเทียบระบบไฟล์บนเครื่องที่แตกต่างกัน:

เมื่อระบบไฟล์ที่มีอยู่บนเครื่องที่แตกต่างกันควรจะเหมือนกัน แต่คุณสงสัยว่าระบบหนึ่งเสียหาย คุณสามารถเปรียบเทียบระบบไฟล์ได้

โพรซีเดอร์ท่อนี้ อธิบายวิธีการเปรียบเทียบแอ็ดทริบิวต์ ของระบบไฟล์ที่ตั้งอยู่บนโฮสต์ปัจจุบันของคุณ (ในสถานการณ์จำลองนี้ เรียกว่า *orig_host*) กับระบบไฟล์เดียวกันบน รีโมตโฮสต์

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

1. ล็อกอินเข้าสู่รีโมตโฮสต์เป็นผู้ใช้ราก ตัวอย่างเช่น:

```
tn juniper.mycompany.com
```

```
AIX เวอร์ชัน 6.1
(C) Copyrights by IBM and by others 1982, 2002. login: root
root's Password:
```

2. ใช้โปรแกรมแก้ไขที่โปรดปรานของคุณเพื่อแก้ไขไฟล์ `.rhosts` ของรีโมตโฮสต์เพื่อเพิ่ม stanza ซึ่งช่วยให้ผู้ใช้รากสามารถดำเนินการคำสั่งรีโมต ที่ปลอดภัยได้ ใช้รูปแบบต่อไปนี้สำหรับ stanza ใหม่:

```
orig_host root
```

ไฟล์ `.rhosts` ที่ได้ อาจ มีลักษณะคล้ายดังต่อไปนี้:

```
NIM.mycompany.com root
nim.mycompany.com root
host.othernetwork.com root
orig_host.mycompany.com root
```

3. บันทึกการเปลี่ยนแปลงของคุณและออกจากการเชื่อมต่อรีโมต
4. ด้วยสิทธิการใช้งานรากบน *orig_host* ให้สร้างอีกไฟล์หนึ่งโดยใช้โปรแกรมแก้ไขที่โปรดปราน สำหรับ สถานการณ์จำลองนี้ ไฟล์ใหม่มีชื่อว่า `compareFS` ตัวอย่างเช่น:

```
vi compareFS
```

5. แทรกข้อความต่อไปนี้ลงในไฟล์นี้ โดยที่ *FSname* คือ ชื่อของระบบไฟล์ซึ่งคุณต้องการเปรียบเทียบ และ *remote_host* คือ ชื่อโฮสต์ซึ่งระบบไฟล์ที่เปรียบเทียบตั้งอยู่:

```
FSname -> remote_host
install -v ;
```

หมายเหตุ: ในบรรทัดคำสั่ง `install` ของไฟล์นี้ ต้องมีช่องว่างระหว่างพารามิเตอร์ `-v` และเครื่องหมายอัฒภาค (`;`)

ตัวอย่างเช่น:

```
/home/jane/* -> juniper.mycompany.com
install -v ;
```

- บันทึกไฟล์และออกจากโปรแกรมแก้ไขไฟล์ compareFS มีการใช้ เป็น *distfile* สำหรับคำสั่ง *rdist* ในขั้นตอนต่อไป
- พิมพ์ดังต่อไปนี้ที่พร้อมต์คำสั่ง:

```
/usr/bin/rdist -f compareFS
```

หรือ ถ้าคุณคิดว่าเอาต์พุตจากการเปรียบเทียบจะมีจำนวนมาก ให้ส่งเอาต์พุตไปยังชื่อไฟล์ ตัวอย่างเช่น:

```
/usr/bin/rdist -f compareFS > compareFS_output
```

เอาต์พุตแสดงรายการความแตกต่างใดๆ ระหว่างระบบไฟล์

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง *rdist*

รูปแบบไฟล์ *rhosts* สำหรับ TCP/IP

การทำความเข้าใจกับคำสั่ง Secure Remote C

การบำรุงรักษาระบบไฟล์

ภารกิจง่ายที่สุดซึ่งคุณอาจต้องการเมื่อบำรุงรักษาระบบไฟล์ มีการจัดกลุ่มไว้ในตารางนี้

ตารางที่ 65. ภารกิจการบำรุงรักษาระบบไฟล์

ภารกิจ	พาด่วน SMIT	คำสั่งหรือไฟล์
ทำสำเนาสำรองตามชื่อไฟล์หรือไดเรกทอรี	smit backup	backup ^{Note 1}
สร้างและทำสำเนาสำรองรูปภาพ JFS2 snapshot	smit backsnap	backsnap ^{Note 1}
แสดงรายการระบบไฟล์ทั้งหมดบนดิสก์	smit lsmttdsk	
แสดงรายการระบบไฟล์บนดิสก์ที่ถอดออกได้	smit lsmttdsk	
แสดงรายการระบบไฟล์ที่ติดตั้ง	smit fs	
ติดตั้งกลุ่มของระบบไฟล์ ^{Note 5}	smit mountg	mount -t GroupName
ติดตั้ง JFS หรือ JFS2 ^{Note 3}	smit mountfs	mount
ติดตั้ง JFS2 snapshot	smit mntsnap	mount -v jfs2 -o snapshot Device MountPoint
ลบ JFS หรือ JFS2	smit rmfs	
ลบ JFS2 snapshot	smit rmsnap	snapshot -d SnapshotDevice
แปลงระบบไฟล์ JFS2 กลับเป็น point-in-time snapshot	smit rollbacksnap	rollback [-s] [-v] [-c] snappedFS snapshotObject
ถอนติดตั้งระบบไฟล์ ^{Note 4}	smit umountfs	
ถอนติดตั้งระบบไฟล์บนดิสก์ที่ถอดออกได้ ^{Note 4}	smit umnttdsk	
ถอนติดตั้งกลุ่มของระบบไฟล์ ^{Note 5}	smit umountg	umount -t GroupName
จัดการโควตาของระบบไฟล์ที่เจอร์นัลที่เพิ่มประสิทธิภาพ	smit j2fsquotas	
เปิดใช้งานหรือปิดใช้งานการจัดการโควตา	smit j2enablequotas	
หยุด/รีสตาร์ทการบังคับใช้ขีดจำกัดโควตา	smit j2enforcequotas	quotaonloff -v

ตารางที่ 65. การจัดการบำรุงรักษาระบบไฟล์ (ต่อ)

ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
แสดงรายการการใช้โควตา	smit j2repquota	repquota -v
คำนวณสถิติการใช้ดิสก์บล็อกและไฟล์ปัจจุบันอีกครั้ง	smit j2quotacheck	quotacheck -v
เพิ่มคลาสซีดจำกัด	smit j2addlimit	j2edlimit -e
แสดง/เปลี่ยนลักษณะของคลาสซีดจำกัด	smit j2changelimit	
ทำให้คลาสซีดจำกัดเป็นซีดจำกัดดีฟอลต์ของระบบไฟล์	smit j2defaultlimit	
กำหนดผู้ใช้หรือกลุ่มให้กับคลาสซีดจำกัด	smit j2assignlimit	
แสดงรายการคลาสซีดจำกัดสำหรับระบบไฟล์	smit j2listlimits	j2edlimit -l '-u'
ลบคลาสซีดจำกัด	smit j2removelimit	

หมายเหตุ:

1. สำหรับอ็อปชัน โปรดอ้างอิงแต่ละคำสั่ง
2. อย่าเปลี่ยนชื่อของระบบไฟล์ที่มีความสำคัญมากต่อระบบ ซึ่งได้แก่ / (ราก) บนโลจิคัลวอลุ่ม 4 (hd4), /usr บน hd2, /var บน hd9var, /tmp บน hd3, และ /blv บน hd5 หากผู้ใช้ระบุ hdm ให้เริ่มต้นที่ hd10
3. ตรวจสอบระบบไฟล์ก่อนการติดตั้ง โดยใช้โปรแกรม “การตรวจสอบระบบไฟล์” ในหน้า 481 หรือรันคำสั่ง fsck
4. หากการถอนติดตั้งล้มเหลว อาจเนื่องมาจากผู้ใช้หรือโปรเซสได้เปิดไฟล์ในระบบไฟล์ที่กำลังถอนติดตั้ง คำสั่ง fuser ช่วยให้คุณสามารถค้นหาผู้ใช้หรือโปรเซสซึ่งอาจทำให้เกิดความล้มเหลว
5. กลุ่มระบบไฟล์คือคอลเล็กชันของระบบไฟล์ซึ่งมีค่า ตัวระบุ type= เหมือนกันในไฟล์ /etc/filesystems

งานที่เกี่ยวข้อง:

“การติดตั้งพื้นที่ว่างจากดิสก์ไดร์ฟอื่น” ในหน้า 410

คุณสามารถได้รับพื้นที่ว่างมากขึ้นบนดิสก์ไดร์ฟได้โดยการติดตั้งพื้นที่ว่างจากไดร์ฟอื่น

การกู้คืนตั้งแต่หนึ่งไฟล์ขึ้นไปจาก JFS2 snapshot ภายนอกทางออนไลน์:

คุณสามารถเปลี่ยนไฟล์ที่เสียหายได้ถ้าคุณมีสำเนาที่ถูกต้องอยู่ใน JFS2 snapshot ภายนอกทางออนไลน์

ใช้โปรแกรมต่อไปนี้เพื่อกู้คืนตั้งแต่หนึ่งไฟล์ขึ้นไปจาก รูปภาพ JFS2 snapshot ภายนอกทางออนไลน์

สำหรับตัวอย่างนี้ สมมติว่า /home/aaa/myfile เป็นไฟล์ที่เสียหายในระบบไฟล์ /home

1. ติดตั้ง snapshot ด้วยคำสั่งที่คล้ายกับดังต่อไปนี้:

```
mount -v jfs2 -o snapshot /dev/mysnaplv /tmp/mysnap
```

2. เปลี่ยนไดเรกทอรีที่มี snapshot ด้วยคำสั่ง ที่คล้ายกับดังต่อไปนี้:

```
cd /tmp/mysnap
```

3. คัดลอกไฟล์ที่ถูกต้องจาก snapshot เพื่อเขียนทับไฟล์ที่เสียหาย ด้วยคำสั่งที่คล้ายกับดังต่อไปนี้:

```
cp aaa/myfile /home/aaa/myfile
```

ตัวอย่าง ก่อนหน้านี้คัดลอกเฉพาะไฟล์ที่ชื่อ myfile ถ้า ต้องการคัดลอกไฟล์ทั้งหมดจาก snapshot ไปยังไดเรกทอรี aaa ให้ใช้คำสั่งคล้ายกับดังต่อไปนี้:

```
cp -R aaa /home/aaa
```

หากต้องการ ตัวอย่างเพิ่มเติมเกี่ยวกับการเปลี่ยนไฟล์ที่เสียหายด้วยรูปภาพ snapshot ให้ดูที่คำอธิบายคำสั่ง cp หรือ cpio ใน *Commands Reference, Volume 1*

การกู้คืนตั้งแต่หนึ่งไฟล์ขึ้นไปจาก JFS2 snapshot ภายในทางออนไลน์:

คุณสามารถเปลี่ยนไฟล์ที่เสียหายได้ถ้าคุณมีสำเนาที่ถูกต้องอยู่ใน JFS2 snapshot ภายในทางออนไลน์

ใช้พรซีเดอร์ต่อไปนี้เพื่อกู้คืนตั้งแต่หนึ่งไฟล์ขึ้นไปจาก รูปภาพ JFS2 snapshot ภายในทางออนไลน์

สำหรับตัวอย่างนี้ สมมติว่า /home/aaa/myfile เป็นไฟล์ที่เสียหายในระบบไฟล์ /home

1. เปลี่ยนไดเรกทอรีที่มี snapshot ด้วยคำสั่ง ที่คล้ายกับดังต่อไปนี้:

```
cd /home/.snapshot/mysnap
```

2. คัดลอกไฟล์ที่ถูกต้องจาก snapshot เพื่อเขียนทับไฟล์ที่เสียหาย ด้วยคำสั่งที่คล้ายกับดังต่อไปนี้:

```
cp aaa/myfile /home/aaa/myfile
```

ตัวอย่าง ก่อนหน้านี้คัดลอกเฉพาะไฟล์ที่ชื่อ myfile ถ้า ต้องการคัดลอกไฟล์ทั้งหมดจาก snapshot ไปยังไดเรกทอรี aaa ให้ใช้คำสั่งคล้ายกับดังต่อไปนี้:

```
cp -R aaa /home/aaa
```

หากต้องการ ตัวอย่างเพิ่มเติมเกี่ยวกับการเปลี่ยนไฟล์ที่เสียหายด้วยรูปภาพ snapshot ให้ดูที่คำอธิบายคำสั่ง cp หรือ cpio ใน *Commands Reference, Volume 1*

ระบบไฟล์บนดิสก์ซีดีรอมและดีวีดี:

แผ่นซีดีและดีวีดีไม่มีการติดตั้งโดยอัตโนมัติ แต่สามารถเปิดใช้งาน คุณลักษณะนี้ได้

เมื่อต้องการเปิดใช้งานคุณลักษณะนี้ ให้ใช้คำสั่ง cdmount เพื่อติดตั้งระบบไฟล์ CDRFS หรือ UDFS ตัวอย่างเช่น:

```
cdmount cd0
```

คุณสามารถติดตั้ง UDFS การอ่าน/เขียนด้วยตนเองโดยใช้คำสั่งต่อไปนี้:

```
mount -V udfs DevName MtPt
```

โดยที่ DevName คือชื่อของดีวีดีไดรฟ์ และ MtPt คือจุดต่อสำหรับระบบไฟล์

การใช้ระบบไฟล์บนสื่อบันทึกออปติคัลการอ่าน/เขียน:

สามารถใช้ระบบไฟล์ CDRFS และ JFS บนสื่อบันทึกออปติคัลการอ่าน/เขียน

สามารถจัดเก็บระบบไฟล์ซีดีรอม (CDRFS) บนสื่อบันทึกออปติคัลการอ่าน /เขียนได้ ถ้าสื่อบันทึกออปติคัลนั้นมีการป้องกันการเขียน เช่นเดียวกับบน ซีดีรอม ตารางต่อไปนี้บอกให้คุณทราบถึงวิธีการเพิ่ม ติดตั้ง หรือถอนติดตั้ง CDRFS บนสื่อบันทึกออปติคัลการอ่าน/เขียน คุณต้องระบุข้อมูลต่อไปนี้เมื่อ ติดตั้งระบบไฟล์:

ไอเท็ม
ชื่ออุปกรณ์
จุดต่อ
ติดตั้งอัตโนมัติ

คำอธิบาย
กำหนดชื่อของอุปกรณ์ที่มีสื่อบันทึก
ระบุไดเรกทอรีที่จะติดตั้งระบบไฟล์
ระบุว่าจะติดตั้งระบบไฟล์โดยอัตโนมัติ เมื่อรีสตาร์ทระบบหรือไม่

ภารกิจ CDRFS บนสื่อบันทึกออปติคัล		
ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
การเพิ่ม CDRFS ¹	smit crcdrfs	1. เพิ่มระบบไฟล์: <code>crfs -v cdrfs -p ro -d DeviceName -m MountPoint -A AutomaticMount</code> 2. ติดตั้งระบบไฟล์: <code>mount MountPoint</code>
การลบ CDRFS ²	1. ถอนติดตั้งระบบไฟล์: <code>smit umountfs</code> 2. ลบระบบไฟล์: <code>smit rmcdrfs</code>	1. ถอนติดตั้งระบบไฟล์: <code>umount FileSystem</code> 2. ลบระบบไฟล์: <code>rmfs MountPoint</code>

หมายเหตุ:

- ตรวจสอบให้แน่ใจว่าสื่อบันทึกออปติคัลการอ่าน/เขียนมีการป้องกันการเขียน
- ต้องถอนติดตั้งระบบไฟล์ CDRFS ก่อนสามารถเอาสื่อบันทึกออปติคัล การอ่าน/เขียนออกได้

JFS นำเสนอระบบไฟล์การอ่าน/เขียนบน สื่อบันทึกออปติคัลที่คล้ายกับบนฮาร์ดดิสก์ คุณต้องมีสิทธิ์ในระบบ ในการสร้างหรืออิมพอร์ตระบบไฟล์การอ่าน/เขียนบนสื่อบันทึกออปติคัลการอ่าน/เขียน (นั่นคือ ล็อกอินของคุณต้องเป็นสมาชิกของกลุ่มระบบ) และคุณต้องมีข้อมูล ต่อไปนี้:

ชื่อกลุ่มวอลุ่ม

ระบุชื่อของกลุ่มวอลุ่ม

ชื่ออุปกรณ์

ระบุชื่อโลจิคัลของออปติคัลไดรฟ์การอ่าน/เขียน

จุดต่อ ระบุไดเรกทอรีที่จะติดตั้งระบบไฟล์

ติดตั้งอัตโนมัติ

ระบุว่าจะติดตั้งระบบไฟล์โดยอัตโนมัติ เมื่อรีสตาร์ทระบบหรือไม่

หมายเหตุ:

- กลุ่มวอลุ่มใดๆ ที่สร้างขึ้นบนสื่อบันทึกออปติคัลการอ่าน/เขียนต้องมีอยู่ในตัวเอง บนสื่อบันทึกนั้น กลุ่มวอลุ่มไม่สามารถขยายเกินหนึ่งออปติคัลดิสก์การอ่าน/เขียน
- เมื่อเข้าถึงระบบไฟล์ที่เจอร์นัลซึ่งสร้างขึ้นก่อนหน้านี้ ชื่อกลุ่มวอลุ่ม ไม่จำเป็นต้องตรงกับชื่อที่ใช้เมื่อสร้างกลุ่มวอลุ่ม

ภารกิจ JFS บนสื่อบันทึกออปติคัล		
ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
เพิ่ม JFS	<ol style="list-style-type: none"> ใส่ออปติคัลดิสก์เข้าในไดรฟ์ สร้างกลุ่มวอลุ่ม (ถ้าจำเป็น): <code>smit mkvg</code> สร้างระบบไฟล์ที่เจอร์นัล: <code>smit crfs</code> 	<ol style="list-style-type: none"> ใส่ออปติคัลดิสก์เข้าในไดรฟ์ สร้างกลุ่มวอลุ่ม (ถ้าจำเป็น): <code>mkvg -f -y VGName -d 1 DeviceName</code> สร้างระบบไฟล์ที่เจอร์นัล: <code>crfs -v jfs -g VGName -a size=SizeFileSystem -m MountPoint -A AutomaticMount -p rw</code> ติดตั้งระบบไฟล์: <code>mount MountPoint</code>
การเข้าถึง JFS ^{Note 1} ที่สร้างขึ้นก่อนหน้านี้	<ol style="list-style-type: none"> ใส่ออปติคัลดิสก์เข้าในไดรฟ์ อิมพอร์ตกลุ่มวอลุ่ม: <code>smit importvg</code> 	<ol style="list-style-type: none"> ใส่ออปติคัลดิสก์เข้าในไดรฟ์ อิมพอร์ตกลุ่มวอลุ่ม: <code>importvg -y VGName DeviceName</code> ติดตั้งระบบไฟล์: <code>mount MountPoint</code>
การลบ JFS ^{Note 2}	<ol style="list-style-type: none"> ถอนติดตั้งระบบไฟล์: <code>smit umountfs</code> ลบระบบไฟล์: <code>smit rmjfs</code> 	<ol style="list-style-type: none"> ถอนติดตั้งระบบไฟล์: <code>umount FileSystem</code> ลบระบบไฟล์: <code>rmfs MountPoint</code>

หมายเหตุ:

- โพรซีเดรนี้เป็นสิ่งจำเป็นในทุกครั้งที่ใส่สื่อบันทึกที่มี ระบบไฟล์ที่เจอร์นัล
- การลบระบบไฟล์ที่เจอร์นัลจะทำลายข้อมูลทั้งหมดที่มีอยู่ในระบบไฟล์นั้น และที่อยู่บนสื่อบันทึกออปติคัลการอ่าน/เขียน

การตรวจสอบระบบไฟล์:

ความไม่สอดคล้องกันสามารถเกิดขึ้นในระบบไฟล์เมื่อระบบหยุด ในขณะที่ระบบไฟล์ยังคงติดตั้งอยู่ หรือเมื่อดิสก์เสียหาย ในสถานการณ์ดังกล่าว สิ่งสำคัญคือการตรวจสอบระบบไฟล์ก่อนติดตั้งระบบไฟล์นั้น

และตรวจสอบระบบไฟล์ของคุณในสถานการณ์ต่อไปนี้ด้วย:

- หลังจากความบกพร่อง ตัวอย่างเช่น ถ้าผู้ใช้ไม่สามารถเปลี่ยนไดเรกทอรีเป็นไดเรกทอรีที่มีสิทธิอนุญาตของผู้ใช้รายนั้น (uid)
- ก่อนทำสำเนาสำรองระบบไฟล์เพื่อป้องกันข้อผิดพลาดและปัญหาการเรียกคืน ที่เป็นไปได้
- เมื่อติดตั้งหรือบูตระบบเพื่อให้แน่ใจว่าไม่มีข้อผิดพลาดเกี่ยวกับไฟล์ ระบบปฏิบัติการ

การตรวจสอบระบบไฟล์ที่ผู้ใช้กำหนด:

เมื่อต้องการตรวจสอบระบบไฟล์ที่ผู้ใช้กำหนด ให้ปฏิบัติตามขั้นตอนต่อไปนี้

1. ถอนติดตั้งระบบไฟล์ที่ผู้ใช้กำหนดซึ่งกำลังตรวจสอบ
2. ตรวจสอบให้แน่ใจว่าคุณมีสิทธิการเขียนบนไฟล์ในระบบไฟล์ มิฉะนั้น `fsck` ไม่สามารถซ่อมแซมไฟล์ที่เสียหายได้แม้ว่าคุณตอบ ใช่ ที่พร้อมต่อซ่อมแซม
3. ใช้พารามิเตอร์ `smit fsck` เพื่อเข้าถึงเมนู ตรวจสอบ ระบบไฟล์
4. ทำอย่างใดอย่างหนึ่งต่อไปนี้:

- ระบุชื่อของแต่ละระบบไฟล์ที่จะตรวจสอบในฟิลด์ ชื่อ ของระบบไฟล์ หรือ
 - เลือกชนิดระบบไฟล์ทั่วไปที่จะตรวจสอบ เช่น ระบบไฟล์ที่เจอร์นัล (JFS) ในฟิลด์ ชนิดของระบบไฟล์
5. หากคุณต้องการจำกัดการตรวจสอบของคุณเฉพาะตัวเลือกที่มีโอกาสมากที่สุด ให้ระบุ ไซ ในฟิลด์ การตรวจสอบด่วน หรือไม? อีพซันการตรวจสอบด่วนตรวจสอบเฉพาะระบบไฟล์ซึ่งมี โอกาสมีความไม่สอดคล้องกัน เช่น ระบบไฟล์ที่ติดตั้ง เมื่อระบบหยุดที่บางจุดในพาส
 6. ระบุชื่อของไฟล์ชั่วคราวบนระบบไฟล์ที่กำลังถูก ตรวจสอบในฟิลด์ ไฟล์ SCRATCH
 7. เริ่มต้นการตรวจสอบระบบไฟล์

การตรวจสอบระบบไฟล์รากและ /usr:

เมื่อต้องการรันคำสั่ง fsck บนระบบไฟล์ / หรือ /usr คุณต้องปิดระบบและรีบูตจากสื่อบันทึกที่ถอดออกได้ เนื่องจากไม่สามารถถอนติดตั้ง ระบบไฟล์ / (ราก) และ /usr จากระบบที่กำลังรันอยู่

โปรซีเดอร์ต่อไปนี้อธิบายวิธีการรัน fsck บน ระบบไฟล์ / และ /usr จาก เซลล์การบำรุงรักษา

1. ปิดระบบของคุณ (ต้องการสิทธิเข้าถึงราก)
2. บูตจากสื่อบันทึกการติดตั้งของคุณ
3. จากเมนู ต้อนรับ ให้เลือกอีพซัน การบำรุงรักษา
4. จากเมนู การบำรุงรักษา ให้เลือกอีพซัน เพื่อเข้าถึงกลุ่มวอลุ่ม
5. เลือกกลุ่มวอลุ่ม rootvg รายการของโลจิคัลวอลุ่มที่เป็นสมาชิกของกลุ่มวอลุ่มซึ่งคุณเลือกแสดงขึ้น
6. เลือก 2 เพื่อเข้าถึงกลุ่มวอลุ่มและ เพื่อเริ่มต้นเซลล์ก่อนติดตั้งระบบไฟล์ในขั้นตอนต่อไป นี้ คุณจะรันคำสั่ง fsck โดยใช้อีพซันและ ชื่ออุปกรณ์ระบบไฟล์ที่เหมาะสม คำสั่ง fsck ตรวจสอบความสอดคล้องกันของระบบไฟล์และการซ่อมแซมระบบไฟล์แบบโต้ตอบ อุปกรณ์ระบบไฟล์ / (ราก) คือ /dev/hd4 และ อุปกรณ์ระบบไฟล์ /usr คือ /dev/hd2
7. เมื่อต้องการตรวจสอบระบบไฟล์ / ให้พิมพ์ดังต่อไปนี้:


```
$ fsck -y /dev/hd4
```

 แนะนำให้ใช้แฟล็ก -y สำหรับผู้ใช้ที่มีประสบการณ์น้อย (ให้ดูที่คำสั่ง fsck)
8. เมื่อต้องการตรวจสอบระบบไฟล์ /usr ให้พิมพ์ดังต่อไปนี้:


```
$ fsck -y /dev/hd2
```
9. เมื่อต้องการตรวจสอบระบบไฟล์อื่นใน rootvg ให้พิมพ์คำสั่ง fsck พร้อมด้วยชื่ออุปกรณ์ที่เหมาะสม อุปกรณ์สำหรับ /tmp คือ /dev/hd3 และอุปกรณ์สำหรับ /var คือ /dev/hd9var
10. เมื่อคุณตรวจสอบระบบไฟล์เสร็จสมบูรณ์แล้ว ให้รีบูตระบบ

การลดขนาดของระบบไฟล์ในกลุ่มวอลุ่มรากของคุณ:

วิธีที่ง่ายที่สุดในการลดระบบไฟล์ *ทั้งหมด* เป็นขนาดเล็กที่สุดคือ การตั้งค่าอีพซัน SHRINK เป็น ไซ เมื่อ เรียกคืนระบบปฏิบัติการพื้นฐานจากสำเนาสำรอง

วิธีที่ง่ายที่สุดในการลดระบบไฟล์ *ทั้งหมด* เป็นขนาดเล็กที่สุดคือ การตั้งค่าอีพซัน SHRINK เป็น ไซ เมื่อ เรียกคืนระบบปฏิบัติการพื้นฐานจากสำเนาสำรอง อีพซัน SHRINK และสถานการณ์จำลองต่อไปนี้ไม่สามารถใช้ใน tandem หากคุณตั้งค่าอีพซัน SHRINK เป็น ไซ หลังจากทำโปรซีเดอร์ต่อไปนี้ การติดตั้งจะยกเลิกการเปลี่ยนแปลงของคุณในไฟล์ /image.data

สถานการณ์จำลองนี้ นำคุณผ่านโปรเซสการทำด้วยตนเองเพื่อลดขนาดของ ระบบไฟล์ rootvg ที่เลือก คุณจะระบุ ระบบไฟล์ที่ไม่ได้ใช้พื้นที่ว่างดิสก์ที่ได้รับจัดสรรครบทั้งหมด จากนั้น จัดสรรใหม่ตามข้อมูลจำนวนของพื้นที่ว่างซึ่งระบบไฟล์ใช้จริง ดังนั้นจึงทำให้มีพื้นที่ว่างมากขึ้นสำหรับการใช้ของกลุ่มวอลุ่มราก ในส่วนหนึ่งของโปรซีเดอร์นี้ คุณจะทำสำเนาสำรองกลุ่มวอลุ่มของคุณและติดตั้งระบบปฏิบัติการ อีกครั้ง โดยใช้การจัดสรรที่แก้ไข

ข้อควรสนใจ: โปรซีเดอร์นี้บังคับให้ปิดระบบ และติดตั้งระบบปฏิบัติการพื้นฐานใหม่ เมื่อใดก็ตามที่คุณติดตั้งระบบปฏิบัติการใดๆ ใหม่ ให้จัดตารางเวลา downtime ใหม่ของคุณเป็นเวลาที่ส่งผลกระทบต่อเวิร์กโหลดของคุณ น้อยที่สุด เพื่อป้องกันการสูญหายของข้อมูลหรือการทำงานที่อาจเกิดขึ้นได้ ก่อนติดตั้งระบบปฏิบัติการ อีกครั้ง ตรวจสอบให้แน่ใจว่าคุณมีสำเนาสำรองที่เชื่อถือได้ ของข้อมูลและแอปพลิเคชันหรือกลุ่มวอลุ่มแบบกำหนดเอง

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

1. สร้างสำเนาสำรองแยกต่างหากของระบบไฟล์ทั้งหมดซึ่ง *ไม่มีอยู่* ใน rootvg สำเนาสำรองแยกต่างหากช่วยให้อุ่นใจถึงคุณภาพของระบบไฟล์ทั้งหมดของคุณ
2. ด้วยสิทธิการใช้งานราก ตรวจสอบว่าระบบไฟล์ใดในกลุ่มวอลุ่มราก ไม่ได้ใช้พื้นที่ว่างดิสก์ที่ได้รับจัดสรร โดยพิมพ์ คำสั่งต่อไปนี้:

```
df -k
```

แฟล็ก **-k** แสดงขนาดระบบไฟล์ในหน่วยกิโลไบต์ ผลลัพธ์ของคุณจะมีลักษณะ คล้ายตัวอย่างต่อไปนี้:

Filesystem	1024-blocks	Free	%Used	Iused	%Iused	Mounted on
/dev/hd4	196608	4976	98%	1944	2%	/
/dev/hd2	1769472	623988	65%	36984	9%	/usr
/dev/hd9var	163840	65116	61%	676	2%	/var
/dev/hd3	65536	63024	4%	115	1%	/tmp
/dev/hd1	49152	8536	83%	832	7%	/home
/proc	-	-	-	-	-	/proc
/dev/hd10opt	32768	26340	20%	293	4%	/opt

โดยดูที่ผลลัพธ์เหล่านี้ คุณสังเกตเห็น บล็อกที่ว่างจำนวนมากและเปอร์เซ็นต์การใช้งานที่ค่อนข้างต่ำ ซึ่งเชื่อมโยงของระบบไฟล์ที่ติดตั้งบน /usr คุณตัดสินใจว่าคุณสามารถรีไซเคิลบล็อกจำนวนมากได้โดยการลดจำนวนของพาร์ติชันที่จัดสรรให้กับระบบไฟล์ /usr

3. ตรวจสอบเนื้อหาของไฟล์ /etc/filesystems เพื่อให้แน่ใจว่าระบบไฟล์ทั้งหมดใน rootvg มีการติดตั้งแล้ว ถ้าไม่ระบบนั้นจะไม่รวมอยู่ในระบบที่ติดตั้งใหม่
4. สร้างไฟล์ /image.data ซึ่ง แสดงรายการระบบไฟล์ทั้งหมดที่ใช้งานอยู่ใน rootvg ซึ่งถูกรวมไว้ในโปรซีเดอร์การติดตั้ง โดยพิมพ์คำสั่งต่อไปนี้:

```
mkszfile
```

5. เปิดไฟล์ /image.data ในโปรแกรมแก้ไข ที่โปรดปรานของคุณ
6. ค้นหาสตริงข้อความ usr เพื่อระบุตำแหน่ง lv_data stanza ที่เกี่ยวข้องกับระบบไฟล์ /usr ใช้ตัวเลขจาก stanza นี้เป็นพื้นฐานในการกำหนดจำนวน ของพาร์ติชันโลจิคัลของระบบไฟล์ /usr ซึ่งคุณ สามารถลดได้ ค่าดีฟอลต์ของแต่ละพาร์ติชันโลจิคัลเพิ่มเติม มีการกำหนดไว้ในรายการ PP_SIZE ของไฟล์ /image.data ไฟล์ /image.data ของคุณ จะมีลักษณะคล้ายตัวอย่างต่อไปนี้:

```
lv_data:
VOLUME_GROUP= rootvg
LV_SOURCE_DISK_LIST= hdisk0
LV_IDENTIFIER= 00042345d300bf15.5
```

```

LOGICAL_VOLUME= hd2
VG_STAT= active/complete
TYPE= jfs
MAX_LPS= 32512
COPIES= 1
LPs= 108
STALE_PPs= 0
INTER_POLICY= minimum
INTRA_POLICY= center
MOUNT_POINT= /usr
MIRROR_WRITE_CONSISTENCY= on/ACTIVE
LV_SEPARATE_PV= yes
PERMISSION= read/write
LV_STATE= opened/syncd
WRITE_VERIFY= off
PP_SIZE= 16
SCHED_POLICY= parallel
PP= 108
BB_POLICY= relocatable
RELOCATABLE= yes
UPPER_BOUND= 32
LABEL= /usr
MAPFILE=
LV_MIN_LPS= 70
STRIPE_WIDTH=
STRIP_SIZE=

```

จำนวนของพาร์ติชันโลจิคัลที่จัดสรรให้กับโลจิคัลวอลุ่มนี้คือ 108 (LPs=108)

- กำหนดจำนวนของพาร์ติชันโลจิคัลที่ต้องการโดยข้อมูล ที่มีอยู่ในระบบไฟล์ /usr โดยใช้ผลลัพธ์ของคุณจากขั้นตอนที่ 2 คุณสามารถแสดงขนาดไฟล์ที่มีอยู่ เฉพาะสำหรับระบบไฟล์ /usr โดยใช้คำสั่งต่อไปนี้:

```
df -k /usr
```

ผลลัพธ์ ทำซ้ำจำนวน (ในหน่วยกิโลไบต์) ที่คุณได้รับสำหรับระบบไฟล์ /usr ในขั้นตอนที่ 2 ตัวอย่างเช่น:

```
Filesystem      1024-blocks      Free %Used   Iused %Iused Mounted on
/dev/hd2         1769472          623988  65%    36984   9% /usr
```

- ลบจำนวนพื้นที่ว่างออกจากจำนวนทั้งหมดของบล็อกขนาด 1024 ที่จัดสรร:

$$1769472 - 623988 = 1145484$$

- บวกเพิ่มจำนวนประเมินของพื้นที่ว่างซึ่งคุณอาจต้องการใช้สนับสนุน การเติบโตที่คาดไว้ในอนาคตในระบบไฟล์นี้ ตัวอย่าง เช่น เพิ่ม 200000 ในผลลัพธ์

$$1145484 + 200000 = 1345484$$

- หารผลลัพธ์ด้วยขนาดพาร์ติชันโลจิคัลในหน่วยไบต์ (16*1024) เพื่อกำหนดจำนวนต่ำสุดของพาร์ติชันโลจิคัลซึ่งคุณ ต้องการ

$$1345484 / 16384 = 82.121826171875$$

ใช้ค่าผลลัพธ์นี้โดยปัดเศษขึ้น เพื่อกำหนดจำนวนของพาร์ติชันโลจิคัลที่ต้องการ อีกครั้ง (LPs=83)

- ในไฟล์ image.data ของคุณ ให้เปลี่ยนฟิลด์ LPs จาก 108 เป็น 83
- ค้นหา fs_data stanza ที่เกี่ยวข้องกับระบบไฟล์ /usr fs_data stanza ของคุณจะมีลักษณะคล้ายตัวอย่าง ต่อไปนี้:

```
fs_data:
FS_NAME= /usr
FS_SIZE= 3538944
FS_MIN_SIZE= 2290968
FS_LV= /dev/hd2
FS_FS= 4096
FS_NBPI= 4096
FS_COMPRESS= no
FS_BF= false
FS_AGSIZE= 8
```

10. คำนวณขนาดระบบไฟล์ (FS_SIZE) โดยการคูณขนาดของพาร์ติชันฟิสิกัล (PP_SIZE) ด้วย 2 (จำนวนบล็อกขนาด 512-ไบต์ที่ใช้โดยพาร์ติชันฟิสิกัล) และคูณด้วยจำนวนของพาร์ติชันโลจิคัล (LPs) จากค่าที่ใช้ในตัวอย่างนี้ การคำนวณเป็นดังนี้:

```
PP_SIZE * 512 blocks * LPs = FS_SIZE
16384 * 2 * 83 = 2719744
```

11. ในไฟล์ image.data ของคุณ ให้เปลี่ยนฟิลด์ FS_SIZE จาก 3538944 เป็น 2719744

12. คำนวณขนาดระบบไฟล์ที่ต่ำสุด (FS_MIN_SIZE) ตามข้อมูลขนาดจริงของข้อมูลปัจจุบันที่ใช้โดยระบบไฟล์ /usr ดังนี้:

- a. คำนวณจำนวนต่ำสุดของพาร์ติชันที่ต้องการจากค่าที่ใช้ในตัวอย่างนี้ การคำนวณเป็นดังนี้:

```
size_in_use (see step 7a) / PP_SIZE = partitions
1145484 / 16384 = 69.914794921875
```

- b. คำนวณขนาดต่ำสุดที่ต้องการโดยพาร์ติชัน จำนวนนั้น โดยการปัดผลลัพธ์การคำนวณก่อนหน้านี้ขึ้นเป็น 70 การคำนวณคือ:

```
PP_SIZE * 512 blocks * partitions = FS_MIN_SIZE
16384 * 2 * 70 = 2293760
```

13. ในไฟล์ image.data ของคุณ ให้เปลี่ยนฟิลด์ FS_MIN_SIZE จาก 2290968 เป็น 2293760

14. บันทึกการแก้ไขของคุณและออกจากโปรแกรมแก้ไข

15. ถอนติดตั้งระบบไฟล์ทั้งหมดที่ไม่ได้อยู่ในกลุ่มวอลุ่ม rootvg

16. หากคุณมีกลุ่มวอลุ่มที่ผู้ใช้กำหนดให้พิมพ์คำสั่งต่อไปนี้ เพื่อ vary off และเอ็กซ์พอร์ต:

```
varyoffvg VGName
exportvg VGName
```

17. ในขณะที่เทปอยู่ในเทปไดรฟ์ ให้พิมพ์คำสั่งต่อไปนี้ เพื่อเริ่มต้นสำเนาสำรองระบบทั้งหมด:

```
mksysb /dev/rmt0
```

สำเนาสำรองชนิดนี้รวมข้อมูลขนาดระบบไฟล์ที่คุณระบุไว้ในไฟล์ /image.data ซึ่งจะใช้ในภายหลังเพื่อติดตั้งระบบของคุณอีกครั้งด้วยขนาดระบบไฟล์ใหม่

หมายเหตุ: เพื่อเริ่มต้นสำเนาสำรองนี้ คุณต้องรันคำสั่ง `mksysb` จากบรรทัดคำสั่ง หากคุณใช้เครื่องมือการจัดการระบบ เช่น SMIT สำเนาสำรองจะสร้างไฟล์ image.data ใหม่ ซึ่งเขียนทับการเปลี่ยนแปลงที่คุณได้ทำ

18. ใช้สำเนาสำรองนี้เพื่อติดตั้งระบบปฏิบัติการอีกครั้งโดยใช้อ็อปชัน ติดตั้งด้วยค่าติดตั้งระบบปัจจุบัน ในระหว่างการจัดตั้ง ตรวจสอบว่าอ็อปชันต่อไปนี้มีการตั้งค่าอย่างเหมาะสม:

- ใช้แม่พิมพ์ ต้องตั้งค่าเป็น `no`
- ย่อระบบไฟล์ ต้องตั้งค่าเป็น `no`

ถ้าคุณต้องการข้อมูลเพิ่มเติมเกี่ยวกับโปรซีเดเจอร์การติดตั้ง ให้ดูที่ การติดตั้งสำเนาสำรองระบบ

19. หลังจากการติดตั้งระบบปฏิบัติการ ให้รีบูตระบบ ในโหมดปกติ ณ จุดนี้ ระบบไฟล์ /usr ถูกปรับขนาด แต่ไม่มีระบบไฟล์ที่ผู้ใช้กำหนด
20. ติดตั้งระบบไฟล์ทั้งหมดโดยพิมพ์คำสั่งต่อไปนี้:

```
mount all
```

หาก คุณได้รับข้อความ Device Busy เกี่ยวกับระบบไฟล์ ที่ติดตั้งแล้ว คุณสามารถละเว้นข้อความเหล่านี้ได้

ณ จุดนี้ ระบบไฟล์ /usr ถูกปรับขนาด กลุ่มวอลุ่มรากมีพื้นที่ว่างมากขึ้น และสามารถใช้ระบบไฟล์ ของคุณได้

หลักการที่เกี่ยวข้อง:

“หน่วยเก็บโลจิคัลวอลุ่ม” ในหน้า 421

โลจิคัลวอลุ่มคือกลุ่มของข้อมูลที่ตั้งอยู่บนฟิสิคัลวอลุ่ม

ข้อมูลที่เกี่ยวข้อง:

การสร้างสำเนาสำรองกลุ่มวอลุ่มรากไว้ในเทปหรือ File

คำอธิบายไฟล์ /image.data

คำสั่ง mkszfiled

คำสั่ง mkysyb

การแก้ไขปัญหาระบบไฟล์

ใช้เมธอดการแก้ไขปัญหาเหล่านี้เพื่อจัดการกับปัญหาพื้นฐานบางอย่าง ที่อาจเกิดขึ้นบนระบบไฟล์ของคุณ หากข้อมูลการแก้ไขปัญหา ไม่ได้ช่วยแก้ไขปัญหา โปรดติดต่อผู้แทนเซอร์วิสของคุณ

การแก้ไขระบบไฟล์ที่โอเวอร์โฟลว์และกำหนดโดยผู้ใช้:

ใช้โปรซีเดเจอร์นี้เพื่อแก้ไขระบบไฟล์ที่โอเวอร์โฟลว์และกำหนดโดยผู้ใช้

1. ลบไฟล์สำเนาสำรองเก่าและแกนไฟล์ออก ตัวอย่างต่อไปนี้ลบไฟล์ *.bak, *.bak, a.out, core, *, หรือ ed.hup ทั้งหมด

```
find / \( -name "*.bak" -o -name core -o -name a.out -o \  
-name "...*" -o -name "*.bak" -o -name ed.hup \) \  
-atime +1 -mtime +1 -type f -print | xargs -e rm -f
```

2. เมื่อต้องการป้องกันไฟล์ไม่ให้โอเวอร์โฟลว์ดีสก์อย่างสม่ำเสมอ ให้รันคำสั่ง skulker เป็นส่วนหนึ่งของโปรเซส cron และลบไฟล์ที่ไม่จำเป็นหรือไฟล์ชั่วคราวออก

คำสั่ง skulker ล้างข้อมูลไฟล์ ในไดเรกทอรี /tmp ไฟล์ที่เก่ากว่าอายุที่ระบุ ไฟล์ a.out แกนไฟล์ และไฟล์ ed.hup คำสั่งนี้มีการรันรายวันเป็นส่วนหนึ่งของโปรซีเดเจอร์การบัญชีที่รันโดยคำสั่ง cron ในระหว่างรอบเวลาที่ใช้งานไม่มากนัก (สมมติว่าคุณเปิดการบัญชี)

cron daemon รันคำสั่งเชลล์ ในวันและเวลาที่ระบุ คำสั่งที่จัดตารางไว้อย่างสม่ำเสมอ เช่น skulker สามารถระบุได้ตามคำแนะนำ ที่มีอยู่ในไฟล์ crontab ส่งไฟล์ crontab ด้วยคำสั่ง crontab เมื่อต้องการแก้ไขไฟล์ crontab คุณต้องมีสิทธิ์ผู้ใช้ราก

งานที่เกี่ยวข้อง:

“การตั้งค่าระบบการบัญชี” ในหน้า 186

คุณสามารถตั้งค่าระบบการบัญชี

การแก้ไขระบบไฟล์ที่เสียหาย:

ระบบไฟล์อาจเกิดความเสียหายได้เมื่อข้อมูลไอ-โหนดหรือซูเปอร์บล็อก สำหรับโครงสร้างไดเรกทอรีของระบบไฟล์เสียหาย

ความเสียหายนี้อาจเกิดขึ้นจากความบกพร่องที่เกี่ยวข้องกับฮาร์ดแวร์ หรือ จากโปรแกรมที่เสียหายซึ่งเข้าถึงข้อมูลไอ-โหนดหรือซูเปอร์บล็อก โดยตรง (โปรแกรมที่เขียนใน assembler และ C สามารถบายพาสระบบปฏิบัติการ และเขียนลงในฮาร์ดแวร์โดยตรงได้) อาการอย่างหนึ่งของระบบไฟล์ที่เสียหายคือ ระบบไม่สามารถระบุตำแหน่ง อ่าน หรือเขียนข้อมูลซึ่งอยู่ในระบบไฟล์เฉพาะ

เพื่อแก้ไขระบบไฟล์ที่เสียหาย คุณต้องวิเคราะห์ปัญหา จากนั้นซ่อมแซมแก้ไขปัญหานั้น คำสั่ง `fsck` ทำการวิเคราะห์ระดับต่ำ และการซ่อมแซม

ข้อมูลต่อไปนี้เป็นโพรซีเจอร์สำหรับการแก้ไข ระบบไฟล์ที่เสียหาย:

1. ด้วยสิทธิการใช้งานราก ถอนติดตั้งระบบไฟล์ที่เสียหายโดยใช้พาร์ตวน SMIT อย่างใดอย่างหนึ่งต่อไปนี้: `smit unmountfs` (สำหรับ ระบบไฟล์บนดิสก์ไดรฟ์คงที่) หรือ `smit unmntdsk` (สำหรับ ระบบไฟล์บนดิสก์ที่ถอดออกได้)
2. เข้าถึงระบบไฟล์ที่เสียหายโดยการรันคำสั่ง `fsck` ในตัวอย่างต่อไปนี้ คำสั่ง `fsck` ตรวจสอบระบบไฟล์ที่ถอนติดตั้ง ซึ่งตั้งอยู่บนอุปกรณ์ `/dev/myfilelv`:

```
fsck /dev/myfilelv
```

คำสั่ง `fsck` จะตรวจสอบและซ่อมแซมแบบโต้ตอบ ระบบไฟล์ที่ไม่สอดคล้องกัน โดยปกติ ระบบไฟล์จะสอดคล้องกัน และคำสั่ง `fsck` เพียงรายงานเกี่ยวกับจำนวนไฟล์ บล็อกที่ใช้ และบล็อกที่ว่างใน ระบบไฟล์ หากระบบไฟล์ไม่สอดคล้อง คำสั่ง `fsck` จะแสดงข้อมูลเกี่ยวกับความไม่สอดคล้องกันที่พบและพร้อมท์ให้คุณเพื่อขอสิทธิในการซ่อมแซม คำสั่ง `fsck` ะมัดระวังเมื่อพยายามจะทำการซ่อมแซม และพยายามหลีกเลี่ยงการดำเนินการที่อาจส่งผลให้เกิดการสูญเสียข้อมูลที่ถูกต้อง อย่างไรก็ตาม ก็มีบางกรณี ที่คำสั่ง `fsck` จะแนะนำ ให้ทำลายไฟล์ที่เสียหายนั้น โปรดอ้างอิงคำอธิบายคำสั่ง `fsck` ใน *Commands Reference, Volume 2* เพื่อดู รายการความไม่สอดคล้องกันที่คำสั่งนี้จะตรวจสอบ

3. หากไม่สามารถซ่อมแซมระบบไฟล์ได้ ให้เรียกคืนจากสำเนาสำรอง

ข้อควรสนใจ: การเรียกคืนระบบไฟล์จากสำเนาสำรองจะทำลาย และแทนที่ระบบไฟล์ใดๆ ที่จัดเก็บไว้ก่อนหน้านี้บนดิสก์

เมื่อต้องการเรียกคืน ระบบไฟล์จากสำเนาสำรอง ให้ใช้พาร์ตวน SMIT `smit restfilesys` หรือชุดของคำสั่งที่แสดงในตัวอย่างต่อไปนี้:

```
mkfs /dev/myfilelv
mount /dev/myfilelv /myfilesys
cd /myfilesys
restore -r
```

ในตัวอย่างนี้ คำสั่ง `mkfs` จัดทำระบบไฟล์ใหม่ บนอุปกรณ์ที่ชื่อ `/dev/myfilelv` และเริ่มต้น เลเบลวลุ่ม เลเบลระบบไฟล์ และสร้างที่อับบล็อก คำสั่ง `mount` สร้าง `/dev/myfilelv` เป็นจุดต่อสำหรับ `myfilesys` และ คำสั่ง `restore` ดึงข้อมูลระบบไฟล์มาจากสำเนาสำรอง

หากสำเนาสำรอง ของคุณจัดทำขึ้นโดยใช้สำเนาสำรองระบบไฟล์ส่วนเพิ่ม คุณต้องเรียกคืน สำเนาสำรองในลำดับระดับ สำเนาสำรองที่เพิ่มขึ้น (ตัวอย่างเช่น 0, 1, 2) เมื่อใช้ `smit restfilesys` เพื่อเรียกคืนทั้งระบบไฟล์ ให้ป้อนไดเรกทอรีเป้าหมาย อุปกรณ์เรียกคืน (ไม่ใช่ `/dev/rfd0`) และจำนวนบล็อกที่จะอ่านในการดำเนินงานอินพุตเดียว

งานที่เกี่ยวข้อง:

“การเรียกคืนไฟล์ผู้ใช้จากรูปภาพสำเนาสำรอง” ในหน้า 34

ถ้าคุณต้องการเรียกคืนรูปภาพสำเนาสำรองที่ถูกทำลายโดยอุบัติเหตุ ปัญหาที่ยากที่สุดของคุณคือการกำหนดเทปสำเนาสำรองที่มีไฟล์นี้ คำสั่ง `restore -T` สามารถใช้เพื่อแสดงรายการเนื้อหาของที่เก็บถาวร ถือเป็นแนวคิดที่ดีในการเรียกคืนไฟล์ในไดเรกทอรี `/tmp` เพื่อให้คุณไม่เขียนทับไฟล์อื่นของผู้ใช้โดยไม่ได้ตั้งใจ

การแก้ไขหมายเลขพิเศษที่เสียหายในซูเปอร์บล็อกระบบไฟล์:

หากซูเปอร์บล็อกของระบบไฟล์เสียหาย จะไม่สามารถเข้าถึง ระบบไฟล์ได้ คุณสามารถแก้ไขหมายเลขพิเศษที่เสียหายในซูเปอร์บล็อก ระบบไฟล์ได้

ความเสียหายส่วนใหญ่ในซูเปอร์บล็อกไม่สามารถซ่อมแซมได้ โพรซีเจอร์ต่อไปนี้ อธิบายวิธีการซ่อมแซมซูเปอร์บล็อกในระบบไฟล์ JFS เมื่อปัญหา เกิดขึ้นจากหมายเลขพิเศษที่เสียหาย หากซูเปอร์บล็อกหลัก เสียหายในระบบไฟล์ JFS2 ให้ใช้คำสั่ง `fsck` เพื่อตัดลอก ซูเปอร์บล็อกรองและซ่อมแซมซูเปอร์บล็อกหลักโดยอัตโนมัติ

ในสถานการณ์จำลองต่อไปนี้ สมมติว่า `/home/myfs` เป็นระบบไฟล์ JFS บนพาร์ติชัน `/dev/lv02`

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

1. ถอนติดตั้งระบบไฟล์ `/home/myfs` ซึ่งคุณ สงสัยว่าจะเสียหาย โดยใช้คำสั่งต่อไปนี้:

```
umount /home/myfs
```

2. เมื่อต้องการยืนยันความเสียหายในระบบไฟล์ ให้ใช้คำสั่ง `fsck` บนระบบไฟล์ ตัวอย่าง:

```
fsck -p /dev/lv02
```

ถ้า ปัญหาคือการเสียหายของซูเปอร์บล็อก คำสั่ง `fsck` จะส่งคืนข้อความอย่างใดอย่างหนึ่งต่อไปนี้:

```
fsck: Not an AIXV5 file system
```

หรือ

```
Not a recognized filesystem type
```

3. ด้วยสิทธิการใช้งานราก ให้ใช้คำสั่ง `od` เพื่อแสดงซูเปอร์บล็อกของระบบไฟล์ ดังแสดงในตัวอย่างต่อไปนี้:

```
od -x -N 64 /dev/lv02 +0x1000
```

โดยที่แฟล็ก `-x` แสดงเอาต์พุตในรูปแบบฐานสิบหก และแฟล็ก `-N` สั่งให้ระบบ จัดรูปแบบอินพุตไม่เกินกว่า 64 ไบต์จากออฟเซตพารามิเตอร์ (+) ซึ่งระบุจุดในไฟล์ซึ่งไฟล์เอาต์พุตเริ่มขึ้น ข้อมูล ต่อไปนี้เป็นเอาต์พุตตัวอย่าง:

```
0001000 1234 0234 0000 0000 0000 4000 0000 000a
0001010 0001 8000 1000 0000 2f6c 7633 0000 6c76
0001020 3300 0000 000a 0003 0100 0000 2f28 0383
0001030 0000 0001 0000 0200 0000 2000 0000 0000
0001040
```

ในเอาต์พุตก่อนหน้าให้จดบันทึกค่าพิเศษที่เสียหาย ที่ `0x1000 (1234 0234)` หากใช้ค่าดีฟอลต์ทั้งหมดเมื่อสร้าง ระบบไฟล์ หมายเลขพิเศษควรจะเป็น `0x43218765` หากค่าดีฟอลต์ใดๆ ถูกยกเลิก หมายเลขพิเศษควรจะเป็น `0x65872143`

4. ใช้คำสั่ง `od` เพื่อตรวจสอบซูเปอร์บล็อกรอง สำหรับหมายเลขพิเศษที่ถูกต้อง คำสั่งตัวอย่างและเอาต์พุตของคำสั่งเป็นดังนี้:

```
$ od -x -N 64 /dev/lv02 +0x1f000
001f000 6587 2143 0000 0000 0000 4000 0000 000a
001f010 0001 8000 1000 0000 2f6c 7633 0000 6c76
001f020 3300 0000 000a 0003 0100 0000 2f28 0383
001f030 0000 0001 0000 0200 0000 2000 0000 0000
001f040
```

จดบันทึกค่าพิเศษที่ถูกต้องที่ 0x1f000

5. คัดลอกซูเปอร์บล็อกของไปยังซูเปอร์บล็อกหลัก คำสั่ง ตัวอย่างและเอาต์พุตเป็นดังนี้:

```
$ dd count=1 bs=4k skip=31 seek=1 if=/dev/lv02 of=/dev/lv02
dd: 1+0 records in.
dd: 1+0 records out.
```

6. ใช้คำสั่ง `fsck` เพื่อทำความสะอาดไฟล์ที่ไม่สอดคล้องกัน ซึ่งเกิดขึ้นจากการใช้ซูเปอร์บล็อกของ ตัวอย่าง:

```
fsck /dev/lv02 2>&1 | tee /tmp/fsck.errs
```

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `fsck`

คำสั่ง `odd`

ดิสก์โอเวอร์โฟลว์

ดิสก์โอเวอร์โฟลว์เกิดขึ้นเมื่อมีไฟล์อยู่มากเกินไปในพื้นที่ว่างซึ่ง จัดสรร กรณีนี้อาจเกิดขึ้นจากโปรเซส runaway ที่สร้างไฟล์ซึ่งไม่จำเป็นจำนวนมาก

คุณสามารถใช้ไพธอนเตอร์ต่อไปนี้เพื่อแก้ไขปัญหา:

หมายเหตุ: คุณต้องมีสิทธิ์ผู้ใช้รากเพื่อลบโปรเซสที่คุณไม่ได้เป็นเจ้าของ

หลักการที่เกี่ยวข้อง:

“คำสั่งสำหรับการทำความสะอาดระบบไฟล์โดยอัตโนมัติ” ในหน้า 409

ใช้คำสั่ง `skulker` เพื่อทำความสะอาดระบบไฟล์โดยการลบไฟล์ที่ไม่ต้องการออก

การระบุโปรเซสปัญหา:

ใช้ไพธอนเตอร์นี้เพื่อแยกโปรเซสปัญหา

1. เมื่อต้องการตรวจสอบสถานะโปรเซสและระบุโปรเซสที่อาจก่อให้เกิดปัญหา ให้พิมพ์:

```
ps -ef | pg
```

คำสั่ง `ps` แสดงสถานะของโปรเซส แฟล็ก `-e` เขียนข้อมูลเกี่ยวกับโปรเซสทั้งหมด (ยกเว้นโปรเซสเคอร์เนล) และแฟล็ก `-f` สร้างรายการฉบับสมบูรณ์ของโปรเซส รวมถึงชื่อคำสั่งและ พารามิเตอร์เมื่อโปรเซสถูกสร้างขึ้น คำสั่ง `pg` จำกัดเอาต์พุตเพียงหน้าเดียวในแต่ละครั้ง เพื่อไม่ให้ข้อมูลหลุดจากหน้าจอ เร็วขึ้น

ตรวจสอบโปรเซสระบบหรือผู้ใช้ที่ผู้ใช้ซอร์สระบบ เช่น เวลา CPU จำนวนมากเกินไป โปรเซสระบบ เช่น `sendmail`, `routed`, และ `lpd` เป็นโปรเซสระบบที่มีแนวโน้มจะ runaways

2. เมื่อต้องการตรวจสอบโปรเซสผู้ใช้ที่ใช้ CPU มากกว่าจำนวนที่คาดไว้ ให้พิมพ์:

```
ps -u
```

3. จดบันทึก ID โปรเซส (PID) ของแต่ละโปรเซสปัญหา

การยุติโปรเซส:

คุณสามารถยุติโปรเซสปัญหาได้

ใช้ไพรซีเตอร์ต่อไปนี้เพื่อยุติโปรเซสปัญหา:

1. ยุติโปรเซสที่ก่อให้เกิดปัญหาโดยพิมพ์:

```
kill -9 PID
```

โดยที่ *PID* คือ ID ของโปรเซสปัญหา

2. ลบไฟล์ที่โปรเซสจัดทำขึ้นโดยพิมพ์:

```
rm file1 file2 file3
```

โดยที่ *file1 file2 file3* แสดงถึง ชื่อของไฟล์ที่เกี่ยวข้องกับโปรเซส

การฟื้นฟูพื้นที่ว่างไฟล์โดยไม่ต้องยุติโปรเซส:

เมื่อต้องการฟื้นฟูบล็อกที่จัดสรรให้กับไฟล์ที่ใช้งานอยู่โดยไม่ต้องยุติโปรเซส ให้เปลี่ยนทิศทางเอาต์พุตของคำสั่งอื่นไปยังไฟล์ การเปลี่ยนทิศทางข้อมูล จะตัดไฟล์ให้สั้นลงและฟื้นฟูบล็อกของหน่วยความจำ

เมื่อไฟล์ที่ใช้งานอยู่ถูกลบออกจากระบบไฟล์ บล็อกที่จัดสรรให้กับไฟล์นั้น ยังคงถูกจัดสรรต่อไปจนกว่าการอ้างอิงที่เปิดล่าสุดจะถูกลบออก อันเป็นผลมาจาก โปรเซสการปิดไฟล์ หรือเนื่องจากการยุติโปรเซสที่เปิดไฟล์ อย่างใดอย่างหนึ่ง หากโปรเซส runaway กำลังเขียน ไปยังไฟล์และไฟล์ถูกลบออก บล็อกที่จัดสรรให้กับไฟล์นั้นยังไม่เป็นอิสระ จนกว่าโปรเซสยุติลง

ตัวอย่างเช่น:

```
$ ls -l
total 1248
-rwxrwxr-x    1 web  staff   1274770 Jul 20 11:19 datafile
$ date > datafile
$ ls -l
total 4
-rwxrwxr-x    1 web  staff      29 Jul 20 11:20 datafile
```

เอาต์พุตของคำสั่ง `date` แทนที่เนื้อหาก่อนหน้านี้ ของไฟล์ `datafile` บล็อกที่รายงานสำหรับ ไฟล์ที่ถูกตัดให้สั้นลงสะท้อนถึงความแตกต่างขนาดตั้งแต่ `1248>` ถึง `4` ถ้าโปรเซส runaway ทำการผนวกข้อมูลเข้ากับไฟล์ที่ถูกตัดใหม่นี้ต่อไป คำสั่ง `ls` ถัดไป จะให้ผลลัพธ์ต่อไปนี้:

```
$ ls -l
total 8
-rwxrwxr-x    1 web  staff   1278866 Jul 20 11:21 datafile
```

ขนาดของไฟล์ `datafile` สะท้อนการผนวก ที่ทำโดยโปรเซส runaway แต่จำนวนบล็อกที่จัดสรรมีเพียงเล็กน้อย ขณะนี้ไฟล์ `datafile` มีหุ้มน้อยภายใน หุ้มนั้นในไฟล์คือ พื้นที่ของไฟล์ที่ไม่มีติดต่อกับบล็อกซึ่งจัดสรรให้กับไฟล์

/ (ราก) โอเวอร์ไฟล์:

ตรวจสอบข้อมูลต่อไปนี้เมื่อระบบไฟล์ราก (/) เต็ม

- ใช้คำสั่งต่อไปนี้เพื่ออ่านเนื้อหาของไฟล์ `/etc/security/failedlogin`:

```
who /etc/security/failedlogin
```

เงื่อนไขของการสร้าง TTY ใหม่อย่างรวดเร็วสามารถสร้างรายการล็อกอินล้มเหลวได้เมื่อต้องการเคลียร์ไฟล์หลังจากการอ่านหรือบันทึกเอาต์พุตให้ดำเนินการคำสั่งต่อไปนี้:

```
cp /dev/null /etc/security/failedlogin
```

- ตรวจสอบไดเรกทอรี /dev เพื่อหาชื่ออุปกรณ์ที่พิมพ์ผิด หากมีการพิมพ์ชื่ออุปกรณ์ไม่ถูกต้อง เช่น rmt0 แทน rmt0 ไฟล์จะถูกสร้างขึ้นใน /dev ที่เรียกว่า rmt0 โดยปกติ คำสั่งจะทำต่อไปจนกว่าจะมีการเติมระบบไฟล์รากทั้งระบบ ก่อนจะล้มเหลว /dev เป็นส่วนหนึ่งของ ระบบไฟล์ราก (/) มองหารายการที่ไม่ใช่อุปกรณ์ (รายการที่ไม่มีหมายเลขหลักหรือรอง) เมื่อต้องการตรวจสอบสถานการณ์นี้ให้ใช้คำสั่งต่อไปนี้:

```
cd /dev  
ls -l | pg
```

ในที่นี้ตั้งเดียวกันกับที่จะบ่งชี้ขนาดไฟล์ของไฟล์ปกติ ไฟล์อุปกรณ์มีสองหมายเลขที่คั่นด้วย เครื่องหมายจุลภาค ตัวอย่างเช่น:

```
crw-rw-rw- 1 root system 12,0 Oct 25 10:19 rmt0
```

หากชื่อไฟล์หรือที่ตั้งขนาดบ่งชี้อุปกรณ์ที่ไม่ถูกต้อง ดังแสดง ในตัวอย่างต่อไปนี้ ให้ลบไฟล์ที่เชื่อมโยง:

```
crw-rw-rw- 1 root system 9375473 Oct 25 10:19 rmt0
```

หมายเหตุ:

- อย่าลบชื่ออุปกรณ์ที่ต้องการในไดเรกทอรี /dev ตัวอย่างเช่นชื่ออุปกรณ์ที่ไม่ถูกต้องคือ ขนาดไฟล์เชื่อมโยงที่ใหญ่กว่า 500 ไบต์
- หากการตรวจสอบระบบกำลังรันอยู่ ในไดเรกทอรี /audit ดีฟอลต์ สามารถเติมอย่างรวดเร็วและต้องการความสนใจ
- ตรวจสอบหาไฟล์ขนาดใหญ่มากซึ่งอาจลบออกได้โดยใช้คำสั่ง `find` ตัวอย่างเช่น เมื่อต้องการค้นหาไฟล์ทั้งหมดในไดเรกทอรีราก (/) ที่มีขนาดใหญ่กว่า 1 MB ให้ใช้คำสั่งต่อไปนี้:

```
find / -xdev -size +2048 -ls | sort -r -n +6
```

คำสั่งนี้ค้นหาไฟล์ทั้งหมดที่ใหญ่กว่า 1 MB และเรียงลำดับไฟล์เหล่านั้นในลำดับย้อนกลับโดยไฟล์ที่มีขนาดใหญ่ที่สุดแสดงขึ้นก่อน แพล็กอื่นๆ สำหรับคำสั่ง `find` เช่น `-newer` อาจเป็น ประโยชน์ในการค้นหาสำหรับข้อมูลรายละเอียด ใหญ่ที่คำอธิบายคำสั่ง ของคำสั่ง `find`

หมายเหตุ: เมื่อ ตรวจสอบไดเรกทอรีราก หมายเลขหลักและรองของอุปกรณ์ใน ไดเรกทอรี /dev จะแสดงด้วย ไฟล์จริง และขนาดไฟล์ หมายเลขหลักและรองซึ่งคั่นด้วย เครื่องหมายจุลภาค สามารถละเว้นได้

ก่อนลบไฟล์ใดๆ ให้ใช้คำสั่งต่อไปนี้เพื่อให้แน่ใจว่าไฟล์ไม่ได้ถูกใช้อยู่ในปัจจุบัน โดยโปรเซสผู้ใช้:

```
fuser filename
```

โดยที่ `filename` คือชื่อ ของไฟล์ที่สงสัยว่ามีขนาดใหญ่ หากไฟล์เปิดในเวลา ที่ ลบออก ไฟล์จะถูกลบออกจากรายการไดเรกทอรีเท่านั้น บล็อกที่จัดสรร ให้กับไฟล์นั้นยังคงไม่เป็นอิสระจนกว่าโปรเซสที่มีไฟล์เปิดจะ ถูก killed

การแก้ไขโอเวอร์ไฟล์ในระบบไฟล์ /var:

ตรวจสอบข้อมูลต่อไปนี้เมื่อระบบไฟล์ /var เต็ม

- คุณสามารถใช้คำสั่ง `find` เพื่อค้นหาไฟล์ขนาดใหญ่ในไดเรกทอรี /var ตัวอย่างเช่น:

```
find /var -xdev -size +2048 -ls | sort -r +6
```

สำหรับข้อมูลรายละเอียด ใหญ่ที่คำอธิบายคำสั่ง ของคำสั่ง `find`

- ตรวจสอบหาไฟล์ที่เก่าหรือที่เหลืออยู่ใน /var/tmp
- ตรวจสอบขนาดของไฟล์ /var/adm/wtmp ซึ่งบันทึก เซสชันล็อกอิน, rlogins และ telnet ทั้งหมด บันทึกจะเพิ่มขึ้นอย่างไม่สิ้นสุด ยกเว้นว่าบัญชีระบบกำลังรัน บัญชีระบบจะล้างข้อมูลบันทึกทุกคืน ไฟล์ /var/adm/wtmp สามารถถูกล้างข้อมูลหรือแก้ไข เพื่อลบข้อมูลเก่าและที่ไม่ต้องการ เมื่อต้องการล้างข้อมูลดังกล่าว ให้ใช้คำสั่งต่อไปนี้:

```
cp /dev/null /var/adm/wtmp
```

เมื่อต้องการแก้ไขไฟล์ /var/adm/wtmp อันดับแรก ให้คัดลอกไฟล์เป็นการชั่วคราว โดยใช้คำสั่งต่อไปนี้:

```
/usr/sbin/acct/fwtmp < /var/adm/wtmp >/tmp/out
```

แก้ไขไฟล์ /tmp/out เพื่อลบรายการที่ไม่ต้องการจากนั้นแทนที่ไฟล์ต้นฉบับด้วยคำสั่งต่อไปนี้:

```
/usr/sbin/acct/fwtmp -ic < /tmp/out > /var/adm/wtmp
```

- ล้างข้อมูลบันทึกข้อผิดพลาดในไอดีเร็กทอรี /var/adm/ras โดยใช้โปรแกรมต่อไปนี้ บันทึกข้อผิดพลาดไม่เคยมีการล้างข้อมูลยกเว้นวาทูก ล้างด้วยตนเอง

หมายเหตุ: ห้ามใช้คำสั่ง cp /dev/null เพื่อล้างข้อมูลบันทึกข้อผิดพลาด ไฟล์ errlog ที่มีความยาวเป็นศูนย์ปิดใช้งานฟังก์ชันการบันทึกข้อผิดพลาดของระบบปฏิบัติการ และต้องถูกแทนที่จากสำเนาสำรอง

1. หยุด daemon ข้อผิดพลาดโดยใช้คำสั่งต่อไปนี้:

```
/usr/lib/errstop
```

2. ลบหรือย้ายล็อกไฟล์ข้อผิดพลาดไปยังระบบไฟล์อื่นโดยใช้คำสั่ง อย่างใดอย่างหนึ่งต่อไปนี้:

```
rm /var/adm/ras/errlog
```

หรือ

```
mv /var/adm/ras/errlog filename
```

โดยที่ filename คือชื่อของไฟล์ errlog ที่ย้าย

หมายเหตุ: ข้อมูลข้อผิดพลาดในอดีตจะถูกลบออก ถ้าคุณลบล็อกไฟล์ข้อผิดพลาด

3. รีสตาร์ท daemon ข้อผิดพลาดโดยใช้คำสั่งต่อไปนี้:

```
/usr/lib/errdemom
```

หมายเหตุ: พิจารณาการจำกัด errlog โดยรันรายการต่อไปนี้ใน cron:

```
0 11 * * * /usr/bin/errclear -d S,0 30
```

```
0 12 * * * /usr/bin/errclear -d H 90
```

- ตรวจสอบว่าไฟล์ trcfile ในไอดีเร็กทอรีนี้มีขนาดใหญ่หรือไม่ หากไฟล์มีขนาดใหญ่และไม่ได้กำลังรันการติดตามในปัจจุบัน คุณสามารถลบไฟล์โดยใช้คำสั่งต่อไปนี้:

```
rm /var/adm/ras/trcfile
```

- หากอุปกรณ์ดัมพ์ของคุณมีการตั้งค่าเป็น hd6 (ซึ่งเป็นค่าดีฟอลต์) อาจมีไฟล์ vmcore* จำนวนหนึ่งอยู่ในไอดีเร็กทอรี /var/adm/ras หากวันที่ของไฟล์ดังกล่าวเก่าหรือคุณไม่ต้องการเก็บไฟล์เหล่านั้นไว้ คุณสามารถลบไฟล์เหล่านั้นได้ด้วยคำสั่ง

rm

- ตรวจสอบไอดีเร็กทอรี /var/spool ซึ่งมีไฟล์ระบบย่อยที่จัดคิว ล้างระบบย่อยที่จัดคิวโดยใช้คำสั่งต่อไปนี้:

```
stopsrc -s qdaemon
```

```
rm /var/spool/lpd/qdir/*
```

```
rm /var/spool/lpd/stat/*
```

```
rm /var/spool/qdaemon/*
```

```
startsrc -s qdaemon
```

- ตรวจสอบไดเรกทอรี `/var/adm/acct` ซึ่งมี เร็กคอร์ดการบัญชี ถ้าการบัญชีกำลังรัน ไดเรกทอรีนี้อาจมีไฟล์ขนาดใหญ่หลายไฟล์
- ตรวจสอบไดเรกทอรี `/var/preserve` เพื่อหาเซสชัน `vi` ที่ยุติโดยทั่วไป ปลอดภัยที่จะลบไฟล์เหล่านี้ หากผู้ใช้ต้องการกู้คืนเซสชัน คุณสามารถใช้คำสั่ง `vi -r` เพื่อแสดงรายการเซสชันที่กู้คืนได้ ทั้งหมด เมื่อต้องการกู้คืนเซสชันเฉพาะ ให้ใช้ `vi -r filename`
- แก้ไขไฟล์ `/var/adm/sulog` ซึ่งบันทึกจำนวนของการใช้คำสั่ง `su` ที่พยายามทำและแสดงว่าความพยายามแต่ละครั้งสำเร็จหรือไม่ นี่เป็นไฟล์ flat และสามารถดูและแก้ไขได้ด้วย โปรแกรมแก้ไขที่โปรดปราน หากไฟล์นี้ถูกลบออก ไฟล์จะถูกสร้างขึ้นใหม่โดยคำสั่ง `su` ที่พยายามครั้งถัดไป แก้ไข `/var/tmp/snmpd.log` ซึ่งบันทึกเหตุการณ์จาก `snmpd` daemon หากไฟล์ถูกลบออก ไฟล์จะถูกสร้างขึ้นใหม่โดย `snmpd` daemon

หมายเหตุ: ขนาดของไฟล์ `/var/tmp/snmpd.log` สามารถจำกัดได้เพื่อไม่ให้เติบโตอย่างไม่สิ้นสุด แก้ไขไฟล์ `/etc/snmpd.conf` เพื่อเปลี่ยนจำนวน (ในหน่วยไบต์) ในส่วนที่เหมาะสมสำหรับขนาด

หลักการที่เกี่ยวข้อง:

“การทำแอคเคาต์ระบบ” ในหน้า 174

ยุติสิทธิ์การทำแอคเคาต์ระบบอนุญาตให้คุณรวบรวมและรายงานในการใช้งานแต่ละคน หรือกลุ่มของรีซอร์สระบบต่างๆ

การแก้ไขระบบไฟล์อื่นและเทคนิคการค้นหาทั่วไป:

ใช้คำสั่ง `find` พร้อมด้วยแฟล็ก `-size` เพื่อระบุตำแหน่งไฟล์ขนาดใหญ่ หรือถ้าระบบไฟล์โอเวอร์โฟลล์เมื่อเร็วๆ นี้ ให้ใช้แฟล็ก `-newer` เพื่อค้นหาไฟล์ที่แก้ไขล่าสุด

เมื่อต้องการจัดทำไฟล์เพื่อให้แฟล็ก `-newer` ค้นหา ให้ใช้คำสั่ง `touch` ต่อไปนี้:

```
touch mmdhmm filename
```

โดยที่ `mm` คือเดือน, `dd` คือวันที่, `hh` คือชั่วโมงในรูปแบบ 24-ชั่วโมง, `mm` คือ นาที, และ `filename` คือชื่อไฟล์ซึ่งคุณกำลังสร้างด้วยคำสั่ง `touch`

หลังจากคุณสร้างไฟล์ touched แล้ว คุณสามารถใช้คำสั่งต่อไปนี้ เพื่อค้นหาไฟล์ขนาดใหญ่ที่ใหม่กว่า:

```
find /filesystem_name -xdev -newer touch_filename -ls
```

คุณยังสามารถใช้คำสั่ง `find` เพื่อระบุตำแหน่งไฟล์ที่มีการเปลี่ยนแปลงในเวลา 24 ชั่วโมงล่าสุดได้ด้วย ดังแสดงในตัวอย่างต่อไปนี้:

```
find /filesystem_name -xdev -mtime 0 -ls
```

การติดตั้ง

การติดตั้งทำให้ระบบไฟล์ ไดรฟ์ ไดเรกทอรี อุปกรณ์ และไฟล์พิเศษพร้อมใช้งานที่ติดตั้งเฉพาะ การติดตั้งนับเป็นวิธีเดียวที่ทำให้ระบบไฟล์สามารถเข้าถึงได้

คำสั่ง `mount` สั่งให้ระบบปฏิบัติการติดตั้งระบบไฟล์ที่ไดเรกทอรีซึ่งระบุ

คุณสามารถติดตั้งไฟล์หรือไดเรกทอรี ถ้าคุณมีสิทธิ์เข้าถึงไฟล์หรือไดเรกทอรีที่กำลังติดตั้งและสิทธิ์การเขียนสำหรับจุดต่อสมาชิกของกลุ่มระบบ ยังสามารถทำการติดตั้งอุปกรณ์ (ซึ่งอุปกรณ์หรือระบบไฟล์มีการติดตั้ง ข้างบนไดเรกทอรี) และการติด

ตั้งชื่ออธิบายไว้ในไฟล์ /etc/filesystems ผู้ใช้ที่มีสิทธิผู้ใช้สามารถติดตั้งระบบไฟล์ได้ตามความต้องการ โดยการตั้งชื่อทั้งอุปกรณ์และไดเรกทอรีบนบรรทัดคำสั่ง ไฟล์ /etc/filesystems ใช้เพื่อกำหนดการติดตั้งที่จะทำโดยอัตโนมัติเมื่อเริ่มต้นระบบ คำสั่ง `mount` ใช้เพื่อติดตั้งหลังจากสแตร์ทอ์ระบบ

จุดต่อ:

จุดต่อ คือไดเรกทอรีหรือไฟล์ซึ่งทำให้ระบบไฟล์ ไดเรกทอรี หรือไฟล์ใหม่สามารถเข้าถึงได้ เมื่อต้องการติดตั้งระบบไฟล์หรือไดเรกทอรี จุดต่อต้องเป็นไดเรกทอรี และเมื่อต้องการติดตั้งไฟล์ จุดต่อ ต้องเป็นไฟล์

โดยปกติ ระบบไฟล์ ไดเรกทอรี หรือไฟล์มีการติดตั้งบนจุดต่อที่ว่าง แต่ไม่ใช่ข้อบังคับ หากไฟล์หรือไดเรกทอรีที่ทำหน้าที่เป็นจุดต่อ มีข้อมูลอยู่ ข้อมูลนั้นจะไม่สามารถเข้าถึงได้ในขณะที่ติดตั้ง ไฟล์หรือไดเรกทอรีอื่น ส่งผลให้ไฟล์หรือไดเรกทอรีที่ติดตั้งมีเนื้อหาเท่าที่เคยมียูก่อนหน้านั้นในไดเรกทอรีนั้น ไดเรกทอรีหรือไฟล์ดั้งเดิม ที่ถูกติดตั้งจะสามารถเข้าถึงได้อีกครั้งหลังจากการติดตั้งบนนั้นถูกเลิกทำ

เมื่อติดตั้งระบบไฟล์บนไดเรกทอรี สิทธิอนุญาตของไดเรกทอรีรากของ ระบบไฟล์ที่ติดตั้งมีความสำคัญมากกว่าสิทธิอนุญาตของ จุดต่อ ข้อยกเว้นหนึ่งคือรายการไดเรกทอรีพาเรนต์ .. (จุดจุด) ใน ไดเรกทอรีที่มีการติดตั้ง เพื่อให้ระบบปฏิบัติการเข้าถึงระบบไฟล์ใหม่ ข้อมูลไดเรกทอรีพาเรนต์ของจุดต่อ ต้องมีอยู่

ตัวอย่างเช่น ถ้าไดเรกทอรีการทำงานปัจจุบันเป็น /home/frank คำสั่ง `cd ..` จะเปลี่ยนไดเรกทอรีการทำงานเป็น /home หากไดเรกทอรี /home/frank เป็นรากของระบบไฟล์ที่ติดตั้ง ระบบปฏิบัติการต้องค้นหาข้อมูลไดเรกทอรีพาเรนต์ใน ไดเรกทอรี /home/frank เพื่อให้คำสั่ง `cd ..` สำเร็จ

สำหรับคำสั่งใดๆ ที่ต้องการข้อมูลไดเรกทอรีพาเรนต์ ผู้ใช้ต้องมีสิทธิการค้นหาในไดเรกทอรีที่มีการติดตั้ง ความล้มเหลวของไดเรกทอรีที่มีการติดตั้งในการมอบสิทธิการค้นหาอาจทำให้เกิดผล ที่คาดการณ์ไม่ได้ โดยเฉพาะอย่างยิ่ง ถ้ามองไม่เห็นสิทธิในไดเรกทอรี ที่มีการติดตั้ง ปัญหาทั่วไปคือความล้มเหลวของคำสั่ง `pwd` หากไม่มีสิทธิการค้นหาในไดเรกทอรีที่มีการติดตั้ง คำสั่ง `pwd` จะ ส่งคืนข้อความนี้:

```
pwd: Permission denied
```

ปัญหานี้สามารถหลีกเลี่ยงได้โดยการตั้งค่าสิทธิอนุญาตในไดเรกทอรีที่มีการติดตั้ง เป็นอย่างน้อย 111 เสมอ

การติดตั้งระบบไฟล์ ไดเรกทอรี และไฟล์:

การติดตั้งมีอยู่สองชนิดคือ การติดตั้งรีโมตและการติดตั้งโลคัล การติดตั้ง รีโมต ทำบนระบบรีโมตซึ่งข้อมูลมีการส่งผ่านบนเส้นโทรคมนาคม ระบบไฟล์รีโมต เช่น ระบบไฟล์เน็ตเวิร์ก (NFS) กำหนดให้เอ็กซ์พอร์ตไฟล์ก่อนสามารถติดตั้งไฟล์นั้นได้ การติดตั้งโลคัล คือ การติดตั้งที่ทำบนระบบโลคัลของคุณ

แต่ละระบบไฟล์มีการเชื่อมโยงกับอุปกรณ์ (โลจิคัลวอลุ่ม) ที่แตกต่างกัน ก่อนคุณสามารถใช้ระบบไฟล์ได้ ต้องเชื่อมต่อระบบไฟล์เข้ากับโครงสร้างไดเรกทอรี ที่มีอยู่ (ระบบไฟล์รากหรือระบบไฟล์อื่นซึ่งเชื่อมต่อ ไว้แล้ว) คำสั่ง `mount` ทำการเชื่อมต่อนี้

ระบบไฟล์ ไดเรกทอรี หรือไฟล์เดียวกันสามารถเข้าถึงได้จากหลายพาร ตัวอย่างเช่น ถ้าคุณมีหนึ่งฐานข้อมูลและมีผู้ใช้ที่ใช้งานข้อมูลนี้อยู่หลายราย อาจเป็นประโยชน์ที่จะมีการติดตั้งฐานข้อมูลเดียวกันในหลายจุด แต่ละการติดตั้งควรมี ชื่อและรหัสผ่านของตนเองสำหรับวัตถุประสงค์ในการติดตามและการแบ่งแยกงาน การดำเนินการนี้ สามารถบรรลุผลได้โดยการติดตั้งระบบไฟล์เดียวกันบนจุดต่อที่แตกต่างกัน ตัวอย่างเช่น คุณสามารถติดตั้งจาก /home/server/database ไปยัง จุดต่อที่ระบุเป็น /home/user1, /home/user2, และ /home/user3:

```
/home/server/database      /home/user1
/home/server/database      /home/user2
/home/server/database      /home/user3
```

ระบบไฟล์ไต่เรียกทอรี หรือไฟล์สามารถทำให้มีอยู่สำหรับผู้ใช้งานต่างๆ โดยการใช้ลิงก์สัญลักษณ์ ลิงก์สัญลักษณ์มีการสร้างขึ้นด้วยคำสั่ง `ln -s` การลิงก์หลายผู้ใช้กับไฟล์ส่วนกลางช่วยให้มั่นใจว่า การเปลี่ยนแปลงทั้งหมดในไฟล์ มีการสะท้อนให้เห็นในแต่ละครั้งที่ผู้ใช้เข้าถึงไฟล์

ตัวควบคุมการติดตั้งอัตโนมัติ:

สามารถตั้งค่าการติดตั้งให้เกิดขึ้นโดยอัตโนมัติในระหว่างการเริ่มต้นระบบ

การติดตั้งอัตโนมัติมีอยู่สองชนิด ชนิดแรกประกอบด้วยการติดตั้งที่จำเป็นต่อบูตและรันระบบ ระบบไฟล์เหล่านี้มีการติดตั้งอย่างชัดเจนโดยโปรเซสบูต Stanzas ของระบบไฟล์ดังกล่าวในไฟล์ `/etc/filesystems` มี `mount = automatic` ชนิดที่สองของการติดตั้งอัตโนมัติคือแบบควบคุมโดยผู้ใช้ ระบบไฟล์เหล่านี้มีการติดตั้งโดยสคริปต์ `/etc/rc` เมื่อออกใช้คำสั่ง `mount all` Stanzas ของการติดตั้งอัตโนมัติที่ควบคุมโดยผู้ใช้นี้มี `mount = true` ใน `/etc/filesystems`

ไฟล์ `/etc/filesystems` ควบคุมการติดตั้งอัตโนมัติ มีการดำเนินการตามลำดับชั้นที่ละหนึ่งจุดติดตั้ง นอกจากนี้ยังสามารถวางในลำดับเฉพาะที่สามารถเปลี่ยนและจัดเรียงใหม่ได้ด้วย สำหรับข้อมูลเพิ่มเติมเกี่ยวกับไฟล์ `/etc/filesystems` ให้ดูที่ `/etc/filesystems`

ไฟล์ `/etc/filesystems` มีการจัดระเบียบเป็น stanzas โดยมีหนึ่ง stanza สำหรับแต่ละการติดตั้ง Stanza อธิบายแอตทริบิวต์ของระบบไฟล์ที่สอดคล้องกัน และวิธีการติดตั้ง ระบบจะติดตั้งระบบไฟล์ในลำดับที่ปรากฏขึ้นในไฟล์ `/etc/filesystems` ข้อมูลต่อไปนี้เป็นตัวอย่างของ stanzas ภายในไฟล์ `/etc/filesystems`:

```
/:
dev=/dev/hd4
vol="root"
mount=automatic
check=false
free=true
vfs=jfs
log=/dev/hd8
type=bootfs

/home:
dev=/dev/hd1
vfs=jfs
log=/dev/hd8
mount=true
check=true
vol="/home"
free=false

/usr:
/dev=/dev/hd2
vfs=jfs
log=/dev/hd8
mount=automatic
```

```
check=false
type=bootfs
vol="/usr"
free=false
```

คุณสามารถแก้ไขไฟล์ `/etc/filesystems` เพื่อควบคุม ลำดับของการติดตั้ง หากการติดตั้งไม่สำเร็จ การติดตั้งใดๆ ในลำดับต่อมาที่กำหนดไว้ในไฟล์ `/etc/filesystems` จะติดตั้งต่อไป ตัวอย่างเช่น ถ้าการติดตั้งของระบบไฟล์ `/home` ไม่สำเร็จ การติดตั้งของระบบไฟล์ `/usr` จะ ดำเนินการติดตั้งต่อไป การติดตั้งอาจไม่สำเร็จด้วยเหตุผลต่างๆ เช่น ข้อผิดพลาด การพิมพ์ การฟังหา หรือปัญหาของระบบ

ความปลอดภัยในการติดตั้งสำหรับเวิร์กสเตชันแบบไม่มีดิสก์:

เวิร์กสเตชันแบบไม่มีดิสก์ต้องมีความสามารถในการสร้างและเข้าถึง ไฟล์พิเศษของอุปกรณ์บนเครื่องรีโมตเพื่อให้มีไดเรกทอรี `/dev` ที่ติดตั้งจากเซิร์ฟเวอร์ เนื่องจากเซิร์ฟเวอร์ไม่สามารถแยกความแตกต่างระหว่างไฟล์พิเศษของ อุปกรณ์ที่ใช้สำหรับไคลเอ็นต์กับที่ใช้สำหรับเซิร์ฟเวอร์ ผู้ใช้บนเซิร์ฟเวอร์จึง อาจสามารถเข้าถึงอุปกรณ์ฟิสิคัลของเซิร์ฟเวอร์โดยใช้ไฟล์พิเศษของอุปกรณ์ไคลเอ็นต์ได้

ตัวอย่างเช่น ความเป็นเจ้าของสำหรับ `tty` มีการตั้งค่าเป็นผู้ใช้ที่กำลัง ใช้ `tty` โดยอัตโนมัติ หาก ID ผู้ใช้บนไคลเอ็นต์และเซิร์ฟเวอร์ไม่เหมือนกัน ผู้ใช้ที่ไม่มีสิทธิบนเซิร์ฟเวอร์สามารถเข้าถึง `tty` ที่กำลังใช้ อยู่โดยผู้ใช้บนเซิร์ฟเวอร์ได้

ผู้ใช้ที่มีสิทธิบนไคลเอ็นต์สามารถสร้างไฟล์พิเศษของอุปกรณ์เพื่อให้ตรงกับ อุปกรณ์ฟิสิคัลบนเซิร์ฟเวอร์และไม่ต้องมีสิทธิพิเศษสำหรับการเข้าถึง จากนั้น ผู้ใช้สามารถใช้แอคเคสท์ที่ไม่มีสิทธิบนเซิร์ฟเวอร์เพื่อเข้าถึง อุปกรณ์ที่ป้องกันตามปกติที่ใช้ไฟล์พิเศษของอุปกรณ์ใหม่ได้

ปัญหาด้านความปลอดภัยที่คล้ายกันเกี่ยวข้องกับการใช้โปรแกรม `setuid` และ `setgid` บนไคลเอ็นต์และเซิร์ฟเวอร์ไคลเอ็นต์ที่ไม่มีดิสก์ต้องสามารถสร้างและรันโปรแกรม `setuid` และ `setgid` บนเซิร์ฟเวอร์สำหรับการดำเนินงานปกติ อีกครั้งที่เซิร์ฟเวอร์ไม่สามารถแยกความแตกต่างระหว่าง โปรแกรมที่ใช้สำหรับเซิร์ฟเวอร์กับที่ใช้สำหรับไคลเอ็นต์

นอกจากนี้ IDs ผู้ใช้และ IDs กลุ่มอาจไม่ตรงกันระหว่างเซิร์ฟเวอร์และไคลเอ็นต์ ดังนั้นผู้ใช้บนเซิร์ฟเวอร์จึงอาจสามารถรันโปรแกรมที่มีความสามารถ ซึ่งไม่ได้ใช้สำหรับผู้ใช้เหล่านั้น

ปัญหามีอยู่เนื่องจากโปรแกรม `setuid` และ `setgid` และไฟล์พิเศษของอุปกรณ์ควรจะใช้ได้เฉพาะบนเครื่องที่สร้างโปรแกรมหรือไฟล์ เหล่านั้นขึ้นเท่านั้น

วิธีแก้ไขคือการใช้อ็อปชันความปลอดภัยของคำสั่ง `mount` ที่จำกัดความสามารถในการใช้อ็อบเจกต์เหล่านี้ อ็อปชันเหล่านี้ยังสามารถใช้ใน `stanzas` ในไฟล์ `/etc/filesystems` ได้ด้วย

อ็อปชัน `nosuid` ในคำสั่ง `mount` ป้องกันการดำเนินการของโปรแกรม `setuid` และ `setgid` ซึ่งเข้าถึงผ่านทาง ระบบไฟล์ที่ติดตั้งที่ได้ อ็อปชันนี้ใช้สำหรับระบบไฟล์ใดๆ ที่กำลังติดตั้งบนโฮสต์เฉพาะสำหรับการใช้งานโดยโฮสต์อื่น (ตัวอย่างเช่น ที่เอ็กซ์พอร์ตสำหรับไคลเอ็นต์ที่ไม่มีดิสก์)

อ็อปชัน `nodev` ในคำสั่ง `mount` ป้องกันการเปิด อุปกรณ์ที่ใช้ไฟล์พิเศษของอุปกรณ์ซึ่งเข้าถึงผ่านทาง ระบบไฟล์ที่ติดตั้งที่ได้ อ็อปชันนี้ยังใช้สำหรับระบบไฟล์ใดๆ ที่กำลังติดตั้งสำหรับการใช้งานโดยโฮสต์อื่น (ตัวอย่างเช่น ที่เอ็กซ์พอร์ตสำหรับไคลเอ็นต์ที่ไม่มีดิสก์)

โดยทั่วไป ผู้ใช้บนเซิร์ฟเวอร์ไม่มีสิทธิเข้าถึงใดๆ ในไดเรกทอรี `/export`

การเอ็กซ์พอร์ตไดเรกทอรี /export/root

ไดเรกทอรี /export/root ต้องถูกเอ็กซ์พอร์ตด้วยสิทธิ์ การอ่าน/เขียน และผู้ใช้รากบนเซิร์ฟเวอร์ต้องมีสิทธิ์เข้าถึงได้อย่างไร้ที่ตาม คุณอาจต้องการติดตั้งไดเรกทอรีนี้ด้วยอ็อปชันต่อไปนี้ ของคำสั่ง **mount**:

ไอเท็ม	คำอธิบาย
nosuid	ป้องกันไม่ให้ผู้ใช้บนเซิร์ฟเวอร์รันโปรแกรม setuid ของ โคลเอ็นต์
nodev	ป้องกันไม่ให้ผู้ใช้เข้าถึงอุปกรณ์เซิร์ฟเวอร์โดยใช้ไฟล์พิเศษอุปกรณ์ของโคลเอ็นต์

ทางเลือกของการติดตั้งไดเรกทอรี /export/root พร้อมด้วยอ็อปชันเหล่านี้คือ หลีกเลี่ยงการให้สิทธิ์เข้าถึงไดเรกทอรี /export/root แก่ผู้ใช้ที่กำลังรันบนเซิร์ฟเวอร์

การเอ็กซ์พอร์ตไดเรกทอรี /export/exec

ไดเรกทอรี /export/exec ถูกเอ็กซ์พอร์ตด้วยสิทธิ์อ่านอย่างเดียว และต้องให้สิทธิ์เข้าถึงราก อย่างไรก็ตาม คุณอาจต้องการติดตั้งไดเรกทอรีนี้ด้วยอ็อปชันต่อไปนี้ ของคำสั่ง **mount**:

ไอเท็ม	คำอธิบาย
nosuid	ป้องกันไม่ให้ผู้ใช้บนเซิร์ฟเวอร์รันโปรแกรม setuid ของ โคลเอ็นต์ หากคุณกำลังเอ็กซ์พอร์ตไดเรกทอรีเซิร์ฟเวอร์ /usr คุณไม่สามารถใช้อ็อปชัน nosuid ได้
nodev	ป้องกันไม่ให้ผู้ใช้เข้าถึงอุปกรณ์เซิร์ฟเวอร์โดยใช้ไฟล์พิเศษอุปกรณ์ของโคลเอ็นต์

การเอ็กซ์พอร์ตไดเรกทอรี /export/share

ไดเรกทอรี /export/share ถูกเอ็กซ์พอร์ตด้วยสิทธิ์อ่านอย่างเดียว และต้องให้สิทธิ์เข้าถึงราก เนื่องจากโดยทั่วไปไดเรกทอรีนี้มีข้อมูลอย่างเดียว (ไม่มีไฟล์ปฏิบัติการหรืออุปกรณ์) คุณจึงไม่ต้องใช้อ็อปชันความปลอดภัยของการติดตั้ง

การเอ็กซ์พอร์ตไดเรกทอรี /export/home

มีหลายวิธีในการติดตั้งไดเรกทอรีผู้ใช้ /home:

- คุณสามารถติดตั้งไดเรกทอรี /export/home/*Clienthostname* ข้างบนไดเรกทอรีโคลเอ็นต์ /home ในกรณีนี้ โคลเอ็นต์ มีสิทธิ์การอ่าน/เขียนและผู้ใช้รากมีสิทธิ์เข้าถึงได้ เพื่อให้มั่นใจถึงความปลอดภัยของ ระบบ ให้ติดตั้งไดเรกทอรี /export/home ด้วยอ็อปชันต่อไปนี้ ของคำสั่ง **mount**:

ไอเท็ม	คำอธิบาย
nosuid	ป้องกันไม่ให้ผู้ใช้บนเซิร์ฟเวอร์รันโปรแกรม setuid ของ โคลเอ็นต์
nodev	ป้องกันไม่ให้ผู้ใช้เข้าถึงอุปกรณ์เซิร์ฟเวอร์โดยใช้ไฟล์พิเศษอุปกรณ์ของโคลเอ็นต์

- คุณสามารถติดตั้งไดเรกทอรี /home บนเซิร์ฟเวอร์ข้างบน ไดเรกทอรี /home ของโคลเอ็นต์ ในกรณีนี้ ไดเรกทอรี /home ถูกเอ็กซ์พอร์ตด้วยสิทธิ์การอ่าน/เขียนและไม่มีการเข้าถึง เพื่อให้มั่นใจถึงความปลอดภัยของระบบ ให้ติดตั้งไดเรกทอรี /home บนทั้ง เซิร์ฟเวอร์และโคลเอ็นต์ด้วยอ็อปชัน **nosuid** และ **nodev** ของคำสั่ง **mount**
- หรือคุณสามารถติดตั้งบนโคลเอ็นต์ แต่ละไดเรกทอรี /home/*UserName* บนเซิร์ฟเวอร์ข้างบนไดเรกทอรี /home/*Username* บน โคลเอ็นต์เพื่อให้ผู้ใช้สามารถล็อกอินเข้าสู่เครื่องอื่นและยังคงมีสิทธิ์เข้าถึง ไดเรกทอรีโฮมของตนได้ ในกรณีนี้ ไดเรกทอรี /home/*Username* ทั้งบนเซิร์ฟเวอร์และโคลเอ็นต์มีการติดตั้งด้วยอ็อปชัน **nosuid** และ **nodev** ของคำสั่ง **mount**

การเอ็กซ์พอร์ตไดเรกทอรี /export/swap

เอ็กซ์พอร์ตไฟล์ /export/swap/*Clienthostname* ด้วยสิทธิการอ่าน/เขียนและสิทธิการ ไม่จำเป็นต้องมีมาตรการรักษาความปลอดภัย ผู้ใช้บนเซิร์ฟเวอร์ไม่มีสิทธิเข้าถึงใดๆ ในไฟล์ /export/swap/*Clienthostname*

การติดตั้งแบบไม่มีดิสก์:

แม้ว่าระบบไฟล์ของเวิร์กสเตชันที่ไม่มีดิสก์มีการติดตั้งจาก ไดเรกทอรีเซิร์ฟเวอร์ / เอ็กซ์พอร์ต ไปยังเครื่องที่ไม่มีดิสก์ แต่ระบบไฟล์ดูเหมือนกับระบบไฟล์บนเครื่อง สแตนด์อะโลน

ข้อมูลต่อไปนี้แสดงความสัมพันธ์ระหว่างเอ็กซ์พอร์ตเซิร์ฟเวอร์และจุดต่อ เวิร์กสเตชันแบบไม่มีดิสก์:

เอ็กซ์พอร์ตเซิร์ฟเวอร์	อิมพอร์ตแบบไม่มีดิสก์
/export/root/ <i>HostName</i>	/ (ราก)
/export/exec/ <i>SPOTName</i>	/usr
/export/home/ <i>HostName</i>	/home
/export/share	/usr/share
/export/dump	ใช้โดยไคลเอ็นต์ที่ไม่มีดิสก์เป็นพื้นที่วางดัมพ์
/export/swap	ใช้โดยไคลเอ็นต์ที่ไม่มีดิสก์เป็นพื้นที่การเลื่อนหน้าแบบรีโมต

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับไดเรกทอรี /export ให้ดูที่ “ไดเรกทอรี /export” ในหน้า 472

โดยทั่วไป ผู้ใช้บนเซิร์ฟเวอร์ไม่มีสิทธิเข้าถึงใดๆ ในไดเรกทอรี /export

การเอ็กซ์พอร์ตไดเรกทอรี /export/root

ไดเรกทอรี /export/root ต้องถูกเอ็กซ์พอร์ตด้วยสิทธิ การอ่าน/เขียน และผู้ใช้รากบนเซิร์ฟเวอร์ต้องมีสิทธิเข้าถึงได้อย่างไรก็ตาม คุณอาจต้องการติดตั้งไดเรกทอรีนี้ด้วยอ็อปชันต่อไปนี้ ของคำสั่ง **mount**:

ไอเท็ม	คำอธิบาย
nosuid	ป้องกันไม่ให้ผู้ใช้บนเซิร์ฟเวอร์รันโปรแกรม setuid ของ ไคลเอ็นต์
nodev	ป้องกันไม่ให้ผู้ใช้เข้าถึงอุปกรณ์เซิร์ฟเวอร์โดยใช้ไฟล์พิเศษอุปกรณ์ของไคลเอ็นต์

ทางเลือกของการติดตั้งไดเรกทอรี /export/root พร้อมด้วยอ็อปชันเหล่านี้คือ หลีกเลี่ยงการให้สิทธิเข้าถึงไดเรกทอรี /export/root แก่ผู้ใช้ที่กำลังรันบนเซิร์ฟเวอร์

การเอ็กซ์พอร์ตไดเรกทอรี /export/exec

ไดเรกทอรี /export/exec ถูกเอ็กซ์พอร์ตด้วยสิทธิอ่านอย่างเดียว และต้องให้สิทธิเข้าถึงราก อย่างไรก็ตาม คุณอาจต้องการติดตั้งไดเรกทอรีนี้ด้วยอ็อปชันต่อไปนี้ ของคำสั่ง **mount**:

ไอเท็ม	คำอธิบาย
nosuid	ป้องกันไม่ให้ผู้ใช้บนเซิร์ฟเวอร์รันโปรแกรม <code>setuid</code> ของ ไคลเอ็นต์ หากคุณกำลังเอ็กซ์พอร์ตไดเรกทอรีเซิร์ฟเวอร์ /usr คุณไม่สามารถใช้อ็อปชัน <code>nosuid</code> ได้
nodev	ป้องกันไม่ให้ผู้ใช้เข้าถึงอุปกรณ์เซิร์ฟเวอร์โดยใช้ไฟล์พิเศษอุปกรณ์ของไคลเอ็นต์

การเอ็กซ์พอร์ตไดเรกทอรี /export/share

ไดเรกทอรี /export/share ถูกเอ็กซ์พอร์ตด้วยสิทธิ์อ่านอย่างเดียว และต้องให้สิทธิ์เข้าถึงราก เนื่องจากโดยทั่วไปไดเรกทอรีนี้มีข้อมูลอย่างเดียว (ไม่มีไฟล์ปฏิบัติการหรืออุปกรณ์) คุณจึงไม่ต้อง ใช้อ็อปชันความปลอดภัยของการติดตั้ง

การเอ็กซ์พอร์ตไดเรกทอรี /export/home

มีหลายวิธีในการติดตั้งไดเรกทอรีผู้ใช้ /home:

- คุณสามารถติดตั้งไดเรกทอรี /export/home/*Clienthostname* ข้างบนไดเรกทอรีไคลเอ็นต์ /home ในกรณีนี้ไคลเอ็นต์ มีสิทธิ์การอ่าน/เขียนและผู้ใช้รากมีสิทธิ์เข้าถึงได้ เพื่อให้มั่นใจถึงความปลอดภัยของระบบ ให้ติดตั้งไดเรกทอรี /export/home ด้วยอ็อปชันต่อไปนี้ ของคำสั่ง `mount`:

ไอเท็ม	คำอธิบาย
nosuid	ป้องกันไม่ให้ผู้ใช้บนเซิร์ฟเวอร์รันโปรแกรม <code>setuid</code> ของ ไคลเอ็นต์
nodev	ป้องกันไม่ให้ผู้ใช้เข้าถึงอุปกรณ์เซิร์ฟเวอร์โดยใช้ไฟล์พิเศษอุปกรณ์ของไคลเอ็นต์

- คุณสามารถติดตั้งไดเรกทอรี /home บนเซิร์ฟเวอร์ข้างบน ไดเรกทอรี /home ของไคลเอ็นต์ ในกรณีนี้ไดเรกทอรี /home ถูกเอ็กซ์พอร์ตด้วยสิทธิ์การอ่าน/เขียนและไม่มีสิทธิ์ราก เพื่อให้มั่นใจถึง ความปลอดภัยของระบบ ให้ติดตั้งไดเรกทอรี /home บนทั้ง เซิร์ฟเวอร์และไคลเอ็นต์ด้วยอ็อปชัน `nosuid` และ `nodev` ของคำสั่ง `mount`
- หรือคุณสามารถติดตั้งบนไคลเอ็นต์แต่ละไดเรกทอรี /home/*UserName* บนเซิร์ฟเวอร์ข้างบนไดเรกทอรี /home/*Username* บน ไคลเอ็นต์เพื่อให้ผู้ใช้สามารถล็อกอินเข้าสู่เครื่องอื่นและยังคงมีสิทธิ์เข้าถึง ไดเรกทอรีโฮมของตนได้ ในกรณีนี้ ไดเรกทอรี /home/*Username* ทั้งบนเซิร์ฟเวอร์และไคลเอ็นต์มีการติดตั้งด้วยอ็อปชัน `nosuid` และ `nodev` ของคำสั่ง `mount`

การเอ็กซ์พอร์ตไดเรกทอรี /export/dump

เอ็กซ์พอร์ตไดเรกทอรี /export/dump/*Clienthostname* ด้วยสิทธิ์การอ่าน/เขียนและสิทธิ์ราก ผู้ใช้บนเซิร์ฟเวอร์ไม่มีสิทธิ์เข้าถึงใดๆ ในไฟล์ /export/dump/*Clienthostname*

การเอ็กซ์พอร์ตไดเรกทอรี /export/swap

เอ็กซ์พอร์ตไฟล์ /export/swap/*Clienthostname* ด้วยสิทธิ์การอ่าน/เขียนและสิทธิ์ราก ไม่จำเป็นต้องมีมาตรการรักษาความปลอดภัย ผู้ใช้บนเซิร์ฟเวอร์ไม่มีสิทธิ์เข้าถึงใดๆ ในไฟล์ /export/swap/*Clienthostname*

ชนิดระบบไฟล์

AIX สนับสนุนระบบไฟล์ หลายชนิด

ชนิดระบบไฟล์มีดังต่อไปนี้:

ระบบไฟล์ที่เจอร์นัล (JFS) หรือระบบไฟล์ที่เจอร์นัลที่เพิ่มประสิทธิภาพ (JFS2)

สนับสนุนชุดของความหมายระบบไฟล์ทั้งหมด ระบบไฟล์เหล่านี้ใช้เทคนิคการเจอร์นัลฐานข้อมูลเพื่อรักษาความสอดคล้องกันของโครงสร้าง ซึ่งช่วยป้องกันไม่ให้ระบบไฟล์เสียหายเมื่อระบบหยุดอย่างผิดปกติ

แต่ละ JFS หรือ JFS2 ตั้งอยู่บนโลจิคัลวอลุ่มที่แยกต่างหาก ระบบปฏิบัติการติดตั้ง ระบบไฟล์ในระหว่างการเริ่มต้น คอนฟิกูเรชันหลายระบบไฟล์นี้ มีประโยชน์สำหรับฟังก์ชันการจัดการระบบ เช่น ทำสำเนาสำรอง เรียกคืน และ ช่อมแซม เนื่องจากมีการแยกส่วนของแผนผังไฟล์เพื่อให้คุณสามารถทำงานบนส่วนนั้นได้

JFS คือชนิดระบบไฟล์พื้นฐานที่สนับสนุนชุดของคำสั่งระบบไฟล์ ทั้งหมด

JFS2 คือชนิดระบบไฟล์พื้นฐานที่สนับสนุนชุดของคำสั่งระบบไฟล์ ทั้งหมด

ความแตกต่างระหว่าง JFS และ JFS2 คือ JFS2 ได้รับการออกแบบมาเพื่อสนับสนุนไฟล์ขนาดใหญ่และระบบไฟล์ขนาดใหญ่

ระบบไฟล์เน็ตเวิร์ก (NFS)

เป็นระบบไฟล์แจกจ่ายที่อนุญาตให้ผู้ใช้เข้าถึงไฟล์และ ไดเรกทอรีซึ่งตั้งอยู่บนรีโมตคอมพิวเตอร์ และใช้ไฟล์และ ไดเรกทอรีเหล่านั้น ได้ราวกับว่าเป็นแบบโลคัล ตัวอย่างเช่น ผู้ใช้สามารถใช้คำสั่งระบบปฏิบัติการ เพื่อสร้าง ลบ อ่าน เขียน และตั้งค่าแอตทริบิวต์ไฟล์สำหรับไฟล์และไดเรกทอรีรีโมต

ระบบไฟล์ซีดีรอม (CDRFS)

อนุญาตให้เข้าถึงเนื้อหาของซีดีรอมผ่านทางอินเตอร์เฟซระบบไฟล์ ปกติ (เปิด อ่าน และปิด)

ระบบไฟล์ดีวีดีรอม (UDFS)

อนุญาตให้เข้าถึงเนื้อหาของดีวีดีผ่านทางอินเตอร์เฟซระบบไฟล์ ปกติ

ข้อมูลที่เกี่ยวข้อง:

ระบบไฟล์เครือข่าย

JFS และ JFS2:

ระบบไฟล์ที่เจอร์นัล (JFS) และระบบไฟล์ที่เจอร์นัลที่เพิ่มประสิทธิภาพ (JFS2) มีการสร้างขึ้นในระบบปฏิบัติการพื้นฐาน ระบบไฟล์ทั้งสองชนิด ลิงก์ข้อมูลไฟล์และไดเรกทอรีเข้ากับโครงสร้างที่ใช้โดย AIX Logical Volume Manager สำหรับหน่วยเก็บ และการดึงข้อมูล

ข้อแตกต่างคือ JFS2 ได้รับการออกแบบมาเพื่อสนับสนุนเคอร์เนล 64-บิตและไฟล์ที่ใหญ่ขึ้น

ส่วนต่อไปนี้อธิบายระบบไฟล์เหล่านี้ ยกเว้นว่ามีกรระบุเป็นอย่างอื่น ส่วนต่อไปนี้จะใช้กับทั้ง JFS และ JFS2 ในลักษณะเดียวกัน

ฟังก์ชัน JFS และ JFS2:

Enhanced Journaled File System (JFS2) คือระบบไฟล์ที่มีความสามารถในการเก็บไฟล์ที่มีขนาดใหญ่กว่า Journaled File System (JFS) ที่มีอยู่

คุณสามารถเลือกที่จะนำ JFS หรือ JFS2 ไปใช้ได้ JFS2 เป็นระบบไฟล์ดีฟอลต์ใน AIX 6.1

หมายเหตุ: สิ่งที่แตกต่างกันจาก ระบบไฟล์ JFS คือระบบไฟล์ JFS2 จะไม่อนุญาตให้ใช้ลิงก์() API บนไฟล์ของชนิดไดเรกทอรี
 ข้อจำกัดนี้อาจส่งผลให้บางแอปพลิเคชัน ที่ดำเนินงานอย่างถูกต้องบนระบบไฟล์ JFS ล้มเหลวบนระบบไฟล์ JFS2

ตารางต่อไปนี้แสดงข้อมูลสรุปของฟังก์ชัน JFS และ JFS2:

ฟังก์ชัน	JFS2	JFS
แฟรกเมนต์และขนาดบล็อก	ขนาดบล็อก (ไบต์): 512, 1024, 2048, 4096 ขนาดของระบบไฟล์ สูงสุดในหน่วยเทระไบต์ (TBs): 4, 8, 16, 32	ขนาดแฟรกเมนต์ (ไบต์): 512, 1024, 2048, 4096 ขนาดของระบบไฟล์ สูงสุดในหน่วยกิกะไบต์ (GBs): 128, 256, 512, 1024
ขนาดของระบบไฟล์สูงสุด	32 TBs	1 TB
ขนาดของระบบไฟล์ต่ำสุด	16 MBs	ใช้ไม่ได้
ขนาดไฟล์สูงสุด	16 TBs	ประมาณ 63.876 GBs
จำนวนของโอ-โหนด	ไดนามิก, ถูกจำกัดโดยพื้นที่ว่างดิสก์	คงที่, ตั้งค่าเมื่อสร้างระบบไฟล์
การจัดระเบียบไดเรกทอรี	แผนผัง B	เชิงเส้น
การบีบอัด	ไม่ใช่	ใช่
โควต้า	ใช่	ใช่
การบันทึกข้อผิดพลาด	ใช่	ใช่

หมายเหตุ:

- ขนาดไฟล์สูงสุดและขนาดระบบไฟล์สูงสุดถูกจำกัดเป็น (1 TB - (ขนาดพาร์ติชันฟิสิกส์)) เมื่อใช้กับเคอร์เนล 32-บิต ตัวอย่างเช่น ถ้าขนาดพาร์ติชันฟิสิกส์ของกลุ่มวอลุ่มเป็น 64 MB ขนาดระบบไฟล์สูงสุด จะเป็น (1 TB - 64 MB) = (1048576 MB - 64 MB) = 1048512 MB ที่เป็นเช่นนี้ เนื่องจากข้อจำกัดที่ใช้เกี่ยวกับขนาดสูงสุดของโลจิคัลวอลุ่ม เมื่อใช้เคอร์เนล 32-บิต
- JFS2 สนับสนุน scheme การบันทึกบล็อกมาตรฐานของ AIX สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการบันทึกข้อผิดพลาด AIX โปรดดูที่ ภาพรวมของการบันทึกข้อผิดพลาด ใน *General Programming Concepts: Writing and Debugging Programs*

การแบ่งเซกเมนต์พื้นที่ว่างดิสก์ JFS และ JFS2:

ระบบไฟล์ UNIX จำนวนมาก จัดสรรพื้นที่ว่างดิสก์ที่ต่อเนื่องในหน่วยเท่ากับขนาดของบล็อกเชิงตรรกะที่ใช้สำหรับส่วนโลจิคัลของไฟล์และไดเรกทอรีเท่านั้น โดยปกติ หน่วยการจัดสรร เหล่านี้เรียกว่าเป็น *ดิสก์บล็อก* และดิสก์บล็อกหนึ่ง ใช้เฉพาะสำหรับการจัดเก็บข้อมูลซึ่งมีอยู่ภายในบล็อกเชิงตรรกะเดี่ยว ของไฟล์หรือไดเรกทอรี

การใช้ขนาดบล็อกเชิงตรรกะที่ค่อนข้างใหญ่ (ตัวอย่างเช่น 4096 ไบต์) และการรักษาการจัดสรรดิสก์บล็อกที่เท่ากับขนาดของบล็อกเชิงตรรกะ มีข้อดีในการลดจำนวนของการดำเนินงานดิสก์ I/O ซึ่งต้องทำ โดยการดำเนินงานระบบไฟล์เดี่ยว ข้อมูลไฟล์หรือไดเรกทอรีมีการจัดเก็บไว้ บนดิสก์ในดิสก์บล็อกขนาดใหญ่จำนวนน้อยแทนการใช้ดิสก์บล็อกขนาดเล็ก จำนวนมาก ตัวอย่างเช่น ไฟล์ที่มีขนาด 4096 ไบต์ หรือน้อยกว่า ได้รับการจัดสรรดิสก์บล็อกขนาด 4096-ไบต์เพียงไบต์เดียวเท่านั้น บล็อกเชิงตรรกะเป็น 4096 ไบต์ ด้วยเหตุนี้ การดำเนินงานอ่านหรือเขียนจึงต้องทำการดำเนินงาน ดิสก์ I/O เดี่ยวเท่านั้นเพื่อเข้าถึงข้อมูลบนดิสก์ หากขนาดบล็อกเชิงตรรกะ เล็กลง ส่งผลให้ต้องจัดสรรข้อมูลจำนวนเดียวกันมากกว่าหนึ่งครั้ง ผลคือต้องการ

มากกว่าหนึ่งการดำเนินงานดิสก์ I/O เพื่อเข้าถึงข้อมูล บล็อกเชิงตรรกะ ขนาดใหญ่และขนาดดิสก์บล็อกเท่ากันยังมีข้อดีในการลดจำนวน ของกิจกรรมการจัดสรรพื้นที่ว่างดิสก์ซึ่งต้องทำเมื่อมีการเพิ่มข้อมูล ใหม่ลงในไฟล์และไดเรกทอรี เนื่องจากดิสก์บล็อกขนาดใหญ่เก็บข้อมูลได้ มากขึ้น

อย่างไรก็ตาม การจำกัดหน่วยการจัดสรรพื้นที่ว่างดิสก์เป็นขนาดบล็อกเชิงตรรกะ อาจทำให้พื้นที่ว่างดิสก์สูญเปล่าในระบบไฟล์ซึ่งมีไฟล์และไดเรกทอรีขนาดเล็ก อยู่เป็นจำนวนมาก พื้นที่ว่างดิสก์สูญเปล่าเกิดขึ้นเมื่อบล็อกเชิงตรรกะ ที่มีคุณค่าของพื้นที่ว่างดิสก์ถูกจัดสรรไปยังบล็อกเชิงตรรกะบางส่วนของไฟล์หรือไดเรกทอรี เนื่องจากบล็อกเชิงตรรกะบางส่วนมีข้อมูลน้อยกว่าบล็อกเชิงตรรกะที่มีคุณค่าเสมอ บล็อกเชิงตรรกะบางส่วนจึงใช้พื้นที่ว่างดิสก์ที่ได้รับจัดสรรเพียง บางส่วนเท่านั้น ส่วนที่เหลือยังคงไม่ได้ใช้เนื่องจากไม่มีไฟล์หรือไดเรกทอรีอื่น สามารถเขียนเนื้อหาลงในพื้นที่ว่างดิสก์ซึ่งถูกจัดสรรแล้วได้ จำนวนทั้งหมดของพื้นที่ว่างดิสก์ที่สูญเปล่าอาจมีจำนวนมากสำหรับระบบไฟล์ที่มีไฟล์และไดเรกทอรีขนาดเล็กจำนวนมาก

Journalled file system (JFS) แบ่งพื้นที่ว่างดิสก์เป็นหน่วยการจัดสรร ที่เรียกว่า *แฟรกเมนต์* Enhanced journalled file system (JFS2) แบ่งแฟรกเมนต์ พื้นที่ว่างดิสก์ออกเป็น *บล็อก* โดยมีวัตถุประสงค์เดียวกันคือ: เพื่อจัดเก็บข้อมูล อย่างมีประสิทธิภาพ

แฟรกเมนต์ JFS มีขนาดเล็กกว่าขนาดการจัดสรรดิสก์ดีฟอลต์ 4096 ไบต์ แฟรกเมนต์ลดพื้นที่ว่างดิสก์ที่สูญเปล่าให้เหลือน้อยที่สุดโดยการจัดเก็บข้อมูล อย่างมีประสิทธิภาพมากขึ้นในบล็อกเชิงตรรกะบางส่วนของไฟล์หรือไดเรกทอรี ลักษณะการทำงานของ การสนับสนุนแฟรกเมนต์ JFS ขึ้นอยู่กับการสนับสนุนแฟรกเมนต์ที่นำเสนอโดย การกระจายซอฟต์แวร์ของ Berkeley (BSD)

JFS2 สนับสนุนขนาดบล็อกระบบไฟล์หลายขนาดคือ 512, 1024, 2048, และ 4096 ขนาดบล็อกเล็กลงช่วยลดพื้นที่ว่างดิสก์ที่สูญเปล่าให้เหลือน้อยที่สุดโดยการจัดเก็บ ข้อมูลอย่างมีประสิทธิภาพมากขึ้นในบล็อกเชิงตรรกะบางส่วนของไฟล์หรือไดเรกทอรี อย่างไรก็ตาม ขนาดบล็อกเล็กลง ส่งผลให้ค่าใช้จ่ายในการดำเนินงานสูงขึ้น ขนาดบล็อก สำหรับ JFS2 มีการระบุในระหว่างการสร้าง ระบบไฟล์ที่แตกต่างกันสามารถมี ขนาดบล็อกที่แตกต่างกัน แต่สามารถใช้ได้เพียงหนึ่งขนาดบล็อกเท่านั้น ในระบบ ไฟล์หนึ่ง

หลักการที่เกี่ยวข้อง:

“การบีบอัดข้อมูล JFS” ในหน้า 509

JFS สนับสนุนระบบไฟล์ที่แบ่งแฟรกเมนต์และบีบอัด ซึ่งประหยัด พื้นที่ว่างดิสก์โดยอนุญาตให้จัดเก็บบล็อกเชิงตรรกะบนดิสก์ในหน่วยหรือ “แฟรกเมนต์” ที่เล็กกว่าขนาดบล็อกเต็ม 4096 ไบต์

แฟรกเมนต์ JFS:

ใน JFS หน่วยการจัดสรรพื้นที่ว่างดิสก์เรียกว่า *แฟรกเมนต์* และสามารถเล็กกว่าขนาดบล็อกเชิงตรรกะ 4096 ไบต์

ด้วยการใช้แฟรกเมนต์ที่เล็กกว่า 4096 ไบต์ ส่งผลให้สามารถจัดเก็บข้อมูลที่มีอยู่ ภายในบล็อกเชิงตรรกะบางส่วนได้อย่างมีประสิทธิภาพมากขึ้น โดยการ ใช้ จำนวนแฟรกเมนต์เท่าที่จำเป็นสำหรับการจัดเก็บข้อมูลเท่านั้น ตัวอย่างเช่น บล็อกเชิงตรรกะบางส่วน ที่มีเพียง 500 ไบต์สามารถได้รับจัดสรรแฟรกเมนต์ขนาด 512 ไบต์ (สมมติว่า ขนาดแฟรกเมนต์เป็น 512 ไบต์) ส่งผลให้ลดจำนวนพื้นที่ว่างดิสก์ที่สูญเปล่าลงได้ เป็นอย่างมาก หากความต้องการหน่วยเก็บของบล็อกเชิงตรรกะบางส่วนเพิ่มขึ้น จะมีการจัดสรรแฟรกเมนต์เพิ่มเติมตั้งแต่หนึ่งแฟรกเมนต์ขึ้นไป

ขนาดแฟรกเมนต์ของระบบไฟล์มีการระบุในระหว่างการสร้าง ขนาดแฟรกเมนต์ ที่ใช้ได้สำหรับ journalled file systems (JFS) คือ 512, 1024, 2048, และ 4096 ไบต์ ระบบไฟล์ที่แตกต่างกันสามารถมีขนาดแฟรกเมนต์ที่แตกต่างกัน แต่สามารถใช้ได้เพียงหนึ่งขนาดแฟรกเมนต์เท่านั้นในระบบไฟล์หนึ่ง ขนาดแฟรกเมนต์ ที่แตกต่างกันยังอาจมีอยู่บนระบบเดียว (เครื่อง) เพื่อให้ผู้ใช้ สามารถเลือกขนาดแฟรกเมนต์ที่เหมาะสมที่สุดสำหรับแต่ละระบบไฟล์

การสนับสนุนแฟร็กเมนต์ JFS นำเสนอมุมมองของระบบไฟล์เป็นชุดแฟร็กเมนต์ที่ต่อเนื่องแทนที่จะเป็นชุดดิสก์บล็อกที่ต่อเนื่อง อย่างไรก็ตาม เพื่อรักษาประสิทธิภาพของการดำเนินงานดิสก์ พื้นที่ว่างดิสก์จึงมักมีการจัดสรร ในหน่วย 4096 ไบต์เพื่อให้ดิสก์บล็อกหรือหน่วยการจัดสรรยังคงเท่ากับ ขนาดของบล็อกเชิงตรรกะ การจัดสรรดิสก์บล็อกในกรณีนี้ สามารถมองเป็นการจัดสรรแฟร็กเมนต์ขนาด 4096 ไบต์ที่ต่อเนื่อง

ทั้งค่าใช้จ่ายในการดำเนินงาน (การค้นหาดิสก์เพิ่มเติม การโอนย้ายข้อมูล และ กิจกรรมการจัดสรร) และการใช้ประโยชน์พื้นที่ว่างดิสก์ได้ดีขึ้น ต่างก็เพิ่มขึ้นเมื่อ ขนาดแฟร็กเมนต์ของระบบไฟล์ลดลง เพื่อรักษาสมดุลที่ดีที่สุดระหว่างค่าใช้จ่ายที่เพิ่มขึ้น และพื้นที่ว่างดิสก์ซึ่งใช้ประโยชน์ได้มากขึ้น จึงมีการใช้ปัจจัยต่อไปนี้กับการสนับสนุนแฟร็กเมนต์ JFS:

- ถ้าเป็นไปได้ มีการรักษาการจัดสรรพื้นที่ว่างดิสก์ของแฟร็กเมนต์ขนาด 4096 ไบต์ สำหรับไฟล์หรือบล็อกเชิงตรรกะของไต่เร็กทอรี
- เฉพาะบล็อกเชิงตรรกะบางส่วนของไฟล์หรือไต่เร็กทอรีที่มีขนาดเล็กกว่า 32KB เท่านั้นสามารถได้รับการจัดสรรแฟร็กเมนต์ที่เล็กกว่า 4096 ไบต์

เมื่อไฟล์และไต่เร็กทอรีภายในระบบไฟล์เติบโตขึ้นจนมีขนาดใหญ่กว่า 32 KB การรักษาการจัดสรรพื้นที่ว่างดิสก์น้อยกว่า 4096 ไบต์สำหรับบล็อกเชิงตรรกะบางส่วนจะไม่มีประโยชน์ การประหยัดพื้นที่ว่างดิสก์ซึ่งคิดเป็นเปอร์เซ็นต์ของพื้นที่ว่างระบบไฟล์ทั้งหมดเพิ่มขึ้นเพียงเล็กน้อย ในขณะที่ต้นทุนประสิทธิภาพ เพิ่มเติมในการรักษาการจัดสรรพื้นที่ว่างดิสก์ขนาดเล็ก ยังคงเท่าเดิม เนื่องจากการจัดสรรพื้นที่ว่างดิสก์ ขนาดที่เล็กกว่า 4096 ไบต์ช่วยให้การใช้พื้นที่ว่างดิสก์มีประโยชน์สูงสุด เมื่อใช้กับไฟล์และไต่เร็กทอรีขนาดเล็ก บล็อกเชิงตรรกะของ ไฟล์และไต่เร็กทอรีที่เท่ากับหรือใหญ่กว่า 32 KB จึงได้รับการจัดสรรแฟร็กเมนต์ ขนาด 4096 ไบต์เสมอ บล็อกเชิงตรรกะบางส่วนใดๆ ที่เชื่อมโยงกับไฟล์หรือ ไต่เร็กทอรีขนาดใหญ่ดังกล่าวยังได้รับการจัดสรรแฟร็กเมนต์ขนาด 4096 ไบต์เช่นกัน

บล็อก JFS2:

ระบบไฟล์ที่เจอร์นัลที่เพิ่มประสิทธิภาพแบ่งเซ็กเมนต์พื้นที่ว่างดิสก์ออกเป็น *บล็อก* JFS2 สนับสนุนขนาดบล็อกระบบไฟล์หลายขนาดคือ 512, 1024, 2048, และ 4096

ระบบไฟล์ที่แตกต่างกันสามารถมี ขนาดบล็อกที่แตกต่างกัน แต่สามารถใช้ได้เพียงหนึ่งขนาดบล็อกเท่านั้นในระบบ ไฟล์หนึ่ง

ขนาดบล็อกเล็กช่วยลดพื้นที่ว่างดิสก์ที่สูญเสียไปให้เหลือน้อยที่สุดโดยการจัดเก็บ ข้อมูลอย่างมีประสิทธิภาพมากขึ้นใน บล็อกเชิงตรรกะบางส่วนของไฟล์หรือไต่เร็กทอรี อย่างไรก็ตาม ขนาดบล็อกเล็กส่งผลกระทบต่อค่าใช้จ่ายในการดำเนินงานสูงขึ้นด้วย นอกจากนี้ ไตรเวอร์อุปกรณ์ยังต้อง นำเสนอดิสก์บล็อกที่เข้าถึงได้และมีขนาดเท่ากับหรือเล็กกว่าขนาดบล็อก ระบบไฟล์ด้วย

เนื่องจากพื้นที่ว่างดิสก์มีการจัดสรรในหน่วยที่เล็กลงสำหรับระบบไฟล์ที่มี ขนาดบล็อกอื่นซึ่งไม่ใช่ 4096 ไบต์ กิจกรรมการจัดสรรจึงอาจเกิดบ่อยขึ้น เมื่อไฟล์หรือไต่เร็กทอรีมีขนาดใหญ่ขึ้นอย่างต่อเนื่อง ตัวอย่างเช่น การดำเนินงานเขียนที่ขยายขนาดของไฟล์ที่มีความยาวเป็นศูนย์กลางกลายเป็น 512 ไบต์ ส่งผลให้มีการจัดสรรหนึ่งบล็อกให้กับไฟล์โดยสมมติว่าขนาดบล็อกคือ 512 ไบต์ หากขนาดไฟล์ขยายเพิ่มขึ้นอีก 512 ไบต์จากการเขียนอีกรายการหนึ่ง ต้องมีการจัดสรรบล็อกเพิ่มเติมให้กับไฟล์ เมื่อใช้ตัวอย่างนี้กับระบบไฟล์ ที่มีบล็อก 4096- ไบต์ การจัดสรรพื้นที่ว่างดิสก์จะเกิดขึ้นเพียงครั้งเดียวเป็นส่วน หนึ่งของการดำเนินงานเขียนครั้งแรก ไม่มีการทำกิจกรรมการจัดสรรเพิ่มเติม เป็นส่วนหนึ่งของการดำเนินงานเขียนที่สองเนื่องจากการจัดสรรบล็อกขนาด 4096- ไบต์ครั้งแรกใหญ่เพียงพอที่จะจัดเก็บข้อมูลซึ่งเพิ่มขึ้นเนื่องจากการดำเนินงานเขียนครั้งที่สอง

ขนาดบล็อกของระบบไฟล์มีการระบุในระหว่างการสร้างระบบไฟล์ด้วย System Management Interface Tool (SMIT) หรือคำสั่ง `crfs` และ `mkfs` การตัดสินใจเกี่ยวกับขนาด บล็อกของระบบไฟล์ที่จะเลือกขึ้นอยู่กับข้อมูลขนาดที่คาดการณ์ ของไฟล์ที่มีในระบบไฟล์

ขนาดบล็อกของระบบไฟล์สามารถระบุโดยใช้ System Management Interface Tool (SMIT) หรือคำสั่ง `lsfs` สำหรับแอปพลิเคชันโปรแกรม สามารถใช้ยูทิลิตี้ `statfs` เพื่อระบุขนาดบล็อกของระบบไฟล์

บล็อกทำหน้าที่เป็นหน่วยพื้นฐานของการจัดสรรพื้นที่ว่างดิสก์ และสภาพการจัดสรรของแต่ละบล็อกภายในระบบไฟล์มีการบันทึกไว้ในแม่ผลการจัดสรรบล็อกของระบบไฟล์ อาจต้องการหน่วยความจำเสมือนและพื้นที่ว่างดิสก์ของระบบไฟล์มากขึ้น เพื่อจัดเก็บแม่ผลการจัดสรรบล็อกสำหรับระบบไฟล์ที่มีขนาดบล็อกเล็กกว่า 4096 ไบต์

หมายเลขตัวแปรของไอ-โหนด:

การแบ่งเซกเมนต์พื้นที่ว่างดิสก์เป็นขนาดที่เล็กกว่า 4096 ไบต์ทำให้การใช้พื้นที่ว่างดิสก์เกิดประโยชน์สูงสุด แต่ส่งผลให้จำนวนของไฟล์และไดเรกทอรีขนาดเล็กซึ่งสามารถจัดเก็บภายในระบบไฟล์เพิ่มมากขึ้น

อย่างไรก็ตาม พื้นที่ว่างดิสก์เป็นเพียงรีซอร์สระบบไฟล์อย่างหนึ่งเท่านั้นที่ต้องการ โดยไฟล์และไดเรกทอรี: แต่ละไฟล์หรือไดเรกทอรียังต้องการดิสก์ไอ-โหนดด้วย

JFS และไอ-โหนด:

JFS อนุญาตให้คุณระบุจำนวนของดิสก์ไอ-โหนดที่สร้างขึ้นภายในระบบไฟล์ได้ในกรณีที่ต้องการดิสก์ไอ-โหนดมากกว่าหรือน้อยกว่าจำนวนดีฟอลต์

จำนวนของดิสก์ไอ-โหนดในการสร้างระบบไฟล์มีการระบุในค่าที่เรียกว่า *จำนวนไบต์ต่อไอ-โหนด* หรือ *NBPI* ตัวอย่างเช่น ค่า NBPI เป็น 1024 ส่งผลให้ดิสก์ไอ-โหนดจะถูกสร้างขึ้นสำหรับทุก 1024 ไบต์ของพื้นที่ว่างดิสก์ระบบไฟล์ อีกวิธีหนึ่งในการดูค่านี้คือ ค่า NBPI น้อย (เช่น 512) ส่งผลให้ไอ-โหนดจำนวนมาก ในขณะที่ค่า NBPI มาก (เช่น 16,384) ส่งผลให้ไอ-โหนดจำนวนน้อย

สำหรับระบบไฟล์ JFS มีการสร้างหนึ่งไอ-โหนดสำหรับทุกไบต์ NBPI ของพื้นที่ว่างกลุ่มการจัดสรรซึ่งจัดสรรให้กับระบบไฟล์ จำนวนไอ-โหนดทั้งหมดในระบบไฟล์จำกัดจำนวนไฟล์ทั้งหมดและขนาดทั้งหมดของระบบไฟล์ กลุ่มการจัดสรรสามารถมีการจัดสรรเป็นบางส่วนได้ แม้ว่าจะยังคงจัดสรรจำนวนทั้งหมดของไอ-โหนดต่อกลุ่มการจัดสรร NBPI เป็นสัดส่วนผกผันกับจำนวนไอ-โหนดทั้งหมดในระบบไฟล์

JFS จำกัดระบบไฟล์ทั้งหมดเป็น $16M (2^{24})$ ไอ-โหนด

ชุดของค่า NBPI ที่ใช้ได้แตกต่างกันไปตามขนาดของกลุ่มการจัดสรร (*agsize*) ค่าดีฟอลต์คือ 8 MB ค่า NBPI ที่อนุญาตคือ 512, 1024, 2048, 4096, 8192 และ 16,384 ที่มี *agsize* เท่ากับ 8 MB *agsize* ที่มีขนาดใหญ่ขึ้นสามารถใช้ได้ ค่าที่อนุญาตสำหรับ *agsize* ได้แก่ 8, 16, 32, และ 64 ช่วงของค่า NBPI ที่อนุญาต จะปรับเพิ่มขึ้นเมื่อ *agsize* เพิ่มขนาด หาก *agsize* มีขนาดเพิ่มขึ้นสองเท่าเป็น 16 MB ช่วงของค่า NBPI ก็จะเพิ่มขึ้นเป็นสองเท่าด้วย: 1024, 2048, 4096, 8193, 16384, และ 32768

ขนาดแฟร็กเมนต์และค่า NBPI มีการระบุในระหว่างการสร้างระบบไฟล์ ด้วย System Management Interface Tool (SMIT), หรือคำสั่ง `crfs` และ `mkfs` การตัดสินใจเกี่ยวกับขนาด แฟร็กเมนต์และจำนวนไอ-โหนดที่จะสร้างสำหรับระบบไฟล์ขึ้นอยู่กับข้อมูลจำนวน ที่คาดการณ์และขนาดของไฟล์ที่มีในระบบไฟล์

คุณสามารถระบุขนาดแฟร็กเมนต์และค่า NBPI โดยใช้ System Management Interface Tool (SMIT) หรือคำสั่ง `lsfs` สำหรับแอปพลิเคชันโปรแกรม ให้ใช้ยูทิลิตี้ `statfs` เพื่อระบุขนาดแฟร็กเมนต์ของระบบไฟล์

JFS2 และไอ-โทนด์:

JFS2 จัดสรรไอ-โทนด์ตามความจำเป็น

หากมีพื้นที่ว่างในระบบไฟล์สำหรับไอ-โทนด์เพิ่มเติม ไอ-โทนด์จะถูกจัดสรรโดยอัตโนมัติ ดังนั้น จำนวนของไอ-โทนด์ที่มีอยู่จึงถูกจำกัดโดยขนาดของตัวระบบไฟล์เอง

ข้อจำกัดขนาด JFS และ JFS2:

คุณกำหนดขนาดสูงสุดสำหรับ JFS เมื่อคุณสร้างระบบไฟล์ การตัดสินใจเกี่ยวกับขนาดที่จะกำหนดสำหรับ JFS ขึ้นอยู่กับประเด็นที่สำคัญหลายอย่าง

ขนาดสูงสุดที่แนะนำสำหรับ JFS2 คือ 16 TBs ขนาดระบบไฟล์ต่ำสุดสำหรับ JFS2 คือ 16 MBs

แม้ว่าระบบไฟล์ที่ใช้หน่วยการจัดสรรที่เล็กกว่า 4096 ไบต์ ต้องการพื้นที่ว่างดิสก์น้อยกว่าระบบไฟล์ที่ใช้หน่วยการจัดสรรดีฟอลต์ขนาด 4096 ไบต์เป็นอย่างมาก แต่การใช้แฟร็กเมนต์ที่เล็กลงอาจทำให้เกิดต้นทุนประสิทธิภาพ

สภาพการจัดสรรของแต่ละแฟร็กเมนต์ (JFS) หรือบล็อก (JFS2) ภายในระบบไฟล์มีการบันทึกไว้ในแม่พาร์ติชันระบบไฟล์ อาจต้องการหน่วยความจำเสมือน และพื้นที่ว่างดิสก์ของระบบไฟล์มากขึ้นเพื่อจัดเก็บแม่พาร์ติชันสำหรับระบบไฟล์ที่มีแฟร็กเมนต์หรือขนาดบล็อกเล็กกว่า 4096 ไบต์

เนื่องจากพื้นที่ว่างดิสก์มีการจัดสรรในหน่วยที่เล็กลงสำหรับระบบไฟล์ที่มีขนาดแฟร็กเมนต์ (JFS) หรือบล็อก (JFS2) อื่นซึ่งไม่ใช่ 4096 ไบต์ ก็กรรม การจัดสรรจึงอาจเกิดบ่อยขึ้นเมื่อไฟล์หรือไดเรกทอรีมีขนาดใหญ่ขึ้นอย่างต่อเนื่อง ตัวอย่างเช่น การดำเนินงานเขียนที่ขยายขนาดของไฟล์ที่มีความยาวเป็นศูนย์ กลายเป็น 512 ไบต์ ส่งผลให้มีการจัดสรรหนึ่งแฟร็กเมนต์หรือบล็อกขนาด 512 ไบต์ให้กับไฟล์ ขึ้นอยู่กับชนิดระบบไฟล์ หากขนาดไฟล์ขยายเพิ่มขึ้นอีก 512 ไบต์จากการเขียนอีกรายการหนึ่ง ต้องมีการจัดสรรแฟร็กเมนต์หรือบล็อกเพิ่มเติมให้กับไฟล์ เมื่อใช้ตัวอย่างนี้กับระบบไฟล์ที่มีแฟร็กเมนต์หรือบล็อก 4096-ไบต์ การจัดสรรพื้นที่ว่างดิสก์จะเกิดขึ้นเพียงครั้งเดียวเป็นส่วนหนึ่งของการดำเนินงานเขียนครั้งแรก ไม่มีการทำกิจกรรมการจัดสรรเพิ่มเติม เป็นส่วนหนึ่งของการดำเนินงานเขียนที่สองเนื่องจากการจัดสรร 4096-ไบต์ครั้งแรกใหญ่เพียงพอที่จะจัดเก็บข้อมูลซึ่งเพิ่มขึ้นเนื่องจากการดำเนินงานเขียนครั้งที่สอง กิจกรรมการจัดสรร อาจถูกลดจำนวนให้เหลือน้อยที่สุดได้ถ้าไฟล์มีการขยายเพิ่มครั้งละ 4096 ไบต์

ประเด็นหนึ่งที่เกี่ยวข้องกับขนาดคือขนาดของบันทึก ระบบไฟล์

สำหรับ JFS ในอินสแตนซ์ส่วนใหญ่ หลายระบบไฟล์ใช้บันทึกทั่วไปซึ่งกำหนด คอนฟิกให้มีขนาด 4 MB ตัวอย่างเช่น หลังจากการติดตั้งครั้งแรก ระบบไฟล์ทั้งหมด ภายในกลุ่มวอลุ่มจะใช้โลจิคัลวอลุ่ม hd8 เป็นบันทึก JFS ทั่วไป ขนาดดีฟอลต์โลจิคัลวอลุ่มพาร์ติชันคือ 4 MB และขนาดบันทึกดีฟอลต์คือ หนึ่งพาร์ติชัน ดังนั้นโดยปกติแล้ว กลุ่มวอลุ่มจึงมีบันทึก JFS ขนาด 4 MB เมื่อระบบไฟล์เกินกว่า 2 GB หรือเมื่อจำนวนทั้งหมดของพื้นที่ว่างระบบไฟล์ ที่ใช้หนึ่งบันทึกเกินกว่า 2 GB ขนาดบันทึกดีฟอลต์อาจจะไม่เพียงพอ ในกรณีใดกรณีหนึ่งนี้ ขนาดบันทึกจะเพิ่มขึ้นเมื่อขนาดระบบไฟล์เพิ่มขึ้น เมื่อขนาดของโลจิคัลวอลุ่มบันทึกเปลี่ยนแปลง ต้องรันคำสั่ง **logform** เพื่อเริ่มต้นบันทึกอีกครั้งก่อนสามารถใช้พื้นที่ว่างใหม่ได้ บันทึก JFS มีข้อจำกัดขนาดสูงสุดไม่เกิน 256 MB

มีข้อจำกัดที่แท้จริงเกี่ยวกับขนาดของระบบไฟล์รวมทั้ง บันทึก JFS หนึ่งสามารถสนับสนุนได้ ตามแนวทาง ความจุทั้งหมดของระบบไฟล์จำนวน หนึ่งล้านล้านไบต์เป็นขีดจำกัดที่แนะนำสำหรับบันทึก JFS หนึ่ง เมื่อเกินกว่า แนวทางนี้หรือใกล้เคียงเกิน หรือเมื่อเกิดข้อผิดพลาดการขาดหน่วยความจำ จากคำสั่ง **logredo** (ซึ่งเรียกโดย คำสั่ง **fsck**) ให้เพิ่มบันทึก JFS เพิ่มเติม แล้ว แบ่งโหลระหว่างบล็อกไฟล์ JFS สองไฟล์

สำหรับ JFS2 ในอินสแตนซ์ส่วนใหญ่ หลายระบบไฟล์ยังใช้บันทึกทั่วไปด้วย เมื่อระบบไฟล์เกินกว่า 2 GB หรือเมื่อจำนวนทั้งหมดของพื้นที่ว่างระบบไฟล์ที่ใช้หนึ่งบันทึกเกินกว่า 2 GB ขนาดบันทึกดีพอลต์อาจจะไม่เพียงพอ ในกรณีใดกรณีหนึ่งนี้ คุณสามารถเพิ่มขนาดบันทึกเมื่อขนาดระบบไฟล์เพิ่มขึ้น หรือคุณสามารถเพิ่มบันทึก JFS2 เพิ่มเติม แล้วแบ่งโหนดระหว่าง ล็อกไฟล์ JFS2 สองไฟล์

ขีดจำกัดขนาด JFS:

ขนาด JFS สูงสุดมีการกำหนดเมื่อระบบไฟล์ถูกสร้างขึ้น NBPI, ขนาดแฟรกเมนต์ และขนาดกลุ่มการจัดสรรเป็นปัจจัยในการกำหนด ขนาดสูงสุด

ขีดจำกัดขนาดระบบไฟล์เป็นค่าต่ำสุดของข้อมูลต่อไปนี้:

$NBPI * 2^{24}$

หรือ

$FragmentSize * 2^{28}$

ตัวอย่างเช่น ถ้าคุณเลือกอัตราส่วน NBPI เป็น 512 ขนาดระบบไฟล์จะถูกจำกัดเป็น 8 GB ($512 * 2^{24} = 8 \text{ GB}$) JFS สนับสนุนค่า NBPI ของ 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, และ 131072

JFS จำกัดระบบไฟล์ทั้งหมดเป็น 16M (2^{24}) ไอบิต

มีการสร้างหนึ่งไอบิตสำหรับทุกไอบิต NBPI ของพื้นที่ว่างกลุ่ม การจัดสรรซึ่งจัดสรรให้กับระบบไฟล์ กลุ่มการจัดสรรสามารถมีการจัดสรรเป็นบางส่วนได้ แม้ว่าจะยังคงจัดสรรจำนวน ทั้งหมดของไอบิตต่อกลุ่มการจัดสรร NBPI เป็นสัดส่วนผกผัน กับจำนวนไอบิตทั้งหมดในระบบไฟล์

JFS แบ่งพื้นที่ว่างระบบไฟล์ออกเป็นกลุ่มของไอบิตและดิสก์บล็อก สำหรับข้อมูลผู้ใช้ กลุ่มเหล่านี้เรียกว่ากลุ่มการจัดสรร ขนาดกลุ่มการจัดสรร สามารถระบุได้เมื่อระบบไฟล์ถูกสร้างขึ้น ขนาดกลุ่มการจัดสรร คือ 8M, 16M, 32M, และ 64M ขนาดกลุ่มการจัดสรรแต่ละขนาดมีช่วง NBPI ที่เชื่อมโยง ช่วงมีการกำหนดโดยตารางต่อไปนี้:

Allocation Group Size in Megabytes	Allowable NBPI Values
8	512, 1024, 2048, 4096, 8192, 16384
16	1024, 2048, 4096, 8192, 16384, 32768
32	2048, 4096, 8192, 16384, 32768, 65536
64	4096, 8192, 16384, 32768, 65536, 131072

JFS สนับสนุนแฟรกเมนต์สี่ขนาดคือ 512, 1024, 2048, และ 4096 หน่วยไอบิตของพื้นที่ว่างดิสก์ต่อกัน JFS รักษา แฟรกเมนต์ แอตเตอร์สในไอบิตและบล็อกโดยอ้อมเป็นตัวเลข 28-บิต แต่ละ แฟรกเมนต์ต้องระบุได้ด้วยตัวเลขตั้งแต่ 0 ถึง (2^{28})

ขีดจำกัดขนาด JFS2:

การทดสอบแสดงว่าระบบไฟล์ JFS2 ขนาดใหญ่มากที่มีไฟล์ขนาดใหญ่สามารถบำรุงรักษาได้ง่ายกว่าระบบไฟล์ที่มีไฟล์ขนาดเล็กจำนวนมาก เมื่อระบบไฟล์ขนาดใหญ่มีไฟล์ขนาดเล็กจำนวนมาก คำสั่ง fsck และการบำรุงรักษาระบบไฟล์อื่น จะใช้เวลาในการรันนาน

ขีดจำกัดขนาดที่แนะนำมีดังต่อไปนี้:

506 AIX เวอร์ชัน 7.2: ระบบปฏิบัติการและการจัดการกับอุปกรณ์

ไอเท็ม
ขนาดระบบไฟล์ JFS2 สูงสุด:
ขนาดไฟล์ JFS2 สูงสุด:
ขนาดระบบไฟล์ JFS2 ต่ำสุด:

คำอธิบาย
32TB
16TB
16MB

การแบ่งแฟรกเมนต์พื้นที่ว่าง JFS:

สำหรับระบบไฟล์ JFS การใช้แฟรกเมนต์ที่เล็กกว่า 4096 ไบต์อาจส่งผลให้การแบ่งแฟรกเมนต์ของพื้นที่ว่างบนดิสก์ดีขึ้น

ตัวอย่างเช่น พิจารณาพื้นที่ของดิสก์ที่แบ่งออกเป็นแปดแฟรกเมนต์ แฟรกเมนต์ละ 512 ไบต์ สมมติว่าไฟล์ที่แตกต่างกันซึ่งต้องการพื้นที่ไฟล์ละ 512 ไบต์ ถูกเขียนลงในแฟรกเมนต์ที่หนึ่ง ทีสี่ ทีห้า และที่เจ็ดในพื้นที่นี้ ของดิสก์ และปล่อยให้แฟรกเมนต์ที่สอง ที่สาม ที่หก และที่แปดว่าง แม้ว่าสี่แฟรกเมนต์ซึ่งแสดงถึงพื้นที่ว่างดิสก์จำนวน 2048 ไบต์ว่างอยู่ แต่ไม่มีการจัดสรรบล็อกเชิงตรรกะบางส่วนที่ต้องการสี่แฟรกเมนต์ (หรือ 2048 ไบต์) สำหรับ แฟรกเมนต์ที่ว่างเหล่านี้ เนื่องจากแฟรกเมนต์ในการจัดสรรหนึ่งต้อง ต่อกัน

เนื่องจากแฟรกเมนต์ที่จัดสรรสำหรับบล็อกเชิงตรรกะไฟล์หรือไดเรกทอรี ต้องต่อกัน การแบ่งแฟรกเมนต์พื้นที่ว่างอาจส่งผลให้การดำเนินงานระบบไฟล์ ที่ต้องการพื้นที่ว่างดิสก์ใหม่ล้มเหลว แม้ว่าจำนวนทั้งหมดของพื้นที่ว่างที่มีอยู่มากเพียงพอสำหรับการดำเนินงาน ตัวอย่างเช่น การดำเนินงานเขียน ที่ขยายไฟล์ความยาวศูนย์เพิ่มหนึ่งบล็อกเชิงตรรกะต้องการการจัดสรร ไบต์ของพื้นที่ว่างดิสก์ต่อกันจำนวน 4096 ไบต์ ถ้าพื้นที่ว่างของระบบไฟล์ มีการแบ่งแฟรกเมนต์และประกอบด้วยแฟรกเมนต์ที่ไม่ต่อกันขนาด 512-ไบต์จำนวน 32 แฟรกเมนต์ หรือเท่ากับพื้นที่ดิสก์ว่างทั้งหมด 16 KB การดำเนินงานเขียนจะล้มเหลวเนื่องจากแฟรกเมนต์ที่ต่อกันแปดแฟรกเมนต์ (หรือพื้นที่ว่างดิสก์ที่ต่อกัน 4096 ไบต์) ไม่เพียงพอสำหรับ การดำเนินงานเขียน

ระบบไฟล์ JFS ที่มีจำนวนพื้นที่ว่างแฟรกเมนต์ซึ่งจัดการไม่ได้ สามารถถูกยกเลิกแบ่งแฟรกเมนต์ด้วยคำสั่ง `defragfs` การรันคำสั่ง `defragfs` มีผลดีต่อประสิทธิภาพ

ไฟล์ที่กระจัดกระจาย:

ไฟล์คือลำดับของบล็อกที่จัดทำดัชนี บล็อกถูกแมปจาก ไอ-โหนดเข้ากับโลจิคัลออฟเซตของไฟล์ที่บล็อกแสดงแทน

ไฟล์ที่มีตั้งแต่หนึ่งดัชนีขึ้นไปและไม่ได้แมปเข้ากับบล็อกข้อมูล เรียกว่าเป็น *ที่จัดสรรแบบกระจัดกระจาย* หรือ *ไฟล์ที่กระจัดกระจาย* ไฟล์ที่กระจัดกระจายจะมีขนาดเชื่อมโยงอยู่ด้วย แต่อาจไม่ได้รับการจัดสรร บล็อกข้อมูลครบทั้งหมดตามความต้องการของขนาด เมื่อต้องการระบุ ว่าไฟล์เป็นแบบจัดสรรแบบกระจัดกระจายหรือไม่ ให้ใช้คำสั่ง `fileplace` คำสั่งจะบ่งชี้บล็อกทั้งหมดในไฟล์ที่ไม่ได้รับการจัดสรรในปัจจุบัน

หมายเหตุ: ในสถานการณ์ส่วนใหญ่ยังสามารถใช้ `du` เพื่อกำหนดว่า จำนวนของบล็อกข้อมูลที่จัดสรรให้กับไฟล์ไม่ตรงกับจำนวนที่ต้องการ เพื่อจัดเก็บไฟล์ในขนาดของไฟล์นั้นหรือไม่ ระบบไฟล์ที่บีบอัดอาจมีลักษณะการทำงานเหมือนกับ ไฟล์ที่ไม่ได้จัดสรรแบบกระจัดกระจาย

ไฟล์ที่กระจัดกระจายมีการสร้างขึ้นเมื่อแอ็พพลิเคชันขยายไฟล์โดยการค้นหา ในที่ตั้งภายนอกดัชนีที่จัดสรรในปัจจุบัน แต่ข้อมูลที่เขียน ไม่ได้ใช้ดัชนีที่กำหนดให้ใหม่ทั้งหมด ขนาดไฟล์ใหม่ สะท้อนการเขียนที่ไกลมากที่สุดลงในไฟล์

การอ่านในส่วนของไฟล์ที่มีบล็อกข้อมูลซึ่งไม่ได้รับการจัดสรรส่งผลให้มีการส่งคืน บัฟเฟอร์ของศูนย์ การเขียนในส่วนของไฟล์ที่มี บล็อกข้อมูลซึ่งไม่ได้รับการจัดสรรส่งผลให้มีการจัดสรรบล็อกข้อมูลที่จำเป็น และข้อมูลที่เขียน

ลักษณะการทำงานอาจส่งผลกระทบต่อคำสั่งจัดการไฟล์หรือการเก็บถาวร ตัวอย่างเช่น คำสั่งต่อไปนี้ไม่รักษาการจัดสรรแบบกระจายของไฟล์:

- cp
- mv
- tar
- cpio

หมายเหตุ: ในกรณีของ mv ข้อมูลนี้ใช้กับการย้ายไฟล์ไปยังระบบไฟล์อื่นเท่านั้น หากไฟล์ถูกย้ายภายในระบบไฟล์เดียวกัน ไฟล์จะยังคงเป็นแบบกระจาย

ผลลัพธ์ของไฟล์ที่คัดลอกหรือเรียกคืนจากคำสั่งก่อนหน้านี้ มีบล็อกข้อมูลที่จัดสรรแต่ละบล็อก ดังนั้นจึงไม่มีลักษณะการกระจาย อย่างไรก็ตาม คำสั่งเก็บถาวรต่อไปนี้รักษาลักษณะการกระจายหรือ ไฟล์ที่กระจายซึ่งใช้งานอยู่:

- backup
- restore
- pax

เนื่องจากไฟล์ที่กระจายอาจ overcommit รีซอร์สของระบบไฟล์ ดังนั้นจึงควรระมัดระวังในการใช้และการบำรุงรักษาไฟล์ชนิดนี้

JFS และไฟล์ขนาดใหญ่:

คุณสามารถสร้างไฟล์ขนาดใหญ่ด้วยชนิดระบบไฟล์ JFS ได้

ระบบไฟล์ JFS2 ทั้งหมดสนับสนุนไฟล์ขนาดใหญ่

ระบบไฟล์ที่เปิดใช้งานไฟล์ขนาดใหญ่สามารถสร้างขึ้นได้ด้วยคำสั่ง crfs และ mkfs ทั้งสองคำสั่งมีอ็อปชัน (bf=true) เพื่อระบุระบบไฟล์ที่เปิดใช้งานไฟล์ขนาดใหญ่ คุณยังสามารถใช้ SMIT เพื่อสร้างระบบไฟล์เหล่านี้

ในระบบไฟล์ที่เปิดใช้งานไฟล์ขนาดใหญ่ ข้อมูลไฟล์ที่จัดเก็บไว้ก่อนถึงออฟเซตไฟล์ 4 MB จะได้รับจัดสรรในบล็อกขนาด 4096-ไบต์ ข้อมูลไฟล์ที่จัดเก็บไว้เกินกว่าออฟเซตไฟล์ 4 MB จะได้รับจัดสรรดิस्कบล็อกขนาดใหญ่ที่มีขนาด 128 KB โดยแท้จริงแล้ว ดิस्कบล็อกขนาดใหญ่คือบล็อกขนาด 4096-ไบต์ที่ต่อกันจำนวน 32 บล็อก

ตัวอย่างเช่น ในระบบไฟล์ปกติไฟล์ 132-MB ต้องการดิस्कบล็อก 33K 4-KB (33 บล็อกโดยอ้อมเดี่ยว โดยแต่ละบล็อกมีดิस्कแอดเดรส 1024 4 KB) ไฟล์ 132-MB ในระบบไฟล์ที่เปิดใช้งานไฟล์ขนาดใหญ่มีดิस्कบล็อก 1024 4-KB และดิस्कบล็อก 1024 128-KB เรขาคณิตไฟล์ขนาดใหญ่ต้องการเพียงสอง บล็อกโดยอ้อมเดี่ยวเท่านั้นสำหรับไฟล์ 132 MB ทั้งไฟล์ขนาดใหญ่และไฟล์ปกติ ต้องการบล็อกโดยอ้อมดับเบิลหนึ่งบล็อก

ดิस्कบล็อกขนาดใหญ่ต้องการบล็อกขนาด 4 KB ที่ต่อกัน 32 บล็อก หากคุณเขียนลงในไฟล์ขนาดใหญ่ เกินกว่า 4 MB ไฟล์ออฟเซตจะล้มเหลวด้วย ENOSPC ถ้าระบบไฟล์ ไม่มีบล็อกขนาด 4 KB ที่ต่อกันซึ่งไม่ได้ใช้จำนวน 32 บล็อก

หมายเหตุ: ระบบไฟล์อาจมีบล็อกที่ว่างนับพันบล็อก แต่ถ้า 32 บล็อกไม่ต่อกัน การจัดสรรจะล้มเหลว

คำสั่ง defragfs จัดระเบียบดิस्कบล็อกอีกครั้งเพื่อจัดเตรียมพื้นที่บล็อกว่างต่อกันซึ่งใหญ่ขึ้น

JFS ต้องเริ่มต้นการจัดสรรดิสก์ใหม่ทั้งหมด JFS เริ่มต้นโพรซีเจอร์ เคอร์เนล kproc ที่ใช้เพื่อทำการจัดสรรไฟล์แรกเริ่มให้เป็น ศูนย์เมื่อติดตั้ง ระบบไฟล์ที่เปิดใช้งานไฟล์ขนาดใหญ่แรกบนระบบ โพรซีเจอร์ kproc ยังคงอยู่หลังจากถอนติดตั้งระบบไฟล์ที่เปิดใช้งานไฟล์ขนาดใหญ่เสร็จเรียบร้อยแล้ว

การบีบอัดข้อมูล JFS:

JFS สนับสนุนระบบไฟล์ที่แบ่งแฟรกเมนต์และบีบอัด ซึ่งประหยัด พื้นที่ว่างดิสก์โดยอนุญาตให้จัดเก็บบล็อกเชิงตรรกะบนดิสก์ในหน่วยหรือ "แฟรกเมนต์" ที่เล็กกว่าขนาดบล็อกเต็ม 4096 ไบต์

การบีบอัดข้อมูลไม่ได้รับการสนับสนุนสำหรับ JFS2

ในระบบไฟล์ที่แบ่งแฟรกเมนต์ เฉพาะบล็อกเชิงตรรกะล่าสุดของไฟล์ที่ไม่ใหญ่กว่า 32KB เท่านั้นถูกจัดเก็บในลักษณะนี้ ดังนั้นการสนับสนุนแฟรกเมนต์จึงมี ประโยชน์สำหรับระบบไฟล์ซึ่งมีไฟล์ขนาดเล็กจำนวนมากเท่านั้น อย่างไรก็ตาม การบีบอัดข้อมูล ช่วยให้สามารถจัดเก็บบล็อกเชิงตรรกะทั้งหมดของไฟล์ทุกขนาดเป็นหนึ่งหรือหลาย แฟรกเมนต์ที่ต่อกันได้โดยเฉลี่ยแล้ว การบีบอัดข้อมูลช่วยประหยัดพื้นที่ว่างดิสก์ได้ประมาณ ครึ่งหนึ่ง

อย่างไรก็ตาม การใช้แฟรกเมนต์และการบีบอัดข้อมูลเพิ่มโอกาสการแบ่ง แฟรกเมนต์พื้นที่ว่างดิสก์ แฟรกเมนต์ที่จัดสรรให้ กับบล็อกเชิงตรรกะ ต้องต่อเนื่องบนดิสก์ ระบบไฟล์ที่มีปัญหาการแบ่งแฟรกเมนต์พื้นที่ว่าง อาจมีความยากลำบากในการระบุ ตำแหน่งแฟรกเมนต์ต่อเนื่องที่มากเพียงพอ สำหรับการจัดสรรบล็อกเชิงตรรกะ แม้ว่าจำนวนทั้งหมดของแฟรกเมนต์ที่ว่าง อาจเกินกว่าความต้องการบล็อกเชิงตรรกะ JFS บรรเทาปัญหาการแบ่งแฟรกเมนต์พื้นที่ว่าง โดยการนำเสนอโปรแกรม defragfs ซึ่ง "ยกเลิกแฟรกเมนต์" ระบบไฟล์โดยการเพิ่มจำนวนของพื้นที่ว่างที่ต่อกัน ยูทิลิตีนี้ สามารถใช้สำหรับระบบไฟล์ที่แบ่งแฟรกเมนต์และที่บีบอัด การประหยัดพื้นที่ว่างดิสก์ ที่ได้จากแฟรกเมนต์และการบีบอัดข้อมูลมีประโยชน์มาก ในขณะที่ ปัญหา การแบ่งแฟรกเมนต์พื้นที่ว่างยังคงสามารถจัดการได้

การบีบอัดข้อมูลใน JFS ปัจจุบันเข้ากันได้กับเวอร์ชันก่อนหน้านี้ ของระบบปฏิบัติการนี้ API ที่เป็นส่วนประกอบของการเรียก ระบบทั้งหมดยังคง เหมือนกันในทั้งสองเวอร์ชันของ JFS

หลักการที่เกี่ยวข้อง:

"การแบ่งแฟรกเมนต์พื้นที่ว่างดิสก์ JFS และ JFS2" ในหน้า 501

ระบบไฟล์ UNIX จำนวนมาก จัดสรรพื้นที่ว่างดิสก์ที่ต่อเนื่องในหน่วยเท่ากับขนาดของบล็อกเชิงตรรกะที่ใช้สำหรับส่วนโลจิก คัลของไฟล์และไดเรกทอรีเท่านั้น โดยปกติ หน่วยการจัดสรร เหล่านี้เรียกว่าเป็น *ดิสก์บล็อก* และดิสก์บล็อกหนึ่ง ใช้เฉพาะ สำหรับการจัดเก็บข้อมูลซึ่งมีอยู่ภายในบล็อกเชิงตรรกะเดียว ของไฟล์หรือไดเรกทอรี

การนำการบีบอัดข้อมูล JFS ไปใช้:

การบีบอัดข้อมูลเป็นแอตทริบิวต์ของระบบไฟล์ซึ่งมีการระบุ เมื่อระบบไฟล์ถูกสร้างขึ้นด้วยคำสั่ง crfs หรือ mkfs คุณสามารถ ใช้ SMIT เพื่อระบุการบีบอัดข้อมูล

ข้อควรสนใจ: ระบบไฟล์ราก (/) ต้อง ไม่ถูกบีบอัด ไม่แนะนำการบีบอัดระบบไฟล์ /usr เนื่องจาก installp ต้องสามารถ คำนวณขนาดสำหรับ อัปเดตและการติดตั้งใหม่ได้อย่างถูกต้อง

การบีบอัดใช้กับไฟล์ปกติและลิงก์สัญลักษณ์ยาวเท่านั้นในระบบไฟล์ ดังกล่าว การสนับสนุนแฟรกเมนต์ยังใช้กับไดเรกทอรี และข้อมูลเมตา ที่ไม่ได้บีบอัดต่อไป แต่ละบล็อกเชิงตรรกะของไฟล์ถูกบีบอัดโดยตัวเอง ก่อนถูกเขียนลงในดิสก์ การบีบอัดใน ลักษณะนี้สนับสนุนการค้นหาและอัปเดตแบบสุ่ม ในขณะที่สูญเสียพื้นที่ดิสก์ที่เป็นอิสระเพียงจำนวนเล็กน้อยเท่านั้นเมื่อ เปรียบเทียบกับการบีบอัดข้อมูลในหน่วยที่ใหญ่ขึ้น

หลังจากการบีบอัด โดยปกติแล้ว บล็อกเชิงตรรกะต้องการพื้นที่ว่างดิสก์น้อยกว่า 4096 ไบต์ บล็อกเชิงตรรกะที่บีบอัดถูกเขียนลงในดิสก์และได้รับจัดสรร เฉพาะจำนวนของแฟรกเมนต์ที่ต่อกันซึ่งต้องการสำหรับการจัดเก็บเท่านั้น หากบล็อกเชิงตรรกะไม่ได้บีบอัด บล็อกนั้นจะถูกเขียนลงในดิสก์ในรูปแบบที่ไม่บีบอัด และได้รับจัดสรรแฟรกเมนต์ที่ต่อกันจำนวน 4096 ไบต์

คำสั่ง `lsfs -q` แสดงค่าปัจจุบันสำหรับการบีบอัด คุณยังสามารถใช้ `SMIT` เพื่อระบุการบีบอัดข้อมูล

หลักการที่เกี่ยวข้อง:

“ลักษณะการทำงานโดยปริยายของการบีบอัดข้อมูล JFS”

เนื่องจากโปรแกรมที่เขียนไฟล์ไม่ได้คาดหวังว่าสภาพการขาดพื้นที่ว่าง (ENOSPC) จะเกิดขึ้นหลังการเขียนสำเร็จ (หรือการจัดเก็บไฟล์ที่แม่พิมพ์สำเร็จ) จึงจำเป็นต้องรับประกันว่าพื้นที่ว่างมีอยู่เมื่อเขียนบล็อกเชิงตรรกะลงในดิสก์

ลักษณะการทำงานโดยปริยายของการบีบอัดข้อมูล JFS:

เนื่องจากโปรแกรมที่เขียนไฟล์ไม่ได้คาดหวังว่าสภาพการขาดพื้นที่ว่าง (ENOSPC) จะเกิดขึ้นหลังการเขียนสำเร็จ (หรือการจัดเก็บไฟล์ที่แม่พิมพ์สำเร็จ) จึงจำเป็นต้องรับประกันว่าพื้นที่ว่างมีอยู่เมื่อเขียนบล็อกเชิงตรรกะลงในดิสก์

ซึ่งทำได้โดยการจัดสรร 4096 ไบต์ให้กับบล็อกเชิงตรรกะเมื่อแก้ไขบล็อกนั้น เป็นครั้งแรก เพื่อให้พื้นที่ว่างดิสก์แม้ว่าบล็อกไม่ได้บีบอัด ถ้าการจัดสรร 4096-ไบต์ไม่มีอยู่ระบบจะส่งคืนเงื่อนไขข้อผิดพลาด ENOSPC หรือ EDQUOT แม้ว่าอาจมีพื้นที่ว่างดิสก์เพียงพอที่จะสนับสนุนบล็อกเชิงตรรกะที่บีบอัด การรายงานสภาพการขาดพื้นที่ว่าง ก่อนถึงกำหนดมักเกิดขึ้นบ่อยที่สุดเมื่อใช้งานเกือบถึงขีดจำกัดโควต้าดิสก์ หรือเกือบเต็มระบบไฟล์

ระบบไฟล์ที่บีบอัดยังอาจห้ามลักษณะการทำงานต่อไปนี้ด้วย:

- เนื่องจากโดยแรกเริ่ม 4096 ไบต์ถูกจัดสรรให้กับบล็อกเชิงตรรกะ ดังนั้นการเรียก ระบบบางอย่างจึงอาจได้รับข้อผิดพลาด ENOSPC หรือ EDQUOT ตัวอย่างเช่น ไฟล์เก่า อาจถูกแม่พิมพ์โดยใช้การเรียกระบบ `mmap` และการดำเนินงานจัดเก็บ เข้าในที่ตั้งซึ่งเขียนก่อนหน้านี้อาจส่งผลให้เกิดข้อผิดพลาด ENOSPC
- ด้วยการบีบอัดข้อมูล บล็อกดิสก์เต็มยังคงถูกจัดสรรให้กับบล็อกที่แก้ไข จนกว่าจะถูกเขียนลงในดิสก์ หากบล็อกมีการจัดสรรที่ `committed` ก่อนหน้านั้นน้อยกว่าหนึ่งบล็อกเต็ม จำนวนของพื้นที่ว่างดิสก์ที่ใช้โดย บล็อกคือผลรวมของสอง การจัดสรรก่อนหน้านี้อย่างไม่เป็นอิสระจนกว่า ไฟล์ (ไอ-โหนด) ถูก `committed` นี่เป็นกรณีของแฟรกเมนต์ปกติ จำนวนของ บล็อกเชิงตรรกะในไฟล์ซึ่งสามารถมีการจัดสรรที่ `committed` ก่อนหน้านี้นี้คือ ไม่เกินหนึ่งสำหรับแฟรกเมนต์ปกติ แต่สามารถมากกว่ากับจำนวนบล็อก ในไฟล์ที่มีการบีบอัด
- ไม่มีรีซอร์สที่ `committed` ก่อนหน้านี้นี้สำหรับบล็อกเชิงตรรกะถูกปล่อยเป็นอิสระ จนกว่าการเรียกระบบ `fsync` หรือ `sync` มีการรันโดยแอปพลิเคชันโปรแกรม
- การเรียกระบบ `stat` บ่งชี้จำนวนของแฟรกเมนต์ที่จัดสรรให้กับไฟล์ จำนวนที่รายงาน ขึ้นอยู่กับ 4096 ไบต์ที่กำลังจัดสรรให้กับบล็อกที่แก้ไขแต่ไม่ได้เขียนและ ขนาดที่บีบอัดของบล็อกที่ไม่ได้แก้ไข รีซอร์สที่ `committed` ก่อนหน้านี้นี้ไม่ถูก นับโดยการเรียกระบบ `stat` การเรียกระบบ `stat` รายงานจำนวนที่ถูกต้องของแฟรกเมนต์ที่จัดสรรหลังจากการดำเนินงาน `commit` ไอ-โหนดถ้าไม่มีการบีบอัดบล็อกที่แก้ไข ในลักษณะคล้ายกัน ดิสก์โควต้า ถูกเรียกเก็บสำหรับการจัดสรรปัจจุบัน เมื่อบล็อกเชิงตรรกะของไฟล์ถูกเขียน ลงในดิสก์ จำนวนของแฟรกเมนต์ที่จัดสรรให้กับดิสก์จะลดลงถ้า มีการบีบอัด ส่งผลให้เปลี่ยนดิสก์โควต้าและผลลัพธ์จาก `stat`

หลักการที่เกี่ยวข้อง:

“การนำการบีบอัดข้อมูล JFS ไปใช้” ในหน้า 509

การบีบอัดข้อมูลเป็นแอตทริบิวต์ของระบบไฟล์ซึ่งมีการระบุ เมื่อระบบไฟล์ถูกสร้างขึ้นด้วยคำสั่ง `crfs` หรือ `mkfs` คุณสามารถใช้ `SMIT` เพื่อระบุการบีบอัดข้อมูล

ขั้นตอนวิธีการบีบอัดข้อมูล JFS:

ขั้นตอนวิธีการบีบอัดเป็นเวอร์ชัน IBM ของ LZ โดยทั่วไป ขั้นตอนวิธี LZ บีบอัดข้อมูลโดยการแสดงถึงการเกิดขึ้นครั้งที่สอง และครั้งต่อมาของสตริงที่กำหนดด้วยตัวชี้ซึ่งระบุที่ตั้ง ของการเกิดขึ้นครั้งแรกของสตริงและความยาว

ที่ตอนต้นของโปรเซสการบีบอัด ยังไม่มีการระบุสตริง ดังนั้นไบต์แรกของข้อมูลเป็นอย่างน้อยต้องถูกแสดงแทนด้วยอักขระ "ดิบ" ที่ต้องการ 9-บิต (0,ไบต์) หลังจากจำนวนข้อมูลที่กำหนดถูกบีบอัดแล้ว สมมติว่าเป็น N ไบต์ ตัวบีบอัดจะค้นหาสตริงที่ยาวที่สุดใน N ไบต์ ซึ่งตรงกับสตริงเริ่มต้นที่ไบต์ซึ่งยังไม่ได้ประมวลผลถัดไป หากรายการตรงกันที่ยาว ที่สุดมีความยาว 0 หรือ 1 ไบต์ถัดไปจะถูกเข้ารหัสเป็นอักขระ "ดิบ" มิฉะนั้น สตริงมีการแสดงแทนเป็นคู่ (ตัวชี้,ความยาว) และจำนวนไบต์ที่ประมวลผลจะเพิ่มขึ้นตามความยาว ตามสถาปัตยกรรม IBM L Z สนับสนุนค่า N ของ 512, 1024, หรือ 2048 IBM L Z ระบุการเข้ารหัสของ คู่ (ตัวชี้,ความยาว) และของอักขระดิบ ตัวชี้เป็นฟิลด์ความยาวคงที่ ขนาด $\log_2 N$ ในขณะที่ความยาวมีการเข้ารหัสเป็นฟิลด์ความยาวแปรผัน

ต้นทุนประสิทธิภาพของการบีบอัดข้อมูล JFS:

เนื่องจากการบีบอัดข้อมูลเป็นส่วนขยายของการสนับสนุนแฟร็กเมนต์ ประสิทธิภาพที่เชื่อมโยงกับแฟร็กเมนต์จึงใช้กับการบีบอัดข้อมูลด้วย

ระบบไฟล์ที่บีบอัดยังส่งผลกระทบต่อประสิทธิภาพการทำงานในวิธีต่อไปนี้ด้วย:

- ระบบไฟล์จำเป็นต้องใช้เวลาในการบีบอัดและคลายบีบอัดข้อมูลนาน ดังนั้นการใช้งานระบบไฟล์แบบบีบอัดอาจถูกจำกัดสำหรับสถานะแวดล้อมของผู้ใช้บางราย
- ไฟล์ปกติ UNIX ส่วนใหญ่ ถูกเขียนเพียงครั้งเดียว แต่บางไฟล์ถูกอัปเดต สำหรับอัปเดต การบีบอัดข้อมูล มีต้นทุน ประสิทธิภาพเพิ่มเติมของการจัดสรร 4096 ไบต์ของพื้นที่ว่างดิสก์เมื่อแก้ไขบล็อกเชิงตรรกะเป็นครั้งแรก จากนั้นจัดสรรพื้นที่ว่างดิสก์อีกครั้งหลังจากบล็อกเชิงตรรกะถูกเขียนลงในดิสก์ กิจกรรมการจัดสรรเพิ่มเติมนี้ไม่จำเป็นสำหรับไฟล์ปกติในระบบไฟล์ที่ไม่ได้บีบอัด
- การบีบอัดข้อมูลเพิ่มจำนวนของรอบตัวประมวลผล สำหรับตัวบีบอัดซอฟต์แวร์ จำนวนรอบของการบีบอัดคือประมาณ 50 รอบ ต่อไบต์ และสำหรับการคลายบีบอัดคือ 10 รอบต่อไบต์

การสำรองแบบออนไลน์ JFS และ JFS2 snapshots:

You can make a point-in-time image of a JFS file system or of a JFS2 file system, which you can then use for backup purposes. อย่างไรก็ตาม มีความแตกต่างในข้อกำหนดและลักษณะการทำงานของรูปภาพนี้สำหรับระบบไฟล์แต่ละชนิด

สำหรับระบบไฟล์ JFS คุณสามารถแบ่งสำเนาเสถียรแบบอ่านอย่างเดียวของสำเนาที่มิร์เรอร์ของระบบไฟล์ โดยปกติ สำเนาที่มิร์เรอร์มีการอัปเดตในทุกเมื่อที่อัปเดต ระบบไฟล์ดั้งเดิม แต่สำเนา point-in-time นี้ไม่เปลี่ยนแปลง ยังคงรักษารูปภาพของ point in time ซึ่งจัดทำสำเนา เมื่อใช้รูปภาพนี้สำหรับการทำสำเนาสำรอง การแก้ไขใดๆ ที่เริ่มขึ้นหลังจากคุณ เริ่มต้นโปรแกรม เพื่อสร้างรูปภาพอาจจะไม่มีอยู่ในสำเนา สำรอง ดังนั้นขอแนะนำให้ดำเนินกิจกรรมต่างๆ ใน ระบบไฟล์ให้น้อยที่สุดขณะที่การแบ่งเกิดขึ้น การเปลี่ยนแปลงใดๆ ที่เกิดขึ้นหลังจากทำการแบ่งอาจไม่มีอยู่ในสำเนาสำรอง

สำหรับระบบไฟล์ JFS2 รูปภาพ point-in-time เรียกว่า *snapshot* Snapshot ยังคงเสถียรและคงสภาพความปลอดภัยเดียวกับที่ระบบไฟล์ดั้งเดิม (เรียกว่า *snappedFS*) มีเมื่อจัดทำ snapshot นอกจากนี้ คุณสามารถสร้าง JFS2 snapshot ได้โดยไม่ต้องถอนติดตั้ง หรือหยุดระบบไฟล์ คุณสามารถใช้ JFS2 snapshot เพื่อใช้ในการสำรองแบบออนไลน์ของ ระบบไฟล์ เพื่อเข้าถึงไฟล์ หรือไคลเอนท์หรือถ้ามีอยู่เมื่อใช้ snapshot หรือเพื่อทำสำเนาสำรองไว้บนสื่อบันทึกที่ถอดออกได้ โปรดทราบข้อมูลต่อไปนี้เกี่ยวกับ JFS2 snapshots:

- รูปภาพ snapshot ของระบบไฟล์ราก (/) หรือ /usr ถูกเขียนทับเมื่อรีบูตระบบ Snapshots ของระบบไฟล์อื่น สามารถสำรองไว้ได้โดยการถอนติดตั้งระบบไฟล์ก่อนรีบูต Snapshots ที่สร้างขึ้นใน AIX 5.2 with 5200-01 สามารถกู้คืนได้ เมื่อ fsck หรือ logredo รีบระบบไฟล์ JFS2 โดยมี snapshot ที่สร้างขึ้นบน AIX 5.2 with 5200-01 snapshot จะถูกสำรองไว้ ระบบไฟล์ที่ถอนติดตั้งโดยสมบูรณ์ด้วย snapshot ที่ AIX 5.2 สร้างขึ้นยังสามารถ กู้คืนได้หลังจากถูกติดตั้งบนระบบ AIX 5.2 with 5200-01 แล้ว
- ไม่แนะนำให้รันคำสั่ง defragfs บนระบบไฟล์ ที่มี snapshots เนื่องจากทุกบล็อกที่ถูกย้ายในระหว่างการยกเลิกแฟรกเมนต์ ต้องถูกคัดลอกไปยัง snapshot ด้วย ซึ่งทำให้ทั้งเสียเวลาและสิ้นเปลือง พื้นที่ว่างใน snapshot โลจิคัลวอลุ่ม
- หาก snapshot ขาดพื้นที่ว่าง snapshots ทั้งหมดสำหรับ snappedFS นั้น จะถูกลบออก ความล้มเหลวนี้เขียนรายการไปยังบันทึกข้อผิดพลาด
- หากการเขียนลงใน snapshot ล้มเหลว snapshots ทั้งหมดสำหรับ snappedFS นั้นจะถูกลบออก ความล้มเหลวนี้เขียนรายการไปยังบันทึกข้อผิดพลาด
- Snapshot ที่สร้างขึ้นหรือเข้าถึงบนระบบ AIX 5.2 with 5200-01 ไม่สามารถเข้าถึงบนระบบ AIX 5.2 ต้องลบ snapshots เหล่านี้ก่อนสามารถ ติดตั้งระบบไฟล์ได้
- ระบบไฟล์ JFS2 ที่มี snapshot บน AIX 5.3 ไม่สามารถ เข้าถึงบนวิธีใดๆ ก่อนหน้า AIX 5.2 with 5200-01 หากระบบจะถูกย้ายกลับ ต้องลบ snapshots ออกก่อนเพื่อให้สามารถ เข้าถึงระบบไฟล์ได้

การสำรองแบบออนไลน์ JFS:

คุณสามารถจัดทำรูปภาพ point-in-time ของระบบไฟล์ JFS ซึ่งจากนั้น คุณสามารถใช้สำหรับวัตถุประสงค์สำเนาสำรอง

สำหรับระบบไฟล์ JFS คุณสามารถแบ่งสำเนาเสตติงแบบอ่านอย่างเดียวของ สำเนาที่มิร์เรอร์ของระบบไฟล์ โดยปกติ สำเนาที่มิร์เรอร์มีการอัปเดตในทุกเมื่อที่อัปเดต ระบบไฟล์ดั้งเดิม แต่สำเนา point-in-time นี้ไม่เปลี่ยน แต่ยังคงรักษาภาพของ point in time ซึ่งจัดทำสำเนา เมื่อใช้รูปภาพนี้สำหรับการทำสำเนาสำรอง การแก้ไขใดๆ ที่เริ่มขึ้นหลังจากคุณ เริ่มต้นโปรซีเดอร์ เพื่อสร้างรูปภาพอาจจะไม่มีอยู่ในสำเนา สำรอง ดังนั้นขอแนะนำให้ดำเนินการกิจกรรมต่างๆ ใน ระบบไฟล์ให้บ่อยที่สุดขณะที่การแบ่งเกิดขึ้น การเปลี่ยนแปลงใดๆ ที่เกิดขึ้นหลังจากทำการแบ่งอาจไม่มีอยู่ในสำเนาสำรอง

JFS2 snapshots:

คุณสามารถจัดทำรูปภาพ point-in-time ของระบบไฟล์ JFS2 ซึ่งจากนั้น คุณสามารถใช้สำหรับวัตถุประสงค์สำเนาสำรอง

รูปภาพ point-in-time ของระบบไฟล์ JFS2 เรียกว่า *snapshot* Snapshot ยังคงเสตติงและคงสิทธิความปลอดภัยเดียวกับที่ระบบไฟล์ดั้งเดิม (เรียกว่า *snappedFS*) มีเมื่อจัดทำ snapshot นอกจากนี้ คุณสามารถสร้าง JFS2 snapshot ได้โดยไม่ต้องถอนติดตั้งระบบไฟล์ หรือหยุดระบบไฟล์ คุณสามารถใช้ JFS2 snapshot เพื่อ:

- เข้าถึงไฟล์หรือไดเรกทอรีถ้ามีอยู่เมื่อใช้ snapshot
- ทำสำเนาสำรองไว้บนสื่อบันทึกที่ถอดออกได้

JFS2 snapshots มีอยู่สองชนิดคือ: ภายในและภายนอก JFS2 snapshot ภายนอกถูกสร้างขึ้นในโลจิคัลวอลุ่มแยกต่างหากจากระบบไฟล์ สามารถติดตั้ง snapshot ภายนอกแยกต่างหากจากระบบไฟล์ที่จุดต่อเฉพาะของ snapshot เองได้

JFS2 snapshot ภายในถูกสร้างขึ้นในโลจิคัลวอลุ่มเดียวกันกับ ระบบไฟล์และจัดสรรบล็อกจากระบบไฟล์ Snapshot ภายในสามารถเข้าถึงได้จากไดเรกทอรี .snapshot ที่มองไม่เห็น ในรากของระบบไฟล์ JFS2 ที่มี snapshot ระบบไฟล์ JFS2 ต้องมีการเปิดใช้งานเพื่อสนับสนุน snapshots ภายในในเวลาสร้าง ระบบไฟล์

JFS2 snapshots ไม่สนับสนุนการตรวจสอบโควตาในระบบไฟล์ คุณไม่สามารถใช้คำสั่ง `repquota` บน snapshot เพื่อกำหนดสภาพของโควตา ข้อมูลโควตา point-in-time จะถูกสงวนไว้ที่คุณ `roll back` รูปภาพระบบไฟล์ไปยังรูปภาพ snapshot โปรดทราบว่าข้อควรพิจารณาต่อไปนี้ใช้เฉพาะสำหรับ JFS2 snapshots ภายนอกและ JFS2 snapshots ภายใน:

- Snapshot ภายนอกที่สร้างขึ้นหรือเข้าถึงบนระบบ AIX 5.2 with 5200-01 ไม่สามารถเข้าถึงบนระบบ AIX 5.2 ต้องลบ snapshots เหล่านี้ก่อนสามารถติดตั้งระบบไฟล์ได้
- ระบบไฟล์ JFS2 ที่มี snapshot บน AIX 5.3 ไม่สามารถเข้าถึงบนรีลีสใดๆ ก่อนหน้า AIX 5.2 with 5200-01 หากระบบจะถูกย้ายกลับ ต้องลบ snapshots ออกก่อนเพื่อให้สามารถเข้าถึงระบบไฟล์ได้
- ไม่แนะนำให้รันคำสั่ง `defragfs` บนระบบไฟล์ JFS2 ที่มี snapshots ภายนอกเนื่องจากทุกบล็อกที่ถูกย้ายในระหว่าง การยกเลิกแฟรกเมนต์ต้องถูกคัดลอกไปยัง snapshot ด้วย ซึ่งทำให้ทั้งเสียเวลา และสิ้นเปลืองพื้นที่ว่างใน snapshot โลจิสติกส์วอลุ่ม
- หาก snapshot ภายนอกขาดพื้นที่ว่าง หรือถ้า snapshot ภายนอก ล้มเหลว snapshots ภายนอกทั้งหมดที่ทำเครื่องหมายสำหรับ snappdFS นั้นจะไม่ถูกต้อง การเข้าถึงเพิ่มเติมใน snapshot จะล้มเหลว ความล้มเหลวเหล่านี้เขียนรายการไปยังบันทึก ข้อผิดพลาด

ข้อควรพิจารณาเกี่ยวกับ JFS2 snapshot ภายใน:

- Snapshots ภายในถูกสงวนไว้เมื่อคำสั่ง `logredo` รันบนระบบไฟล์ JFS2 ที่มี snapshot ภายใน
- Snapshots ภายในจะถูกลบออกถ้าคำสั่ง `fsck` ต้องแก้ไขระบบไฟล์ JFS2 เพื่อซ่อมแซม
- หาก snapshot ภายในขาดพื้นที่ว่าง หรือถ้าการเขียนไปยัง snapshot ภายใน ล้มเหลว snapshots ภายในทั้งหมดที่ทำเครื่องหมายสำหรับ snappdFS นั้นจะไม่ถูกต้อง การเข้าถึงเพิ่มเติมใน snapshots ภายในจะล้มเหลว ความล้มเหลวเหล่านี้เขียนรายการไปยังบันทึก ข้อผิดพลาด
- Snapshots ภายในไม่สามารถติดตั้งแยกต่างหากได้ คุณสามารถเข้าถึง snapshots ภายในในไอดีเร็กทอรี .snapshot ของรากของระบบ ไฟล์ได้ทันทีหลังจากที่ snapshot ถูกสร้างขึ้น ในเวลาต่อมา คุณสามารถเข้าถึง snapshots ภายในผ่านทางเซิร์ฟเวอร์ NFS ได้โดยไม่ต้องเอ็กซ์พอร์ต จุดต่อแยกต่างหากสำหรับ snapshot
- Snapshots ภายในไม่เข้ากันกับ AIX รีลีส ก่อนหน้า AIX 6.1 ระบบไฟล์ JFS2 ที่สร้างขึ้นเพื่อสนับสนุน snapshots ภายในไม่สามารถแก้ไขบน รีลีสก่อนหน้าของ AIX
- ระบบไฟล์ JFS2 ที่สร้างขึ้นเพื่อสนับสนุน snapshots ภายในยังสามารถเปิดใช้งานเพื่อสนับสนุน Extended Attributes เวอร์ชัน 2 ได้ด้วย
- ระบบไฟล์ JFS2 ที่มี snapshots ภายในไม่สามารถใช้กับ Data Management Application Programming Interface (DMAPI)
- คุณไม่สามารถใช้คำสั่ง `defragfs` กับระบบไฟล์ JFS2 ที่มี snapshots ภายใน
- ไม่มีการส่งคืนไอดีเร็กทอรี .snapshot จากการเรียกระบบ `readdir()` ลักษณะเช่นนี้ช่วยป้องกันการเยี่ยมชม snapshots โดยไม่ได้ตั้งใจ การเรียกระบบ หรือคำสั่งใดๆ ที่ขึ้นอยู่กับ การเรียกระบบ `readdir()` ล้มเหลวด้วยไอดีเร็กทอรี .snapshot (ตัวอย่างเช่น คำสั่ง `/bin/pwd` และการเรียกระบบ `getcwd()` ของไอดีเร็กทอรี .snapshot ไม่พบ ไอดีเร็กทอรีพารেন্ট)

ความเข้ากันได้และการย้าย:

ระบบไฟล์ JFS เข้ากันได้อย่างเต็มที่ภายใน AIX 5.1 และ AIX 5.2 เวอร์ชันที่สนับสนุนก่อนหน้านี้ ของระบบปฏิบัติการเข้ากันได้กับ JFS ปัจจุบัน แม้ว่าระบบไฟล์ที่ไม่มีขนาดแฟรกเมนต์ดีฟอลต์ ค่า NBPI หรือขนาดกลุ่มการจัดสรร อาจต้องการความใส่ใจเป็นพิเศษย้ายไปยังเวอร์ชันก่อนหน้านี้

หมายเหตุ: ระบบไฟล์ JFS ด้รับสนับสนุนดิสก์ ที่มีขนาดเซ็กเตอร์ 4 KB ดังนั้น เมื่อคุณสร้างระบบไฟล์ หรือการดำเนินการสำรองข้อมูลให้ตรวจสอบว่า ดิสก์ไม่ใช่เซ็กเตอร์ ที่มีขนาด 4 KB

ระบบไฟล์ JFS2 ยกเว้น snapshots สามารถเข้ากันได้ภายใน AIX 5.1 และ AIX 5.2 แต่ไม่เข้ากันกับเวอร์ชันก่อนหน้าของระบบปฏิบัติการ ระบบไฟล์ JFS2 ที่มี snapshots ไม่ได้รับการสนับสนุนใน AIX 5.1 ตรวจสอบให้แน่ใจว่าอนติติดตั้งระบบไฟล์ JFS2 ทั้งหมดโดยสมบูรณ์แล้วเสมอ ก่อนการแปลงเป็นเวอร์ชันก่อนหน้าของ AIX เนื่องจาก คำสั่ง **logredo** อาจไม่รันบนระบบไฟล์ที่สร้างขึ้นสำหรับวิธีอื่นในภายหลัง

หมายเหตุ: ระบบไฟล์ JFS2 ที่สร้างขึ้นหรือที่แปลงเป็นรูปแบบ v2 ไม่สามารถเข้าถึงได้นับวิธีก่อนหน้าของ AIX

รายการต่อไปนี้อธิบายลักษณะที่อาจก่อให้เกิดปัญหาเกี่ยวกับระบบไฟล์ที่สร้างขึ้น ภายใต้เวอร์ชันก่อนหน้าของระบบปฏิบัติการ:

รูปภาพระบบไฟล์ JFS

รูปภาพระบบไฟล์ JFS ใดๆ ที่สร้างขึ้นด้วยขนาดแฟร็กเมนต์ดีฟอลต์และค่า NBPI เป็น 4096 ไบต์ และขนาดกลุ่มการจัดสรรดีฟอลต์ (agsize) เป็น 8 สามารถแลกเปลี่ยนกับรูปภาพระบบไฟล์ JFS ที่สร้างขึ้นภายใต้ AIX 4.3 และเวอร์ชันใหม่กว่าของระบบปฏิบัติการนี้ได้โดยไม่ต้องทำกิจกรรมการย้ายพิเศษใดๆ

หมายเหตุ: JFS2 Snapshots: JFS2 snapshots ที่สร้างขึ้นหรือที่เข้าถึงใน AIX 5L™ Version 5.2 with the 5200-01 Recommended Maintenance package ไม่สามารถเข้าถึงได้นับวิธีก่อนหน้า ต้องลบ snapshots เหล่านี้ก่อนสามารถติดตั้งระบบไฟล์ได้

ทำสำเนาสำรองและเรียกคืนระหว่างระบบไฟล์ JFS

ลำดับการทำสำเนาสำรองและเรียกคืนสามารถทำระหว่างระบบไฟล์ JFS ที่มีขนาดบล็อกแตกต่างกันได้อย่างไรก็ตาม เนื่องจากการใช้ดิสก์ที่เพิ่มมากขึ้น การดำเนินงานเรียกคืนจึงอาจล้มเหลวได้เนื่องจากขนาดบล็อกที่วางถ้านขนาดบล็อกของระบบไฟล์ต้นทางเล็กกว่าขนาดบล็อกของระบบไฟล์เป้าหมาย นี่เป็นเรื่องที่ต้องใส่ใจเป็นพิเศษสำหรับลำดับการทำสำเนาสำรองและเรียกคืน ระบบไฟล์ทั้งระบบ และอาจเกิดขึ้นแม้ในกรณีที่ขนาดทั้งหมดของระบบไฟล์ของระบบไฟล์เป้าหมายใหญ่กว่าขนาดทั้งหมดของระบบไฟล์ต้นทาง

ในขณะที่ลำดับการทำสำเนาสำรอง และเรียกคืนสามารถทำจากระบบไฟล์ที่บีบอัดไปยังระบบที่ไม่บีบอัด หรือระหว่างระบบไฟล์ที่บีบอัดซึ่งมีขนาดแฟร็กเมนต์ต่างกัน แต่เนื่องจากการใช้ดิสก์ที่มีประสิทธิภาพมากขึ้นของระบบไฟล์ที่บีบอัด การดำเนินงานเรียกคืนจึงอาจล้มเหลวได้เนื่องจากการขาดแคลนพื้นที่ว่างดิสก์ นี่เป็นเรื่องที่ต้องใส่ใจเป็นพิเศษสำหรับลำดับการทำสำเนาสำรองและเรียกคืนระบบไฟล์ทั้งระบบ และอาจเกิดขึ้นแม้ในกรณีที่ขนาดทั้งหมดของระบบไฟล์ของระบบไฟล์เป้าหมายใหญ่กว่าขนาดทั้งหมดของระบบไฟล์ต้นทาง

ข้อจำกัดไดรวเวอร์อุปกรณ์ JFS และ JFS2

ไดรวเวอร์อุปกรณ์ต้องนำเสนอดีสก์บล็อกที่เข้าถึงได้และมีขนาดเท่ากับ หรือเล็กกว่าขนาดแฟร็กเมนต์ของระบบไฟล์ JFS หรือขนาดบล็อก JFS2 ตัวอย่างเช่น ถ้าระบบไฟล์ JFS มีการจัดทำขึ้นสำหรับผู้ใช้ที่มีไดรวเวอร์อุปกรณ์ดีสก์ RAM ไดรวเวอร์ต้องอนุญาตให้บล็อกขนาด 512 ไบต์จัดเก็บระบบไฟล์ที่มี แฟร็กเมนต์ขนาด 512 ไบต์ หากไดรวเวอร์อนุญาตให้เข้าถึงได้เฉพาะระดับหน้าเท่านั้น จะสามารถใช้ได้เฉพาะ JFS ที่มีขนาดแฟร็กเมนต์เป็น 4096 ไบต์เท่านั้น

การคัดลอก JFS ไปยังฟิลิคัลวอลุ่มอื่น:

คุณสามารถคัดลอกระบบไฟล์ JFS ไปยังฟิลิคัลวอลุ่มอื่นในขณะที่ คงบูรณภาพของระบบไฟล์ไว้ได้

สถานการณ์จำลองต่อไปนี้อธิบายการคัดลอกระบบไฟล์ JFS ไปยัง ฟิลิคัลวอลุ่มอื่นในขณะที่คงบูรณภาพของระบบไฟล์ไว้

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

เมื่อต้องการคัดลอก JFS ไปยังฟิลิคัลวอลุ่มอื่นในขณะที่คงบูรณภาพของระบบไฟล์ไว้ให้ทำต่อไปนี้:

1. หยุดกิจกรรมของระบบไฟล์ซึ่งคุณต้องการคัดลอก ยกเว้นว่า แอ็พพลิเคชันที่กำลังใช้ระบบไฟล์นั้นอยู่ หรือระบบไฟล์อยู่ในสภาพที่คุณรู้จัก คุณไม่สามารถทราบสิ่งที่อยู่ในข้อมูลของ สำเนาสำรอง
2. มีร์เรอร์โลจิคัลวอลุ่ม โดยพิมพ์พาราด่วน SMIT ต่อไปนี้ บนบรรทัดคำสั่ง:

```
smit mklvcopy
```

3. คัดลอกระบบไฟล์ โดยใช้คำสั่งต่อไปนี้:

```
chfs -a splitcopy=/backup -a copy=2 /testfs
```

พารามิเตอร์ `splitcopy` สำหรับแฟล็ก `-a` ส่งผลให้คำสั่งแบ่งสำเนาที่มีร์เรอร์ของระบบไฟล์ และติดตั้ง แบบอ่านอย่างเดียวที่จุดต่อใหม่ การดำเนินการนี้แนะนำสำเนาของระบบไฟล์ พร้อมกับข้อมูลเมตาที่เจอร์นัลสอดคล้องกันซึ่งสามารถใช้สำหรับวัตถุประสงค์สำเนาสำรอง

4. ถ้าคุณต้องการย้ายสำเนาที่มีร์เรอร์ไปยังจุดต่ออื่น ให้ใช้พาราด่วน SMIT ต่อไปนี้:

```
smit cplv
```

ณ จุดนี้สามารถใช้สำเนาของระบบไฟล์ได้แล้ว

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `mklv`

ระบบไฟล์ CD-ROM และระบบไฟล์ UDF:

ระบบไฟล์ CD-ROM (CDRFS) เป็นการอิมพลีเมนต์ ระบบไฟล์โลคัลแบบอ่านอย่างเดียวที่ถูกเก็บบนสื่อบันทึก CD-ROM สื่อบันทึก CD-RW ถ้ามีการป้องกันการบันทึก และสื่อบันทึก DVD-ROM ขนาดไฟล์ CDRFS สูงสุดคือ 2 GB ไม่ว่าจะใช้สื่อบันทึกใด ระบบไฟล์ Universal Disk Format (UDF) คือการอิมพลีเมนต์ระบบไฟล์โลคัล ที่ถูกเก็บเป็นแบบอ่านอย่างเดียวบนสื่อบันทึก DVD-ROM หรืออ่าน-เขียน ในสื่อบันทึก DVD-RAM

CD ถูกเมาท์โดยอัตโนมัติตามดีฟอลต์แต่คุณลักษณะนี้สามารถปิดใช้งานได้ หากคุณลักษณะถูกปิดใช้งาน ให้ใช้คำสั่ง `cdmount` เพื่อติดตั้งระบบไฟล์ CDRFS

AIX สนับสนุน วอลุ่ม CDRFS และรูปแบบโครงสร้างไฟล์ต่อไปนี้:

ชนิด

ISO 9660:1988(E) มาตรฐาน

High Sierra Group Specification

Rock Ridge Group Protocol

CD-ROM eXtended Architecture File Format (ในรูปแบบเซกเตอร์โหมด 2 แบบฟอร์ม 1 เท่านั้น)

คำอธิบาย

CDRFS สนับสนุน ISO 9660 ระดับ 3 ของการแลกเปลี่ยนและระดับ 1 ของการนำไปใช้

ก่อนหน้านี้ ISO 9660 และนำเสนอความเข้ากันได้ย้อนหลังกับ ซีดีรอมก่อนหน้านี้

ระบุส่วนขยายของ ISO 9660 ที่ปฏิบัติตาม ISO 9660 มาตรฐาน โดยสมบูรณ์ และนำเสนอความหมายระบบไฟล์ POSIX ทั้งหมด ตามข้อมูล System Use Sharing Protocol (SUSP) และ Rock Ridge Interchange Protocol (RRIP) เพื่อเปิดใช้งานการติดตั้ง/เข้าถึงซีดีรอม ด้วยระบบไฟล์ UNIX อื่น

รูปแบบไฟล์ CD-ROM eXtended Architecture (XA) ระบุ ส่วนขยายของ ISO 9660 ที่ใช้ในแอ็พพลิเคชันมัลติมีเดียบนซีดีรอม ตัวอย่างเช่น Photo CD

สำหรับวอลุ่มและรูปแบบโครงสร้างไฟล์ทั้งหมด ใช้ข้อจำกัด ต่อไปนี้:

- ชุดวอลุ่มแบบวอลุ่มเดี่ยวเท่านั้น
- ไฟล์ non-interleaved เท่านั้น

CDRFS อาศัยไดรเวอร์อุปกรณ์ซีดีรอมที่ใช้งานอยู่ เพื่อนำเสนอความโปร่งใสของรูปแบบเซกเตอร์ฟิลิคัล (CD-ROM โหมด 1 และ CD-ROM XA โหมด 2 แบบฟอร์ม 1) และรูปแบบหลายเซกชันของ ดิสก์ (การแม่พชุดตัวอธิบายวอลุ่มจากพื้นที่การรับรู้วอลุ่ม ของเซกชันล่าสุด)

หมายเหตุ: CDRFS ต้องถูกถอนติดตั้งออกจากระบบก่อนคุณสามารถเอา สื่อบันทึกซีดีรอมออกได้

ยังมีระบบไฟล์อื่นที่สนับสนุนอีกคือ UDFS ซึ่ง ระบบไฟล์แบบอ่านอย่างเดียวถูกเก็บไว้บนสื่อบันทึก DVD-ROM UDFS ต้องถูกถอนติดตั้งออกจากระบบ ก่อนคุณสามารถเอาสื่อบันทึกออกได้ AIX สนับสนุนรูปแบบ UDFS เวอร์ชัน 1.50, 2.00, และ 2.01

UDFS ต้องมีการเอ็กซ์พอร์ตโดยใช้ NFS ในโหมดอ่านอย่างเดียว ไม่สนับสนุนการเขียนใน NFS ที่ติดตั้ง UDFS

เมื่อต้องการใช้คำสั่ง `cdmount` เพื่อติดตั้ง UDFS การอ่าน/เขียนโดยอัตโนมัติ ให้แก้ไขไฟล์ `cdromd.conf` คุณยังสามารถติดตั้ง UDFS การอ่าน/เขียนด้วยตนเองได้ด้วยโดยใช้คำสั่ง `mount`

ไดเร็กทอรี

ไดเร็กทอรี คือชนิดที่ไม่ซ้ำกันของไฟล์ซึ่งมีเฉพาะ ข้อมูลที่จำเป็นสำหรับการเข้าถึงไฟล์หรือไดเร็กทอรีอื่น ส่งผลให้ไดเร็กทอรีใช้พื้นที่ว่างน้อยกว่าไฟล์ชนิดอื่น

ระบบไฟล์ ประกอบด้วยกลุ่มของไดเร็กทอรีและไฟล์ภายใน ไดเร็กทอรี โดยทั่วไประบบไฟล์มีการแสดงแทนโดยใช้แผนผังผัดพัน ไดเร็กทอรีรากซึ่งบ่งชี้ด้วยสัญลักษณ์ทับ (/) กำหนดระบบไฟล์ และปรากฏขึ้นที่ด้านบนสุดของไดอะแกรมแผนผังระบบไฟล์

สาขาไดเร็กทอรีต่อลงมาจากไดเร็กทอรีราก ในไดอะแกรมแผนผัง และมีทั้งไฟล์และไดเร็กทอรีย่อย การแตกสาขา สร้างพาทที่ไม่ซ้ำกันผ่านทางโครงสร้างไดเร็กทอรีไปยังทุกอ็อบเจกต์ใน ระบบไฟล์

คอลเล็กชันของไฟล์มีการจัดเก็บไว้ในไดเร็กทอรี คอลเล็กชันของไฟล์เหล่านี้มักเกี่ยวข้องกับซึ่งกันและกัน ดังนั้นการจัดเก็บคอลเล็กชันไว้ใน โครงสร้างของไดเร็กทอรีจึงช่วยในการจัดระเบียบ

ไฟล์ คือคอลเล็กชันของข้อมูลที่สามารถถูกอ่าน หรือเขียนได้ ไฟล์สามารถเป็นโปรแกรมซึ่งคุณสร้างขึ้น ข้อมูลที่คุณเขียน ข้อมูล ซึ่งคุณจัดหา หรืออุปกรณ์ที่คุณใช้ คำสั่ง เครื่องพิมพ์ เทอร์มินัล การโต้ตอบ และแอปพลิเคชันโปรแกรม ล้วนมีการจัดเก็บไว้ในไฟล์ทั้งหมด ซึ่งช่วยให้ผู้ใช้สามารถเข้าถึง อิลิเมนต์ที่หลากหลายของระบบในวิธีรูปแบบเดียวกันและให้ความยืดหยุ่นสูงสุด ในระบบไฟล์

ไดเร็กทอรีอนุญาตให้คุณจัดกลุ่มไฟล์และไดเร็กทอรีอื่น เพื่อจัดระเบียบระบบไฟล์เป็นลำดับชั้นจำเพาะ ซึ่งทำให้โครงสร้างระบบไฟล์ มีความยืดหยุ่นและความลึก

ไดเร็กทอรีประกอบด้วยรายการไดเร็กทอรี แต่ละรายการมีชื่อไฟล์หรือไดเร็กทอรีย่อยและหมายเลขอ้างอิงโหนดดัชนี (หมายเลข *ไอ-โหนด*) เพื่อเพิ่มความเร็วและประสิทธิภาพในการใช้พื้นที่ว่างดิสก์ ข้อมูลในไฟล์จึงมีการจัดเก็บไว้ในที่ตั้งต่างๆ ในหน่วยความจำของคอมพิวเตอร์ หมายเลขไอ-โหนดมีแอดเดรสที่ใช้เพื่อระบุตำแหน่งบล็อกที่กระจายทั้งหมดของข้อมูลซึ่งเชื่อมโยงกับ ไฟล์ หมายเลขไอ-โหนดยังบันทึกข้อมูลอื่นๆ เกี่ยวกับไฟล์ รวมถึง เวลาของการแก้ไขและการเข้าถึง โหมดการเข้าถึง จำนวนลิงก์ เจ้าของไฟล์ และชนิดไฟล์

มีชุดพิเศษของคำสั่งควบคุมไดเร็กทอรี ตัวอย่างเช่น คุณสามารถลิงก์ หลายชื่อของไฟล์กับหมายเลขไอ-โหนดเดียวกันได้ โดยการสร้างรายการไดเร็กทอรี ด้วยคำสั่ง `ln`

เนื่องจากไดเร็กทอรีมักมีข้อมูลที่ไม่ควรจะมีอยู่สำหรับผู้ใช้ทุกรายของระบบ ดังนั้นจึงสามารถป้องกันการเข้าถึงไดเร็กทอรีได้ โดยการตั้งค่าสิทธิอนุญาตของไดเร็กทอรี คุณสามารถควบคุมบุคคลที่มีสิทธิเข้าถึงไดเร็กทอรี และยังสามารถกำหนดผู้ใช้ (ถ้ามี) ซึ่งสามารถเปลี่ยนข้อมูล ภายในไดเร็กทอรีได้ด้วย

หลักการที่เกี่ยวข้อง:

“โหมตการเข้าถึงไฟล์และไดเร็กทอรี” ในหน้า 323

ทุกไฟล์มีเจ้าของ สำหรับไฟล์ใหม่ ผู้ใช้ที่สร้างไฟล์เป็นเจ้าของของไฟล์นั้น เจ้าของกำหนด โหมตการเข้าถึงให้กับไฟล์ โหมตการเข้าถึงมอบสิทธิการอ่าน การแก้ไข หรือการดำเนินการในไฟล์ให้แก่ผู้ใช้ระบบรายอื่น เฉพาะเจ้าของไฟล์หรือผู้ใช้ที่มีสิทธิการใช้งานรากเท่านั้น สามารถเปลี่ยนโหมตการเข้าถึงของไฟล์ได้

ชนิดของไดเร็กทอรี:

ไดเร็กทอรีสามารถกำหนดขึ้นโดยระบบปฏิบัติการ โดยผู้ดูแล ระบบ หรือโดยผู้ใช้

ไดเร็กทอรีที่ระบบกำหนดมีประเภทเฉพาะของ ไฟล์ระบบ เช่น คำสั่ง ที่ด้านบนสุดของลำดับชั้นระบบไฟล์ คือไดเร็กทอรี / (ราก) ที่ระบบกำหนด โดยปกติ ไดเร็กทอรี / (ราก) มีไดเร็กทอรีที่เกี่ยวข้องกับระบบมาตรฐานต่อไปนี้:

ไอเท็ม	คำอธิบาย
/dev	มีไฟล์พิเศษสำหรับอุปกรณ์ I/O
/etc	มีไฟล์สำหรับการเริ่มต้นระบบและการจัดการระบบ
/home	มีไดเร็กทอรีล็อกอินสำหรับผู้ใช้ระบบ
/tmp	มีไฟล์ที่เป็นไฟล์ชั่วคราวและถูกลบออกโดยอัตโนมัติหลังจากจำนวนวันที่ระบุ
/usr	มี lpp, include, และไดเร็กทอรีระบบอื่น
/usr/bin	มีโปรแกรมที่ผู้ใช้ดำเนินการได้

บางไดเร็กทอรี เช่น ไดเร็กทอรีล็อกอินหรือโฮม (\$HOME) มีการกำหนดขึ้นและกำหนดเองโดยผู้ดูแลระบบ เมื่อคุณล็อกอินเข้าสู่ระบบปฏิบัติการ ไดเร็กทอรีล็อกอินเป็นไดเร็กทอรี ปัจจุบัน

ไดเร็กทอรีซึ่งคุณสร้างขึ้นเรียกว่าไดเร็กทอรี *ที่ผู้ใช้กำหนด* ไดเร็กทอรีเหล่านี้ช่วยให้คุณจัดระเบียบและรักษาไฟล์ของคุณได้

การจัดระเบียบไดเร็กทอรี:

ไดเร็กทอรีมีไฟล์ ไดเร็กทอรีย่อย หรือส่วนผสมของทั้งสอง อย่าง *ไดเร็กทอรีย่อย* คือไดเร็กทอรีภายในไดเร็กทอรี ไดเร็กทอรีที่มีไดเร็กทอรีย่อยเรียกว่า *ไดเร็กทอรีพาเรนต์*

แต่ละไดเร็กทอรีมีรายการสำหรับไดเร็กทอรีพาเรนต์ ซึ่งไดเร็กทอรีถูกสร้างขึ้น .. (จุดจุด) และรายการสำหรับ ตัวไดเร็กทอรีเอง . (จุด) ในรายการไดเร็กทอรี ส่วนใหญ่ไฟล์เหล่านี้ถูกซ่อนไว้

แผนผังไดเร็กทอรี

โครงสร้างระบบไฟล์ของไดเร็กทอรีอาจกลายเป็นเรื่องซับซ้อนได้โดยง่าย ฉะนั้น ควร พยายามรักษาโครงสร้างไฟล์ และไดเร็กทอรีให้ง่ายที่สุดเท่าที่เป็นไปได้ สร้างไฟล์ และไดเร็กทอรีด้วยชื่อที่จดจำได้ง่าย ซึ่งทำให้การทำงานกับไฟล์ง่ายขึ้น

ไดเร็กทอรีพาเรนต์

แต่ละไดเร็กทอรี ยกเว้น / (ราก) มีหนึ่งไดเร็กทอรีพาเรนต์ และอาจมีไดเร็กทอรีชายน

ไดเร็กทอรีโฮม

เมื่อคุณล็อกอิน ระบบจะวางคุณไว้ในไดเร็กทอรีที่เรียกว่า *ไดเร็กทอรีโฮม* หรือไดเร็กทอรีล็อกอินของคุณ ไดเร็กทอรี

ดังกล่าวมีการตั้งค่าโดยผู้ดูแลระบบสำหรับผู้ดูแลระบบ ไดรฟ์ทอริโฮมเป็นที่เก็บไฟล์ส่วนบุคคลของคุณ โดยปกติ ไดรฟ์ทอริซึ่งคุณสร้างขึ้นสำหรับการใช้งานของคุณเองจะเป็นไดรฟ์ทอริย่อยของไดรฟ์ทอริโฮมของคุณ เมื่อต้องการกลับไปยังไดรฟ์ทอริโฮมในเวลาใดๆ ให้พิมพ์คำสั่ง `cd` และกด `Enter` ที่พร้อม

ไดรฟ์ทอริการทำงาน

คุณกำลังทำงานภายในไดรฟ์ทอริเสมอ ไดรฟ์ทอริใดก็ตามซึ่งคุณกำลังทำงานอยู่ในปัจจุบันเรียกว่าไดรฟ์ทอริปัจจุบัน หรือ *การทำงาน* คำสั่ง `pwd` (แสดง ไดรฟ์ทอริการทำงาน) รายงานชื่อของไดรฟ์ทอริการทำงานของคุณ ใช้คำสั่ง `cd` เพื่อเปลี่ยนไดรฟ์ทอริการทำงาน

ระเบียบการตั้งชื่อไดรฟ์ทอริ:

ชื่อของแต่ละไดรฟ์ทอริต้องไม่ซ้ำกันภายในไดรฟ์ทอริ ซึ่งจัดเก็บ ลักษณะเช่นนี้ทำให้มั่นใจว่าไดรฟ์ทอริมีชื่อพาธที่ไม่ซ้ำกันในระบบไฟล์

ไดรฟ์ทอริใช้ระเบียบการตั้งชื่อเดียวกันกับ ไฟล์ ดังที่อธิบายใน “ระเบียบการตั้งชื่อไฟล์” ในหน้า 210

คำย่อไดรฟ์ทอริ:

คำย่อเป็นวิธีที่สะดวกในการระบุไดรฟ์ทอริบางรายการ

ข้อมูลต่อไปนี้เป็นรายการของคำย่อ:

คำย่อ	ความหมาย
.	ไดรฟ์ทอริทำงานปัจจุบัน
..	ไดรฟ์ทอริข้างบนไดรฟ์ทอริทำงานปัจจุบัน (พาเรนต์ของ ไดรฟ์ทอริปัจจุบัน)
~	ไดรฟ์ทอริโฮมของคุณ (ไม่เป็นจริงสำหรับเซลล์เป้าหมาย สำหรับข้อมูล เพิ่มเติม ให้ดูที่ “เซลล์เป้าหมาย” ในหน้า 283)
\$HOME	ไดรฟ์ทอริโฮมของคุณ (เป็นจริงสำหรับเซลล์ทั้งหมด)

ชื่อพาธไดรฟ์ทอริ:

แต่ละไฟล์และไดรฟ์ทอริสามารถเข้าถึงโดยใช้พาธเฉพาะ ซึ่งรู้จักว่า เป็น *ชื่อพาธ* ผ่านทางโครงสร้างแผนผังระบบไฟล์ ชื่อพาธระบุที่ตั้งของไดรฟ์ทอริหรือไฟล์ภายในระบบไฟล์

หมายเหตุ: ชื่อพาธต้องยาวไม่เกิน 1023 อักขระ

ระบบไฟล์ใช้ชื่อพาธประเภทต่อไปนี้:

ไอเท็ม	คำอธิบาย
ชื่อพาธสัมบูรณ์	ติดตามพาธจากไดรฟ์ทอริ / (root) ชื่อพาธสัมบูรณ์ ขึ้นด้วยสัญลักษณ์ทับ (/) เสมอ
ชื่อพาธที่สัมพันธ์	ติดตามพาธจากไดรฟ์ทอริปัจจุบันผ่านทางพาเรนต์หรือ ไดรฟ์ทอริย่อยและไฟล์

ชื่อพาธสัมบูรณ์แสดงถึงชื่อที่สมบูรณ์ ของไดรฟ์ทอริหรือไฟล์ตั้งแต่ด้านล่างของไดรฟ์ทอริ / (ราก) ไม่ว่าคุณกำลังทำงานในตำแหน่งใดในระบบไฟล์ คุณสามารถค้นหาไดรฟ์ทอริ หรือไฟล์ได้เสมอโดยการระบุชื่อพาธสัมบูรณ์ของไดรฟ์ทอริหรือไฟล์นั้น ชื่อพาธสัมบูรณ์ขึ้นต้นด้วยเครื่องหมายทับ (/) สัญลักษณ์ที่แสดงถึงไดรฟ์ทอริ ราก ชื่อพาธ /A/D/9 เป็นชื่อพาธสัมบูรณ์สำหรับ 9 เครื่องหมายทับแรก (/) แสดงถึงไดรฟ์ทอริ / (ราก) ซึ่งเป็นตำแหน่งเริ่มต้น สำหรับการค้นหา ส่วนที่เหลือของชื่อพาธกำหนดทิศทางการค้นหาไปยัง A, จากนั้น D, และสุดท้ายไปยัง 9

สามารถมีไฟล์ที่ชื่อ 9 สองไฟล์ได้ เนื่องจากชื่อพาสสัมบูรณ์ไปยังไฟล์ให้ชื่อที่ไม่ซ้ำกันสำหรับแต่ละไฟล์ภายในระบบไฟล์ชื่อพาส /A/D/9 และ /C/E/G/9 ระบบไฟล์ที่ไม่ซ้ำกันสองไฟล์ที่ชื่อ 9

สิ่งที่ไม่เหมือนกับชื่อพาสแบบเต็มคือชื่อพาสที่สัมพันธ์ระบบไดเรกทอรีหรือไฟล์ตามข้อมูลไดเรกทอรีทำงานปัจจุบัน สำหรับชื่อพาสสัมพันธ์ คุณสามารถใช้สัญลักษณ์จุดสองจุด (..) เพื่อเลื่อนขึ้นในลำดับชั้นของระบบไฟล์จุดสองจุด (..) แทนค่าไดเรกทอรีพารেন্ট เนื่องจากชื่อพาสที่สัมพันธ์ระบบพาสที่เริ่มต้นในไดเรกทอรีปัจจุบัน ดังนั้นจึงไม่ได้ขึ้นต้นด้วยเครื่องหมายทับ (/) ชื่อพาสที่สัมพันธ์ใช้เพื่อระบุชื่อของไฟล์ในไดเรกทอรีปัจจุบัน หรือชื่อพาสของไฟล์หรือไดเรกทอรีข้างบนหรือข้างล่างระดับของไดเรกทอรีปัจจุบันในระบบไฟล์ หาก D เป็น ไดเรกทอรีปัจจุบัน ชื่อพาสที่สัมพันธ์สำหรับการเข้าถึง 10 คือ F/10 อย่างไรก็ตาม ชื่อพาสสัมบูรณ์เป็น /A/D/F/10 เสมอ นอกจากนี้ชื่อพาสที่สัมพันธ์สำหรับการเข้าถึง 3 คือ .././B/3

คุณยังสามารถแสดงถึงชื่อของไดเรกทอรีปัจจุบันโดยใช้สัญลักษณ์จุด (.) ได้ด้วย. สัญลักษณ์จุด (.) ใช้กันโดยทั่วไปเมื่อรันโปรแกรมที่อ่านชื่อไดเรกทอรีปัจจุบัน

การสร้างไดเรกทอรี (คำสั่ง `mkdir`):

ใช้คำสั่ง `mkdir` เพื่อสร้างหนึ่งหรือหลาย ไดเรกทอรีที่ระบุโดยพารามิเตอร์ *Directory*

แต่ละไดเรกทอรีใหม่มีชุดรายการมาตรฐาน (.) และจุดจุด (..). คุณสามารถระบุสิทธิอนุญาตสำหรับ ไดเรกทอรีใหม่โดยใช้แฟล็ก `-m Mode`

เมื่อคุณสร้าง ไดเรกทอรี ไดเรกทอรีจะถูกสร้างขึ้นภายในไดเรกทอรีปัจจุบันหรือไดเรกทอรีทำงาน ยกเว้นว่าคุณระบุชื่อพาสสัมบูรณ์ไปยังที่ตั้งอื่นในระบบไฟล์

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง `mkdir`:

- เมื่อต้องการสร้างไดเรกทอรีใหม่ชื่อ Test ในไดเรกทอรีทำงานปัจจุบันด้วยสิทธิอนุญาตดีฟอลต์ให้พิมพ์ดังต่อไปนี้:
`mkdir Test`
- เมื่อต้องการสร้างไดเรกทอรีชื่อ Test ด้วยสิทธิอนุญาต `rw-r-xr-x` ในไดเรกทอรี `/home/demo/sub1` ที่สร้างขึ้นก่อนหน้านี้ให้พิมพ์ดังต่อไปนี้:
`mkdir -m 755 /home/demo/sub1/Test`
- เมื่อต้องการสร้างไดเรกทอรีชื่อ Test ด้วยสิทธิอนุญาตดีฟอลต์ในไดเรกทอรี `/home/demo/sub2` ให้พิมพ์ดังต่อไปนี้:
`mkdir -p /home/demo/sub2/Test`

แฟล็ก `-p` สร้างไดเรกทอรี `/home`, `/home/demo`, และ `/home/demo/sub2` ถ้ายังไม่มีไดเรกทอรีดังกล่าว

ให้ดูที่คำสั่ง `mkdir` ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

การย้ายหรือการเปลี่ยนชื่อไดเรกทอรี (คำสั่ง `mvdir`):

ใช้คำสั่ง `mvdir` เพื่อย้ายหรือเปลี่ยนชื่อไดเรกทอรี

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง `mvdir`:

- เมื่อต้องการย้ายไดเรกทอรีให้พิมพ์ดังต่อไปนี้:
`mvdir book manual`

คำสั่งนี้ย้ายไดเรกทอรี book ภายใต้ไดเรกทอรีที่ชื่อ manual ถ้ามีไดเรกทอรี manual อยู่มีฉะนั้น ไดเรกทอรี book จะถูกเปลี่ยนชื่อเป็น manual

- เมื่อต้องการย้ายและเปลี่ยนชื่อไดเรกทอรี ให้พิมพ์ดังต่อไปนี้:

```
mmdir book3 proj4/manual
```

ถ้ามีไดเรกทอรีที่ชื่อ manual อยู่แล้ว คำสั่งนี้จะย้าย book3 และ เนื้อหาไปยัง proj4/manual อธิบายอีกอย่างคือ book3 กลายเป็น ไดเรกทอรีย่อยของ proj4/manual ถ้า manual ไม่มีอยู่ คำสั่งนี้จะเปลี่ยนชื่อไดเรกทอรี book3 เป็น proj4/manual

ให้ดูที่คำสั่ง **mmdir** ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงไดเรกทอรีปัจจุบัน (คำสั่ง pwd):

ใช้คำสั่ง **pwd** เพื่อเขียนชื่อพาธแบบเต็มของไดเรกทอรี ปัจจุบันของคุณ (จากไดเรกทอรี / (ราก)) ไปยังเอาต์พุตมาตรฐาน

ไดเรกทอรีทั้งหมดมีการแบ่งด้วยเครื่องหมายทับ (/) ไดเรกทอรี / (ราก) มีการแสดงแทนโดยเครื่องหมายทับแรก (/) และชื่อไดเรกทอรีลำดับสุดท้ายคือไดเรกทอรีปัจจุบันของคุณ

ตัวอย่าง เช่น เมื่อต้องการแสดงไดเรกทอรีปัจจุบันของคุณ ให้พิมพ์ดังต่อไปนี้:

```
pwd
```

ชื่อพาธ แบบเต็มของไดเรกทอรีปัจจุบันของคุณแสดงขึ้นคล้ายกับดังต่อไปนี้:

```
/home/thomas
```

การเปลี่ยนเป็นไดเรกทอรีอื่น (คำสั่ง cd):

ใช้คำสั่ง **cd** เพื่อย้ายจากไดเรกทอรีปัจจุบันของคุณ ไปยังไดเรกทอรีอื่น คุณต้องมีสิทธิ์การดำเนินการ (ค้นหา) ในไดเรกทอรีที่ระบุ

หากคุณไม่ได้รับูพารามิเตอร์ *ไดเรกทอรี* คำสั่ง **cd** จะย้ายคุณไปยังไดเรกทอรีลือกอื่น (\$HOME ใน สภาวะแวดล้อม **ksh** และ **bsh** หรือ \$home ในสภาวะแวดล้อม **csh**) ถ้าชื่อไดเรกทอรีที่ระบุเป็น ชื่อพาธแบบเต็ม ชื่อนั้นจะกลายเป็นไดเรกทอรีปัจจุบัน ชื่อพาธแบบเต็มขึ้นต้นด้วย เครื่องหมายสแลช (/) ที่บ่งชี้ถึงไดเรกทอรี / (root) จุด (.) ที่บ่งชี้ถึงไดเรกทอรีปัจจุบัน หรือจุดสองจุด (..) บ่งชี้ไดเรกทอรีพารেন্ট หากชื่อไดเรกทอรี ไม่ใช่ชื่อพาธแบบเต็ม คำสั่ง **cd** จะค้นหาว่าสัมพันธ์กับพาธใดในพาธที่ระบุ ตัวแปรเซลล์ **\$CDPATH** (หรือตัวแปร **\$cdpath csh**) ตัวแปรนี้มีไวยากรณ์เหมือนกับ และมีความหมายคล้ายกับตัวแปรเซลล์ **\$PATH** (หรือตัวแปร **\$path csh**)

ข้อมูลต่อไปนี้ เป็นตัวอย่างวิธีการใช้คำสั่ง **cd**:

- เมื่อต้องการเปลี่ยนไดเรกทอรีโฮมของคุณ ให้พิมพ์ดังต่อไปนี้:

```
cd
```

- เมื่อต้องการเปลี่ยนเป็นไดเรกทอรี /usr/include ให้พิมพ์ ดังต่อไปนี้:

```
cd /usr/include
```

- เมื่อต้องการลงไปในหนึ่งระดับในแผนผังไดเรกทอรีเป็นไดเรกทอรี sys ให้พิมพ์ดังต่อไปนี้:

```
cd sys
```

หากไดเรกทอรีปัจจุบันเป็น /usr/include และมีไดเรกทอรีย่อยชื่อ sys ดังนั้น /usr/include/sys จะกลายมาเป็นไดเรกทอรีปัจจุบัน

- เมื่อต้องการขึ้นไปหนึ่งระดับในแผนผังไดเรกทอรีให้พิมพ์ดังต่อไปนี้:

```
cd ..
```

ชื่อไฟล์พิเศษจุดสองจุด (..) โปรดอ้างอิงถึงไดเรกทอรีที่อยู่ด้านบนไดเรกทอรีปัจจุบัน นั่นคือไดเรกทอรีพารেন্ট

ให้ดูที่คำสั่ง `cd` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

การคัดลอกไดเรกทอรี (คำสั่ง `cp`):

ใช้คำสั่ง `cp` เพื่อสร้างสำเนาของเนื้อหาไฟล์ หรือไดเรกทอรีที่ระบุโดยพารามิเตอร์ *SourceFile* หรือ *SourceDirectory* เข้าในไฟล์ หรือไดเรกทอรีที่ระบุโดยพารามิเตอร์ *TargetFile* หรือ *TargetDirectory*

หากมีไฟล์ที่ระบุเป็น *TargetFile* สำเนาจะ เขียนทับเนื้อหาดั้งเดิมของไฟล์ ถ้าคุณกำลังคัดลอกมากกว่า หนึ่ง *SourceFile* เป้าหมายต้องเป็นไดเรกทอรี

เพื่อวางสำเนาของ *SourceFile* เข้าในไดเรกทอรีให้ระบุพาทไปยังไดเรกทอรีที่มีอยู่สำหรับพารามิเตอร์ *TargetDirectory* ฟิล์รักษาชื่อตามลำดับของไฟล์ไว้ เมื่อถูกคัดลอกไปยังไดเรกทอรี ยกเว้นว่าคุณระบุชื่อไฟล์ใหม่ที่ ตอนท้ายของพาท คำสั่ง `cp` ยังคัดลอกทั้งไดเรกทอรี เข้าในไดเรกทอรีอื่นถ้าคุณระบุแฟล็ก `-r` หรือ `-R`

ข้อมูลต่อไปนี้ เป็นตัวอย่างวิธีการใช้คำสั่ง `cp`:

- เมื่อต้องการคัดลอกไฟล์ทั้งหมดในไดเรกทอรี /home/accounts/customers/orders ไปยังไดเรกทอรี /home/accounts/customers/shipments ให้พิมพ์ดังต่อไปนี้:

```
cp /home/accounts/customers/orders/* /home/accounts/customers/shipments
```

คำสั่งนี้ คัดลอกไฟล์ แต่ไม่คัดลอกไดเรกทอรีจากไดเรกทอรี orders เข้าในไดเรกทอรี shipments

- เมื่อต้องการคัดลอกไดเรกทอรี รวมถึงไฟล์และไดเรกทอรีย่อยทั้งหมดของไดเรกทอรีนั้น ไปยังไดเรกทอรีอื่น ให้พิมพ์ดังต่อไปนี้:

```
cp -R /home/accounts/customers /home/accounts/vendors
```

คำสั่งนี้ คัดลอกไดเรกทอรี customers รวมถึงไฟล์และไดเรกทอรีย่อยของ ไดเรกทอรีนั้นทั้งหมด และไฟล์ในไดเรกทอรีย่อยเหล่านั้น เข้าในไดเรกทอรี vendors

ให้ดูที่คำสั่ง `cp` ใน *Commands Reference, Volume 1* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงผลเนื้อหาของไดเรกทอรี (คำสั่ง `ls`):

ใช้คำสั่ง `ls` เพื่อแสดงผลเนื้อหาของไดเรกทอรี

คำสั่ง `ls` เขียนเนื้อหาของแต่ละ *ไดเรกทอรี* ที่ระบุ หรือชื่อของแต่ละ *ไฟล์* ที่ระบุ พร้อมกับข้อมูลอื่นใดซึ่งคุณร้องขอโดยใช้แฟล็ก ไปยังเอาต์พุตมาตรฐาน หากคุณไม่ได้ระบุ *ไฟล์* หรือ *ไดเรกทอรี* คำสั่ง `ls` จะแสดงผลเนื้อหาของไดเรกทอรีปัจจุบัน

โดยคำตีพอลต์ คำสั่ง Is แสดงข้อมูลทั้งหมดในลำดับตามตัวอักษรตามชื่อไฟล์ ถ้าคำสั่ง ถูกเรียกใช้โดยผู้ใช้ที่มีสิทธิแบบผู้ใช้ root คำสั่งจะใช้แฟล็ก -A ตามคำตีพอลต์ และแสดงรายการทั้งหมดยกเว้นจุด (.) และจุดสองจุด (.). เมื่อต้องการแสดงรายการทั้งหมดของไฟล์ รวมถึงรายการที่ขึ้นต้นด้วยจุด (.) ให้ใช้คำสั่ง Is -a

คุณสามารถจัดรูปแบบเอาต์พุต ในวิธีต่อไปนี้:

- แสดงหนึ่งรายการต่อบรรทัด โดยใช้แฟล็ก -l
- แสดงรายการในหลายคอลัมน์ โดยการระบุแฟล็ก -C หรือ -x แฟล็ก -C เป็นรูปแบบตีพอลต์เมื่อเอาต์พุตเป็น tty
- แสดงรายการในชุดที่คั่นด้วยเครื่องหมายจุลภาคโดยการระบุแฟล็ก -m

เพื่อกำหนดจำนวนของตำแหน่งอักขระในเส้นเอาต์พุต คำสั่ง Is จะใช้ตัวแปรสถานะแวดล้อม \$COLUMNS ถ้าไม่ได้ตั้งค่าตัวแปรนี้ คำสั่งจะอ่านไฟล์ terminfo หากคำสั่ง Is ไม่สามารถกำหนดจำนวนของตำแหน่งอักขระ โดยเมธอดอย่างใดอย่างหนึ่งเหล่านี้ คำสั่งจะใช้คำตีพอลต์เป็น 80

ข้อมูล ที่แสดงขึ้นด้วยแฟล็ก -e และ -l มีการตีความ ดังนี้:

อักขระแรกของแต่ละรายการอาจเป็นอย่างใดอย่างหนึ่งต่อไปนี้:

ไอเท็ม	คำอธิบาย
d	รายการเป็นไดเรกทอรี
b	รายการเป็นไฟล์พิเศษบล็อก
c	รายการเป็นไฟล์พิเศษอักขระ
l	รายการเป็นลิงก์สัญลักษณ์
p	รายการเป็นไฟล์พิเศษไฟฟ์ first-in, first-out (FIFO)
s	รายการเป็นโลคัลซ็อกเก็ต
-	รายการเป็นไฟล์ธรรมดา

อักขระเก้าตัวถัดไปจะถูกแบ่งออกเป็นสามชุด ชุดละสาม อักขระ อักขระสามตัวแรกแสดงสิทธิอนุญาตของเจ้าของไฟล์หรือไดเรกทอรี อักขระสามตัวถัดไปแสดงสิทธิอนุญาตของผู้ใช้อื่นในกลุ่ม อักขระสามตัวสุดท้ายแสดงสิทธิอนุญาตของบุคคลอื่นใดๆ ที่มีสิทธิ เข้าถึงไฟล์ อักขระสามตัวในแต่ละชุดแสดงสิทธิการอ่าน เขียน และดำเนินการ ในไฟล์ สิทธิการดำเนินการในไดเรกทอรีอนุญาตให้คุณค้นหาไฟล์ที่ระบุใน ไดเรกทอรีได้

สิทธิอนุญาตมีการบ่งชี้ดังนี้:

ไอเท็ม	คำอธิบาย
r	ให้สิทธิการอ่าน
t	เฉพาะเจ้าของไดเรกทอรีหรือเจ้าของไฟล์สามารถลบหรือเปลี่ยนชื่อไฟล์ ภายในไดเรกทอรีนั้นได้ แม้ว่าบุคคลอื่นมีสิทธิการเขียนในไดเรกทอรี
w	ให้สิทธิการเขียน (แก้ไข)
x	ให้สิทธิการดำเนินการ (ค้นหา)
-	ไม่ให้สิทธิอนุญาตที่สอดคล้องกัน

ข้อมูลที่แสดงขึ้นด้วยแฟล็ก -e เหมือนกับข้อมูลที่แสดงขึ้น ด้วยแฟล็ก -l ยกเว้นการเพิ่มอักขระลำดับที่ 11th ซึ่งมีการตีความ ดังนี้:

ไอเท็ม คำอธิบาย

- + บ่งชี้ว่าไฟล์มีข้อมูลความปลอดภัยที่ขยาย ตัวอย่างเช่น ไฟล์ อาจมีแอตทริบิวต์ ACL, TCB, หรือ TP ที่ขยายอยู่ในโหมด
- บ่งชี้ว่าไฟล์ไม่มีข้อมูลความปลอดภัยที่ขยาย

เมื่อมีการแสดงขนาดของไฟล์ในไดเรกทอรี คำสั่ง `ls` จะแสดงจำนวนบล็อกทั้งหมด รวมถึงบล็อกทางอ้อม

โปรดดูตัวอย่างต่อไปนี้:

- เมื่อต้องการแสดงรายการไฟล์ทั้งหมดในไดเรกทอรีปัจจุบัน ให้พิมพ์ดังต่อไปนี้:

```
ls -a
```

คำสั่งนี้แสดงรายการไฟล์ทั้งหมด รวมถึง

- จุด (.)
 - จุดจุด (..)
 - ไฟล์อื่นที่มีชื่อซึ่งอาจจะหรืออาจจะไม่ขึ้นต้นด้วยจุด (.)
- เมื่อต้องการแสดงข้อมูลรายละเอียด ให้พิมพ์ดังต่อไปนี้:

```
ls -l chap1 .profile
```

คำสั่งนี้แสดงรายการแบบยาวที่มีข้อมูลรายละเอียดเกี่ยวกับ `chap1` และ `.profile`

- เมื่อต้องการแสดงข้อมูลรายละเอียดเกี่ยวกับไดเรกทอรี ให้พิมพ์ดังต่อไปนี้:

```
ls -d -l . manual manual/chap1
```

คำสั่งนี้แสดงรายการแบบยาวสำหรับไดเรกทอรี `.` และ `manual` และสำหรับไฟล์ `manual/chap1` โดยไม่มีแฟล็ก `-d` คำสั่งนี้จะแสดงรายการไฟล์ในไดเรกทอรี `.` และ `manual` แทน ข้อมูลรายละเอียดเกี่ยวกับตัวไดเรกทอรีเอง

ให้ดูที่คำสั่ง `ls` ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

การลบหรือการเอาไดเรกทอรีออก (คำสั่ง `rmdir`):

ใช้คำสั่ง `rmdir` เพื่อลบไดเรกทอรีที่ระบุ โดยพารามิเตอร์ *Directory* ออกจากระบบ

ไดเรกทอรีต้องว่าง (สามารถมีเฉพาะ `.` และ `..` เท่านั้น) ก่อนคุณสามารถลบไดเรกทอรีได้ และคุณต้องมีสิทธิ์การเขียนในไดเรกทอรีพารেন্ট ใช้คำสั่ง `ls -aDirectory` เพื่อตรวจสอบว่าไดเรกทอรีว่างหรือไม่

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง `rmdir`:

- เมื่อต้องการทำให้ไดเรกทอรีว่างและลบออก ให้พิมพ์ดังต่อไปนี้:

```
rm mydir/* mydir/*.*
rmdir mydir
```

คำสั่งนี้ลบเนื้อหาของ `mydir` จากนั้นลบไดเรกทอรีที่ว่าง คำสั่ง `rm` แสดงข้อความแสดงข้อผิดพลาดเกี่ยวกับการพยายามลบจุดไดเรกทอรี (.) และจุดจุด (..) จากนั้นคำสั่ง `rmdir` จะลบทิ้งและตัวไดเรกทอรีเอง

หมายเหตุ: `rm mydir/* mydir/*.*` ลบไฟล์ที่ชื่อไม่ได้ขึ้นต้นด้วยจุดออกก่อน จากนั้นลบไฟล์ที่ชื่อขึ้นต้นด้วยจุด คำสั่ง `ls` ไม่ได้แสดงรายการ ชื่อไฟล์ที่ขึ้นต้นด้วยจุดยกเว้นว่าคุณใช้แฟล็ก `-a`

- เมื่อต้องการลบไดเรกทอรี /tmp/jones/demo/mydir และ ไดเรกทอรีทั้งหมดข้างใต้นั้น ให้พิมพ์ดังต่อไปนี้:

```
cd /tmp
rmdir -p jones/demo/mydir
```

คำสั่งลบไดเรกทอรี jones/demo/mydir ออกจากไดเรกทอรี /tmp หากไดเรกทอรีไม่ว่าง หรือคุณไม่มีสิทธิ์การเขียนในไดเรกทอรีเมื่อไดเรกทอรีจะถูกลบออก คำสั่งจะ ยุติด้วยข้อความแสดงข้อผิดพลาดที่เหมาะสม

ให้ดูที่คำสั่ง `rmdir` ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

การเปรียบเทียบเนื้อหาของไดเรกทอรี (คำสั่ง `dircmp`):

ใช้คำสั่ง `dircmp` เพื่อเปรียบเทียบสองไดเรกทอรีที่ระบุโดยพารามิเตอร์ *ไดเรกทอรี1* และ *ไดเรกทอรี2* และเขียน ข้อมูลเกี่ยวกับเนื้อหาของไดเรกทอรีไปยังเอาต์พุตมาตรฐาน

อันดับแรก คำสั่ง `dircmp` เปรียบเทียบชื่อไฟล์ในแต่ละไดเรกทอรี ถ้ามีชื่อไฟล์เดียวกันอยู่ในทั้งสองไดเรกทอรี คำสั่ง `dircmp` จะเปรียบเทียบเนื้อหาของทั้งสองไฟล์

ในเอาต์พุตคำสั่ง `dircmp` แสดง ไฟล์ที่ไม่ซ้ำในแต่ละไดเรกทอรี จากนั้นแสดงไฟล์ที่มีชื่อเหมือนกัน ในทั้งสองไดเรกทอรี แต่มีเนื้อหาต่างกัน ถ้าไม่มีการระบุแฟล็ก จะแสดงไฟล์ที่มีเนื้อหาเหมือนกันเช่นเดียวกับชื่อที่เหมือนกัน ในทั้งสองไดเรกทอรี

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้คำสั่ง `dircmp`:

- เมื่อต้องการสรุปความแตกต่างระหว่างไฟล์ในไดเรกทอรี `proj.ver1` และ `proj.ver2` ให้พิมพ์ดังต่อไปนี้:

```
dircmp proj.ver1 proj.ver2
```

คำสั่งนี้ แสดงข้อมูลสรุปความแตกต่างระหว่างไดเรกทอรี `proj.ver1` และ `proj.ver2` รายการสรุปแยกไฟล์ที่พบเฉพาะในหนึ่งไดเรกทอรีหรืออีกไดเรกทอรี และที่พบในทั้งสองไดเรกทอรี ถ้าไฟล์พบในทั้งสองไดเรกทอรี คำสั่ง `dircmp` จะทำหมายเหตุว่าเหมือนกันหรือไม่

- เมื่อต้องการแสดงรายละเอียดความแตกต่างระหว่างไฟล์ในไดเรกทอรี `proj.ver1` และ `proj.ver2` ให้พิมพ์ดังต่อไปนี้:

```
dircmp -d -s proj.ver1 proj.ver2
```

แฟล็ก `-s` ยับยั้งข้อมูลเกี่ยวกับไฟล์ที่เหมือนกัน แฟล็ก `-d` แสดง `diff` แสดงแต่ละไฟล์ที่ต่างกัน ที่พบในทั้งสองไดเรกทอรี

ให้ดูที่คำสั่ง `dircmp` ใน *Commands Reference, Volume 2* สำหรับ ไวยากรณ์ทั้งหมด

ข้อมูลสรุปคำสั่งสำหรับระบบไฟล์และไดเรกทอรี:

ข้อมูลต่อไปนี้เป็นคำสั่งสำหรับระบบไฟล์และไดเรกทอรี คำสั่ง สำหรับโปรแกรมจัดการไดเรกทอรี และรายการของคำย่อไดเรกทอรี

ตารางที่ 66. ข้อมูลสรุปคำสั่งสำหรับระบบไฟล์

ไอเท็ม	คำอธิบาย
df	รายงานข้อมูลเกี่ยวกับพื้นที่ว่างบนระบบไฟล์

ตารางที่ 67. คำย่อไอดีเร็กทอรี

ไอเท็ม	คำอธิบาย
.	ไอดีเร็กทอรีทำงานปัจจุบัน
..	ไอดีเร็กทอรีข้างบนไอดีเร็กทอรีทำงานปัจจุบัน (ไอดีเร็กทอรีพารেন্ট)
~	ไอดีเร็กทอรีโฮมของคุณ (ไม่เป็นจริงสำหรับเซลล์เป้าหมาย สำหรับข้อมูลเพิ่มเติม ให้ดูที่ “เซลล์เป้าหมาย” ในหน้า 283)
\$HOME	ไอดีเร็กทอรีโฮมของคุณ (เป็นจริงสำหรับเซลล์ทั้งหมด)

ตารางที่ 68. ข้อมูลสรุปคำสั่งสำหรับโปรแกรมจัดการไอดีเร็กทอรี

ไอเท็ม	คำอธิบาย
cd	เปลี่ยนไอดีเร็กทอรีปัจจุบัน
cp	คัดลอกไฟล์หรือไอดีเร็กทอรี
dircomp	เปรียบเทียบสองไอดีเร็กทอรีและเนื้อหาของไฟล์ปกติ
ls	แสดงเนื้อหาของไอดีเร็กทอรี
mkdir	สร้างไอดีเร็กทอรีใหม่ตั้งแต่หนึ่งไอดีเร็กทอรีขึ้นไป
mvdir	ย้าย (เปลี่ยนชื่อ) ไอดีเร็กทอรี
pwd	แสดงชื่อพารของไอดีเร็กทอรีทำงาน
rmdir	ลบไอดีเร็กทอรี

ผู้จัดการเวิร์กโหลด

Workload Manager (WLM) ออบแบบมาเพื่อให้ ผู้ดูแลระบบสามารถควบคุมวิธีการที่ตัวจัดตารางเวลา virtual memory manager (VMM) และระบบย่อยดิสก์ I/O จัดสรรรีซอร์สให้กับโปรเซสได้มากขึ้น

คุณสามารถใช้ WLM เพื่อป้องกันไม่ให้คลาสอื่นของงานรบกวนซึ่งกันและกัน และเพื่อจัดสรรรีซอร์สตามข้อมูลความต้องการของกลุ่มผู้ใช้ที่แตกต่างกัน

ข้อควรสนใจ: การใช้ที่มีประสิทธิภาพของ WLM จำเป็นต้องมีความรู้อย่างกว้างขวางเกี่ยวกับโปรเซสระบบที่มีอยู่และประสิทธิภาพ หากผู้ดูแลระบบกำหนดคอนฟิก WLM ด้วยค่าที่มากเกินไปหรือค่าที่ไม่ถูกต้อง อาจทำให้ประสิทธิภาพระบบลดลงอย่างมาก

โดยหลักแล้ว WLM มีไว้สำหรับใช้กับระบบขนาดใหญ่ ระบบขนาดใหญ่มักใช้สำหรับการรวมเซิร์ฟเวอร์ ซึ่งเวิร์กโหลดจากระบบเซิร์ฟเวอร์ที่แตกต่างกัน จำนวนมาก (เช่น เครื่องพิมพ์ ฐานข้อมูล ผู้ใช้ทั่วไป และระบบการประมวลผล ธุรกิจ) จะถูกรวมเข้าเป็นระบบขนาดใหญ่ระบบเดียวเพื่อลดต้นทุนในการบำรุง รักษา ระบบ เวิร์กโหลดเหล่านี้มักรบกวนซึ่งกันและกัน และมีเป้าหมายและข้อตกลงเซอร์วิสแตกต่างกัน

WLM ยังนำเสนอการแยกระหว่างชุมชนผู้ใช้ที่มีลักษณะการทำงานของระบบ แตกต่างกันอย่างมากด้วย ซึ่งสามารถป้องกันความขัดแย้งของเวิร์กโหลดที่มีลักษณะการทำงาน บางอย่าง (ตัวอย่างเช่น งานแบบโต้ตอบหรือที่ใช้ CPU ต่ำ) เนื่องจากเวิร์กโหลดที่มีลักษณะการทำงานอื่น (ตัวอย่างเช่น งานแบตช์หรือที่ใช้หน่วยความจำสูง) ได้อย่างมีประสิทธิภาพ

นอกจากนี้ WLM รวมอยู่ในระบบย่อยการบัญชีเพื่อให้ผู้ใช้สามารถจัดทำบัญชี การใช้รีซอร์สต่อคลาส WLM เพิ่มเติมจากการบัญชีมาตรฐานต่อ ผู้ใช้หรือกลุ่ม

หลักการที่เกี่ยวข้อง:

“การทำแอคเคาต์ระบบ” ในหน้า 174

ยูทิลิตี้การทำแอคเคาต์ระบบอนุญาตให้คุณรวบรวมและรายงาน ในการใช้งานแต่ละคน หรือกลุ่มของรีซอร์สระบบต่างๆ

แนวคิดการจัดการเวิร์กโหลด

ด้วย WLM คุณสามารถสร้างคลาสที่แตกต่างของเซิร์ฟเวอร์สำหรับงาน ตลอดจนระบุแอตทริบิวต์สำหรับคลาสเหล่านั้น

แอตทริบิวต์เหล่านี้ระบุจำนวนต่ำสุดและสูงสุดของ CPU, หน่วยความจำฟิสิคัล และปริมาณงานดิสก์ I/O ที่จะจัดสรรให้กับคลาส จากนั้น WLM กำหนดงานให้กับคลาสโดยอัตโนมัติโดยใช้กฎการกำหนดคลาสที่ระบุ โดยผู้ดูแลระบบ กฎการกำหนดเหล่านี้ขึ้นอยู่กับค่าของชุดของ แอตทริบิวต์สำหรับโปรเซส ผู้ดูแลระบบหรือผู้ใช้ที่มีเอกสิทธิ์ยังสามารถกำหนดงานให้กับคลาสด้วยตนเอง ซึ่งยกเลิกการกำหนดอัตโนมัติได้ด้วย

หลักการที่เกี่ยวข้อง:

“การจัดการ Workload Manager” ในหน้า 533

Workload Manager (WLM) ช่วยให้ผู้ดูแลระบบสามารถควบคุมวิธีการที่ตัวจัดตารางเวลาและ virtual memory manager (VMM) จัดสรรรีซอร์สให้กับโปรเซสได้มากขึ้น คุณสามารถใช้ WLM เพื่อป้องกันไม่ให้คลาสอื่นของงานรบกวนซึ่งกันและกัน และเพื่อจัดสรรรีซอร์สตามข้อมูลความต้องการของกลุ่มผู้ใช้ที่แตกต่างกัน

คำศัพท์สำหรับการจัดการเวิร์กโหลด:

คำศัพท์ทั่วไปที่เชื่อมโยงกับการจัดการเวิร์กโหลดมีการแสดงรายการและ อธิบายในตารางนี้

ไอเท็ม	คำอธิบาย
คลาส	<i>คลาส</i> คือคอลเล็กชันของโปรเซสและเรดที่เชื่อมโยง คลาสมีการจำกัดรีซอร์สหนึ่งชุดและส่วนแบ่งใช้เป้าหมาย <i>คลาส</i> ใช้เพื่ออธิบายทั้งคลาสย่อยและซูเปอร์คลาส
ซูเปอร์คลาส	<i>ซูเปอร์คลาส</i> คือคลาสที่มีคลาสย่อยเชื่อมโยง ด้วย โปรเซสไม่สามารถเป็นสมาชิกของซูเปอร์คลาสโดยไม่ได้เป็นสมาชิกของคลาสย่อย ซูเปอร์คลาสมีชุดของกฎการกำหนดคลาสที่กำหนดโปรเซสที่จะ ถูกกำหนดให้กับซูเปอร์คลาส ซูเปอร์คลาสนี้มีชุดของค่าการจำกัดรีซอร์ส และส่วนแบ่งใช้รีซอร์สเป้าหมายที่กำหนดจำนวนรีซอร์ส ซึ่งสามารถใช้ได้โดยโปรเซสซึ่งเป็นสมาชิกของซูเปอร์คลาสรีซอร์สเหล่านี้ ถูกแบ่งใช้ระหว่างคลาสย่อยตามข้อมูลค่าการจำกัดรีซอร์ส และส่วนแบ่งใช้รีซอร์สเป้าหมายของคลาสย่อย
คลาสย่อย	<i>คลาสย่อย</i> คือคลาสที่เชื่อมโยงกับหนึ่งซูเปอร์คลาสเท่านั้น ทุกโปรเซสในคลาสย่อยเป็นสมาชิกของซูเปอร์คลาสด้วย คลาสย่อย มีสิทธิ์เข้าถึงรีซอร์สที่มีอยู่สำหรับซูเปอร์คลาสเท่านั้น คลาสย่อย มีชุดของกฎการกำหนดคลาสที่กำหนดว่า โปรเซสใดที่กำหนดให้กับซูเปอร์คลาส จะเป็นสมาชิกของคลาสย่อย คลาสย่อยยังมีชุดของค่าการจำกัดรีซอร์ส และส่วนแบ่งใช้รีซอร์สเป้าหมายที่กำหนดจำนวนรีซอร์ส ซึ่งสามารถใช้ได้โดยโปรเซสในคลาสย่อย ค่าการจำกัดรีซอร์ส และส่วนแบ่งใช้รีซอร์สเป้าหมายเหล่านี้บ่งชี้จำนวนของรีซอร์สที่มีอยู่ บนซูเปอร์คลาส (เป้าหมายสำหรับซูเปอร์คลาส) ที่สามารถใช้ได้โดยโปรเซส ในคลาสย่อย การจัดการ WLM สามารถทำได้โดยใช้ SMIT หรืออินเตอร์เฟซบรรทัดคำสั่ง WLM
กลไกการจัดประเภท	<i>กลไกการจัดประเภท</i> คือชุดของกฎการกำหนดคลาสที่กำหนดว่า จะกำหนดโปรเซสใดให้กับคลาสใด (ซูเปอร์คลาส หรือคลาสย่อยภายในซูเปอร์คลาส)
กฎการกำหนดคลาส	<i>กฎการกำหนดคลาส</i> บ่งชี้ค่าภายในชุดของแอตทริบิวต์โปรเซส ที่ส่งผลให้โปรเซสถูกกำหนดให้กับคลาสเฉพาะ (ซูเปอร์คลาสหรือคลาสย่อยภายในซูเปอร์คลาส)
ค่าแอตทริบิวต์โปรเซส	<i>ค่าแอตทริบิวต์โปรเซส</i> คือค่าที่โปรเซสมีสำหรับ แอตทริบิวต์โปรเซส แอตทริบิวต์โปรเซสสามารถรวมถึงแอตทริบิวต์ เช่น ID ผู้ใช้, ID กลุ่ม, และชื่อพารามิเตอร์
ค่าการจำกัดรีซอร์ส	<i>ค่าการจำกัดรีซอร์ส</i> คือชุดของค่าที่ WLM เก็บรักษาไว้ เพื่อใช้เป็นชุดของค่าการรีซอร์สขีดจำกัดเหล่านี้เป็นอิสระจากขีดจำกัดรีซอร์สที่ระบุ ด้วยรูทีนย่อย <code>setrlimit</code> โดยสมบูรณ์
ส่วนแบ่งใช้รีซอร์สเป้าหมาย	<i>ส่วนแบ่งใช้รีซอร์สเป้าหมาย</i> คือส่วนแบ่งใช้ของรีซอร์สที่มีอยู่ บนคลาส (คลาสย่อยหรือซูเปอร์คลาส) มีการใช้ส่วนแบ่งใช้เหล่านี้พร้อมกับ ส่วนแบ่งใช้คลาสอื่น (คลาสย่อยหรือซูเปอร์คลาส) ที่ระดับเดียวกันเพื่อกำหนดการแจกจ่ายรีซอร์สที่ต้องการระหว่างคลาสที่ ระดับนั้น

ไอเท็ม	คำอธิบาย
ค่าการใช้รีซอร์ส	<i>ค่าการใช้รีซอร์ส</i> คือจำนวนรีซอร์สที่โปรเซส หรือชุดของโปรเซสกำลังใช้อยู่ในปัจจุบันในระบบ โดยจะเป็นหนึ่งโปรเซส หรือชุดของโปรเซสขึ้นอยู่กับขอบเขตของการรวบรวม รีซอร์สของโปรเซส
ขอบเขตของการรวบรวมรีซอร์ส	<i>ขอบเขตของการรวบรวมรีซอร์ส</i> คือระดับซึ่งรวบรวมการใช้รีซอร์ส และระดับซึ่งนำค่าการจำกัดรีซอร์ส ไปใช้ ขอบเขตนี้อาจเป็นระดับของแต่ละโปรเซสในคลาส ระดับของ ผลรวมของทุกโปรเซสในคลาสที่เป็นของผู้ใช้แต่ละราย หรือระดับของ ผลรวมของทุกโปรเซสในคลาส ขอบเขตเดียวที่สนับสนุนในปัจจุบัน คือขอบเขตคลาส
คุณสมบัติคลาสโปรเซส	<i>คุณสมบัติคลาสโปรเซส</i> คือชุดของคุณสมบัติที่กำหนดให้กับ โปรเซสตามข้อมูลคลาส (คลาสย่อยหรือซูเปอร์คลาส) ซึ่งโปรเซส ถูกกำหนดให้
การอนุญาตในคลาส	<i>การอนุญาตในคลาส</i> คือชุดของกฎที่บ่งชี้ผู้ใช้และกลุ่ม ที่ได้รับอนุญาตให้ทำการดำเนินงานบนคลาสหรือโปรเซส และเธรดในคลาส ค่านี้รวมถึงการอนุญาตในการกำหนดโปรเซสให้กับคลาส ด้วยตนเอง หรือการสร้างคลาสย่อยของซูเปอร์คลาส
ระดับคลาส	<i>ค่า ระดับคลาส</i> คือตำแหน่งของคลาสภายในลำดับชั้น ของการจำกัดรีซอร์สที่ต้องการสำหรับคลาสทั้งหมด ทุกคลาสในระดับจะใช้รีซอร์สตามขีดจำกัด (รวมถึงเป้าหมายรีซอร์ส) ที่ต้องการก่อน จะนำเสนอรีซอร์สส่วนที่เหลือให้กับคลาสในระดับต่ำกว่า ระดับสามารถระบุได้ ทั้งที่ระดับซูเปอร์คลาสและคลาสย่อย รีซอร์สมีการนำเสนอให้กับซูเปอร์คลาส ตามข้อมูลระดับ ภายในซูเปอร์คลาส รีซอร์สจะถูกมอบให้กับคลาสย่อย ตามค่าระดับภายในซูเปอร์คลาส ดังนั้น ระดับซูเปอร์คลาสจึงเป็น ตัวสร้างความแตกต่างที่สำคัญในการแจกจ่ายรีซอร์ส และระดับคลาสย่อยเป็น ตัวสร้างความแตกต่างเล็กน้อยเพิ่มเติมภายในซูเปอร์คลาส

คลาสสำหรับการจัดการเวิร์กโหลด:

WLM อนุญาตให้ผู้ดูแลระบบกำหนดคลาส และกำหนดชุดของแอตทริบิวต์ และขีดจำกัดรีซอร์สสำหรับแต่ละคลาส

โปรเซสถูกกำหนดให้กับคลาสตามเกณฑ์ที่ระบุโดยผู้ดูแลระบบ สิทธิในรีซอร์สและขีดจำกัดมีการบังคับใช้ที่ ระดับคลาส เมธอดนี้ในการกำหนดคลาสของเซอร์วิสและการควบคุม การใช้รีซอร์สของแต่ละคลาสของแอ็พพลิเคชัน ช่วยป้องกันแอ็พพลิเคชัน ที่มีรูปแบบการใช้รีซอร์สแตกต่างกันมากไม่ให้รบกวนซึ่งกันและกัน เมื่อแบ่งใช้เซิร์ฟเวอร์เดียว

WLM สนับสนุนลำดับชั้นของคลาสที่มีสองระดับ:

- รีซอร์สของระบบมีการแจกจ่ายระหว่างซูเปอร์คลาสตาม สิทธิในรีซอร์สของแต่ละซูเปอร์คลาส ผู้ดูแลระบบเป็นผู้ กำหนด สิทธิในรีซอร์ส
- แต่ละซูเปอร์คลาสสามารถมีคลาสย่อย รีซอร์สที่จัดสรรให้กับ ซูเปอร์คลาสมีการแจกจ่ายระหว่างคลาสย่อยตามสิทธิในรีซอร์ส ที่กำหนดให้กับแต่ละคลาสย่อย
- ผู้ดูแลระบบสามารถมอบหมายหน้าที่การจัดการคลาสย่อยของ แต่ละซูเปอร์คลาสให้กับ *ผู้ดูแลซูเปอร์คลาส* หรือให้กับกลุ่มผู้ดูแล ซูเปอร์คลาส
- WLM supports up to 69 superclasses (64 user-defined) and 64 subclasses per superclass (61 user-defined).
- ขึ้นอยู่กับความต้องการขององค์กร ผู้ดูแลระบบสามารถ เลือกที่จะใช้เฉพาะซูเปอร์คลาส หรือใช้ทั้งซูเปอร์คลาสและคลาสย่อย

หมายเหตุ: ตลอดคำอธิบายเกี่ยวกับ WLM นี้ คำว่า *คลาส* ใ้กับทั้ง ซูเปอร์คลาสและคลาสย่อย หากคำอธิบายใ้กับชนิดคลา สอย่างใดอย่างหนึ่งเท่านั้น จะมีการระบุชนิดนั้นอย่างชัดเจน

การกำหนดโปรเซสให้กับคลาสสำหรับการจัดการเวิร์กโหลด:

โปรเซสถูกกำหนดให้กับคลาส โดยใช้กฎการกำหนดคลาสที่ระบุโดยผู้ดูแลระบบ เกณฑ์การจัดประเภทพิจารณาจาก ค่าของชุดแอตทริบิวต์ของโปรเซส เช่น ID ผู้ใช้, ID กลุ่ม ชื่อของไฟล์แอ็พพลิเคชัน ชนิดของโปรเซส และแอ็พพลิเคชันแท็ก

ใช้ชุดของกฎที่กำหนดไว้เพื่อกำหนดซูเปอร์คลาสที่จะกำหนดโปรเซสให้ หากซูเปอร์คลาสนี้มีการกำหนดคลาสย่อย จะมีกฎอีกชุดหนึ่งสำหรับซูเปอร์คลาสนี้เพื่อกำหนดคลาสย่อยที่จะกำหนดโปรเซสให้ โปรเซสการกำหนดออตโนมิตินี้ยังพิจารณาถึงแอตทริบิวต์ **inheritance** ของทั้งซูเปอร์คลาสและคลาสย่อยด้วย

การกำหนดคลาสออตโนมิตินี้ทำเมื่อโปรเซสเรียกรูทีนย่อย **exec** การกำหนดคลาสจะถูกประเมินใหม่เมื่อโปรเซสใช้รูทีนย่อยที่สามารถเปลี่ยนแอตทริบิวต์โปรเซสซึ่งใช้สำหรับวัตถุประสงค์การจัดประเภท ตัวอย่างคือรูทีนย่อย **setuid, setgid, setpri,** และ **plock**

นอกเหนือจากการกำหนดคลาสออตโนมิตินี้แล้ว ผู้ใช้ที่มีสิทธิเหมาะสม สามารถกำหนดโปรเซสหรือกลุ่มของโปรเซสให้กับซูเปอร์คลาสหรือคลาสย่อย เฉพาะด้วยตนเองได้ด้วย

หลักการที่เกี่ยวข้อง:

“แอตทริบิวต์คลาส” ในหน้า 548

แสดงรายการแอตทริบิวต์ทั้งหมดของคลาส WLM

การควบคุมรีซอร์ส:

WLM อนุญาตการจัดการรีซอร์สในสองวิธีคือ เป็นเปอร์เซ็นต์ ของรีซอร์สที่มีอยู่ หรือเป็นการใช้รีซอร์สทั้งหมด

รีซอร์สที่สามารถควบคุมในแบบเปอร์เซ็นต์มีดังต่อไปนี้:

- การใช้ตัวประมวลผลของเธรดชนิด SCHED_OTHER ในคลาส นี้เป็น ผลรวมของวงจรตัวประมวลผลทั้งหมดที่ใช้โดยทุกเธรดในคลาส เธรดลำดับความสำคัญที่ไม่สามารถปรับได้ ดังนั้น จึงไม่สามารถเปลี่ยนได้ และเธรดประเภทนี้สามารถใช้รีซอร์สเกินกว่าเป้าหมายการใช้ตัวประมวลผลได้
- การใช้หน่วยความจำฟิสิคัลของโปรเซสในคลาส นี้เป็นผลรวม ของหน่วยความจำทั้งหมดที่เป็นสมาชิกของโปรเซสในคลาส
- ดิสก์ I/O แบนด์วิธของคลาส นี้เป็นแบนด์วิธ (ในบล็อก 512-ไบต์ ต่อวินาที) ของ I/Os ทั้งหมดที่เริ่มต้นโดยเธรดในคลาสบนอุปกรณ์ดิสก์แต่ละรายการ ที่เข้าถึงโดยคลาส

รีซอร์สที่สามารถควบคุมในแบบการใช้ทั้งหมดจัดอยู่ในหมวดหมู่อย่างใดอย่างหนึ่ง จากสองหมวดหมู่นี้: ผลรวมคลาสหรือผลรวมโปรเซส หมวดหมู่ผลรวมคลาส รวมถึง:

จำนวนโปรเซสในคลาส

นี่เป็นจำนวนของโปรเซสที่ใช้งานอยู่ในคลาส ณ เวลาหนึ่ง

จำนวนเธรดในคลาส

นี่เป็นจำนวนของเธรดที่ใช้งานอยู่ในคลาส ณ เวลาหนึ่ง

จำนวนบล็อกอินในคลาส

นี่เป็นจำนวนของบล็อกอินเซสชันที่ใช้งานอยู่ในคลาส ณ เวลาหนึ่ง

หมวดหมู่ผลรวมโปรเซสรวมถึง:

เวลา CPU ทั้งหมด

นี่เป็นเวลา CPU สะสมทั้งหมดสำหรับโปรเซสเดียว

ดิสก์ I/O ทั้งหมด

นี่เป็นบล็อกสะสมทั้งหมดของดิสก์ I/O สำหรับโปรเซสเดียว

เวลาเชื่อมต่อทั้งหมด

นี่เป็นเวลาทั้งหมดที่ล็อกอินเซสชันสามารถใช้งานอยู่ได้

สิทธิในรีซอร์ส:

WLM อนุญาตให้ผู้ดูแลระบบระบุสิทธิในรีซอร์สต่อคลาส ได้โดยอิสระสำหรับรีซอร์สแต่ละชนิด

สิทธิเหล่านี้สามารถระบุโดยการบ่งชี้ต่อไปนี้:

- เป้าหมายสำหรับการใช้งานของรีซอร์สชนิดต่างๆ เป้าหมายนี้มีการระบุเป็น ส่วนแบ่งใช้ ส่วนแบ่งใช้มีการระบุเป็นจำนวนสัมพันธ์ของการใช้งานระหว่าง คลาสต่างๆ ตัวอย่างเช่น ถ้าสองคลาสมีส่วนแบ่งใช้ของ CPU เป็น 1 และ 3 ตามลำดับ และเป็นเพียงสองคลาสที่ใช้งานอยู่ ณ เวลานั้น เป้าหมายเปอร์เซ็นต์ของ คลาสที่ใช้โดย WLM สำหรับการควบคุม CPU จะเป็น 25% และ 75% ตามลำดับ เปอร์เซ็นต์เป้าหมายมีการคำนวณสำหรับคลาสต่างๆ ในแต่ละระดับตามข้อมูล จำนวนของส่วนแบ่งใช้ที่ใช้งานอยู่ในระดับและจำนวนของรีซอร์สที่มีอยู่ใน ระดับ
- ขีดจำกัดต่ำสุดและสูงสุด ขีดจำกัดเหล่านี้มีการระบุเป็นเปอร์เซ็นต์ ของรีซอร์สทั้งหมดที่มีอยู่ WLM สนับสนุนขีดจำกัดสูงสุดสองประเภทคือ:
 - ขีดจำกัดสูงสุดซอฟต์แวร์บ่งชี้จำนวนสูงสุดของรีซอร์สที่สามารถทำให้มีอยู่ เมื่อมีการช่วงชิงรีซอร์ส ค่าสูงสุดนี้อาจ เกินได้ ถ้าไม่มีการช่วงชิง นั่นคือถ้าไม่มีส่วนอื่นต้องการ รีซอร์สนั้น
 - ขีดจำกัดสูงสุดฮาร์ดบ่งชี้จำนวนสูงสุดของรีซอร์สที่สามารถทำให้มีอยู่ ไม่ว่าจะมีการช่วงชิงรีซอร์สหรือไม่ก็ตาม อย่างไรก็ตาม เเรดที่มีลำดับความสำคัญคงที่ไม่อยู่ภายใต้กฎเหล่านี้ ดังนั้นจึง สามารถใช้รีซอร์สเกินขีดจำกัดได้
- ขีดจำกัดทั้งหมดขีดจำกัดทั้งหมดมีการบังคับใช้อย่างเข้มงวด ถ้าโปรเซสใช้รีซอร์สเกิน ขีดจำกัดปริมาณการใช้ทั้งหมด อย่างไรก็ตามอย่างหนึ่ง โปรเซสจะถูกยุติ ถ้าคลาสเกิน ขีดจำกัดทั้งหมดอย่างใดอย่างหนึ่ง การดำเนินงานใดๆ ที่อาจส่งผลให้เกิดการสร้าง อีกรีนสแตนท์หนึ่งของรีซอร์สนั้นจะล้มเหลว

ในกรณีส่วนใหญ่ ขีดจำกัดสูงสุดซอฟต์แวร์เพียงพอที่จะทำให้มั่นใจว่ามีการปฏิบัติตาม และบังคับใช้สิทธิในรีซอร์ส การใช้ขีดจำกัดสูงสุดฮาร์ดอาจส่งผลให้เกิด รีซอร์สระบบที่ไม่ได้ใช้งาน เนื่องจากมีการบังคับใช้อย่างเข้มงวดแม้ในเวลาที่ไม่มีการช่วงชิงรีซอร์ส ดังนั้น ต้องพิจารณาอย่างรอบคอบเมื่อใช้ ขีดจำกัดสูงสุดฮาร์ด เนื่องจากอาจส่งผลกระทบต่อประสิทธิภาพของระบบ หรือ แอ็พพลิเคชันเป็นอย่างมากถ้าตั้งค่าต่ำเกินไป ขีดจำกัดทั้งหมดควรใช้ด้วยความระมัดระวังเช่นเดียวกัน เนื่องจากอาจส่ง ผลให้เกิดการยุติโปรเซสหรือความล้มเหลวของฟังก์ชัน ที่ต้องการ

ในโหมด active WLM พยายามทำให้คลาสที่ใช้งานอยู่ใกล้เคียงกับเป้าหมาย เนื่องจากมีข้อจำกัดสองสามข้อเกี่ยวกับค่าของขีด จำกัดต่างๆ ดังนั้นผลรวม ของขีดจำกัดบนคลาสทั้งหมดจึงอาจเกินกว่า 100% ในกรณีนี้ ถ้าคลาสทั้งหมดใช้งานอยู่ คลาสทั้งหมดไม่สามารถใช้งานถึงขีดจำกัด WLM ควบคุมปริมาณการใช้ตัวประมวลผลโดยการปรับลำดับความสำคัญที่จัดตาราง เวลาของเเรดที่มีลำดับความสำคัญไม่คงที่ในระบบตามวิธีการของคลาสซึ่ง เเรดเป็นสมาชิก โดยสัมพันธ์กับขีดจำกัดและเป้าหมาย แนวทางนี้ รับประกันปริมาณการใช้ตัวประมวลผลเฉลี่ยในรอบระยะเวลาที่กำหนด แทน ปริมาณการใช้ตัวประมวลผลในช่วงเวลาที่สั้นมากๆ (ตัวอย่างเช่น 10 ms)

ตัวอย่างเช่น ถ้าคลาส A เป็นเพียงคลาสเดียวที่ใช้งานอยู่โดยมีค่าต่ำสุด ตัวประมวลผลเป็น 0% และเป้าหมายตัวประมวลผล เป็น 60 ส่วนแบ่งใช้ คลาส A จะได้รับตัวประมวลผล 100% หากคลาส B ที่มีขีดจำกัดต่ำสุดตัวประมวลผลเป็น 0% และเป้าหมายตัวประมวลผล เป็น 40 ส่วนแบ่งใช้ เริ่มทำงาน ผลคือการใช้ตัวประมวลผลของคลาส A จะลดลง อย่างมากเป็น 60% และ การใช้ตัวประมวลผลของคลาส B จะเพิ่มขึ้นจาก 0% เป็น 40% ระบบรักษาสมดุลการใช้ตัวประมวลผลอยู่ที่ 60% และ 40% ตามลำดับ เสมอ

ตัวอย่างนี้สมมติว่าไม่มีการช่วงชิงหน่วยความจำระหว่างคลาส ภายใต้สภาพการทำงานปกติขีดจำกัดที่คุณตั้งค่าสำหรับตัวประมวลผลและหน่วยความจำเป็นอิสระจากกัน ตัวอย่างเช่น คลาสอาจไม่สามารถใช้งานถึงเป้าหมาย หรือแม้แต่การจัดสรรตัวประมวลผลต่ำสุดถ้าขีดจำกัดสูงสุดในการใช้หน่วยความจำต่ำเกินไปเมื่อเปรียบเทียบกับชุดการทำงานของคลาสนั้น

เพื่อช่วยกำหนดนิยามคลาสและขีดจำกัดคลาสให้ละเอียดมากขึ้นสำหรับชุดของ แอ็พพลิเคชันที่กำหนด WLM จึงจัดเตรียมเครื่องมือการรายงาน `wlmstat` ซึ่งแสดงจำนวนรีซอร์สที่ใช้อยู่ในปัจจุบันโดยแต่ละคลาส และยังสามารถนำเสนอเครื่องมือการแสดงผลแบบกราฟิกที่ชื่อ `wlmmon` สำหรับการมอนิเตอร์ระบบอีกด้วย

ขีดจำกัดหน่วยความจำเสมือนของ Workload Manager:

ขีดจำกัดหน่วยความจำเสมือนของ Workload Manager (WLM) เป็น เครื่องมือของผู้ดูแลระบบในการป้องกันการด้อยลงหรือความล้มเหลวของระบบ เนื่องจากการเลื่อนหน้ามากเกินไป โดยวิธีการระบุขีดจำกัดหน่วยความจำเสมือนบนคลาสหรือโปรเซส

เมื่อเกินขีดจำกัด WLM จะดำเนินการโดยทำอย่างใดอย่างหนึ่งต่อไปนี้:

- การทำลายโปรเซสทั้งหมดภายใต้คลาส WLM ที่เกินขีดจำกัด
- การทำลายเฉพาะโปรเซสที่ทำให้การใช้ของคลาส WLM เกิน ขีดจำกัด
- การทำลายโปรเซสที่เกินขีดจำกัดโปรเซส

สามารถระบุขีดจำกัดหน่วยความจำเสมือนสำหรับคลาสที่ผู้ใช้กำหนดใดๆ คลาสย่อยดีพอลต์ใดๆ ภายใต้ซูเปอร์คลาสที่ผู้ใช้กำหนด และซูเปอร์คลาสดีพอลต์

สำหรับวัตถุประสงค์ด้านการบัญชี WLM จะพิจารณาเฉพาะส่วนต่อไปนี้ เป็น หน่วยความจำเสมือนเมื่อกำหนดการใช้ของคลาสหรือโปรเซสทั้งหมดสำหรับ WLM:

- ซีป
- ข้อมูลที่เริ่มต้นตัวโหลด, BSS, โลบารีแบบแบ่งใช้, และเซ็กเมนต์ที่โหลดส่วนตัว
- พื้นที่ UBLOCK และ mmap
- หน้าพื้นที่ว่างผู้ใช้ขนาดใหญ่และที่ปัก

ผู้ดูแลระบบสามารถระบุขีดจำกัดหน่วยความจำเสมือน WLM สำหรับคลาสหรือสำหรับแต่ละโปรเซสในคลาส เมื่อใช้งานเกินขีดจำกัดคลาส WLM สามารถทำลายโปรเซสทั้งหมดที่กำหนดให้กับคลาส หรือทำลายเฉพาะโปรเซสที่ทำให้เกิน ขีดจำกัด ทั้งนี้ขึ้นอยู่กับว่าแอ็พพลิเคชันบีบอัดคลาส `vmenforce` มีการ ตั้งค่าเป็น `class` หรือ `proc` ตามลำดับ ลักษณะการทำงานดีพอลต์คือทำลายเฉพาะโปรเซสที่ทำให้เกิน ขีดจำกัด ขีดจำกัดโปรเซสจะถูกทำลายถ้าการใช้หน่วยความจำเสมือนของโปรเซส เเลยขีดจำกัด

โหมดการดำเนินงานสำหรับ Workload Manager:

WLM สามารถใช้เพื่อควบคุมปริมาณการใช้รีซอร์สเป็นเปอร์เซ็นต์ต่อคลาส ผลรวมต่อคลาส หรือผลรวมต่อโปรเซส การควบคุมรีซอร์สทุกชนิด สามารถเปิดใช้งานได้โดยรัน WLM ในโหมด active

หรือคุณสามารถเริ่มต้นโหมดของ WLM ที่จัดประเภทโปรเซสใหม่และที่มีอยู่ และมอนิเตอร์การใช้รีซอร์สของคลาสต่างๆ โดยไม่พยายาม ควบคุมการใช้งานนี้ โหมดนี้เรียกว่า โหมด *passive*

สามารถใช้โหมด passive เมื่อกำหนดคอนฟิก WLM บนระบบใหม่เพื่อตรวจสอบการจำกัดประเภทและกฎการกำหนด และเพื่อสร้างบรรทัดฐานของการใช้รีซอร์สสำหรับคลาสต่างๆ เมื่อ WLM ไม่ได้ควบคุมการจัดการตัวประมวลผลและหน่วยความจำ ซึ่งควรให้ข้อมูลพื้นฐานสำหรับผู้ดูแลระบบ ในการตัดสินใจเกี่ยวกับวิธีการใช้ส่วนแบ่งใช้รีซอร์สและขีดจำกัดรีซอร์ส (ถ้าต้องการ) เพื่อสนับสนุนแอปพลิเคชันที่สำคัญและจำกัดงานที่สำคัญน้อยกว่าเพื่อให้บรรลุผลตามเป้าหมายทางธุรกิจ

หากเวลาตัวประมวลผลเป็นรีซอร์สเพียงรายการเดียวที่คุณสนใจจะควบคุม คุณสามารถเลือกที่จะรัน WLM ในโหมด active สำหรับตัวประมวลผลและโหมด passive สำหรับรีซอร์สอื่นทั้งหมด โหมดนี้เรียกว่าโหมด *cpu อย่างเดียว* ถ้าคุณต้องการ ควบคุมเปอร์เซ็นต์ต่อคลาส แต่ไม่ต้องการควบคุมชนิดผลรวมรีซอร์สทั้งสองชนิด การบัญชีรีซอร์สทั้งหมดและการควบคุมสามารถปิดใช้งานสำหรับผลรวมต่อคลาส ผลรวมต่อโปรเซส หรือทั้งสองอย่าง ในทุกโหมด คุณมีตัวเลือก การปิดใช้งานการผูกชุดรีซอร์ส

การควบคุมไดนามิกของ Workload Manager:

เมื่อ WLM ใช้งานอยู่ สามารถแก้ไขพารามิเตอร์ใดๆ ของคอนฟิกูเรชัน ปัจจุบันได้ตลอดเวลา รวมถึงแอตทริบิวต์ของคลาส ส่วนแบ่งใช้และขีดจำกัด รีซอร์ส กฎการกำหนด และการเพิ่มคลาสใหม่ หรือการลบ คลาสที่มีอยู่

ซึ่งสามารถทำได้หลายวิธี เช่น:

- การแก้ไขไฟล์คุณสมบัติของคอนฟิกูเรชันที่ใช้งานอยู่ในปัจจุบัน (ไดเรกทอรี ที่ชี้โดยลิงก์สัญลักษณ์ /etc/wlm/current) และการรีเฟรช WLM โดยใช้คำสั่ง `wlmcntrl` เพื่อใช้ พารามิเตอร์ใหม่
- การสร้างอีกคอนฟิกูเรชันหนึ่งที่มีชุดของพารามิเตอร์ที่แตกต่างอื่น และการอัปเดต WLM เพื่อโหลดพารามิเตอร์ของคอนฟิกูเรชันใหม่ และทำให้เป็น คอนฟิกูเรชันปัจจุบัน
- การแก้ไขพารามิเตอร์บางตัวของคอนฟิกูเรชันที่ใช้งานอยู่ในปัจจุบัน โดยใช้อินเตอร์เฟซบรรทัดคำสั่ง WLM (คำสั่ง `mkclass`, `chclass`, และ `rmclass`)
- การแก้ไขพารามิเตอร์บางตัวของคอนฟิกูเรชันที่ใช้งานอยู่ในปัจจุบัน จากแอปพลิเคชันโดยใช้ WLM APIs

การสลับโดยอัตโนมัติไปยังคอนฟิกูเรชันใหม่ในเวลาที่เหมาะสมสามารถทำได้ โดยใช้ชุดคอนฟิกูเรชัน ชุดคอนฟิกูเรชันช่วยให้ ผู้ดูแลระบบสามารถระบุชุดของคอนฟิกูเรชันที่จะใช้ และช่วงเวลา ซึ่งจะใช้แต่ละชุด

เครื่องมือการมอนิเตอร์:

ใช้คำสั่ง WLM เหล่านี้เพื่อแสดงสถิติ WLM และมอนิเตอร์ การดำเนินงานของ WLM

- คำสั่ง `wlmstat` มุ่งเน้นข้อความและแสดงสถิติ เป็นข้อความ (เปอร์เซ็นต์ของการใช้รีซอร์สต่อคลาสสำหรับรีซอร์สทุกชนิด ที่จัดการโดย WLM)
- คำสั่ง `wlmmmon` ให้มุมมองแบบกราฟิกของการใช้รีซอร์ส ต่อคลาสและการควบคุม WLM
- คำสั่ง `wlmpperf` เป็นเครื่องมือทางเลือกที่มีอยู่ใน กล้องเครื่องมือประสิทธิภาพและนำเสนอความสามารถเพิ่มเติม เช่น เรียก คอร์ด ระยะยาวและการเล่นซ้ำ

ความเข้ากันได้ย้อนหลังใน Workload Manager:

ดีฟอลต์เอาต์พุตของคำสั่ง `wlmstat` แสดงรายการเฉพาะซูเปอร์คลาส และคล้ายกับรายการของเวอร์ชันก่อนหน้านี้

ตัวอย่างเช่น:

```
# wlmstat
      CLASS CPU MEM DKIO
Unclassified 0 0 0
  Unmanaged 0 0 0
    Default 0 0 0
    Shared 0 2 0
    System 2 12 0
    class1 0 0 0
    class2 0 0 0
#
```

หากบางซูเปอร์คลาสมีคลาสย่อยที่กำหนดโดยผู้ดูแลระบบ WLM จะมีการแสดงรายการคลาสย่อย ตัวอย่างเช่น:

```
# wlmstat
      CLASS CPU MEM DKIO
Unclassified 0 0 0
  Unmanaged 0 0 0
    Default 0 0 0
    Shared 0 2 0
    System 3 11 7
    class1 46 0 0
class1.Default 28 0 0
class1.Shared 0 0 0
  class1.sub1 18 0 0
    class2 48 0 0
#
```

เอาต์พุตเหมือนกับเมื่อคุณใช้คำสั่ง `ps` สำหรับโปรเซสในซูเปอร์คลาสที่ไม่มีคลาสย่อย คำสั่ง `ps` แสดงรายการชื่อซูเปอร์คลาสเป็นชื่อคลาสของโปรเซส

การบัญชีต่อคลาส:

ยูทิลิตี้ระบบการบัญชี AIX ช่วยให้คุณสามารถรวบรวมและรายงานการใช้รีซอร์สระบบหลายชนิด โดยเรียงตามผู้ใช้ กลุ่ม หรือ คลาส WLM

เมื่อเปิดบัญชีกระบวนการ ระบบปฏิบัติการจะบันทึกสถิติ เกี่ยวกับการใช้รีซอร์สของโปรเซสในไฟล์การบัญชีเมื่อมีโปรเซสเร็กคอร์ดบัญชีนี้ มีคีย์ตัวเลข 64 บิต ที่แสดงถึงชื่อของคลาส WLM ที่โปรเซสเป็นสมาชิก

ระบบย่อยการบัญชีใช้คีย์ 64-บิตแทนชื่อคลาสแบบเต็ม 34-อักขระ เพื่อประหยัดพื้นที่ (มีฉะนั้น การเปลี่ยนแปลงจะทำให้ขนาดของเร็กคอร์ดการบัญชี เพิ่มขึ้นจริงเป็นสองเท่า) เมื่อคำสั่งการบัญชีรันเพื่อดึงข้อมูล ต่อโปรเซส คีย์จะถูกแปลกลับเป็นชื่อคลาสโดยใช้รูนที่ระบุข้างบน การแปลงนี้ใช้ชื่อคลาสที่อยู่ในไฟล์คอนฟิกูเรชัน WLM ในปัจจุบัน ดังนั้น ถ้าคลาสถูกลบออกไปแล้วในระหว่างที่เขียน เร็กคอร์ดการบัญชี เมื่อโปรเซสสิ้นสุดลง และในเวลาที่ รันรายงานการบัญชี จะไม่พบชื่อคลาสที่สอดคล้องกับคีย์ และคลาสแสดงขึ้นเป็น Unknown

เมื่อต้องการเก็บเร็กคอร์ดที่ถูกต้องของการใช้รีซอร์สของคลาสที่ลบออกในระหว่าง รอบระยะเวลาบัญชี ให้ทำอย่างใดอย่างหนึ่งต่อไปนี้:

- แทนการลบคลาส ให้เก็บชื่อคลาสไว้ในไฟล์คลาส และลบคลาสออกจากไฟล์กฎเพื่อให้ไม่สามารถกำหนดโปรเซสให้กับคลาสนั้นได้ จากนั้น คุณสามารถลบคลาสหลังจากสร้างรายงานการบัญชีขึ้นแล้ว เมื่อสิ้นสุดรอบระยะเวลาบัญชี
- หรือลบคลาสออกจากคอนฟิกูเรชันที่คลาสเป็นสมาชิกอยู่ และเก็บ ชื่อคลาสไว้ในไฟล์คลาสในคอนฟิกูเรชัน "dummy" (คอนฟิกูเรชันที่ไม่เคยเรียกใช้) จนกว่าสร้างเร็กคอร์ดการบัญชีสำหรับรอบระยะเวลาเสร็จแล้ว

หลักการที่เกี่ยวข้อง:

“การทำแควเคเตอร์ระบบ” ในหน้า 174

ยุติสิทธิ์การทำแควเคเตอร์ระบบอนุญาตให้คุณรวบรวมและรายงาน ในการใช้งานแต่ละคน หรือกลุ่มของรีซอร์สระบบต่างๆ

การจัดการ Workload Manager

Workload Manager (WLM) ช่วยให้ผู้ดูแลระบบสามารถควบคุมวิธีการที่ตัวจัดตารางเวลาและ virtual memory manager (VMM) จัดสรรรีซอร์สให้กับโปรเซสได้มากขึ้น คุณสามารถใช้ WLM เพื่อป้องกันไม่ให้คลาสอื่นของงานรบกวนซึ่งกันและกัน และเพื่อจัดสรรรีซอร์สตามข้อมูลความต้องการของกลุ่มผู้ใช้ที่แตกต่างกัน

WLM ช่วยให้คุณสามารถสร้างคลาสที่แตกต่างของเซอร์วิสสำหรับงาน ตลอดจนระบุแอตทริบิวต์สำหรับคลาสเหล่านั้น แอตทริบิวต์เหล่านี้ระบุจำนวนต่ำสุดและสูงสุดของ CPU, หน่วยความจำฟิสิคัล และปริมาณงานดิสก์ I/O ที่จัดสรรให้กับคลาส จากนั้น WLM กำหนดงานให้กับคลาสโดยอัตโนมัติโดยใช้กฎการกำหนดคลาสที่ระบุ โดยผู้ดูแลระบบ กฎการกำหนดเหล่านี้ขึ้นอยู่กับค่าของชุดของ แอตทริบิวต์สำหรับโปรเซส ผู้ดูแลระบบหรือผู้ใช้ที่มีเอกสิทธิ์ ยังสามารถกำหนดงานให้กับคลาสด้วยตนเอง ซึ่งยกเลิกการกำหนดอัตโนมัติได้ด้วย

WLM เป็นส่วนหนึ่งของระบบปฏิบัติการพื้นฐานและมีการติดตั้งพร้อมกับ ระบบปฏิบัติการพื้นฐาน แต่เป็นเซอร์วิสทางเลือก ต้องกำหนดคอนฟิก WLM ให้เหมาะกับสภาวะแวดล้อมของระบบ ซึ่งเริ่มต้นเมื่อคุณต้องการใช้ และหยุดเมื่อคุณต้องการพักไว้หรือสิ้นสุดเซอร์วิส WLM

ส่วนนี้มีโพรซีเจอร์สำหรับการกำหนดคอนฟิก WLM ด้วยคลาส และกฎที่เหมาะสมกับไซต์ของคุณ และคำแนะนำสำหรับการแก้ไขปัญหา ลักษณะการใช้รีซอร์สที่ไม่ได้คาดไว้

ข้อควรสนใจ: การกึ่งในส่วนนี้สมมติว่าคุณมีความคุ้นเคยกับแนวคิด WLM การใช้ที่มีประสิทธิภาพของ WLM จำเป็นต้องมีความรู้อย่าง กว้างขวางเกี่ยวกับโปรเซสระบบที่มีอยู่และประสิทธิภาพ หากผู้ดูแลระบบกำหนดคอนฟิก WLM ด้วยค่าที่มากเกินไปหรือค่าที่ไม่ถูกต้อง อาจทำให้ประสิทธิภาพระบบ ลดลงอย่างมาก

หลักการที่เกี่ยวข้อง:

“แนวคิดการจัดการเวิร์กโหลด” ในหน้า 526

ด้วย WLM คุณสามารถสร้างคลาสที่แตกต่างของเซอร์วิสสำหรับงาน ตลอดจนระบุแอตทริบิวต์สำหรับคลาสเหล่านั้น

การเปิดและการปิด Workload Manager:

WLM เป็นเซอร์วิสทางเลือกที่ต้องเริ่มต้นและหยุด

ขอแนะนำให้คุณใช้หนึ่งในอินเทอร์เฟซการจัดการระบบ, SMIT, เพื่อเริ่มหรือหยุด WLM

- เมื่อต้องการเริ่มต้นหรือหยุด WLM โดยใช้ SMIT ให้ใช้พารามิเตอร์ smit wlmmanage

ความแตกต่างที่สำคัญระหว่างอ็อปชันเหล่านี้คือความถาวร ใน SMIT, คุณสามารถเริ่มหรือหยุด WLM ได้สามวิธี:

เซสชันปัจจุบัน

หากคุณร้องขอให้หยุด WLM ด้วยอ็อปชันนี้ WLM จะถูกหยุดสำหรับเซสชันนี้ เท่านั้นและรีสตาร์ทเมื่อรีบูตครั้งถัดไป หากคุณร้องขอให้เริ่มต้นด้วยอ็อปชันนี้ WLM จะเริ่มต้นขึ้นสำหรับเซสชันนี้เท่านั้นและไม่รีสตาร์ทเมื่อ รีบูตครั้งถัดไป

รีบูตครั้งถัดไป

หากคุณร้องขอให้หยุด WLM ด้วยอ็อปชันนี้ WLM จะยังคงรันอยู่สำหรับ เซสชันนี้เท่านั้นและ จะไม่รีสตาร์ทเมื่อรีบูตครั้งถัดไป หากคุณร้องขอให้เริ่มต้นด้วยอ็อปชันนี้ WLM จะไม่มีอยู่สำหรับเซสชันนี้ แต่จะเริ่มต้นขึ้นเมื่อรีบูตครั้งถัดไป

ทั้งสองอย่าง

หากคุณร้องขอให้หยุด WLM ด้วยอ็อปชันนี้ WLM จะถูกหยุดสำหรับ เซสชันนี้เท่านั้นและ จะไม่รีสตาร์ทเมื่อรีบูตครั้งถัดไป หากคุณร้องขอให้เริ่มต้นด้วยอ็อปชันนี้ WLM จะเริ่มต้นขึ้นสำหรับเซสชันนี้เท่านั้น และ จะ รีสตาร์ทเมื่อรีบูตครั้งถัดไป

คุณยังสามารถใช้คำสั่ง `wlmcntrl` แต่คำสั่ง `wlmcntrl` อนุญาตให้คุณเริ่มต้นหรือหยุด WLM สำหรับเซสชันปัจจุบันเท่านั้น ถ้าคุณต้องการใช้อินเตอร์เฟซบรรทัดคำสั่งและคุณต้องการให้การเปลี่ยนแปลงยังคงมีผลบังคับใช้เมื่อรีบูตเครื่อง คุณต้องแก้ไขไฟล์ `/etc/inittab`

WLM สามารถใช้เพื่อควบคุมปริมาณการใช้รีซอร์สเป็นเปอร์เซ็นต์ต่อคลาส ผลรวมต่อคลาส หรือผลรวมต่อโปรเซส การควบคุมรีซอร์สทุกชนิด สามารถเปิดใช้งานโดยรัน WLM ในโหมด *active* หรือคุณสามารถ เริ่มต้นโหมดของ WLM ที่จัดประเภทโปรเซสใหม่และที่มีอยู่และมอนิเตอร์ การใช้รีซอร์สของคลาสต่างๆ โดยไม่พยายามควบคุม การใช้งานนี้ โหมดนี้เรียกว่าโหมด *passive* หากเวลา CPU เป็น รีซอร์สเพียงรายการเดียวที่คุณสนใจจะควบคุม คุณสามารถเลือกที่จะรัน WLM ในโหมด *active* สำหรับ CPU และโหมด *passive* สำหรับรีซอร์สอื่นทั้งหมด โหมดนี้ เรียกว่าโหมด *cpu อย่างเดียว*

โปรเซสทั้งหมดที่มีอยู่ในระบบก่อน WLM เริ่มต้นขึ้นจะถูกจัดประเภท ตามกฎการกำหนดที่โหลดเข้าไปใหม่ และถูกมอนิเตอร์โดย WLM

คุณสมบัติ Workload Manager:

คุณสามารถระบุคุณสมบัติสำหรับ WLM โดยใช้ SMIT, อินเตอร์เฟซบรรทัดคำสั่ง WLM หรือโดยการสร้างไฟล์ flat ASCII อินเตอร์เฟซ SMIT ใช้คำสั่ง WLM เพื่อบันทึกข้อมูลในไฟล์ flat ASCII เดียวกันที่เรียกว่า *ไฟล์คุณสมบัติ+*

ชุดของไฟล์คุณสมบัติ WLM กำหนดคอนฟิกูเรชัน WLM คุณสามารถสร้าง ชุดของไฟล์คุณสมบัติได้หลายชุด เพื่อกำหนดคอนฟิกูเรชันที่แตกต่างกัน ของการจัดการเวิร์กโหลด คอนฟิกูเรชันเหล่านี้ตั้งอยู่ในไดเรกทอรีย่อย ของ `/etc/wlm` ไฟล์คุณสมบัติ WLM ที่อธิบายซูเปอร์คลาสของคอนฟิกูเรชัน *Config* ได้แก่ *คลาส*, *คำอธิบาย*, *ขีดจำกัด*, *ส่วนแบ่งใช้* และ *กฎ* ของไฟล์ใน `/etc/wlm/Config` และไฟล์คุณสมบัติที่อธิบายคลาสย่อยของซูเปอร์คลาส *Super* ของ คอนฟิกูเรชันนี้เป็น *คลาส*, *ขีดจำกัด*, *ส่วนแบ่งใช้* และ *กฎ* ของไฟล์ ในไดเรกทอรี `/etc/wlm/Config/Super` เฉพาะผู้ใช้รากสามารถเริ่มต้น หรือหยุด WLM หรือสลับจากคอนฟิกูเรชันหนึ่งไปยังอีกคอนฟิกูเรชันหนึ่งได้

ไฟล์คุณสมบัติมีการตั้งชื่อดังนี้:

ไอเท็ม	คำอธิบาย
คลาส	นิยามคลาส
คำอธิบาย	ข้อความคำอธิบายคอนฟิกูเรชัน
การจัดกลุ่ม	การจัดกลุ่มค่าแอตทริบิวต์
ขีดจำกัด	ขีดจำกัดคลาส
ส่วนแบ่งใช้	ส่วนแบ่งใช้เป้าหมายของคลาส
กฎ	กฎการกำหนดคลาส

คำสั่งในการส่งไฟล์คุณสมบัติ WLM ซึ่งเป็น `wlmcntrl` และคำสั่ง WLM อื่น อนุญาตให้ผู้ใช้ระบุชื่อไดเรกทอรีอื่นสำหรับไฟล์คุณสมบัติ WLM ได้ ซึ่งช่วยให้คุณเปลี่ยนคุณสมบัติ WLM ได้โดยไม่ต้องเปลี่ยนไฟล์คุณสมบัติ WLM ดิฟเฟอเรนซ์

ลิงก์สัญลักษณ์ /etc/wlm/current ซึ่งไปยัง ไดรฟ์ทอริที่มีไฟล์คอนฟิกูเรชันปัจจุบัน อัปเดต ลิงก์นี้ด้วยคำสั่ง `wlmcntrl` เมื่อคุณเริ่มต้น WLM ด้วยคอนฟิกูเรชันหรือชุดคอนฟิกูเรชันที่ระบุ ไฟล์คอนฟิกูเรชัน ตัวอย่างที่จัดส่งมาพร้อมกับระบบปฏิบัติการมีอยู่ใน /etc/wlm/standard

การสร้างการจัดกลุ่มค่าแอตทริบิวต์:

คุณสามารถจัดกลุ่มค่าแอตทริบิวต์และแสดงแทนค่าดังกล่าวด้วยค่าเดียวในไฟล์ `rules` ได้ การจัดกลุ่มค่าแอตทริบิวต์เหล่านี้มีการกำหนดไว้ในไฟล์ `groupings` ภายในไดเรกทอรีคอนฟิกูเรชัน WLM

โดยค่าดีฟอลต์ คอนฟิกูเรชันไม่มีไฟล์ `groupings` ไม่มีคำสั่งหรืออินเทอร์เฟซการจัดการในการสร้างไฟล์นี้ เมื่อต้องการสร้างและใช้ การจัดกลุ่มค่าแอตทริบิวต์ให้ใช้ไพรซีเดนต์ต่อไปนี้:

1. ด้วยสิทธิการใช้งานราก ให้เปลี่ยนเป็นไดเรกทอรีคอนฟิกูเรชันที่เหมาะสม ดังแสดงในตัวอย่างต่อไปนี้:

```
cd /etc/wlm/MyConfig
```

2. ใช้โปรแกรมแก้ไขที่โปรดปรานของคุณเพื่อสร้างและแก้ไขไฟล์ที่ชื่อ `groupings` ตัวอย่างเช่น:

```
vi groupings
```

3. กำหนดแอตทริบิวต์และค่าที่เชื่อมโยงโดยใช้ รูปแบบต่อไปนี้:

```
attribute = value, value, ...
```

ค่าทั้งหมดต้องคั่นด้วยเครื่องหมายจุลภาค ช่องว่างไม่ใช่สิ่งสำคัญ สามารถใช้ช่วงและ wild cards ได้ ตัวอย่างเช่น:

```
trusted = user[0-9][0-9], admin*
nottrusted = user23, user45
shell = /bin/?sh, \
        /bin/sh, \
        /bin/tcsh
rootgroup=system,bin,sys,security,cron,audit
```

4. บันทึกไฟล์

5. เมื่อต้องการใช้การจัดกลุ่มแอตทริบิวต์ภายในเกณฑ์การเลือกสำหรับ คลาสให้แก้ไขไฟล์ `rules` ชื่อการจัดกลุ่ม แอตทริบิวต์ต้องมีสัญลักษณ์ดอลลาร์ (\$) นำหน้าเพื่อรวมค่าที่สอดคล้องกัน หรือนำหน้าด้วยเครื่องหมายอัศเจรีย์ (!) เพื่อแยกค่าดังกล่าว เครื่องหมายอัศเจรีย์ ไม่สามารถใช้ในสมาชิกของกลุ่ม (ขั้นตอนที่ 3) และเป็นตัวแก้ไขเพียงอย่างเดียวที่สามารถใช้นำหน้าการจัดกลุ่มในไฟล์กฎนี้ ในตัวอย่างต่อไปนี้ เครื่องหมายดอกจัน (*) บ่งชี้บรรทัดข้อคิดเห็น:

```
*class resvd user group application type tag
classA - $trusted,!$nottrusted - - -
classB - - - $shell,!/bin/zsh - -
classC - - $rootgroup - - -
```

6. บันทึกไฟล์

ณ จุดนี้ กฎการจัดหมวดหมู่รวมการจัดกลุ่มค่า แอตทริบิวต์ เมื่อมีการแจ่งส่วนกฎ ถ้าอิลิเมนต์ขึ้นต้นด้วย \$ ระบบจะค้นหาอิลิเมนต์นั้นภายในไฟล์ `groupings` หาก อิลิเมนต์มีโครงสร้างไม่ถูกต้อง หรือถ้าไม่มีไฟล์ `groupings` ระบบจะแสดงข้อความคำเตือนและประมวลผลกฎอื่น ต่อไป

การสร้างชุดคอนฟิกูเรชันตามเวลา:

คุณสามารถสร้างชุดของคอนฟิกูเรชันพิเศษ และกำหนดแต่ละคอนฟิกูเรชัน ภายในชุดเป็นวันและเวลาซึ่งคุณต้องการให้คอนฟิกูเรชันเฉพาะ มีผลบังคับใช้

ชุดเหล่านี้ที่เรียกว่า ชุดคอนฟิกูเรชัน ตามเวลา แยกออกจากคอนฟิกูเรชันปกติโดยสิ้นเชิงแต่สามารถเข้ากันได้ คุณสามารถใช้คำสั่ง `wlmcntrl -u` เพื่อสลับระหว่าง ชุดคอนฟิกูเรชันพิเศษและคอนฟิกูเรชันปกติได้ตามต้องการ

เมื่อใช้ ชุดคอนฟิกูเรชัน นั้นคือคุณเชื่อมโยงคอนฟิกูเรชันที่มีอยู่ โดยปกติแล้ว เข้ากับช่วงเวลาเฉพาะ เนื่องจากสามารถใช้ได้เพียงหนึ่งคอนฟิกูเรชันเท่านั้น ในเวลาใดๆ แต่ละช่วงเวลาที่จะบู๊จิ้งต้องไม่ซ้ำกัน ช่วงเวลาไม่สามารถซ้อนเหลื่อม กันหรือเป็นข้อมูลซ้ำ

`wlmd` daemon แจ้งเตือน WLM เมื่อคอนฟิกูเรชันที่ระบุออกนอกช่วงเวลาและต้องใช้คอนฟิกูเรชัน อื่น เฉพาะผู้ใช้รากสามารถจัดการกับช่วงเวลาเหล่านี้ ซึ่งมีการ ระบุภายในไดเรกทอรีของชุดคอนฟิกูเรชันในไฟล์ ASCII ที่เรียกว่า `.times`

ใช้โปรแกรมต่อไปนี้เพื่อสร้างชุดคอนฟิกูเรชันตามเวลา:

1. ด้วยสิทธิการใช้นามราก ให้สร้างไดเรกทอรีชุดคอนฟิกูเรชัน จากนั้น เปลี่ยนไปยังไดเรกทอรีนั้น ตัวอย่างเช่น:

```
mkdir /etc/wlm/MyConfigSet
cd /etc/wlm/MyConfigSet
```

2. ใช้โปรแกรมแก้ไขที่โปรดปรานเพื่อสร้างไฟล์ `.times` ของชุด คอนฟิกูเรชัน และระบุคอนฟิกูเรชันและช่วงเวลาในรูปแบบต่อไปนี้:

```
ConfigurationName:
    time = "N-N,HH:MM-HH:MM"
```

หรือ

```
ConfigurationName:
    time = -
```

(ไม่มีการระบุค่าเวลา) โดยที่ *N* คือ ตัวเลขที่แสดงถึงวันของสัปดาห์ในช่วง 0 (วันอาทิตย์) ถึง 6 (วันเสาร์), *HH* แสดงถึงชั่วโมงในช่วง 00 (เที่ยงคืน) ถึง 23 (11 p.m.) และ *MM* แสดงถึง นาทีในช่วง 00 ถึง 59 คุณสามารถระบุวันอย่างเดียวหรือไม่ระบุเลย ค่าชั่วโมง 24 ถูกต้องสำหรับชั่วโมงสิ้นสุดของวัน หากว่าค่า นาทีเป็น 00 หากคุณพิมพ์เครื่องหมายขีด (-) แทนช่วงเวลา สำหรับคอนฟิกูเรชันเฉพาะ คอนฟิกูเรชันนั้นจะใช้เมื่อช่วงเวลาของ คอนฟิกูเรชันอื่นไม่มีผลบังคับใช้ สามารถระบุคอนฟิกูเรชันที่ไม่มีช่วงเวลา ได้เพียงหนึ่งคอนฟิกูเรชันเท่านั้น

ตัวอย่างเช่น:

```
conf1:
    time =
conf2:
    time = "1-5,8:00-17:00"
conf2
    time = "6-0,14:00-17:00"
conf3
    time = "22:00-6:00"
```

3. ใช้คำสั่ง `wlmcntrl -u` เพื่ออัปเดต WLM ด้วยชุดคอนฟิกูเรชันใหม่ ตัวอย่างเช่น:

```
wlmcntrl -u /etc/wlm/MyConfigSet
```

ณ จุดนี้ คอนฟิกูเรชันปัจจุบันของ WLM เป็นชุดคอนฟิกูเรชันตามเวลา ใหม่ของคุณ

คุณยังสามารถใช้คำสั่ง `confsetcntrl` และ `lswlmconf` เพื่อสร้างและ จัดการชุดคอนฟิกูเรชันได้ด้วย ตัวอย่างเช่น:

เมื่อต้องการสร้างชุดคอนฟิกูเรชัน `confset1` ด้วยคอนฟิกูเรชันดีฟอลต์ของ `conf1` ให้ใช้คำสั่ง ต่อไปนี้:

```
confsetcntl -C confset1 conf1
```

เมื่อต้องการเพิ่ม conf2 ลงใน confset1 และทำให้เป็นคอนฟิกูเรชันที่ใช้งานอยู่ตั้งแต่เวลา 8:00 AM ถึง 5:00 PM ทุกวัน ให้ใช้คำสั่งต่อไปนี้:

```
confsetcntl -d confset1 -a conf2 "0-6,08:00-17:00"
```

เมื่อต้องการทำให้ ชุดคอนฟิกูเรชันนี้เป็นคอนฟิกูเรชันที่ใช้งานอยู่ให้ใช้คำสั่งต่อไปนี้:

```
wlmcntl -d confset1
```

การสร้างชุดรีซอร์ส:

การใช้ชุดรีซอร์ส (rsets) นับเป็นวิธีที่มีประสิทธิภาพในการแยกเวิร์กโหลดที่เกี่ยวข้องกับ CPU ออกจากกัน โดยการแยกสองเวิร์กโหลดที่แตกต่างกัน ออกเป็นสองคลาสและให้ชุดย่อยที่แตกต่างกันของ CPU กับแต่ละคลาสทำให้คุณมั่นใจได้ว่าสองเวิร์กโหลดจะไม่มีทางช่วงชิงรีซอร์ส CPU ระหว่างกัน แม้ว่าทั้งสองเวิร์กโหลดยังคงช่วงชิงหน่วยความจำฟิสิคัลและ I/O แบบดวิวิ

วิธีง่ายที่สุดในการสร้างชุดรีซอร์สคือ ใช้อินเตอร์เฟซ SMIT (พาด่วน `smit addrsetcntl`) หรือคำสั่ง `mkrset`

เพื่อช่วยแนะนำ ตัวอย่างต่อไปนี้แสดงแต่ละขั้นตอนของการสร้างและ การตั้งชื่อชุดรีซอร์สบนระบบ 4-ทาง เป้าหมายคือการสร้างชุดรีซอร์ส ที่มีตัวประมวลผล 0 ถึง 2 และใช้ในคอนฟิกูเรชัน WLM เพื่อจำกัด โพรเซสทั้งหมดของซูเปอร์คลาสในสามตัวประมวลผลเหล่านี้

1. ด้วยสิทธิการใช้งานราก ให้ดูบล็อกการสร้างที่มีอยู่ (ซึ่งใช้ในการสร้าง ชุดรีซอร์ส) โดยใช้คำสั่งต่อไปนี้:

```
lsrset -av
```

เอาต์พุต สำหรับตัวอย่างนี้เป็นดังต่อไปนี้:

T	Name	Owner	Group	Mode	CPU	Memory	Resources
r	sys/sys0	root	system	r-----	4	98298	sys/sys0
r	sys/node.00000	root	system	r-----	4	98298	sys/sys0
r	sys/mem.00000	root	system	r-----	0	98298	sys/mem.00000
r	sys/cpu.00003	root	system	r-----	1	0	sys/cpu.00003
r	sys/cpu.00002	root	system	r-----	1	0	sys/cpu.00002
r	sys/cpu.00001	root	system	r-----	1	0	sys/cpu.00001
r	sys/cpu.00000	root	system	r-----	1	0	sys/cpu.00000

ในเอาต์พุต `sys/sys0` แสดงถึงทั้งระบบ (ในกรณีนี้ คือ SMP 4-ทาง) เมื่อคลาส WLM ไม่ได้ระบุแอตทริบิวต์ `rset` นี้จะเป็นชุดดีฟอลต์ที่โปรเซสสามารถเข้าถึงได้

2. สร้างและตั้งชื่อชุดรีซอร์สโดยใช้พาด่วน SMIT ต่อไปนี้:

```
smit addrsetcntl
```

สำหรับ ตัวอย่างนี้ กรอกข้อมูลในฟิลด์ดังนี้:

พื้นที่ว่างชื่อ

admin

ชื่อชุดรีซอร์ส

proc0_2

รีเซ็ต เลือกบรรทัดที่สอดคล้องกับหน่วยความจำและ CPUs 0 ถึง 2 (sys/cpu.00000 ถึง sys.cpu.00002) จากรายการ

ฟิลด์อื่นทั้งหมด

เลือกจากรายการ

เมื่อคุณป้อนข้อมูลในฟิลด์เสร็จแล้วและออกจาก SMIT admin/proc0_2 rset จะถูกสร้างขึ้นใน /etc/rsets

3. เมื่อต้องการใช้ rset ใหม่ ให้เพิ่มลงในโครงสร้างข้อมูลเคอร์เนลโดยใช้พาด่วน SMIT ต่อไปนี้:

```
smit reloadrsetcntl
```

เมนูนี้ช่วยให้ทางเลือกในการรีโหลดฐานข้อมูลเป็น now, at next boot หรือ both เนื่องจากนี่เป็นครั้งแรกที่คุณใช้ชุดรีเซ็ตใหม่ ให้เลือก both เพื่อโหลด rset นี้ในตอนนี้อย่างไรก็ตามหลังจากรีบูตแต่ละครั้ง (หากคุณเปลี่ยน rset ที่มีอยู่ คุณอาจต้องเลือก now)

4. เพิ่ม rset ใหม่ลงในคลาส WLM โดยใช้พาด่วน SMIT ต่อไปนี้:

```
smit wlmclass_gal
```

เลือก คลาส (ในตัวอย่างนี้ super1) แล้วเลือก admin/proc0_2 จาก รายการที่มีอยู่สำหรับฟิลด์ ชุดรีเซ็ต หลังจาก คุณเลือกและออกจาก SMIT ไฟล์ classes บน ดิสก์จะเปลี่ยน

5. ทำอย่างใดอย่างหนึ่งต่อไปนี้:

- ถ้า WLM กำลังรัน ให้อัปเดตคอนฟิกูเรชันโดยใช้พาด่วน SMIT ต่อไปนี้:

```
smit wlmupdate
```

- ถ้า WLM ไม่ได้กำลังรัน ให้เริ่มต้น WLM โดยใช้พาด่วน SMIT ต่อไปนี้:

```
smit wlmstart
```

6. มอนิเตอร์ผลกระทบของชุดรีเซ็ตใหม่บนคลาส ตัวอย่างเช่น:

- a. เริ่มต้นลูป 90 CPU (โปรแกรมที่ดำเนินการลูปไม่สิ้นสุด) ในคลาส super1

- b. พิมพ์ wlmstat บนบรรทัดคำสั่ง เอาต์พุต สำหรับตัวอย่างนี้เป็นดังต่อไปนี้

```
CLASS CPU MEM BIO
Unclassified 0 0 0
Unmanaged 0 0 0
Default 8 0 0
Shared 0 0 0
System 0 0 0
super1 75 0 0
super2 0 0 0
super2.Default 0 0 0
super2.Shared 0 0 0
super2.sub1 0 0 0
super2.sub2 0 0 0
```

เอาต์พุตนี้แสดงว่าโปรเซสที่โยงกับ 90 CPU ซึ่งถ้าไม่มีการจำกัดจะใช้ CPU 100% ขณะนี้ใช้ CPU เพียง 75% เนื่องจากชุดรีเซ็ตจำกัดให้รันบน CPUs 0 ถึง 2 เท่านั้น

- c. เมื่อต้องการตรวจสอบชุดรีเซ็ตซึ่งโปรเซส (ที่ระบุโดย PID) สามารถเข้าถึงได้ ให้ใช้พาด่วน SMIT ต่อไปนี้:

```
smit lsrsetproc
```

ป้อน PID ของโปรเซสซึ่งคุณสนใจ หรือเลือกจากรายการ เอาต์พุต ต่อไปนี้เป็นเอาต์พุตของลูโป้โปรเซสอย่างใดอย่างหนึ่ง:

```
CPU Memory Resources
3 98298 sys/mem.00000 sys/cpu.00002 sys/cpu.00001 sys/cpu.00000
```

อย่างไรก็ตาม โปรเซสจากคลาสที่ไม่ได้ระบุแอตทริบิวต์ rset ใช้ชุดรีซอร์ส Default ที่รวมตัวประมวลผลทั้งหมด ยกเว้นตัวประมวลผลเหล่านั้นที่เป็นส่วนหนึ่งของชุดรีซอร์ส เฉพาะ โปรเซสที่ไม่ได้เป็นของคลาสใดๆ ใช้คลาส System (ถ้าเป็นโปรเซสรูท) หรือคลาส Default (ถ้าเป็นโปรเซสที่ไม่ใช่รูท) คลาสเหล่านี้อาจมีชุดรีซอร์ส ที่กำหนดไว้สำหรับคลาส

เอาต์พุตต่อไปนี้มาจากโปรเซส init ซึ่ง อยู่ในคลาสที่ไม่ได้ระบุชุดรีซอร์ส:

```
CPU Memory Resources
4 98298 sys/sys0
```

ณ จุดนี้มีชุดรีซอร์สของคุณและกำลังถูกใช้โดยคลาสอย่างน้อย หนึ่งคลาสภายใน WLM

หมายเหตุ: WLM จะไม่ตั้งค่าส่วนต่อพ่วง rset สำหรับโปรเซสที่มีการผูกพันที่ย่อย bindprocessor หรือ ส่วนต่อพ่วง rset อื่นในปัจจุบัน เมื่อส่วนต่อพ่วงอื่นไม่มีอยู่ก็ต่อไป WLM จะกำหนด rset โดยอัตโนมัติ

หมายเหตุ: ชุดรีซอร์สสามารถสร้างได้สำหรับคลาส WLM ใดๆ ซึ่งรวมถึงคลาส Default และ System

หลักการที่เกี่ยวข้อง:

“วิธีสรีชุดรีซอร์สใน Workload Manager” ในหน้า 564

เซอร์วิสวิธีสรี rset ช่วยให้ผู้ใช้และระบบ สามารถกำหนดและตั้งชื่อชุดรีซอร์ส เพื่อให้ผู้ใช้หรือแอปพลิเคชันอื่นสามารถใช้ชุดรีซอร์สนั้นได้

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง lsrset

การกำหนดคอนฟิก Workload Manager เพื่อรวม เวิร์กโหลด:

Workload Manager (WLM) ช่วยให้คุณ สามารถควบคุมรีซอร์สที่ใช้โดยงานบนระบบของคุณได้

ดีฟอลต์ WLM คอนฟิกูเรชันเพิ่มเพลตมีอยู่บนทุกระบบปฏิบัติการ AIX ที่ติดตั้ง โพรซีเดอร์ต่อไปนี้อัปเดต WLM คอนฟิกูเรชันเพิ่มเพลตเพื่อนำนโยบายการจัดการรีซอร์สไปใช้บนเซิร์ฟเวอร์แบบแบ่งใช้ สามารถใช้คอนฟิกูเรชันที่ได้ เป็นจุดเริ่มต้นสำหรับการทดสอบ วิธีการกำหนดคอนฟิก WLM โดยละเอียดของคุณ จะขึ้นอยู่กับข้อกำหนดเวิร์กโหลดและนโยบายสำหรับสถานะแวดล้อมของคุณ

หมายเหตุ:

1. การใช้ที่มีประสิทธิภาพของ WLM จำเป็นต้องมีความรู้อย่างกว้างขวางเกี่ยวกับโปรเซสระบบที่มีอยู่ และประสิทธิภาพ การทำซ้ำการทดสอบและการปรับอาจจะเป็นสิ่งจำเป็นก่อน คุณสามารถจัดทำคอนฟิกูเรชันที่ใช้งานได้ดีสำหรับเวิร์กโหลดของคุณ หากคุณกำหนดคอนฟิก WLM ด้วยค่าที่มากเกินไปหรือค่าที่ไม่ถูกต้อง คุณอาจทำให้ประสิทธิภาพระบบ ลดลงอย่างมาก
2. โปรเซสของการกำหนดคอนฟิก WLM จะง่ายขึ้นเมื่อคุณ ทราบแอตทริบิวต์การจัดประเภทตั้งแต่หนึ่งรายการขึ้นไปของโปรเซสอยู่แล้ว (ตัวอย่างเช่น ผู้ใช้ กลุ่ม หรือชื่อแอปพลิเคชัน) หาก你不คุ้นเคยกับ การใช้รีซอร์สในปัจจุบัน ให้ใช้เครื่องมือ เช่น topas เพื่อระบุโปรเซสที่เป็น ผู้ใช้รีซอร์สหลักและใช้ข้อมูลที่ได้เป็นจุดเริ่มต้น สำหรับการกำหนดคลาสและกฎต่างๆ

3. สถานการณ์จำลองต่อไปนี้จะสมมติว่าคุณคุ้นเคยกับ แนวคิด Workload Manager พื้นฐานดังอธิบายไว้ใน “แนวคิดการจัดการเวิร์กโหลด” ในหน้า 526

ไฟล์คอนฟิกูเรชัน WLM มีอยู่ในไดเรกทอรี `/etc/wlm/ConfigurationName` แต่ละคลาสย่อยของแต่ละซูเปอร์คลาสมีการกำหนดไว้ในไฟล์คอนฟิกูเรชันที่ชื่อ `/etc/wlm/ConfigurationName/SuperClassName` สำหรับข้อมูลเพิ่มเติมเกี่ยวกับไฟล์เหล่านี้ให้ดูที่ *Files Reference*

ในโพธิ์เตอร์ต่อไปนี้จะรวมเวิร์กโหลดจากเซิร์ฟเวอร์แผนกที่แยกต่างหากกัน สองเครื่องเข้าเป็นเซิร์ฟเวอร์ที่ใหญ่ขึ้นเครื่องเดียว ตัวอย่างนี้แก้ไขไฟล์การกำหนดคอนฟิก แต่คุณยังสามารถสร้างคอนฟิกูเรชันโดยใช้ SMIT (ใช้พาด่วน `smit wlmconfig_create`) โดยคร่าวๆ ในโพธิ์เตอร์นี้ คุณจะทำต่อไปนี้:

1. ระบุความต้องการรีซอร์สของแอปพลิเคชันซึ่งคุณต้องการรวม การทำเช่นนี้จะช่วยให้คุณกำหนดจำนวนแอปพลิเคชันซึ่งคุณสามารถย้ายไปยัง เซิร์ฟเวอร์ที่ใหญ่ขึ้นได้
2. กำหนดระดับ ตลอดจนจนส่วนแบ่งรีซอร์สและขีดจำกัด เพื่อเริ่มต้นการทดสอบ กับเวิร์กโหลดที่รวม
3. ปรับคอนฟิกูเรชันให้ละเอียดมากขึ้นจนกว่าคุณได้รับผลที่ต้องการ

The information in this how-to scenario was tested using specific versions of AIX. The results you obtain might vary significantly depending on your version and level of AIX.

ขั้นตอนที่ 1 ระบุความต้องการแอปพลิเคชัน

ในสถานการณ์จำลองนี้ เวิร์กโหลดเป็นแบบปกติซึ่งคุณอาจเห็นบนเซิร์ฟเวอร์ฐานข้อมูล สมมติว่างานแบ่งออกเป็นหมวดหมู่ทั่วไปต่อไปนี้:

ผู้ฟัง นี่เป็นโปรเซสที่โดยเวลาส่วนใหญ่แล้ว พักอยู่และตื่นขึ้นมาเป็นครั้งคราว เพื่อตอบกลับคำร้องขอ แม้ว่าโปรเซสเหล่านี้ไม่ได้ใช้รีซอร์สมากนัก แต่เวลาตอบกลับสามารถเป็นประเด็นที่สำคัญมาก

ผู้ปฏิบัติงาน

นี่เป็นโปรเซสที่ทำงานในนามของคำร้องขอ ไม่ว่าคำร้องขอ จะเป็นแบบโลคัลหรือรีโมต โปรเซสเหล่านี้อาจใช้เวลา CPU และหน่วยความจำเป็นจำนวนมาก

ผู้รายงาน

นี่เป็นโปรเซสที่ทำการกิจอัตโนมัติ โปรเซสหมวดนี้อาจต้องการ เวลา CPU หรือหน่วยความจำจำนวนมาก และอาจทำให้เวลาการตอบกลับช้าลง

มอนิเตอร์

นี่เป็นโปรเซสที่โดยปกติแล้ว รันเป็นครั้งคราวเพื่อตรวจสอบสถานะของ ระบบหรือแอปพลิเคชัน โปรเซสเหล่านี้อาจใช้รีซอร์สจำนวนมาก แต่ในช่วงเวลาสั้นๆ เท่านั้น

คำสั่ง นี่เป็นคำสั่งหรือแอปพลิเคชันอื่นซึ่งผู้ใช้ระบบอาจรันในเวลาใด ก็ได้ ความต้องการรีซอร์สของหมวดหมู่นี้ไม่สามารถคาดการณ์ได้

นอกเหนือจากงานนี้แล้ว งานที่จัดตารางเวลายังแบ่งออกเป็น หมวดหมู่อย่างใดอย่างหนึ่งต่อไปนี้:

SysTools

นี่เป็นโปรเซสที่ทำการกิจอัตโนมัติ งานเหล่านี้ไม่สำคัญมากต่อ การดำเนินงานระบบแต่ต้องรันเป็นครั้งคราวภายในข้อจำกัดเวลาบางอย่าง

SysBatch

นี่เป็นโปรเซสที่รันไม่บ่อยนัก ไม่สำคัญมากต่อการดำเนินงานระบบ และไม่จำเป็นต้องเสร็จภายในเวลา

ขั้นตอนแรกในการสร้างคอนฟิกูเรชันคือการกำหนด คลาสและกฎต่างๆ ในขั้นตอนต่อไปนี้ คุณจะใช้หมวดหมู่ทั้งหมดที่ไปซึ่งแสดงรายการข้างบนเพื่อกำหนดคลาสของคุณ ใช้โพธิ์เตอร์ต่อไปนี้:

1. จัดทำคอนฟิกูเรชันใหม่ภายในไดเร็กทอรี /etc/wlm ที่เรียกว่า MyConfig โดยใช้คำสั่งต่อไปนี้:

```
mkdir /etc/wlm/MyConfig
```

2. คัดลอกไฟล์เทมเพลตเข้าในไดเร็กทอรี /etc/wlm/MyConfig โดยใช้คำสั่งต่อไปนี้:

```
cp -pr /etc/wlm/template/* /etc/wlm/MyConfig
```

3. หากต้องการสร้างซูเปอร์คลาส ให้ใช้โปรแกรมแก้ไขโปรดของคุณแก้ไขไฟล์ /etc/wlm/MyConfig/classes เพื่อให้มีสิ่งดังต่อไปนี้:

System:

Default:

DeptA:

DeptB:

SysTools:

SysBatch:

จากจุดเริ่มต้น คุณกำหนดหนึ่งซูเปอร์คลาสสำหรับแต่ละแผนก (เนื่องจากสองแผนกจะแบ่งใช้เซิร์ฟเวอร์) ซูเปอร์คลาส SysTool และ SysBatch จะจัดการงานที่จัดตารางเวลาตามโครงสร้างในหมวดหมู่ทั่วไปข้างบน ซูเปอร์คลาส ระบบ และ ดีฟอลต์ มีการกำหนดไว้เสมอ

4. ภายในไดเร็กทอรี MyConfig ให้สร้างไดเร็กทอรีสำหรับแต่ละซูเปอร์คลาส DeptA และ DeptB (เมื่อสร้างคอนฟิกูเรชัน คุณต้องสร้างไดเร็กทอรีสำหรับทุกซูเปอร์คลาสที่มีคลาสย่อย) ในขั้นตอนต่อไปนี้ คุณกำหนดหาคาสย่อย (หนึ่งคลาสย่อยสำหรับแต่ละหมวดหมู่) สำหรับแต่ละซูเปอร์คลาสของแผนก
5. เมื่อต้องการสร้างคลาสย่อยสำหรับแต่ละหมวดหมู่ทั่วไปของงาน ให้แก้ไขไฟล์ /etc/wlm/MyConfig/DeptA/classes และ /etc/wlm/MyConfig/DeptB/classes เพื่อให้มีข้อมูลต่อไปนี้:

Listen:

Work:

Monitor:

Report:

Command:

หมายเหตุ: เนื้อหาของไฟล์ classes สามารถแตกต่างกันสำหรับแต่ละซูเปอร์คลาส

หลังจากระบุคลาสแล้ว ในขั้นตอนต่อไปนี้ คุณสามารถสร้างกฎการจัดหมวดหมู่ที่จะใช้เพื่อจัดประเภทโปรเซสที่ระดับซูเปอร์คลาสและคลาสย่อย เพื่อให้ง่าย จึงสมมติว่าแอ็พพลิเคชันทั้งหมดรันจากที่ตั้งซึ่งรู้จัก นั่นคือโปรเซสทั้งหมดจากแผนกหนึ่งรันภายใต้กลุ่ม deptA UNIX และโปรเซสจากแผนกอื่นรันภายใต้กลุ่ม deptB UNIX

6. เมื่อต้องการสร้างกฎการกำหนดซูเปอร์คลาส ให้แก้ไขไฟล์ /etc/wlm/MyConfig/rules เพื่อให้มีข้อมูลต่อไปนี้:

```
DeptA - - deptA - -
DeptB - - deptB - -
SysTools - root,bin - /usr/sbin/tools/* -
SysBatch - root,bin - /usr/sbin/batch/* -
System - root - - -
Default - - - - -
```

หมายเหตุ: หากสามารถรันมากกว่าหนึ่งอินสแตนซ์ของแอ็พพลิเคชันเดียวกันและแอ็ททริบิวต์การจัดประเภททั้งหมด (นอกเหนือจาก แท็ก) เหมือนกัน ให้ใช้คำสั่ง `wlmassign` หรือรูทีนย่อย `wlm_set_tag` เพื่อแยกความแตกต่างโดยกำหนดให้กับคลาสที่แตกต่างกัน

- เมื่อต้องการสร้างกฎคลาสย่อยที่เฉพาะเจาะจงมากขึ้น ให้สร้างไฟล์ `/etc/wlm/MyConfig/DeptA/rules` และ `/etc/wlm/MyConfig/DeptB/rules` ที่มีเนื้อหาต่อไปนี้:

```
Listen - - - /opt/myapp/bin/listen* -
Work - - - /opt/myapp/bin/work* -
Monitor - - - /opt/bin/myapp/bin/monitor -
Report - - - /opt/bin/myapp/report* -
Command - - - /opt/commands/* -
```

- เมื่อต้องการกำหนดลักษณะการใช้รีซอร์สของแต่ละคลาสให้เริ่มต้น WLM ในโหมด `passive` โดยใช้คำสั่งต่อไปนี้:

```
wlmcntrl -p -d MyConfig
```

หลังจาก เริ่มต้น WLM ในโหมด `passive` แล้ว คุณสามารถรันแต่ละแอ็พพลิเคชันแยกกัน ในช่วงแรกเพื่อให้ได้มุมมองความต้องการรีซอร์สที่ละเอียดมากขึ้น จากนั้น คุณสามารถรัน แอ็พพลิเคชันทั้งหมดพร้อมกันเพื่อกำหนดการโต้ตอบระหว่างคลาสทั้งหมด ให้ดียิ่งขึ้น

อีกเมธอดหนึ่งในการระบุความต้องการรีซอร์สของแอ็พพลิเคชัน อาจเป็น การรัน WLM ในโหมด `passive` ก่อนบนเซิร์ฟเวอร์แยกต่างหาก ซึ่งคุณกำลังรวมแอ็พพลิเคชัน ข้อเสียของแนวทางนี้คือ คุณจะต้องสร้างคอนฟิกูเรชันขึ้นใหม่บนระบบที่ใหญ่ขึ้น และเปอร์เซ็นต์ของรีซอร์สที่ต้องการจะแตกต่างบนระบบ ที่ใหญ่ขึ้น

ขั้นตอนที่ 2. กำหนดระดับ ส่วนแบ่งใช้ และขีดจำกัด

คอนฟิกูเรชัน WLM เป็นการนำนโยบายการจัดการรีซอร์สไปใช้ การรัน WLM ในโหมด `passive` ให้ข้อมูลที่ช่วยให้คุณกำหนดว่านโยบายการจัดการรีซอร์สของคุณเหมาะสมสำหรับเวิร์กโหลดที่กำหนดหรือไม่ ขณะนี้ คุณสามารถกำหนดระดับ ส่วนแบ่งใช้ และขีดจำกัดเพื่อควบคุมเวิร์กโหลดของคุณตามนโยบายการจัดการรีซอร์สได้

สำหรับสถานการณ์จำลองนี้ สมมติว่าคุณมีความต้องการต่อไปนี้:

- คลาส ระบบ ต้องมีลำดับความสำคัญสูงสุดและเป็นการเข้าถึงที่รับประกันถึง เปอร์เซ็นต์ของรีซอร์สระบบได้ตลอดเวลา
- คลาส SysTools ต้องมีการเข้าถึงเปอร์เซ็นต์เฉพาะของรีซอร์สได้ตลอดเวลา แต่ต้องไม่มากเกินไปจนส่งผลกระทบต่อแอ็พพลิเคชันที่กำลังรันใน DeptA และ DeptB
- คลาส SysBatch ไม่สามารถเกี่ยวข้องกับงานอื่นใดบน ระบบ
- DeptA จะได้รับ 60% ของรีซอร์สที่มีอยู่ (หมายถึงรีซอร์สที่มีอยู่สำหรับคลาสที่มีส่วนแบ่งใช้) และ DeptB จะได้รับ 40% ภายใน DeptA และ DeptB:
 - โพรเซสในคลาส ฟัง ต้องตอบกลับคำร้องขอด้วยเวลาแฝงต่ำ แต่ต้องไม่ใช้รีซอร์สมากนัก
 - คลาส งาน ต้องสามารถใช้รีซอร์สส่วนใหญ่ได้ คลาส มอนิเตอร์ และ คำสั่ง ต้องใช้รีซอร์สบางส่วน แต่น้อยกว่าคลาส งาน
 - คลาส รายงาน ไม่สามารถเกี่ยวข้องกับงานอื่นใด

ในโพรซีเจอร์ต่อไปนี้ คุณกำหนดระดับ ส่วนแบ่งใช้ และขีดจำกัด:

1. เมื่อต้องการสร้างระดับซูเปอร์คลาสให้ใช้โปรแกรมแก้ไขที่โปรดปรานเพื่อแก้ไขไฟล์ `/etc/wlm/MyConfig/classes` ให้มีข้อมูลต่อไปนี้:

System:

Default:

DeptA:

```
localshm = yes
adminuser = adminA
authuser = adminA
inheritance = yes
```

DeptB:

```
localshm = yes
adminuser = adminB
authuser = adminB
inheritance = yes
```

SysTools:

```
localshm = yes
```

SysBatch:

```
tier = 1
localshm = yes
```

ซูเปอร์คลาส SysBatch ถูกวางไว้ในระดับ 1 เนื่องจากคลาสนี้มีงานลำดับความสำคัญต่ำมากซึ่งคุณไม่ต้องการให้เกี่ยวข้องกับงานส่วนที่เหลือบนระบบ (เมื่อไม่ได้ระบุระดับ คลาสดีฟอลต์เป็นระดับ 0) การจัดการซูเปอร์คลาสของแต่ละแผนกมีการกำหนดโดยแอ็ททริบิวต์ `adminuser` และ `authuser` แอ็ททริบิวต์ การรับสิทธิ์การเปิดใช้งานสำหรับ DeptA และ DeptB โพรเซสใหม่ทั้งหมดที่เริ่มต้นในคลาส ที่มีการรับสิทธิ์จะยังคงถูกจัดประเภทในคลาสนั้น

2. เมื่อต้องการสร้างระดับคลาสย่อยสำหรับงานแต่ละกลุ่ม ให้แก้ไขไฟล์ `/etc/wlm/MyConfig/DeptA/classes` และ `/etc/wlm/MyConfig/DeptB/classes` เพื่อให้มีข้อมูลต่อไปนี้:

Listen:

Work:

Monitor:

Report:

```
tier = 1
```

Command:

3. เมื่อต้องการกำหนดส่วนแบ่งใช้แรกเริ่มสำหรับซูเปอร์คลาสให้แก้ไขไฟล์ `/etc/wlm/MyConfig/shares` เพื่อให้มีข้อมูลต่อไปนี้:

DeptA:

```
CPU = 3
memory = 3
```

DeptB:

```
CPU = 2
memory = 2
```

เนื่องจากคุณกำหนด CPU ทั้งหมดให้กับ 5 ส่วนแบ่งใช้ โพรเซส DeptA จะมีสิทธิเข้าถึงสามจากห้าส่วนแบ่งใช้ (หรือ 60%) ของรีซอร์ส CPU ทั้งหมด และโพรเซส DeptB จะมีสิทธิเข้าถึงสองจากห้าส่วนแบ่งใช้ (หรือ 40%) เนื่องจากคุณไม่ได้กำหนดส่วนแบ่งใช้ให้กับคลาส SysTools, ระบบ, และ ดีฟอลต์ เป้าหมายการใช้รีซอร์สของคลาสดังกล่าวจึงยังคงเป็นอิสระจากจำนวน ส่วนแบ่งใช้ที่ใช้งานอยู่ ส่งผลให้คลาสเหล่านั้นมีลำดับความสำคัญสูงกว่า DeptA และ DeptB ในการเข้าถึงรีซอร์ส (จนกว่าใช้ถึงขีดจำกัดของแต่ละคลาส) คุณไม่ได้กำหนดส่วนแบ่งใช้ให้กับ คลาส SysBatch เนื่องจากคลาสนั้น เป็นเพียงซูเปอร์คลาสเดียวในระดับ 1 ดังนั้น การกำหนดส่วนแบ่งใช้ใดๆ จึงไม่มีส่วนเกี่ยวข้องกับงานในคลาส SysBatch สามารถใช้ได้เฉพาะรีซอร์สที่ไม่ได้ใช้โดยคลาสทั้งหมดในระดับ 0 เท่านั้น

- เมื่อต้องการกำหนดส่วนแบ่งใช้แรกเริ่มสำหรับคลาสย่อย ให้แก้ไขไฟล์ `/etc/wlm/MyConfig/DeptA/shares` และ `/etc/wlm/MyConfig/DeptB/shares` เพื่อให้มีข้อมูลต่อไปนี้:

Work:

```
CPU = 5
memory = 5
```

Monitor:

```
CPU = 4
memory = 1
```

Command:

```
CPU = 1
memory = 1
```

เนื่องจากคุณไม่ได้กำหนดส่วนแบ่งใช้ให้กับคลาส พัง คลาสนี้จึงมีการเข้าถึงลำดับความสำคัญสูงสุด (ในซูเปอร์คลาส) ในรีซอร์ส เมื่อคลาสต้องการรีซอร์ส คุณกำหนดจำนวนส่วนแบ่งใช้มากที่สุดให้กับคลาสงาน เนื่องจาก คลาสนี้มีความต้องการรีซอร์สมากที่สุด จากนั้น คุณกำหนด ส่วนแบ่งใช้ให้กับคลาส มอนิเตอร์ และ คำสั่ง ตามลักษณะการทำงานที่สังเกตได้และความสำคัญสัมพันธ์ คุณไม่ได้กำหนดส่วนแบ่งใช้ให้กับคลาส รายงาน เนื่องจากเป็นเพียงคลาสย่อยเดียวในระดับ 1 ดังนั้นการกำหนดส่วนแบ่งใช้ใดๆ จึงไม่มีส่วนเกี่ยวข้องกับงานในคลาส รายงาน สามารถใช้ได้เฉพาะรีซอร์สที่ไม่ได้ใช้โดย คลาสย่อยในระดับ 0 เท่านั้น

ในขั้นตอนต่อไปนี้ ของตัวอย่างนี้ คุณกำหนดขีดจำกัดให้กับคลาสที่ไม่ได้รับการกำหนดส่วนแบ่งใช้ (คุณยังสามารถ กำหนดขีดจำกัดให้กับคลาสที่มีส่วนแบ่งใช้ได้ด้วย ให้ดูที่ การจัดการ รีซอร์สด้วย WLM สำหรับข้อมูลเพิ่มเติม)

- เมื่อต้องการกำหนดขีดจำกัดให้กับซูเปอร์คลาส ให้แก้ไขไฟล์ `/etc/wlm/MyConfig/limits` เพื่อให้มีข้อมูลต่อไปนี้:

Default:

```
CPU = 0%-10%;100%
memory = 0%-10%;100%
```

SysTools:

```
CPU = 0%-10%;100%
memory = 0%-5%;100%
```

System:

```
CPU = 5%-50%;100%
memory = 5%-50%;100%
```

คุณกำหนดขีดจำกัดสูงสุดซอฟต์แวร์ให้กับ คลาส ระบบ, SysTools, และ ดีฟอลต์ เพื่อป้องกันไม่ให้คลาสดังกล่าวเกี่ยวข้องกับอย่างมากกับงานอื่นบนระบบ คุณกำหนดขีดจำกัดต่ำสุดให้กับ คลาส ระบบ สำหรับ CPU และหน่วยความจำเนื่องจาก คลาสนี้มีโปรเซส ที่จำเป็นสำหรับการดำเนินงานระบบ และต้องสามารถใช้จำนวนรีซอร์ส ที่รับประกันได้

6. เมื่อต้องการกำหนดขีดจำกัดให้กับคลาสย่อย ให้แก้ไขไฟล์ `/etc/wlm/MyConfig/DeptA/limits` และ `/etc/wlm/MyConfig/DeptB/limits` เพื่อให้มีข้อมูลต่อไปนี้:

Listen:

```
CPU = 10%-30%;100%
memory = 10%-20%;100%
```

Monitor:

```
CPU = 0%-30%;100%
memory = 0%-30%;100%
```

หมายเหตุ: ขีดจำกัดสามารถแตกต่างกันสำหรับไฟล์คลาสย่อยแต่ละไฟล์

คุณกำหนดขีดจำกัดสูงสุดขอพต์ให้กับ คลาส ฟัง และ มอนิเตอร์ เพื่อป้องกันไม่ให้คลาสดังกล่าวเกี่ยวข้อง อย่างมากกับ คลาสย่อยอื่นในซูเปอร์คลาสเดียวกัน โดยเฉพาะอย่างยิ่ง คุณไม่ต้องการให้ระบบ ยอมรับคำร้องของานต่อไปภายในคลาสรงาน ถ้าคลาสรงาน ไม่มีสิทธิเข้าถึงรีซอร์สที่จะใช้ประมวลผลงานที่ร้องขอ คุณยังกำหนด ขีดจำกัดต่ำสุดให้กับคลาส ฟัง ด้วยเพื่อให้มั่นใจถึงเวลาตอบกลับที่รวดเร็ว ขีดจำกัดต่ำสุดสำหรับหน่วยความจำทำให้มั่นใจว่าหน้าที่ใช้โดยคลาสนี้จะไม่ ถูก ขโมยโดยการแทนที่หน้า ส่งผลให้เวลาการดำเนินการเร็วขึ้น ขีดจำกัดต่ำสุด สำหรับ CPU ช่วยให้เห็นใจว่าเมื่อสามารถ รันโปรเซสเหล่านี้ได้ โปรเซสจะมี การเข้าถึงลำดับความสำคัญสูงสุด (ในซูเปอร์คลาส) ในรีซอร์ส CPU

ขั้นตอนที่ 3. ปรับคอนฟิกูเรชัน Workload Manager โดยละเอียด

1. มอนิเตอร์ระบบโดยใช้คำสั่ง `wlmstat` และตรวจสอบว่า การควบคุมที่ทำโดย WLM เหมาะสมกับเป้าหมายของคุณ และไม่ได้ทำให้บาง แอปพลิเคชันขาดรีซอร์สอย่างไม่เหมาะสมในขณะที่แอปพลิเคชันอื่นได้รับรีซอร์สมากกว่าที่ควรจะได้รับ หากเป็นกรณีนี้ ให้ปรับส่วนแบ่งใช้และรีเฟรช WLM
2. เมื่อคุณมอนิเตอร์และปรับส่วนแบ่งใช้ ขีดจำกัด และหมายเลขระดับ ให้ตัดสินใจว่าคุณต้องการมอบหมายการจัดการ คลาสย่อยของบางซูเปอร์คลาสหรือ ทุกซูเปอร์คลาส จากนั้น ผู้ดูแลระบบสามารถมอนิเตอร์และตั้งค่า ส่วนแบ่งใช้คลาสย่อย ขีดจำกัด และหมายเลขระดับ

ผู้ดูแลระบบของแต่ละซูเปอร์คลาสสามารถทำซ้ำโปรเซสนี้สำหรับ คลาสย่อยของแต่ละซูเปอร์คลาสได้ ความแตกต่างเพียง อยางเดียวคือ WLM ไม่สามารถรัน ในโหมด `passive` ที่ระดับคลาสย่อยเท่านั้น คอนฟิกูเรชันคลาสย่อยและการปรับต้องทำ ด้วย WLM ในโหมด `active` วิธีหนึ่งที่ไม่กระทบต่อผู้ใช้ และแอปพลิเคชันในซูเปอร์คลาสคือ การเริ่มต้นหมายเลขระดับ และ ส่วนแบ่งใช้ และขีดจำกัดสำหรับคลาสย่อยที่ค่าดีฟอลต์ ('-' (ยัติภังค์) สำหรับส่วนแบ่งใช้, 0% สำหรับค่าต่ำสุด, และ 100% สำหรับค่าสูงสุดขอพต์และฮาร์ด) ด้วยค่าที่ตั้งเหล่านี้ WLM จะไม่ควบคุมการจัดสรรรีซอร์สระหว่างคลาสย่อย

สำหรับข้อมูลเพิ่มเติม

- Workload Manager
- การจัดการเวิร์กโหลด
- การวิเคราะห์การจัดการเวิร์กโหลด ใน *Performance management*
- คำอธิบายเกี่ยวกับไฟล์ คลาส, การจำกัด, กฎ, และ ส่วนแบ่งใช้ใน *Files Reference*
- `topas`, `wlmassign`, `wlmcheck`, `wlmcntrl` และ `wlmstat`
- คำอธิบายรูทีนย่อย WLM โดยเฉพาะ `wlm_set_tag`

หลักการที่เกี่ยวข้อง:

“การตั้งค่า Workload Manager” ในหน้า 564

นิยามคลาส แอ็ททริบิวต์คลาส ส่วนแบ่งใช้และขีดจำกัด และกฎการกำหนดคลาสอัตโนมัติ สามารถป้อนได้โดยใช้ SMIT หรือ อินเทอร์เน็ตคำสั่ง WLM นิยามและกฎเหล่านี้มีการเก็บไว้ในไฟล์ข้อความปกติ ซึ่งยังสามารถสร้างหรือแก้ไขโดยใช้ โปรแกรมแก้ไขข้อความได้ด้วย

คลาส

Workload Manager ช่วยคุณควบคุม การจัดสรรรีซอร์สระบบ โดยการกำหนดคลาสของเซิร์ฟเวอร์และการจัดสรร รีซอร์สให้กับ แต่ละคลาสเหล่านี้

แต่ละคลาสมีชุดของแอ็ททริบิวต์ที่กำหนดสิทธิในรีซอร์สของคลาส และลักษณะการทำงานอื่น ทุกโปรเซสบนระบบถูกจัด ประเภทเป็น คลาสเซิร์ฟเวอร์ และต้องปฏิบัติตามการบังคับใช้เกี่ยวกับสิทธิในรีซอร์ส และลักษณะการทำงานของคลาสนั้น โปรเซสมีการกำหนดให้กับคลาส ด้วยตนเองโดยใช้ การกำหนดด้วยตนเอง หรือโดยอัตโนมัติตามกฎการจัดหมวดหมู่ที่ผู้ใช้ กำหนด ได้อย่างหนึ่ง

WLM สนับสนุนคลาสสองระดับคือ: *ซูเปอร์คลาส* และ *คลาสย่อย* ซูเปอร์คลาสได้รับมอบสิทธิในรีซอร์สตามรีซอร์สระบบที่มี อยู่ และคลาสย่อยได้รับมอบสิทธิในรีซอร์สที่สัมพันธ์กับสิทธิของ ซูเปอร์คลาสที่เชื่อมโยง คุณสามารถเลือกที่จะกำหนดคลาส ย่อยเพื่อให้ สามารถควบคุมโปรเซสในซูเปอร์คลาสได้ละเอียดมากยิ่งขึ้น คุณยังสามารถ มอบหมายความรับผิดชอบในการ กำหนดคลาสย่อยโดยการระบุ admin หรือ admin สำหรับซูเปอร์คลาส

สำหรับทั้งระดับซูเปอร์คลาสและคลาสย่อย คุณสามารถกำหนดคลาส ส่วนแบ่งใช้ และขีดจำกัดรีซอร์ส และกฎต่างๆ ได้โดยใช้ SMI หรืออินเทอร์เน็ตคำสั่ง แอ็พพลิเคชันสามารถใช้ WLM APIs นิยามคอนฟิกูเรชัน มีการเก็บไว้ในชุดของไฟล์ข้อความที่เรียกว่า *ไฟล์คุณสมบัติ WLM*

ชื่อคลาสยาวได้สูงสุด 16 อักขระและสามารถมีตัวอักษรตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก ตัวเลข และขีดเส้นใต้ (_) เท่านั้น สำหรับคอนฟิกูเรชัน WLM ที่กำหนด ชื่อซูเปอร์คลาสแต่ละชื่อต้องไม่ซ้ำกัน ชื่อคลาสย่อยแต่ละชื่อต้องไม่ซ้ำกันภายใน ซูเปอร์ คลาสนั้น แต่อาจตรงกับชื่อคลาสย่อยในซูเปอร์คลาสอื่นได้ เพื่อระบุทุกคลาสย่อยโดยไม่ซ้ำกัน ชื่อเต็มของคลาสย่อยจึง ประกอบด้วย ชื่อซูเปอร์คลาสและชื่อคลาสย่อยที่แบ่งด้วยจุด ตัวอย่างเช่น: *Super.Sub*

ซูเปอร์คลาส:

ผู้ดูแลระบบสามารถกำหนดได้สูงสุด 64 ซูเปอร์คลาส

นอกจากนี้ มีการสร้างห้าซูเปอร์คลาสขึ้นโดยอัตโนมัติดังต่อไปนี้:

ซูเปอร์คลาส ดีฟอลต์

เป็นซูเปอร์คลาสดีฟอลต์และมีการกำหนดเสมอ โปรเซสที่ไม่ใช่รากทั้งหมด และไม่ได้มีการกำหนดให้กับซูเปอร์ คลาสเฉพาะโดยอัตโนมัติจะถูกกำหนด ให้กับซูเปอร์คลาส ดีฟอลต์ คุณยังสามารถกำหนดโปรเซสอื่นให้กับซูเปอร์ คลาส ดีฟอลต์ ได้ ด้วยโดยการระบุกฎการกำหนดเฉพาะ

ซูเปอร์คลาส ระบบ

มีโปรเซสที่มีเอกสิทธิ์ทั้งหมด (ราก) ซึ่งกำหนดให้ ถ้าโปรเซสเหล่านั้น ไม่ได้ถูกกำหนดโดยกฎให้กับคลาสเฉพาะ ซูเปอร์คลาสนี้ยังรวบรวมหน้า หน่วยความจำที่เป็นสมาชิกของเซ็กเมนต์หน่วยความจำเคอร์เนลและเคอร์เนลโปร เซส คุณยังสามารถกำหนดโปรเซสอื่นให้กับซูเปอร์คลาส ระบบ ได้ด้วย โดยการระบุกฎการกำหนดเฉพาะสำหรับซู เปอร์คลาสนี้ ซูเปอร์คลาสนี้มีขีดจำกัดต่ำสุดของ หน่วยความจำเป็น 1% โดยค่าดีฟอลต์

ซูเปอร์คลาส แบบแบ่งใช้

ได้รับหน้าหน่วยความจำที่แบ่งใช้โดยโปรเซสในมากกว่าหนึ่งซูเปอร์คลาส รวมถึงหน้าในพื้นที่หน่วยความจำแบบแบ่งใช้และหน้าในไฟล์ซึ่งใช้โดยโปรเซสในมากกว่าหนึ่งซูเปอร์คลาส (หรือในคลาสย่อยของซูเปอร์คลาสที่แตกต่างกัน) หน่วยความจำและไฟล์แบบแบ่งใช้ที่ใช้โดยหลายโปรเซสซึ่งทั้งหมดเป็นสมาชิกของซูเปอร์คลาสเดียว (หรือคลาสย่อยของซูเปอร์คลาสเดียวกัน) จะมีการเชื่อมโยงเข้ากับซูเปอร์คลาสนั้น เฉพาะถ้าโปรเซสจากซูเปอร์คลาสอื่นเข้าถึงพื้นที่หน่วยความจำหรือไฟล์แบบแบ่งใช้ หน้าจะถูกวางไว้ในซูเปอร์คลาสแบบแบ่งใช้ ซูเปอร์คลาสนี้สามารถมีได้เฉพาะส่วนแบ่งใช้และขีดจำกัด หน่วยความจำฟิสิกส์ที่ใช้กับซูเปอร์คลาส และไม่สามารถมีส่วนแบ่งใช้หรือขีดจำกัดสำหรับรีซอร์สชนิดอื่น คลาสย่อย หรือกฎการกำหนดที่ระบุ การที่เซ็กเมนต์หน่วยความจำซึ่งแบ่งใช้โดยโปรเซสในคลาสย่อยที่แตกต่างกันของซูเปอร์คลาสเดียวกันจะถูกจัดประเภทเข้าในคลาสย่อยแบบแบ่งใช้หรือยังคงอยู่ในคลาสย่อยดั้งเดิม ขึ้นอยู่กับค่าของแอตทริบิวต์ `localshm` ของคลาสย่อยดั้งเดิม

ซูเปอร์คลาสที่ไม่ได้จัดประเภท

เป็นการจัดสรรหน่วยความจำสำหรับโปรเซสที่ไม่ได้จัดประเภท โปรเซสที่มีอยู่ในเวลาที่เริ่มต้น WLM จะถูกจัดประเภทตามกฎการกำหนดของคอนฟิกูเรชัน WLM ที่โหลด ในระหว่างการจัดประเภทครั้งแรกนี้ หน้าหน่วยความจำทั้งหมดซึ่งแนบกับแต่ละโปรเซสจะถูก "เรียกเก็บ" ไปยังซูเปอร์คลาสที่โปรเซสเป็นสมาชิกอยู่ (เมื่อไม่ได้แบ่งใช้หรือแบ่งใช้โดยโปรเซสในซูเปอร์คลาสเดียวกัน) หรือไปยังซูเปอร์คลาสแบบแบ่งใช้เมื่อแบ่งใช้โดยโปรเซสในซูเปอร์คลาสต่างกัน

อย่างไรก็ตาม ไม่สามารถโยงบางหน้าเข้ากับโปรเซสใดโดยตรง (และเข้ากับคลาสใดๆ) ในเวลาของการจัดประเภทนี้ และหน่วยความจำส่วนนี้จะถูกเรียกเก็บไปยังซูเปอร์คลาสที่ไม่ได้จัดประเภท ส่วนใหญ่ของหน่วยความจำนี้จะถูกจัดประเภทอีกครั้งอย่างถูกต้องในภายหลัง เมื่อมีการเข้าถึงโดยโปรเซส หรืออาจถูกรีเส็ตและจัดสรรให้กับโปรเซสอีกครั้งหลังจาก WLM เริ่มต้นขึ้นแล้ว ไม่มีโปรเซสในซูเปอร์คลาสที่ไม่ได้จัดประเภทซูเปอร์คลาสนี้สามารถมีได้เฉพาะส่วนแบ่งใช้และขีดจำกัดหน่วยความจำฟิสิกส์ที่ใช้กับซูเปอร์คลาส และไม่สามารถมีส่วนแบ่งใช้หรือขีดจำกัดสำหรับรีซอร์สชนิดอื่น คลาสย่อย หรือกฎการกำหนดที่ระบุ

ซูเปอร์คลาสที่ไม่ได้จัดการ

มีการกำหนดซูเปอร์คลาสพิเศษที่ชื่อ *ที่ไม่ได้จัดการ* เสมอ ไม่มีการกำหนดโปรเซสให้กับคลาสนี้ คลาสนี้จะสะสมการใช้หน่วยความจำสำหรับหน้าที่ปักทั้งหมดในระบบซึ่งถูกจัดการโดย WLM การใช้ CPU ของโปรเซส `waitproc` ไม่มีการสะสมในคลาสใดๆ เพื่อป้องกันไม่ให้ระบบมีการใช้ 100% ซูเปอร์คลาสนี้ไม่สามารถมีส่วนแบ่งใช้หรือขีดจำกัดสำหรับรีซอร์ส ทุกชนิด คลาสย่อย หรือกฎการกำหนดที่ระบุ

คลาสย่อย:

ผู้ดูแลระบบหรือผู้ดูแลซูเปอร์คลาสสามารถกำหนดได้สูงสุด 61 คลาสย่อย

นอกจากนี้ มีการกำหนดสองคลาสย่อยพิเศษคือ *ดีพอลต์* และ *แบบแบ่งใช้* เสมอ

คลาสย่อย ดีพอลต์

เป็นคลาสย่อยดีพอลต์และมีการกำหนดเสมอ โปรเซสทั้งหมดที่ไม่ได้มีการกำหนดให้กับคลาสย่อยเฉพาะของซูเปอร์คลาสโดยอัตโนมัติจะถูกกำหนดให้กับคลาสย่อย *ดีพอลต์* คุณยังสามารถกำหนดโปรเซสอื่นให้กับคลาสย่อย *ดีพอลต์* ได้ ด้วยโดยการระบุกฎการกำหนดเฉพาะ

คลาสย่อย แบบแบ่งใช้

ได้รับหน้าหน่วยความจำทั้งหมดที่ใช้โดยโปรเซสในมากกว่าหนึ่งคลาสย่อยของซูเปอร์คลาส รวมถึงหน้าในพื้นที่หน่วยความจำแบบแบ่งใช้และหน้าในไฟล์ซึ่งใช้โดยโปรเซสในมากกว่าหนึ่งคลาสย่อยของซูเปอร์คลาสเดียวกัน หน่วยความจำและไฟล์แบบแบ่งใช้ที่ใช้โดยหลายโปรเซสซึ่งทั้งหมดเป็นสมาชิกของคลาสย่อยหนึ่ง จะมีการเชื่อมโยง

เข้ากับคลาสย่อยนั้น เฉพาะถ้า โพรเซสจากคลาสย่อยอื่นของซูเปอร์คลาสเดียวกันเข้าถึงพื้นที่หน่วยความจำ หรือไฟล์แบบแบ่งใช้ หนาจะถูกวางไว้ในคลาสย่อย แบบแบ่งใช้ของซูเปอร์คลาส ไม่มีโปรเซสในคลาสย่อย แบบแบ่งใช้ คลาสย่อยนี้ สามารถมีได้เฉพาะส่วนแบ่งใช้และขีดจำกัดหน่วยความจำฟิสิกัลที่ใช้กับคลาสย่อย และไม่สามารถมีส่วนแบ่งใช้หรือขีดจำกัดสำหรับรีซอร์ซชนิดอื่นหรือ กฎการกำหนดที่ระบุ การที่เช็กเมนต์หน่วยความจำซึ่งแบ่งใช้โดยโปรเซสในคลาสย่อยที่แตกต่างกันของซูเปอร์คลาสเดียวกันจะถูกจัดประเภทเข้าในคลาสย่อย แบบแบ่งใช้ หรือยังคงอยู่ในคลาสย่อยดั้งเดิม ขึ้นอยู่กับค่าของแอตทริบิวต์ localshm ของคลาสย่อยดั้งเดิม

แอตทริบิวต์คลาส:

แสดงรายการแอตทริบิวต์ทั้งหมดของคลาส WLM

ชื่อคลาส

ชื่อคลาสยาวได้สูงสุด 16 อักขระและสามารถมีตัวอักษรตัวพิมพ์ใหญ่และ ตัวพิมพ์เล็ก ตัวเลข และขีดเส้นใต้ (_) เท่านั้น

ระดับ ตัวเลขตั้งแต่ 0 ถึง 9 ที่ใช้เพื่อจัดลำดับความสำคัญในการจัดสรรรีซอร์ซ ระหว่างคลาสต่างๆ

การสืบทอด

ระบุว่าโปรเซสชาวนั้นได้รับสืบทอดการกำหนดคลาสมาจาก พาเรนต์หรือไม่

localshm

ป้องกันเช็กเมนต์หน่วยความจำที่เป็นของคลาสหนึ่งไม่ให้ย้าย ไปยังคลาสแบบแบ่งใช้

ผู้ดูแลระบบ (adminuser, admingroup, authgroup) (ซูเปอร์คลาสเท่านั้น)

มอบหมายหน้าที่การจัดการซูเปอร์คลาส

การอนุญาต (authuser, authgroup)

มอบหมายสิทธิในการกำหนดโปรเซสให้กับคลาสด้วยตนเอง

ชุดรีซอร์ซ (rset)

จำกัดชุดของรีซอร์ซซึ่งคลาสที่กำหนดมีสิทธิเข้าถึงได้ในแง่ของ CPUs (ชุดตัวประมวลผล)

delshm ลบเช็กเมนต์หน่วยความจำแบบแบ่งใช้ถ้าโปรเซสที่อ้างอิงล่าสุด ถูกทำลายเนื่องจากขีดจำกัดหน่วยความจำเสมือน

vmeforce

บ่งชี้ว่าจะทำลายโปรเซสทั้งหมดในคลาส หรือเฉพาะ โปรเซสที่ละเมิด เมื่อคลาสใช้งานถึงขีดจำกัดหน่วยความจำเสมือน

io_priority

ระบุลำดับความสำคัญที่กำหนดให้กับคำร้องขอ I/O ซึ่งออกใช้โดยเรดที่จัดประเภทเป็นคลาส ลำดับความสำคัญนี้ใช้เพื่อจัดลำดับความสำคัญของบัฟเฟอร์ I/O ที่ระดับอุปกรณ์ หากอุปกรณ์หน่วยเก็บไม่สนับสนุนการใช้ลำดับความสำคัญ I/O ลำดับความสำคัญจะถูกข้าม ลำดับความสำคัญ I/O ที่ถูกต้องอยู่ในช่วง ตั้งแต่ 0 ถึง 15

หลักการที่เกี่ยวข้อง:

“การกำหนดโปรเซสให้กับคลาสสำหรับการจัดการเวิร์กโหลด” ในหน้า 527

โปรเซสถูกกำหนดให้กับคลาส โดยใช้กฎการกำหนดคลาส ที่ระบุโดยผู้ดูแลระบบ เกณฑ์การจัดประเภทพิจารณาจาก ค่าของชุดแอตทริบิวต์ของโปรเซส เช่น ID ผู้ใช้, ID กลุ่ม ชื่อของไฟล์แอสพลีเคชัน ชนิดของโปรเซส และแอสพลีเคชันแท็ก

แอ็ททริบิวต์ระดับ:

ระดับแสดงถึงลำดับในการจัดสรรรีซอร์สระบบ ให้กับคลาส WLM

ผู้ดูแลระบบสามารถกำหนดคลาสได้มากถึง 10 ระดับ ซึ่งมีหมายเลขตั้งแต่ 0 ถึง 9 โดยที่ 0 เป็นระดับที่สูงที่สุดหรือสำคัญมากที่สุด จำนวนรีซอร์สที่มีอยู่ สำหรับระดับ 0 คือรีซอร์สระบบที่มีอยู่ทั้งหมด จำนวนรีซอร์สที่มีอยู่ สำหรับระดับที่ต่ำกว่า (หมายเลขสูงขึ้น) คือจำนวนรีซอร์สที่ไม่ได้ใช้ โดยระดับที่สูงกว่าทั้งหมด เพอร์เซ็นต์ปริมาณการใช้เป้าหมายสำหรับคลาส ขึ้นอยู่กับจำนวนของส่วนแบ่งใช้ที่ใช้งานอยู่ในระดับของคลาส และจำนวนรีซอร์ส ที่มีอยู่สำหรับระดับ เนื่องจากระดับ 0 เป็นเพียงระดับเดียวที่รับประกันได้ว่า มีรีซอร์สอยู่เสมอ จึงแนะนำว่าโปรเซสที่จำเป็นต้องการดำเนินงานของระบบควร ถูกจัดประเภทให้อยู่ในคลาสในระดับนี้ หากไม่มีการระบุระดับสำหรับคลาส คลาสจะถูกวางไว้ในระดับ 0

แอ็ททริบิวต์ระดับสามารถระบุได้ทั้งที่ระดับซูเปอร์คลาสและคลาสย่อย ระดับซูเปอร์คลาสใช้เพื่อระบุลำดับความสำคัญในการจัดสรรรีซอร์สระหว่าง ซูเปอร์คลาสต่างๆ ระดับคลาสย่อยใช้เพื่อระบุลำดับความสำคัญในการจัดสรรรีซอร์สระหว่าง คลาสย่อยของซูเปอร์คลาสเดียวกัน ไม่มีความสัมพันธ์ระหว่าง ระดับย่อยของซูเปอร์คลาสที่แตกต่างกัน

แอ็ททริบิวต์การสืบทอด:

แอ็ททริบิวต์ **inheritance** ของคลาส บ่งชี้ว่าโปรเซสในคลาสควรถูกจัดประเภทใหม่โดยอัตโนมัติหรือไม่ เมื่อแอ็ททริบิวต์การจัดประเภทอย่างใดอย่างหนึ่งของโปรเซสเปลี่ยนแปลง

เมื่อมีการสร้างโปรเซสใหม่ด้วยวิธีที่น้อย fork โปรเซสนั้นจะได้รับสืบทอดคลาสของพารেন্টโดยอัตโนมัติ ไม่ว่าจะการสืบทอดมีการ เปิดใช้งานหรือไม่ก็ตาม ข้อยกเว้นหนึ่งคือเมื่อโปรเซสพารেন্টมีแท็ก, **inherit tag at fork** มีการตั้งค่าเป็น off และการสืบทอดคลาสเป็น off สำหรับคลาสของพารেন্ট ในกรณีนี้ โปรเซสขายนจะถูกจัดประเภทใหม่ ตามกฎการจัดหมวดหมู่

เมื่อการสืบทอดไม่ได้เปิดใช้งานสำหรับคลาสโปรเซสใดๆ ในคลาสจะถูก จัดประเภทโดยอัตโนมัติตามกฎการจัดหมวดหมู่ หลังจากการเรียกเซอร์วิสใดๆ ที่เปลี่ยนแอ็ททริบิวต์โปรเซสซึ่งให้อยู่ในกฎ การเรียกที่ใช้กันมากที่สุด ในบรรดาการเรียกเหล่านี้คือรุ่นที่น้อย exec แต่รุ่นที่น้อยอื่นที่สามารถเปลี่ยนการจัดประเภทรวมถึง **setuid, setgid, plock, setpri,** และ **wlm_set_tag** เมื่อเปิดใช้งานการสืบทอด โปรเซสไม่ต้องการการจัดประเภทใหม่ ตามกฎการจัดหมวดหมู่ และจะยังคงคลาสปัจจุบันของโปรเซสไว้ การกำหนดค่าด้วยตนเองมีลำดับความสำคัญสูงกว่าการสืบทอด และสามารถใช้ในการจัดประเภทโปรเซสที่อยู่ในคลาสซึ่งมีการเปิดใช้งานการสืบทอดอีกครั้ง

ค่าที่ระบุสำหรับแอ็ททริบิวต์ **inheritance** อาจเป็น ใช่ หรือ ไม่ใช่ หากไม่ได้ระบุ การสืบทอดจะไม่เปิดใช้งานสำหรับ คลาส

แอ็ททริบิวต์นี้สามารถระบุได้ทั้งที่ระดับซูเปอร์คลาสและคลาสย่อย สำหรับคลาสย่อยของซูเปอร์คลาสที่กำหนด:

- หากแอ็ททริบิวต์ **inheritance** มีการตั้งค่าเป็น ใช่ ทั้งที่ ระดับซูเปอร์คลาสและคลาสย่อย ขายนของโปรเซสในคลาสย่อยจะยังคงอยู่ในคลาสย่อยเดียวกัน
- หากแอ็ททริบิวต์ **inheritance** มีการตั้งค่าเป็น ใช่ สำหรับ ซูเปอร์คลาส และเป็น ไม่ (หรือไม่ได้ระบุ) สำหรับคลาสย่อย ขายนของโปรเซส ในคลาสย่อยจะยังคงอยู่ในซูเปอร์คลาสเดียวกัน และจะถูกจัดประเภทอยู่ใน คลาสย่อยอย่างใดอย่างหนึ่งตามกฎการกำหนดสำหรับซูเปอร์คลาส
- หากแอ็ททริบิวต์ **inheritance** เป็น ไม่ (หรือไม่ได้ระบุ) สำหรับซูเปอร์คลาส และมีการตั้งค่าเป็น ใช่ สำหรับคลาสย่อย ขายนของโปรเซส ในคลาสย่อยจะถูกส่งไปยังกฎการกำหนดอัตโนมัติสำหรับ ซูเปอร์คลาส
 - ถ้าโปรเซสถูกจัดประเภทโดยกฎในซูเปอร์คลาสเดียวกัน โปรเซสนั้น จะยังคงอยู่ในคลาสย่อย (จะไม่ถูกส่งไปยังกฎการกำหนดของ คลาสย่อย)

- ถ้าโปรเซสถูกจัดประเภทโดยกฎของซูเปอร์คลาสในซูเปอร์คลาสที่แตกต่างกัน ระบบจะใช้กฎการกำหนดคลาสย่อยของซูเปอร์คลาสใหม่ เพื่อกำหนดคลาสย่อยของซูเปอร์คลาสใหม่ซึ่งโปรเซส จะถูกกำหนดให้
- ถ้าแอตทริบิวต์ **inheritance** ของทั้งซูเปอร์คลาสและคลาสย่อย มีการตั้งค่าเป็น **ไม่** (หรือไม่ได้ระบุ) ซายน์ของโปรเซสในคลาสย่อย จะถูกส่งไปยังการกำหนดอัตโนมัติมาตรฐาน

แอตทริบิวต์ localshm:

แอตทริบิวต์ **localshm** สามารถระบุได้ ทั้งที่ระดับซูเปอร์คลาสและคลาสย่อย

แอตทริบิวต์ **localshm** ใช้เพื่อป้องกันไม่ให้ย้ายเซ็กเมนต์ หน่วยความจำของคลาสหนึ่งไปยังซูเปอร์คลาสหรือคลาสย่อย **แบบแบ่งใช้** เมื่อเข้าถึงโดยโปรเซสในคลาสอื่น ค่าที่เป็นไปได้สำหรับ แอตทริบิวต์นี้คือ **ใช่** หรือ **ไม่ใช่** ค่า **yes** หมายความว่า เซ็กเมนต์หน่วยความจำแบบแบ่งใช้ในคลาสนี้ต้องยังคงเป็นแบบโลคัลสำหรับคลาส และไม่ถูกย้ายไปยังคลาส **แบบแบ่งใช้** ที่เหมาะสม ค่า **no** เป็น ค่าดีฟอลต์เมื่อไม่ได้ระบุแอตทริบิวต์

เซ็กเมนต์หน่วยความจำมีการจัดประเภทตามความผิดพลาดของเพจ เมื่อมีการสร้างเซ็กเมนต์ขึ้น เซ็กเมนต์จะถูกทำเครื่องหมายว่าเป็นสมาชิกของซูเปอร์คลาส **ที่ไม่ได้จัดประเภท** ในความผิดพลาดของเพจแรก บนเซ็กเมนต์ เซ็กเมนต์นี้ถูกจัดประเภทเข้าในคลาสเดียวกันกับ โปรเซส faulting ถ้าในเวลาต่อมา โปรเซสที่เป็นสมาชิกของคลาสอื่น ที่แตกต่างจากเซ็กเมนต์ page faults บนเซ็กเมนต์นี้ WLM จะพิจารณาว่า จำเป็นต้องจัดประเภทเซ็กเมนต์อีกครั้งเข้าในคลาส **แบบแบ่งใช้** (ซูเปอร์คลาสหรือคลาสย่อย) ที่เหมาะสมหรือไม่ หากโปรเซส faulting และเซ็กเมนต์เป็นสมาชิกของ ซูเปอร์คลาสที่แตกต่างกัน อย่างไรก็ดี อย่างหนึ่งต่อไปนี้เกิดขึ้น:

- ถ้าซูเปอร์คลาสของเซ็กเมนต์มีการตั้งค่าแอตทริบิวต์ **localshm** เป็น **yes** เซ็กเมนต์ยังคงอยู่ในซูเปอร์คลาสปัจจุบันของตน ถ้าคลาสย่อยของเซ็กเมนต์มีการตั้งค่าแอตทริบิวต์ **localshm** เป็น **yes** เซ็กเมนต์ยังคงอยู่ในคลาสย่อยปัจจุบันของตน ถ้าแอตทริบิวต์ **localshm** ของซูเปอร์คลาสมีการตั้งค่าเป็น **yes** แต่ แอตทริบิวต์ของคลาสย่อยมีการตั้งค่าเป็น **no** เซ็กเมนต์ จะไปอยู่ใน คลาสย่อย **แบบแบ่งใช้** ของซูเปอร์คลาสปัจจุบัน
- ถ้าซูเปอร์คลาสของเซ็กเมนต์มีการตั้งค่าแอตทริบิวต์ **localshm** เป็น **no** เซ็กเมนต์จะไปอยู่ในซูเปอร์คลาส **แบบแบ่งใช้** ค่านี้ เป็นการดำเนินการดีฟอลต์

ถ้าโปรเซส faulting และเซ็กเมนต์เป็นสมาชิกของคลาสย่อยอื่น ของซูเปอร์คลาสเดียวกัน และคลาสย่อยของเซ็กเมนต์มีการตั้งค่าแอตทริบิวต์ **localshm** เป็น **yes** เซ็กเมนต์ยังคงอยู่ในคลาสปัจจุบัน (ซูเปอร์คลาส และคลาสย่อย) มิฉะนั้น เซ็กเมนต์จะไปอยู่ในคลาสย่อย **แบบแบ่งใช้** ของ ซูเปอร์คลาส

แน่นอนว่า ถ้าโปรเซส faulting และเซ็กเมนต์เป็นสมาชิกของคลาสเดียวกัน (ซูเปอร์คลาสเดียวกันและคลาสย่อยเดียวกัน) จะไม่มีการจัดประเภทเซ็กเมนต์ อีกครั้ง ไม่ว่าค่าของแอตทริบิวต์ **localshm** เป็นอะไรก็ตาม

แอตทริบิวต์ผู้ดูแลระบบ:

แอตทริบิวต์ **adminuser** และ **admingroup** ใช้เพื่อมอบหมายการจัดการซูเปอร์คลาสให้กับผู้ใช้หรือกลุ่มผู้ใช้

หมายเหตุ: แอตทริบิวต์เหล่านี้ถูกต้องสำหรับซูเปอร์คลาสเท่านั้น

แอตทริบิวต์ **adminuser** ระบุชื่อของ ผู้ใช้ (ดังแสดงใน /etc/passwd) ที่ได้รับอนุญาตให้ทำภารกิจ การจัดการบนซูเปอร์คลาส
แอตทริบิวต์ **admingroup** ระบุชื่อของ กลุ่มผู้ใช้ (ดังแสดงใน /etc/group) ที่ได้รับอนุญาตให้ทำภารกิจ การจัดการบนซูเปอร์คลาส

ใช้ได้เพียงหนึ่งค่า (ผู้ใช้หรือกลุ่ม) เท่านั้นสำหรับแต่ละแอ็ททริบิวต์ สามารถระบุผู้ใช้หรือกลุ่ม อย่างใดอย่างหนึ่ง ไม่ระบุ หรือระบุทั้งสองอย่าง ผู้ใช้หรือกลุ่มจะมีสิทธิ ทำดังต่อไปนี้:

- สร้างและลบคลาสย่อย
- เปลี่ยนแอ็ททริบิวต์ และส่วนแบ่งใช้และขีดจำกัดรีซอร์สสำหรับคลาสย่อย
- กำหนด ลบ หรือแก้ไขกฎการกำหนดคลาสย่อย
- รีเฟรช (อัปเดต) คอนฟิกูเรชัน WLM ที่ใช้งานอยู่สำหรับซูเปอร์คลาส

แอ็ททริบิวต์การอนุญาต:

แอ็ททริบิวต์ **authuser** และ **authgroup** ถูกต้องสำหรับคลาสทั้งหมด แอ็ททริบิวต์เหล่านี้ใช้เพื่อระบุผู้ใช้หรือกลุ่มที่ได้รับอนุญาต ให้กำหนดโปรเซสให้กับคลาสด้วยตนเองได้ (ซูเปอร์คลาสหรือคลาสย่อย)

เมื่อกำหนดโปรเซส (หรือกลุ่มโปรเซส) ให้กับซูเปอร์คลาสด้วยตนเอง จะมีการใช้กฎการกำหนดสำหรับซูเปอร์คลาสเพื่อกำหนดคลาสย่อยของ ซูเปอร์คลาสที่จะได้รับการกำหนดแต่ละโปรเซส

ใช้ได้เพียงหนึ่งค่า (ผู้ใช้หรือกลุ่ม) เท่านั้นสำหรับแต่ละแอ็ททริบิวต์ สามารถระบุผู้ใช้หรือกลุ่ม อย่างใดอย่างหนึ่ง ไม่ระบุ หรือระบุทั้งสองอย่าง

แอ็ททริบิวต์ซุตรีซอร์ส:

แอ็ททริบิวต์ซุตรีซอร์ส (ที่เรียกว่า *rset*) สามารถระบุ สำหรับทุกคลาส ค่าเป็นชื่อของซุตรีซอร์สที่กำหนดโดย ผู้ดูแลระบบ

แอ็ททริบิวต์ *rset* แสดงถึงชุดย่อยของรีซอร์ส CPU ที่มีอยู่ บนระบบ (ชุดตัวประมวลผล) ค่าดีฟอลต์คือ "ระบบ" ซึ่งให้สิทธิเข้าถึงใน รีซอร์ส CPU ทั้งหมดที่มีอยู่บนระบบ ข้อจำกัดเพียงอย่างเดียวคือ ถ้ามีการระบุ *rset* สำหรับคลาสย่อย ชุดของ CPUs ในชุด ต้องเป็นชุดย่อยของ CPUs ที่มีอยู่บนซูเปอร์คลาส (สำหรับข้อมูลรายละเอียด ใหญ่ที่คำสั่ง **mkrset**)

หมายเหตุ: ควรพิจารณาการกำหนดซุตรีซอร์สให้กับคลาสใดๆ ที่ไม่ได้อยู่ในระดับ 0 อย่างรอบคอบ เนื่องจากระดับที่ต่ำกว่ามีสิทธิเข้าถึงเฉพาะรีซอร์สที่ไม่ได้ใช้โดย ระดับที่สูงกว่าเท่านั้น ดังนั้น การจำกัดคลาสที่ไม่ใช่ระดับ 0 สำหรับชุดย่อยของ CPUs บนระบบอาจส่งผลให้เกิดการขาดแคลนถ้าไม่มีเวลา CPU บน CPUs เหล่านั้น

การจัดประเภทโปรเซสใน Workload Manager

ใน WLM สามารถจัดประเภทโปรเซสได้สองวิธี

- โปรเซสถูกกำหนดโดยอัตโนมัติโดยใช้กฎการกำหนดเมื่อแอ็ททริบิวต์ การจัดประเภทโปรเซสเปลี่ยนไป เมื่อ WLM กำลังรันในโหมด active การกำหนด อัตโนมัตินี้มีผลบังคับใช้เสมอ (ไม่สามารถปิดได้) นี่เป็นวิธีที่ใช้กันมากที่สุดในการจัดประเภทโปรเซส
- โปรเซสหรือกลุ่มโปรเซสที่เลือกสามารถถูกกำหนดให้กับคลาส ด้วยตนเอง โดยผู้ใช้ที่มีสิทธิที่จำเป็นในทั้งโปรเซสและคลาสเป้าหมาย การกำหนดด้วยตนเองสามารถทำได้โดยใช้คำสั่ง WLM ซึ่ง สามารถเรียกใช้โดยตรง หรือผ่านทาง SMIT หรือโดยแอ็พพลิเคชันที่ไใช้ฟังก์ชันของ WLM Application Programming Interface การกำหนดด้วยตนเองนี้ยกเลิกการกำหนดอัตโนมัติและการสืบทอด

การกำหนดคลาสอัตโนมัติใน Workload Manager:

การกำหนดโปรเซสให้กับคลาสโดยอัตโนมัติใช้ชุดของ กฎการกำหนดคลาสที่ระบุโดยผู้ดูแลระบบ WLM

กฎการกำหนดมีอยู่สองระดับ:

- ชุดของกฎการกำหนดที่ระดับคอนฟิเกอรัล WLM ซึ่งใช้เพื่อกำหนด ซูเปอร์คลาสที่จะกำหนดโปรเซสให้
- ในลักษณะคล้ายกัน แต่ละซูเปอร์คลาสที่มีการกำหนดคลาสย่อยมีชุดของ กฎการกำหนดซึ่งใช้เพื่อกำหนดคลาสย่อยของซูเปอร์คลาสที่จะกำหนดโปรเซสให้

กฎการกำหนดที่ทั้งสองระดับสร้างขึ้นจากค่าของชุดของ แอ็ททริบิวต์โปรเซส แอ็ททริบิวต์เหล่านี้มีดังนี้:

- ID ผู้ใช้โปรเซส
- ID กลุ่มโปรเซส
- ชื่อพารามิเตอร์ของแอ็พพลิเคชัน (โปรแกรม) ที่ดำเนินการ
- ชนิดของโปรเซส (ตัวอย่างเช่น 32bit หรือ 64bit)
- โปรเซสแท็ก

แท็กเป็นแอ็ททริบิวต์โปรเซส ที่กำหนดเป็นสตริงอักขระซึ่งโปรแกรม สามารถตั้งค่าให้กับแอ็พพลิเคชันได้โดยใช้ WLM API

การจัดประเภทมีการทำในทุกครั้งที่แอ็ททริบิวต์เปลี่ยนไปโดยการเปรียบเทียบ ค่าของแอ็ททริบิวต์โปรเซสกับรายการของค่าที่เป็นไปได้ซึ่งกำหนดไว้ในไฟล์กฎการกำหนดคลาส (ที่เรียกว่า rules) การเปรียบเทียบ กำหนดว่ากฎใดตรงกับค่าปัจจุบันของแอ็ททริบิวต์โปรเซส

กฎการกำหนดคลาสเป็นสตริงข้อความที่ประกอบด้วยฟิลด์ต่อไปนี้ โดยแบ่งด้วยช่องว่างตั้งแต่หนึ่งช่องขึ้นไป:

ไอเท็ม	คำอธิบาย
ชื่อ	ต้องมีชื่อของคลาสซึ่งมีการกำหนดไว้ในไฟล์คลาส ที่สอดคล้องกับระดับของไฟล์ rules (ซูเปอร์คลาส หรือคลาสย่อย) ชื่อคลาสสามารถมีตัวอักษรตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก ตัวเลข และขีดเส้นใต้เท่านั้น และยาวได้สูงสุด 16 อักขระ ไม่สามารถระบุกฎการกำหนดสำหรับระบบที่มีการกำหนดคลาส ไม่ได้จัดประเภท ไม่ได้จัดการ และ แบ่งใช้
สงวนไว้	สงวนไว้สำหรับการขยายในอนาคต ค่าต้องเป็นยัติภังค์ (-) และ ต้องแสดงขึ้น
ผู้ใช้	สามารถมียัติภังค์ (-) หรือชื่อผู้ใช้ที่ถูกต้องอย่างน้อยหนึ่งชื่อ (ดังที่กำหนดในไฟล์ /etc/passwd) รายการประกอบด้วยชื่อตั้งแต่หนึ่งชื่อขึ้นไป โดยแบ่งด้วยเครื่องหมายจุลภาค (,) สามารถใส่เครื่องหมายอัศเจรีย์ (!) นำหน้า ชื่อเพื่อแยกผู้ใช้ที่กำหนดออกจากคลาส สามารถระบุรูปแบบ เพื่อจับคู่ชุดของชื่อผู้ใช้ โดยใช้ไวยากรณ์การจับคู่รูปแบบ Korn เซลล์ หากไม่มีชื่อผู้ใช้ที่ถูกต้อง กฎจะถูกละเว้น
กลุ่ม	สามารถมียัติภังค์ (-) หรือชื่อกลุ่มที่ถูกต้องอย่างน้อยหนึ่งชื่อ (ดังที่กำหนดในไฟล์ /etc/group) รายการประกอบด้วยชื่อตั้งแต่หนึ่งชื่อขึ้นไป โดยแบ่งด้วยเครื่องหมายจุลภาค (,) สามารถใส่เครื่องหมายอัศเจรีย์ (!) นำหน้า ชื่อเพื่อแยกกลุ่มที่กำหนดออกจากคลาส สามารถระบุรูปแบบ เพื่อจับคู่ชุดของชื่อกลุ่ม โดยใช้ไวยากรณ์การจับคู่รูปแบบ Korn เซลล์ หากไม่มีชื่อกลุ่มที่ถูกต้อง กฎจะถูกละเว้น
แอ็พพลิเคชัน	สามารถมียัติภังค์ (-) หรือรายการของชื่อพารามิเตอร์แอ็พพลิเคชัน นี่เป็นชื่อพารามิเตอร์แอ็พพลิเคชัน (โปรแกรม) ที่ดำเนินการโดยโปรเซส ซึ่งรวมอยู่ในคลาสชื่อแอ็พพลิเคชันจะเป็นชื่อพารามิเตอร์แบบเต็ม หรือรูปแบบ Korn เซลล์ที่จับคู่ชื่อพารามิเตอร์อย่างใดอย่างหนึ่ง รายการประกอบด้วย ชื่อพารามิเตอร์ตั้งแต่หนึ่งชื่อขึ้นไป โดยแบ่งด้วยเครื่องหมายจุลภาค (,) สามารถใส่เครื่องหมายอัศเจรีย์ (!) นำหน้า ชื่อเพื่อแยกแอ็พพลิเคชันที่กำหนด
	ต้องพบอย่างน้อยหนึ่งแอ็พพลิเคชัน ในรายการเมื่อโหลด มิฉะนั้น กฎจะถูกละเว้น กฎที่ถูกละเว้นเนื่องจาก เหตุผลนี้ในตอนแรกอาจมีผลบังคับใช้ในภายหลังได้ ถ้ามีการติดตั้งระบบไฟล์ ซึ่งมีหนึ่งหรือหลายแอ็พพลิเคชันในรายการ

ไอเท็ม
ชนิด

คำอธิบาย

สามารถมีขั้วติดกัน (-) หรือรายการของแอตทริบิวต์โปรเซส ค่าที่เป็นไปได้สำหรับแอตทริบิวต์เหล่านี้คือ:

- **32bit**: โปรเซสเป็นโปรเซส 32-บิต
- **64bit**: โปรเซสเป็นโปรเซส 64-บิต
- **plock**: โปรเซสที่เรียกกรูทีนย่อย **plock** เพื่อปกป้องหน่วยความจำ
- **fixed**: โปรเซสเป็นโปรเซสลำดับความสำคัญที่ระบุ (SCHED_FIFO หรือ SCHED_RR)

ชนิด **fixed** ใช้สำหรับวัตถุประสงค์การจัดประเภทเท่านั้น WLM ไม่ได้ควบคุมการใช้ตัวประมวลผลของโปรเซสลำดับความสำคัญที่ระบุ หรือเธรด เนื่องจากโปรเซสลำดับความสำคัญที่ระบุอาจทำให้โปรเซสอื่นในคลาสเกิดความขาดแคลน จึงมีการนำเสนอแอตทริบิวต์การจัดประเภทนี้ เพื่อให้สามารถแยกงานเหล่านี้ได้ แอตทริบิวต์นี้ยังสามารถใช้เพื่อรายงานปริมาณการใช้ของโปรเซสดังกล่าวได้ด้วย

แท็ก

ค่าของฟิลด์ **type** อาจเป็นส่วนผสมของแอตทริบิวต์ข้างบนตั้งแต่หนึ่งแอตทริบิวต์ขึ้นไป โดยแบ่งด้วย เครื่องหมายบวก (+) ค่า **32bit** และ **64bit** ต้องไม่เกิดขึ้นพร้อมกัน สามารถมีขั้วติดกัน (-) หรือรายการของแอ็พพลิเคชันแท็ก แอ็พพลิเคชันแท็ก เป็นสตริงของตัวอักษรและตัวเลขสูงสุด 30 อักขระ รายการประกอบด้วย ค่าแอ็พพลิเคชันแท็กตั้งแต่หนึ่งค่าขึ้นไป โดยแบ่งด้วยเครื่องหมายจุลภาค

แอตทริบิวต์ ผู้ใช้กลุ่ม แอ็พพลิเคชัน และ แท็ก สามารถใช้เป็นการจัดกลุ่ม ค่าแอตทริบิวต์ได้

เมื่อโปรเซสถูกสร้างขึ้น (fork) โปรเซสยังคงอยู่ในคลาสเดียวกันกับพารেন্ট การจัดประเภทใหม่เกิดขึ้นเมื่อโปรเซสใหม่ออกใช้การเรียกระบบซึ่งสามารถ แกะไขแอตทริบิวต์อย่างใดอย่างหนึ่งของโปรเซสที่ใช้สำหรับการจัดประเภท ตัวอย่าง เช่น **exec**, **setuid** (และการเรียกที่เกี่ยวข้อง), **setgid** (และการเรียกที่เกี่ยวข้อง), **setpri** และ **plock**

ในการจัดประเภทโปรเซส WLM จะตรวจสอบไฟล์ **rules** ระดับบนสุด เพื่อดูคอนฟิกูเรชันที่ใช้งานอยู่เพื่อกำหนดว่าโปรเซสเป็นสมาชิกของซูเปอร์คลาสใด สำหรับกฎแต่ละข้อในไฟล์ WLM จะตรวจสอบค่าปัจจุบันของแอตทริบิวต์โปรเซส โดยเปรียบเทียบกับค่าและรายการของค่าที่ระบุในกฎ การตรวจสอบกฎ เรียงตามลำดับที่ปรากฏขึ้นในไฟล์ เมื่อพบรายการที่ตรงกัน โปรเซสจะถูกกำหนดให้กับซูเปอร์คลาสที่มีชื่ออยู่ในฟิลด์แรกของกฎ จากนั้น ตรวจสอบ ฟิลด์กฎของซูเปอร์คลาสในวิธีเดียวกัน เพื่อกำหนดคลาสย่อย ซึ่งควรจะกำหนดโปรเซสให้

โปรเซสจะตรงกับกฎข้อใดข้อหนึ่ง ถ้าแอตทริบิวต์แต่ละรายการของโปรเซสตรงกับฟิลด์ที่สอดคล้องกันในกฎ ข้อมูลต่อไปนี้ เป็นรายการของเกณฑ์ที่ใช้เพื่อกำหนดว่า ค่าของแอตทริบิวต์ตรงกับค่าใน ฟิลด์ของฟิลด์ **rules** หรือไม่:

- ถ้าฟิลด์ในไฟล์กฎมีค่าเป็นขั้วติดกัน (-) ค่าใดๆ ของ แอตทริบิวต์โปรเซสที่สอดคล้องกันจะเป็นรายการที่ตรงกัน
- สำหรับแอตทริบิวต์ทั้งหมดยกเว้น **type** ถ้าค่าของ แอตทริบิวต์โปรเซสตรงกับค่าใดค่าหนึ่งในรายการในฟิลด์กฎที่ไม่ได้แยก (นำหน้าด้วย "!") ถือว่าการตรงกันเกิดขึ้น
- สำหรับแอตทริบิวต์ **type** ถ้าค่าใดค่าหนึ่ง ในกฎประกอบด้วยค่าตั้งแต่สองค่าขึ้นไปและแบ่งด้วยเครื่องหมายบวก (+) โปรเซสจะตรงกันเฉพาะถ้าลักษณะของโปรเซสตรงกับค่าทั้งหมด

ทั้งที่ระดับซูเปอร์คลาสและคลาสย่อย WLM ตรวจสอบกฎโดยเรียงตามลำดับ ที่ปรากฏขึ้นในไฟล์ **rules** และจัดประเภท โปรเซสในคลาสที่สอดคล้องกับกฎแรกซึ่งโปรเซส ตรงกัน ด้วยเหตุนี้ ลำดับของกฎในไฟล์กฎจึงมีความสำคัญ มาก โปรดใช้ความระมัดระวังเมื่อคุณสร้างหรือแก้ไขไฟล์กฎ

การกำหนดคลาสด้วยตนเองใน Workload Manager:

โปรเซสหรือกลุ่มของโปรเซสสามารถถูกกำหนดให้กับซูเปอร์คลาส และ/หรือคลาสย่อยด้วยตนเองได้โดยใช้ SMIT หรือคำสั่ง **wlmassign**

โปรดดูที่ `wlmassign` สำหรับข้อมูลเพิ่มเติม แอ็พพลิเคชัน สามารถกำหนดโปรเซสโดยใช้ฟังก์ชัน `wlm_assign` API

เพื่อกำหนดโปรเซสให้กับคลาสด้วยตนเอง หรือเพื่อยกเลิกการกำหนดด้วยตนเอง ที่มีอยู่ ผู้ใช้ต้องมีเอกสิทธิ์ในระดับที่เหมาะสม การกำหนดด้วยตนเอง สามารถทำหรือยกเลิกแยกต่างหากกันที่ระดับซูเปอร์คลาส ระดับคลาสย่อย หรือทั้งสองระดับ การกำหนดนี้มีการระบุโดยแฟล็กสำหรับอินเตอร์เฟซ การเขียนโปรแกรม และชุดของอ็อปชันสำหรับอินเตอร์เฟซบรรทัดคำสั่งที่ใช้โดยเครื่องมือการจัดการ WLM ดังนั้น สามารถกำหนดโปรเซสด้วยตนเองให้กับ ซูเปอร์คลาสอย่างเดียว คลาสย่อยอย่างเดียว หรือให้กับซูเปอร์คลาสและคลาสย่อยของซูเปอร์คลาสนั้น ในกรณีหลังสุด การกำหนดคู่สามารถทำพร้อมกัน (ด้วยคำสั่งหรือการเรียก API เดียว) หรือในเวลาที่แตกต่างกันโดยผู้ใช้ที่แตกต่างกัน

การกำหนดมีความยืดหยุ่นมาก แต่สามารถทำให้เกิดความสับสนได้ ข้อมูลต่อไปนี้ เป็นสองตัวอย่าง ของกรณีที่เป็นไปได้

ตัวอย่างที่ 1: การกำหนดครั้งแรกของโปรเซส

ผู้ดูแลระบบกำหนด `Process1` จาก `superclassA` ให้กับ `superclassB` ด้วยตนเอง (การกำหนดระดับซูเปอร์คลาสอย่างเดียว) กฎการกำหนดอัตโนมัติสำหรับคลาสย่อยของ `superclassB` จะ ถูกใช้โดย WLM เพื่อกำหนดคลาสย่อยสุดท้าย ซึ่งจะกำหนดโปรเซสให้ `Process1` ถูก กำหนดให้กับ `superclassB.subclassA` และมีแฟล็กว่า มีการกำหนด "ซูเปอร์คลาสอย่างเดียว"

ผู้ใช้ที่มีเอกสิทธิ์ที่เหมาะสม กำหนด `Process2` จากคลาสปัจจุบัน `superclassA.subclassA` ให้กับ คลาสย่อยใหม่ในซูเปอร์คลาสเดียวกัน `superclassA.subclassB` `Process2` ถูก กำหนดให้กับคลาสย่อยใหม่และมีแฟล็กว่ามีการกำหนด "คลาสย่อยอย่างเดียว"

ผู้ดูแลระบบ WLM ของคลาสย่อยของ `superclassB` กำหนด `Process1` อีกครั้งด้วยตนเองให้กับ `subclassC` ซึ่งเป็นอีกคลาสย่อยหนึ่งของ `superclassB` `Process1` ถูก จัดประเภทใหม่เป็น `superclassB.subclassC` และขณะนี้ มีการแฟล็กว่ามีการกำหนดทั้งระดับซูเปอร์คลาสและคลาสย่อย

ตัวอย่างที่ 2: การกำหนดใหม่หรือการยกเลิกการกำหนดด้วยตนเอง

การกำหนดใหม่และการยกเลิกการกำหนดด้วยตนเองที่ระดับคลาสย่อย มีความซับซ้อนน้อย และมีผลต่อการกำหนดระดับคลาสย่อยเท่านั้น

สมมติว่า ผู้ดูแลระบบต้องการให้ `Process2` อยู่ใน ซูเปอร์คลาสที่มีรีซอร์สมากขึ้น และตัดสินใจกำหนด `Process2` ให้กับ `superclassC` ด้วยตนเอง ในตัวอย่างที่ 1 `Process2` ถูกกำหนดให้กับ `subclassB` ของ `superclassA` ด้วยตนเอง โดยมีการกำหนด "คลาสย่อยอย่างเดียว" เนื่องจาก `Process2` ถูก กำหนดให้กับซูเปอร์คลาสอื่น การกำหนดด้วยตนเองที่ทำก่อนหน้านี้จึง ไม่มีความหมายและถูกยกเลิก `Process2` ขณะนี้ มีการกำหนด ด้วยตนเอง "ซูเปอร์คลาสอย่างเดียว" ให้กับ `superclassC` และถ้าไม่มีการสืบทอด จะถูกกำหนดให้กับคลาสย่อย `superclassC` โดยใช้กฎการกำหนดอัตโนมัติ

ขณะนี้ ผู้ดูแลระบบตัดสินใจ ที่จะยุติการกำหนดด้วยตนเองจาก `Process1` ให้กับ `superclassB` การกำหนดด้วยตนเอง "ระดับซูเปอร์คลาส" ของ `Process1` ถูก ยกเลิก และถ้าไม่มีการสืบทอด `Process1` จะถูกกำหนด ให้กับซูเปอร์คลาส โดยใช้กฎการกำหนดอัตโนมัติระดับบนสุด

หาก กฎยังไม่เปลี่ยนแปลง `Process1` จะถูกกำหนดให้ กับ `superclassA` และการกำหนดด้วยตนเองระดับคลาสย่อย ให้กับ `superclassB.subclassC` กลายเป็นไม่มีความหมายและ ถูกยกเลิก

ถ้าด้วยเหตุผลบางอย่าง กฎระดับบนสุดกำหนด *Process1* ให้กับ *superclassB* การกำหนดระดับคลาสย่อยให้กับ *superclassB.subclassC* ยังคง ถูกต้องและมีผลบังคับใช้ ขณะนี้ *Process1* มีการกำหนดด้วยตนเอง " คลาสย่อยอย่างเดียว"

อัปเดตใน Workload Manager:

เมื่อมีการอัปเดต WLM (โดยใช้คำสั่ง `wlmcntrl -u`) คอนฟิกูเรชันที่อัปเดตสามารถโหลดกฎการจัดหมวดหมู่ชุดใหม่ได้

เมื่อเกิดกรณีเช่นนี้ โพรเซสมักถูกจัดประเภทใหม่โดยใช้กฎใหม่ WLM ไม่ได้จัดประเภทโพรเซสที่กำหนดด้วยตนเองหรือที่อยู่ใน คลาสซึ่งเปิดใช้งานการสืบทอดซ้ำอีกครั้ง ยกเว้นว่าคลาสของโพรเซสนั้น ไม่มีอยู่ในคอนฟิกูเรชันใหม่

ข้อควรพิจารณาด้านความปลอดภัยสำหรับ Workload Manager:

เพื่อกำหนดโพรเซสให้กับคลาส หรือเพื่อยกเลิกการกำหนดด้วยตนเองก่อน หน้าที่ ผู้ใช้ต้องมีสิทธิ์ทั้งบนโพรเซสและบนคลาส เป้าหมาย

ข้อจำกัดเหล่านี้แปลเป็นกฎต่อไปนี้:

- ผู้ใช้สามารถกำหนดทุกโพรเซสให้กับคลาสใดก็ได้
- ผู้ใช้ที่มีเอกสิทธิ์การจัดการในคลาสย่อยของซูเปอร์คลาสที่กำหนด (นั่นคือ ชื่อผู้ใช้หรือกลุ่มตรงกับชื่อผู้ใช้หรือกลุ่มที่ระบุ ในแอตทริบิวต์ `adminuser` และ `admingroup` ของซูเปอร์คลาส) สามารถกำหนดทุกโพรเซสอีกครั้งด้วยตนเองจากคลาสย่อยอย่างใดอย่างหนึ่งของซูเปอร์คลาสนี้ให้กับคลาสย่อยอื่นของซูเปอร์คลาส
- ผู้ใช้สามารถกำหนดโพรเซสของตนเองด้วยตนเอง (โพรเซสที่เชื่อมโยงกับ ID ผู้ใช้จริงหรือที่ใช้อยู่เดียวกัน) ให้กับคลาสย่อยซึ่งผู้ใช้มีเอกสิทธิ์การกำหนดด้วยตนเอง (นั่นคือ ชื่อผู้ใช้หรือกลุ่มตรงกับชื่อผู้ใช้หรือกลุ่มที่ระบุ ในแอตทริบิวต์ `authuser` และ `authgroup` ของ ซูเปอร์คลาสหรือคลาสย่อย)

เพื่อแก้ไขหรือยุติการกำหนดด้วยตนเอง ผู้ใช้ต้องมีเอกสิทธิ์ในระดับเดียวกันกับ บุคคลที่ออกใช้การกำหนดด้วยตนเองล่าสุด เป็นอย่างน้อย

การจัดการรีซอร์สด้วย Workload Manager

WLM มอนิเตอร์และควบคุมการใช้รีซอร์สสำหรับแต่ละคลาส ของเรดและโพรเซสที่ใช้งานอยู่บนระบบ คุณสามารถตั้งค่าขีด จำกัดต่ำสุด หรือสูงสุดต่อคลาสสำหรับรีซอร์สแต่ละชนิดที่จัดการโดย WLM ตลอดจน ค่าเป้าหมายต่อคลาสสำหรับรีซอร์สแต่ละรายการ

เป้าหมายนี้เป็นตัวแทนของจำนวนรีซอร์สที่เหมาะสมที่สุด สำหรับงานในคลาส ส่วนแบ่งใช้และขีดจำกัดที่ระดับซูเปอร์คลาส หมายถึงจำนวนทั้งหมดของรีซอร์สแต่ละรายการที่มีอยู่บนระบบ ที่ระดับคลาสย่อย ส่วนแบ่งใช้และขีดจำกัดหมายถึงจำนวนของรีซอร์สแต่ละรายการ ที่มีอยู่ในซูเปอร์คลาสซึ่งคลาสย่อยเป็นสมาชิกอยู่ (เป้าหมายซูเปอร์คลาส) ลำดับชั้นของคลาสเป็นวิธีในการแบ่งรีซอร์สระบบระหว่างกลุ่มของผู้ใช้ (ซูเปอร์คลาส) และมอบหมายหน้าที่การจัดการส่วนแบ่งใช้ของรีซอร์สนี้ให้กับผู้ดูแลซูเปอร์คลาส จากนั้น ผู้ดูแลซูเปอร์คลาสแต่ละรายสามารถ แจกจ่ายรีซอร์สจำนวนนี้อีกครั้งระหว่างผู้ใช้ในกลุ่ม โดยการสร้างคลาสย่อยและการกำหนดสิทธิ์ในรีซอร์สสำหรับคลาสย่อยเหล่านี้

ชนิดรีซอร์สใน Workload Manager:

WLM จัดการรีซอร์สสามชนิดโดยคำนวณเป็นเปอร์เซ็นต์ ปริมาณการใช้

ไอเท็ม การใช้ CPU ของเซิร์ฟเวอร์ในคลาส การใช้หน่วยความจำฟิสิกส์ของโปรเซสใน คลาส ดิสก์ I/O แบบดวิธของคลาส	คำอธิบาย นี่เป็นผลรวมของวงจร CPU ทั้งหมดที่ใช้โดยทุกเซิร์ฟเวอร์ในคลาส นี่เป็นผลรวมของหน่วยความจำทั้งหมดที่เป็นสมาชิกของโปรเซสในคลาส นี่เป็นแบนด์วิธ (ในบล็อก 512-ไบต์ต่อวินาที) ของ I/Os ทั้งหมดที่เริ่มต้น โดยเซิร์ฟเวอร์ในคลาสบนอุปกรณ์ดิสก์ แต่ละรายการที่เข้าถึงโดยคลาส
---	--

ทุกวินาที WLM คำนวณการใช้ต่อคลาสสำหรับรีซอร์สแต่ละรายการ ในระหว่างวินาทีล่าสุด เป็นเปอร์เซ็นต์ของรีซอร์สทั้งหมดที่มีอยู่ ดังนี้:

- สำหรับ CPU จำนวนทั้งหมดของเวลา CPU ที่มีอยู่ในทุกวินาทีเท่ากับ 1 วินาทีคูณจำนวน CPUs บนระบบ ตัวอย่างเช่น บน SMP แปรทาง ถ้าเซิร์ฟเวอร์ทั้งหมดของคลาสใช้เวลา CPU รวม 2 วินาที ในระหว่างวินาทีล่าสุด จะแสดงเป็นเปอร์เซ็นต์ได้ดังนี้ $2/8 = 25\%$ เปอร์เซ็นต์ที่ใช้โดย WLM สำหรับการควบคุมเป็นค่าเฉลี่ยลดลงในช่วงเวลาสองสามวินาที ของการใช้รีซอร์สต่อวินาที "ณ ขณะหนึ่ง" นี้
- สำหรับหน่วยความจำฟิสิกส์ จำนวนทั้งหมดของหน่วยความจำฟิสิกส์ที่มีอยู่สำหรับ โปรเซส ณ เวลาที่กำหนดใดๆ เท่ากับจำนวนทั้งหมดของหน่วยความจำทาง ฟิสิกส์ที่มีอยู่บนระบบ ลบจำนวนของหน่วยความจำที่ปัก หน้าที่ยังไม่ได้ถูกจัดการโดย WLM เนื่องจากหน้าเหล่านี้สามารถถูกขโมยจากคลาสและมอบให้กับ คลาสอื่นเพื่อควบคุมการใช้หน่วยความจำ การใช้หน่วยความจำของ คลาสเป็นอัตราส่วนของจำนวนของหน่วยความจำที่ไม่ได้ปักซึ่งเป็นของ โปรเซสทั้งหมดในคลาสต่อจำนวนหน่วยความจำที่มีอยู่บนระบบ ซึ่งระบุเป็นเปอร์เซ็นต์
- สำหรับดิสก์ I/O ปัญหาหลักคือการกำหนดแบนด์วิธที่มีอยู่และเป็นประโยชน์ สำหรับอุปกรณ์ เมื่อดิสก์อยู่ 100% ปริมาณงานของดิสก์ ในหน่วยบล็อกต่อวินาที มีความแตกต่างกันมากระหว่างแอพลิเคชันหนึ่งที่กำลังทำ I/Os ตามลำดับเมื่อเปรียบเทียบกับหลายแอพลิเคชันที่กำลังสร้าง I/Os แบบสุ่ม หากคุณใช้ปริมาณงานสูงสุดที่วัดสำหรับ กรณี I/O ตามลำดับ (เป็นค่าของ I/O แบบดวิธที่มีอยู่สำหรับ อุปกรณ์) เพียงอย่างเดียวในการคำนวณเปอร์เซ็นต์การใช้ของอุปกรณ์ภายใต้ I/Os แบบสุ่ม อาจทำให้คุณเข้าใจผิดว่าอุปกรณ์มีการใช้ 20% แต่ในข้อเท็จจริงมีการใช้ 100%

เพื่อให้ได้ค่าเปอร์เซ็นต์ที่ต้องการและน่าเชื่อถือมากขึ้นของการใช้ดิสก์ต่อคลาส WLM จะใช้สถิติที่จัดเตรียมโดยดิสก์ไดรเวอร์ (ที่แสดงโดยใช้คำสั่ง `iostat`) ซึ่งให้ค่าเปอร์เซ็นต์ของเวลาที่อุปกรณ์ยุ่งในระหว่างวินาทีล่าสุดสำหรับ อุปกรณ์ดิสก์แต่ละรายการ WLM นับจำนวนบล็อกทั้งหมดซึ่งถูกอ่าน หรือเขียนบนอุปกรณ์ในระหว่างวินาทีล่าสุดโดยคลาสทั้งหมดที่เข้าถึง อุปกรณ์ และจำนวนบล็อกที่อ่านหรือเขียนโดยแต่ละคลาส และเปอร์เซ็นต์ของการใช้ของอุปกรณ์ จากนั้น WLM คำนวณ เปอร์เซ็นต์ของปริมาณงานดิสก์ที่ใช้โดยแต่ละคลาส

ตัวอย่างเช่น ถ้าจำนวนบล็อกทั้งหมดที่อ่านหรือเขียนในระหว่าง วินาทีล่าสุดคือ 1000 และอุปกรณ์ยุ่ง 70% นี้หมายความว่า คลาส ที่อ่านหรือเขียน 100 บล็อกใช้ 7% ของดิสก์แบนด์วิธ ในลักษณะคล้ายกับ เวลา CPU (รีซอร์สที่ใช้ซ้ำได้อีกรายการหนึ่ง) ค่าที่ใช้โดย WLM สำหรับการควบคุมดิสก์ I/O เป็นค่าเฉลี่ยลดลงในช่วงสองสามวินาทีของเปอร์เซ็นต์ ต่อวินาทีเหล่านี้เช่นกัน

สำหรับรีซอร์สดิสก์ I/O ส่วนแบ่งใช้และขีดจำกัดใช้กับอุปกรณ์ดิสก์แต่ละรายการ ที่เข้าถึงโดยคลาส การควบคุมมีการทำอย่างเป็นอิสระสำหรับ แต่ละอุปกรณ์ ซึ่งหมายความว่าคลาสอาจใช้รีซอร์สมากกว่าสิทธิที่ได้รับบนอุปกรณ์หนึ่ง และ I/Os ไปยังอุปกรณ์นี้จะถูกควบคุม ในขณะที่ใช้น้อยกว่าสิทธิที่ได้รับบน อีกดิสก์หนึ่งและ I/Os ไปยังอุปกรณ์อื่นนี้จะไม่ถูก จำกัด

WLM supports accounting and regulation of resources on the basis of total consumption. มีรีซอร์สสองชนิดที่สามารถควบคุมในลักษณะนี้ได้คือ: ผลรวมคลาสและผลรวมโปรเซส

ผลรวมคลาส

สามารถระบุขีดจำกัดต่อคลาสสำหรับจำนวนของโปรเซส เซิร์ฟเวอร์ และล็อกอินเซสชันในคลาส ขีดจำกัดเหล่านี้มีการระบุเป็นตัวเลขสัมบูรณ์ สำหรับรีซอร์สแต่ละรายการที่สามารถมีอยู่ในคลาส ณ ขณะหนึ่ง ขีดจำกัดเหล่านี้ มีการบังคับใช้

อย่างเข้มงวด เมื่อคลาสใช้งานถึงขีดจำกัดสำหรับรีซอร์สอย่างใดอย่างหนึ่งเหล่านี้ ความพยายามใดๆ ในการสร้างอินสแตนซ์อื่นของรีซอร์สจะล้มเหลว การดำเนินงานจะยังคงล้มเหลวต่อไปสำหรับโปรเซสใดๆ ในคลาสจนกว่าคลาสใช้รีซอร์สต่ำกว่าขีดจำกัดที่ระบุ

ผลรวมโปรเซส

สามารถระบุขีดจำกัดต่อโปรเซสสำหรับจำนวนทั้งหมดของเวลา CPU, บล็อกของดิสก์ I/O, และเวลาเชื่อมต่อสำหรับบล็อกอินเซชัน ขีดจำกัดเหล่านี้มีการระบุที่ระดับคลาส แต่ใช้กับแต่ละโปรเซสในคลาสที่ละโปรเซส (แต่ละโปรเซสสามารถใช้จำนวนนี้ได้) ค่าปริมาณการใช้เหล่านี้เป็นค่าสะสม และแสดงถึงจำนวนทั้งหมดของรีซอร์สแต่ละรายการที่ถูกใช้โดยโปรเซสในระหว่างอายุการใช้งานของโปรเซส หลังจากโปรเซสใช้เกินขีดจำกัดทั้งหมดสำหรับรีซอร์สใดๆ โปรเซสจะถูกยุติลง จะมีการส่งสัญญาณ SIGTERM ไปยังโปรเซส และถ้าโปรเซสดักจับสัญญาณนี้และไม่ออก หลังจากระยะเวลาพอนันานาน 5 วินาที จะมีการส่งสัญญาณ SIGKILL เมื่อ ล็อกอินเซชันใช้งานถึง 90% ของขีดจำกัดเวลาเชื่อมต่อ จะมีการเขียนข้อความ คำเตือนไปยังเทอร์มินัลการควบคุมเพื่อเตือนผู้ว่า เซชันจะถูกยุติในไม่ช้า

ส่วนแบ่งใช้เป้าหมายใน Workload Manager:

เปอร์เซ็นต์ปริมาณการใช้รีซอร์สเป้าหมาย (หรือที่ต้องการ) สำหรับคลาส ถูกกำหนดโดยจำนวนส่วนแบ่งใช้ของรีซอร์สเฉพาะสำหรับคลาสนั้น

ส่วนแบ่งใช้แสดงถึงจำนวนของรีซอร์สเฉพาะซึ่งคลาสควรจะได้รับ โดยสัมพันธ์กับคลาสนั้นในระดับ เปอร์เซ็นต์เป้าหมายของคลาสสำหรับ รีซอร์สเฉพาะมีการคำนวณแบบง่ายเป็น จำนวนส่วนแบ่งใช้หารด้วยจำนวน ของส่วนแบ่งใช้ที่ใช้งานอยู่ในระดับ หากมีการใช้ขีดจำกัดด้วย เป้าหมายจะถูก จำกัดเป็นช่วง [ค่าต่ำสุด, ค่าสูงสุดซอฟต์แวร์] ถ้าเป้าหมายที่คำนวณได้อยู่ในช่วงนี้ เป้าหมาย จะถูกกำหนดเป็นขอบเขตสูงสุด/ต่ำสุดที่เหมาะสม (โปรดดูที่ขีดจำกัดรีซอร์ส) จำนวนของส่วนแบ่งใช้ที่ใช้งานอยู่คือจำนวนส่วนแบ่งใช้ทั้งหมดของคลาสทั้งหมดที่มีโปรเซสซึ่งใช้งานอยู่อย่างน้อยหนึ่งโปรเซสในชั้น เนื่องจากจำนวนของส่วนแบ่งใช้ที่ใช้งานอยู่ เป็นแบบไดนามิก ดังนั้นจึงเป็นเป้าหมาย หากคลาสเป็นคลาสที่ใช้งานอยู่เพียงคลาสเดียวในระดับ เป้าหมายจะเป็น 100% ของจำนวนรีซอร์สที่มีอยู่ในระดับนั้น

ตัวอย่างเช่น สมมติว่ามีคลาสซูเปอร์คลาสที่ใช้งานอยู่สามคลาสในระดับ 0-A, B, และ C โดยมีส่วนแบ่งใช้สำหรับรีซอร์สเฉพาะเป็น 15, 10, และ 5, ตามลำดับ เป้าหมายจะเป็น:

$$\text{target(A)} = 15/30 = 50\%$$

$$\text{target(B)} = 10/30 = 33\%$$

$$\text{target(C)} = 5/30 = 17\%$$

ถ้าบางครั้งในเวลาต่อมา คลาส B เปลี่ยนเป็นไม่ได้ใช้งาน (ไม่มีโปรเซสที่ใช้งานอยู่) เป้าหมายของคลาส A และ C จะถูกปรับโดยอัตโนมัติ:

$$\text{target(A)} = 15/20 = 75\%$$

$$\text{target(C)} = 5/20 = 25\%$$

ดังที่คุณเห็น ส่วนแบ่งใช้แสดงถึงเปอร์เซ็นต์ที่ปรับเองซึ่งช่วยให้สามารถ จัดสรรรีซอร์สให้กับคลาสได้อย่างเท่าเทียมกัน หรือสามารถนำรีซอร์สจากคลาสนั้น มาใช้เมื่อคลาสนั้นใช้งาน/ไม่ได้ใช้งาน

เพื่อให้มีความยืดหยุ่นสูง จำนวนส่วนแบ่งใช้สำหรับคลาสสามารถเป็น ตัวเลขใดๆ ระหว่าง 1 ถึง 65535 ส่วนแบ่งใช้สามารถระบุได้สำหรับซูเปอร์คลาส และคลาสย่อย สำหรับซูเปอร์คลาส ส่วนแบ่งใช้สัมพันธ์กับซูเปอร์คลาสที่ใช้งานอยู่อื่น ทั้งหมดใน

ระดับเดียวกัน สำหรับคลาสย่อย ส่วนแบ่งใช้สัมพันธ์กับ คลาสย่อยที่ใช้งานอยู่อื่นทั้งหมดในซูเปอร์คลาสเดียวกัน ในระดับเดียวกัน ส่วนแบ่งใช้สำหรับคลาสย่อยของซูเปอร์คลาสหนึ่งไม่มีความสัมพันธ์กับส่วนแบ่งใช้สำหรับ คลาสย่อยของอีกซูเปอร์คลาสหนึ่ง

ในบางกรณี อาจต้องการทำให้เป้าหมายของคลาสเป็นอิสระจาก จำนวนของส่วนแบ่งใช้ที่ใช้งานอยู่ เมื่อต้องการทำเช่นนี้ สามารถระบุค่า “-” สำหรับจำนวนส่วนแบ่งใช้ในกรณีนี้ จะไม่มีการควบคุมการใช้รีซอร์สนั้นในคลาส ซึ่งหมายความว่าไม่มีส่วนแบ่งใช้ และเป้าหมายไม่ได้ขึ้นอยู่กับ จำนวนของส่วนแบ่งใช้ที่ใช้งานอยู่ เป้าหมายจะถูกตั้งค่าเป็น (รีซอร์สที่มีอยู่ในระดับ - ผลรวมของค่าต่ำสุดสำหรับคลาสอื่นทั้งหมดในระดับ) เป้าหมายนี้ หรือปริมาณการใช้จริง (ค่าใดก็ตามที่ต่ำกว่า) คือปริมาณที่เหลืออยู่หลังจากการใช้ ของคลาสอื่นในระดับเดียวกัน

ตัวอย่างเช่น สมมติว่าคลาส A, B, C, และ D มีส่วนแบ่งใช้สำหรับรีซอร์สเฉพาะ เป็น “-”, 200, 150, และ 100, ตามลำดับ ทุกคลาสใช้งานอยู่ และคลาส A ใช้ 50% ของรีซอร์ส:

```
target(A) = unregulated = 100%
target(B) = 200/450 * available = 44% * 50% = 22%
target(C) = 150/450 * available = 33% * 50% = 17%
target(D) = 100/450 * available = 22% * 50% = 11%
```

เนื่องจากไม่มีการควบคุมคลาส A และคลาส A ใช้รีซอร์สเป็นจำนวน 50% ของ รีซอร์สที่มีอยู่ คลาสที่เหลือจึงมีรีซอร์สเหลือเพียง 50% เท่านั้น และมีการคำนวณ เป้าหมายจากค่าเปอร์เซ็นต์นี้ เนื่องจากคลาส A จะใช้รีซอร์สต่ำกว่าเป้าหมาย (100%) เสมอ คลาส A จะมีลำดับความสำคัญสูงกว่าคลาสอื่นทั้งหมดในระดับเดียวกัน ซึ่งใช้รีซอร์ส ในระดับหรือสูงกว่าเป้าหมายเสมอ (ให้ดูที่ “ลำดับความสำคัญคลาสใน Workload Manager” ในหน้า 561 สำหรับ ข้อมูลเพิ่มเติม)

หมายเหตุ: การไม่ควบคุมคลาสในการใช้รีซอร์สไม่เหมือนกับการวางคลาส ไว้ในระดับที่สูงขึ้น ลักษณะการทำงานต่อไปนี้ที่แสดงที่นี่ เป็นจริงสำหรับ คลาสที่ไม่ได้ควบคุม (ในระดับเดียวกัน) และไม่เป็นจริงถ้าคลาสถูกวางไว้ในระดับที่สูงขึ้น:

- เนื่องจากส่วนแบ่งใช้มีการกำหนดในแบบต่อรีซอร์ส จึงอาจไม่ควบคุมคลาส สำหรับการ ใช้รีซอร์สบางรายการ และควบคุมคลาสสำหรับรีซอร์สอื่น
- ค่านี้ถึงขีดจำกัดต่ำสุดสำหรับคลาสอื่นในระดับเดียวกัน ระดับที่สูงกว่า ไม่คำนึงถึงค่าต่ำสุดซึ่งระบุในระดับที่ต่ำกว่า
- แม้ในกรณีที่ไม่มีการจำกัดต่ำสุดสำหรับคลาสที่มีส่วนแบ่งใช้ ปริมาณการใช้ของคลาสที่ไม่ได้ควบคุมยังคงขึ้นอยู่กับคลาสที่มีส่วนแบ่งใช้ เนื่องจากคลาสกำลังช่วงชิงรีซอร์สบางส่วนที่มีอยู่ในระดับ จำเป็นต้องมี การทดลองเพื่อดูลักษณะการทำงานสำหรับเวิร์กโหลดที่กำหนด

หากไม่ได้ระบุจำนวนส่วนแบ่งใช้ จะใช้ค่าดีฟอลต์ “-” และจะไม่ควบคุมคลาสสำหรับการใช้รีซอร์สนั้น โปรดทราบว่าในเวอร์ชันแรกของ WLM ค่าส่วนแบ่งใช้ดีฟอลต์ ถ้าไม่มีการระบุ คือ 1

ส่วนแบ่งใช้มีการระบุต่อคลาสสำหรับรีซอร์สทุกชนิด ส่วนแบ่งใช้ มีการระบุใน stanzas ของไฟล์ shares ตัวอย่างเช่น:

```
shares
classname:
  CPU      = 2
  memory   = 4
  diskIO   = 3
```

ข้อกำหนดคุณลักษณะขีดจำกัดรีซอร์สใน Workload Manager:

นอกเหนือจากการใช้ส่วนแบ่งใช้เพื่อกำหนดสิทธิในรีซอร์สสัมพันธ์แล้ว WLM นำเสนอความสามารถในการระบุขีดจำกัดรีซอร์สสำหรับคลาสขีดจำกัดรีซอร์ส ช่วยให้ผู้ใช้และระบบสามารถควบคุมการจัดสรรรีซอร์สได้มากขึ้น ขีดจำกัดเหล่านี้มีการระบุเป็นเปอร์เซ็นต์ และสัมพันธ์กับจำนวนรีซอร์สที่มีอยู่ในระดับที่คลาสสังกัดอยู่

ขีดจำกัดสำหรับการควบคุมเป็นเปอร์เซ็นต์มีอยู่สามชนิดดังนี้:

ค่าต่ำสุด

นี่เป็นจำนวนรีซอร์สต่ำสุดที่ควรจะทำให้มีอยู่ในคลาส หากปริมาณการใช้จริงของคลาสต่ำกว่าค่านี้ คลาสจะได้รับมอบสิทธิเข้าถึงลำดับความสำคัญสูงสุดในรีซอร์ส ค่าที่เป็นไปได้คือ 0 ถึง 100 โดยที่ 0 เป็นค่าดีฟอลต์ (ถ้าไม่มีการระบุ)

ค่าสูงสุดซอฟต์แวร์

นี่เป็นจำนวนรีซอร์สสูงสุดที่คลาสสามารถใช้ได้เมื่อมีการช่วงชิงรีซอร์สนั้น หากปริมาณการใช้ของคลาสเกินกว่าค่านี้ คลาสจะได้รับมอบลำดับความสำคัญต่ำสุดในระดับ หากไม่มีการช่วงชิงรีซอร์ส (จากคลาสอื่นในระดับเดียวกัน) คลาสจะได้รับอนุญาตให้ใช้รีซอร์สได้มากตามที่ต้องการ ค่าที่เป็นไปได้คือ 1 ถึง 100 โดยที่ 100 เป็นค่าดีฟอลต์ (ถ้าไม่มีการระบุ)

ค่าสูงสุดฮาร์ด

นี่เป็นจำนวนรีซอร์สสูงสุดที่คลาสสามารถใช้ได้ แม้ว่าไม่มีการช่วงชิงรีซอร์ส หากคลาสใช้รีซอร์สจนถึงขีดจำกัดนี้ คลาสจะไม่ได้รับอนุญาตให้ใช้รีซอร์สเพิ่มเติมอีกจนกว่าเปอร์เซ็นต์ปริมาณการใช้จะลดลงต่ำกว่าขีดจำกัด ค่าที่เป็นไปได้คือ 1 ถึง 100 โดยที่ 100 เป็นค่าดีฟอลต์ (ถ้าไม่มีการระบุ)

ค่าขีดจำกัดรีซอร์สมีการระบุอยู่ในไฟล์ขีดจำกัดรีซอร์ส โดยเรียงตามชนิดรีซอร์สภายใน stanzas ของแต่ละคลาสขีดจำกัดมีการระบุเป็นช่วงค่าต่ำสุด ถึงค่าสูงสุดซอฟต์แวร์ ที่แบ่งด้วยยัติภังค์ (-) และละเว้นช่องว่าง ค่าสูงสุดฮาร์ดมีการระบุหลังจากค่าสูงสุดซอฟต์แวร์ และแบ่งโดยเครื่องหมายอัฒภาค (;) ค่าขีดจำกัดแต่ละค่ามีเครื่องหมายเปอร์เซ็นต์ (%) ตามหลัง

ข้อมูลต่อไปนี้เป็นตัวอย่างการใช้ไฟล์กฎ:

- หากผู้ใช้ joe จากกลุ่ม acct3 ดำเนินการ /bin/vi โพรเซสจะถูกวางไว้ในซูเปอร์คลาส acctg
- ถ้าผู้ใช้ sue จากกลุ่ม dev ดำเนินการ /bin/emacs โพรเซสจะถูกวางไว้ในซูเปอร์คลาส devlt (ID กลุ่มตรงกัน) แต่จะไม่ถูกจัดประเภทเข้าในคลาสย่อย editors เนื่องจากผู้ใช้ sue ถูกแยกออกจาก คลาสนั้น โพรเซสจะอยู่ใน devlt โดยค่าดีฟอลต์
- เมื่อผู้ดูแลระบบ DB เริ่มต้น /usr/sbin/oracle ด้วย ID ผู้ใช้ของ oracle และ ID กลุ่มของ dbm เพื่อใช้ฐานข้อมูล DB1 โพรเซสจะถูกจัดประเภท เข้าในซูเปอร์คลาสดีฟอลต์ เฉพาะเมื่อโพรเซส ตั้งค่าแท็กเป็น _DB1 เท่านั้น โพรเซสจะถูกกำหนดให้กับซูเปอร์คลาส db1

ขีดจำกัดมีการระบุสำหรับรีซอร์สทุกชนิด ต่อคลาส ใน stanzas ของไฟล์ limits ตัวอย่างเช่น:

```
shares
classname:
  CPU      = 0%-50%;80%
  memory   = 10%-30%;50%
```

ในตัวอย่างนี้ ไม่มีการตั้งค่าขีดจำกัดสำหรับดิสก์ I/O โดยใช้ค่าดีฟอลต์ระบบ ข้อมูลนี้แปลเป็นดังต่อไปนี้:

```
diskIO    = 0%-100%;100%
```

ตัวอย่างก่อนหน้านี้นี้ทั้งหมดสมมติว่าซูเปอร์คลาสและคลาสย่อย ที่อธิบายไม่ได้มีการตั้งค่าแอ็ททริบิวต์การสืบทอดเป็น ใช้ หากเปิดใช้งานการสืบทอด โพรเซสใหม่ จะได้รับสืบทอดซูเปอร์คลาสและคลาสย่อยจากพาเรนต์

ข้อมูลต่อไปนี้เป็นข้อจำกัดเท่าที่ WLM ใช้ในค่าขีดจำกัด รีซอร์ส:

- ขีดจำกัดต่ำสุดต้องน้อยกว่าหรือเท่ากับขีดจำกัดสูงสุดซอฟต์แวร์
- ขีดจำกัดสูงสุดซอฟต์แวร์ต้องน้อยกว่าหรือเท่ากับขีดจำกัดสูงสุด ฮาร์ด
- ผลรวมของค่าต่ำสุดของซูเปอร์คลาสทั้งหมดภายในระดับต้องไม่เกิน 100
- ผลรวมของค่าต่ำสุดของคลาสย่อยทั้งหมดของซูเปอร์คลาสที่กำหนดภายในระดับ ต้องไม่เกิน 100

เมื่อคลาสที่มีขีดจำกัดหน่วยความจำฮาร์ดใช้รีซอร์สจนถึงขีดจำกัดนี้และร้องขอ หน้าเพิ่มเติม ขั้นตอนวิธีการแทนที่หน้า VMM (LRU) จะเริ่มต้นขึ้นและ "ขโมย" หน้าจากคลาสที่ถูกจำกัด ส่งผลให้ลดจำนวนหน้าลงต่ำกว่า ค่าสูงสุดฮาร์ด ก่อนนำเสนอหน้าใหม่ ลักษณะการทำงานนี้ถูกต้อง แต่กิจกรรม การเลื่อนหน้าพิเศษซึ่งสามารถเกิดขึ้นได้ในตำแหน่งที่มีหน้าว่างจำนวนมาก ส่งผลกระทบต่อประสิทธิภาพการทำงานโดยทั่วไปของระบบ ขอแนะนำให้ใช้ขีดจำกัดหน่วยความจำต่ำสุดสำหรับคลาสอื่น ก่อนการใช้ค่าสูงสุดหน่วยความจำฮาร์ด สำหรับคลาสใดๆ

เนื่องจากคลาสที่ใช้รีซอร์สต่ำกว่าค่าต่ำสุดมีลำดับความสำคัญสูงสุดในระดับ จึงควรรักษาค่าต่ำสุดให้อยู่ในระดับที่เหมาะสม โดยพิจารณาจาก ความต้องการรีซอร์สของคลาสอื่นในระดับเดียวกัน

ข้อจำกัดผลรวมของขีดจำกัดต่ำสุดภายในระดับที่ต้องน้อยกว่า หรือเท่ากับ 100 หมายความว่าคลาสในระดับที่มีลำดับความสำคัญสูงสุดจะได้รับ อนุญาตให้ใช้รีซอร์สจนถึงขีดจำกัดต่ำสุดเสมอ WLM ไม่ได้รับประกันว่า คลาสจะใช้รีซอร์สได้ถึงขีดจำกัดต่ำสุดอย่างแท้จริง ทั้งนี้ขึ้นอยู่กับวิธีการใช้รีซอร์สของโปรเซส ในคลาส และขีดจำกัดอื่นที่มีผลบังคับใช้ ตัวอย่าง เช่น คลาสอาจไม่ได้รับสิทธิ์ใน CPU ถึงค่าต่ำสุดเนื่องจากคลาสไม่มี หน่วยความจำเพียงพอ

สำหรับหน่วยความจำฟิสิกัล การตั้งค่าขีดจำกัดหน่วยความจำต่ำสุดเป็นการป้องกัน หน้าหน่วยความจำของโปรเซสของคลาส (อย่างน้อยสำหรับที่อยู่ในระดับที่มีลำดับความสำคัญสูงสุด) คลาสไม่ควรมีหน้าที่ขโมยมาเมื่อคลาสใช้รีซอร์สต่ำกว่า ขีดจำกัดต่ำสุด ยกเว้นว่าคลาสที่ใช้งานอยู่ทั้งหมดอยู่ต่ำกว่าขีดจำกัดต่ำสุด และหนึ่งในคลาสเหล่านั้นร้องขอหน้าเพิ่มเติม คลาสในระดับสูงสุดไม่ควรมีหน้าที่ขโมยมาเมื่ออยู่ต่ำกว่าค่าต่ำสุด การตั้งค่าขีดจำกัดหน่วยความจำต่ำสุด สำหรับคลาสของงานแบบโต้ตอบช่วยให้มั่นใจว่า หน้าทั้งหมดของคลาสนั้นจะไม่ ถูกขโมยระหว่างการเรียกใช้ที่ต่อกัน (แม้ว่าหน่วยความจำมีจำนวนจำกัด) และปรับปรุงเวลาการตอบกลับ

ข้อควรสนใจ: การใช้ขีดจำกัดสูงสุดฮาร์ดอาจส่งผลกระทบต่อประสิทธิภาพ ของระบบหรือแอ็พพลิเคชัน ถ้าใช้อย่างไม่เหมาะสม เนื่องจากการใช้ขีดจำกัดฮาร์ดอาจส่งผลให้เกิดรีซอร์สระบบที่ไม่ได้ใช้งาน ดังนั้น ในกรณีส่วนใหญ่ ขีดจำกัดสูงสุดซอฟต์แวร์จึงเหมาะสมมากกว่า การใช้งานอย่างหนึ่งสำหรับขีดจำกัดสูงสุดฮาร์ดอาจเป็น ขีดจำกัดปริมาณการใช้ของระดับที่สูงกว่า เพื่อทำให้มีรีซอร์สบางส่วนอยู่สำหรับ ระดับที่ต่ำกว่า แม้ว่า การวางแอ็พพลิเคชันที่ต้องการรีซอร์สไว้ในระดับที่สูงกว่า อาจเป็นทางออกที่ดีกว่าก็ตาม

ค่าจำกัดทั้งหมดสามารถระบุไว้ในไฟล์ค่าจำกัดพร้อมค่า และหน่วยดังที่สรุปข้อมูลในตารางต่อไปนี้:

ตารางที่ 69. ค่าจำกัดรีซอร์สสำหรับ Workload Manager

รีซอร์ส	หน่วยที่ใช้ได้	หน่วยดีฟอลต์	ค่าสูงสุด	ค่าต่ำสุด
totalCPU	s, m, h, d, w	s	$2^{30} - 1$ s	10 s
totalDiskIO	KB, MB, TB, PB, EB	KB	$(2^{63} - 1) * 512/1024$ KB	1 MB
totalConnectTime	s, m, h, d, w	s	$2^{63} - 1$ s	5 m
totalProcesses	-	-	$2^{63} - 1$	2
totalThreads	-	-	$2^{63} - 1$	2
totalLogins	-	-	$2^{63} - 1$	1

หมายเหตุ: ตัวระบุหน่วยไม่คำนึงถึงตัวพิมพ์ s = วินาที, m = นาที, h = ชั่วโมง, d = วัน, w = สัปดาห์, KB = กิโลไบต์, MK = เมกะไบต์, ... เป็นต้น

Stanza ชิดจำกัดตัวอย่างมีดังนี้:

```
BadUserClass:
    totalCPU = 1m
    totalConnectTime = 1h
```

ชิดจำกัดทั้งหมดสามารถระบุได้โดยใช้ค่าใดๆ ในตารางข้างบนโดยมีข้อจำกัดต่อไปนี้:

- ถ้าระบุ ค่าของ totalThreads ต้องเท่ากับค่าของ totalProcesses เป็นอย่างน้อย
- ถ้าระบุ totalThreads และไม่ได้ระบุ totalProcesses ชิดจำกัดสำหรับ totalProcesses จะถูกตั้งค่าเป็นค่าของ totalThreads

ชิดจำกัดทั้งหมดสามารถระบุได้ทั้งที่ระดับซูเปอร์คลาสและคลาสย่อย เมื่อตรวจสอบชิดจำกัด จะมีการตรวจสอบชิดจำกัดคลาสย่อยก่อนชิดจำกัด ซูเปอร์คลาส ถ้าระบุทั้งสองชิดจำกัด จะมีการบังคับใช้ค่าที่ต่ำกว่า หาก ชิดจำกัดคลาสย่อยที่ระบุมากกว่าชิดจำกัดซูเปอร์คลาสที่เชื่อมโยง จะมีการออกค่าเตือนเมื่อโหลดคอนฟิกูเรชัน แต่จะถูก โหลด นี่เป็นสิ่งสำคัญสำหรับชิดจำกัดทั้งหมดของคลาสเนื่องจากชิดจำกัดเป็น ค่าสัมบูรณ์ (ไม่ใช่สัมพันธ์กับซูเปอร์คลาส) และคลาสย่อยหนึ่งอาจใช้รีซอร์สทั้งหมดที่มีอยู่ในซูเปอร์คลาส หากไม่ได้ระบุ ค่าดีฟอลต์สำหรับชิดจำกัดทั้งหมดทุกค่าจะเป็น "-" ซึ่งหมายความว่าไม่มีชิดจำกัด โดยค่าดีฟอลต์ การบัญชีทั้งหมดและการควบคุม คลาสและโปรเซสจะมีการเปิดใช้งานเมื่อ WLM กำลังรัน สามารถใช้อ็อปชัน -T [class]proc ของคำสั่ง wlmctrl เพื่อ ปิดใช้งานการบัญชีทั้งหมดและการควบคุม

ลำดับความสำคัญคลาสใน Workload Manager:

WLM จัดสรรรีซอร์สให้กับคลาสโดยการกำหนดลำดับความสำคัญให้กับ แต่ละคลาสสำหรับรีซอร์สแต่ละรายการ

ลำดับความสำคัญของคลาสสามารถเปลี่ยนแปลงได้ และขึ้นอยู่กับระดับ ส่วนแบ่ง ใช้ ชิดจำกัด และปริมาณการใช้ในปัจจุบันของคลาส ณ เวลาที่กำหนดใดๆ คลาสที่มีลำดับความสำคัญ สูงสุดจะได้รับสิทธิในรีซอร์สก่อน ที่ลำดับความสำคัญสูงสุด ระดับแสดงถึงช่วงที่ไม่มีการซ้อนเหลื่อมกันของลำดับความสำคัญคลาส คลาสในระดับ 0 จะได้รับการรับประกันเสมอว่า มีลำดับความสำคัญสูงกว่า คลาสในระดับ 1 (ยกเว้นว่าเกินค่าสูงสุดฮาร์ด)

เมื่อกำหนดลำดับความสำคัญคลาส WLM จะบังคับใช้ข้อจำกัดกับคลาส ในลำดับต่อมา (สูงสุดไปต่ำสุด):

ขีดจำกัดฮาร์ด

ถ้าปริมาณการใช้ของคลาสเกินกว่าค่าสูงสุดฮาร์ดสำหรับรีซอร์ส คลาส จะได้รับการกำหนดลำดับความสำคัญต่ำสุดที่เป็นไปได้สำหรับรีซอร์สนั้น และจะถูก ปฏิเสธการเข้าถึงจนกว่าปริมาณการใช้ลดลงต่ำกว่าขีดจำกัดนี้

ระดับ ในกรณีที่ไม่มีขีดจำกัดฮาร์ด ลำดับความสำคัญของคลาสจะถูกโยกเข้ากับ ค่าต่ำสุดและสูงสุดที่ใช้ได้สำหรับลำดับความสำคัญของระดับ

ขีดจำกัดซอฟต์แวร์

ถ้าปริมาณการใช้ของคลาสต่ำกว่าค่าต่ำสุดของขีดจำกัดซอฟต์แวร์สูงสุดสำหรับ รีซอร์ส คลาส จะได้รับการกำหนดลำดับความสำคัญสูงสุดในระดับ ถ้าปริมาณการใช้ของคลาสมากกว่าซอฟต์แวร์สูงสุด คลาส จะได้รับการกำหนดลำดับความสำคัญต่ำสุด ในระดับ

ส่วนแบ่งใช้

ใช้เพื่อคำนวณเป้าหมายปริมาณการใช้ของคลาสสำหรับรีซอร์สแต่ละรายการ ลำดับความสำคัญคลาสเพิ่มขึ้นเมื่อปริมาณการใช้ของคลาสดลดลงต่ำกว่าเป้าหมาย และลำดับความสำคัญคลาสดลดลงเมื่อปริมาณการใช้เพิ่มขึ้นมากกว่าเป้าหมาย โปรดทราบว่าขีดจำกัดซอฟต์แวร์มีความสำคัญ มากกว่าและลำดับความสำคัญของคลาสจะถูกพิจารณาจากขีดจำกัดซอฟต์แวร์ เมื่อใช้ได้

แม้ว่าสามารถใช้ได้ทั้งส่วนแบ่งใช้และขีดจำกัดสำหรับแต่ละคลาสและสำหรับ รีซอร์สแต่ละรายการ แต่จะสามารถคาดการณ์ผลลัพธ์ได้ดีขึ้นถ้ามีการใช้เพียงหนึ่ง เกณฑ์เท่านั้นสำหรับแต่ละคลาส

ชุดรีซอร์สตัวประมวลผลที่ใช้เฉพาะตัว:

Exclusive use processor resource sets (XRSETs) ช่วยให้ผู้ใช้ดูแล ระบบสามารถรับประกันว่ามีรีซอร์สสำหรับงานที่สำคัญ XRSET เป็นชุดรีซอร์สที่ระบุ ซึ่งเปลี่ยนลักษณะการทำงานของ CPUs ทั้งหมดที่รวมอยู่ เมื่อ CPU เป็นแบบ ใช้เฉพาะตัว CPU จะรันเฉพาะโปรแกรมที่เกี่ยวข้องอย่างชัดเจนเท่านั้น

การสร้าง XRSET

คุณต้องเป็นผู้ใช้รากเพื่อสร้าง XRSET ใช้คำสั่ง `mkrset` เพื่อสร้างชุดรีซอร์สในเนมสเปซ `sysxrset` ตัวอย่างเช่น คำสั่ง `mkrset -c 1-3 sysxrset/set1` สร้าง XRSET สำหรับ CPUs 1, 2, และ 3 รูทีนย่อย `rs_register_name()` ยังสามารถใช้เพื่อสร้าง XRSET ได้เช่นกัน

การพิจารณาว่า XRSETs มีการกำหนดไว้บนระบบแล้วหรือไม่

คำสั่ง `lsrset -v -n sysxrset` แสดง XRSETs ทั้งหมดที่กำหนดไว้บนระบบ (ในปัจจุบัน ไม่มี API การเขียนโปรแกรมสำหรับการทำเช่นนั้น)

การลบ XRSET

คุณต้องเป็นผู้ใช้รากเพื่อลบ XRSET คำสั่ง `rmrset` ลบ XRSET รูทีนย่อย `rs_discard_name()` ยังสามารถใช้เพื่อลบ XRSET ได้เช่นกัน

การรีบูตระบบ

เมื่อคุณรีบูตระบบ XRSETs ใดๆ ที่ตั้งค่าไว้จะถูกลบออกจาก รีจิสทรีและไม่มีผลบังคับใช้อีกต่อไป

การระบุงานสำหรับ XRSETs

มีหลายวิธีในการทำเครื่องหมายงานเป็นมีสิทธิ์ใช้ ตัวประมวลผลที่ใช้เฉพาะตัว สามารถใช้คำสั่ง `attachrset` และ `execrset` เพื่อระบุชุดรีซอร์สที่มีตัวประมวลผลที่ใช้เฉพาะตัว ชุดรีซอร์สที่มีตัวประมวลผลที่ใช้เฉพาะตัวสามารถเชื่อมโยงกับคลาส WLM ได้งานที่จัดประเภทเป็นคลาส WLM ดังกล่าวจะใช้ตัวประมวลผลที่ใช้เฉพาะตัว ซึ่งระบุในชุดรีซอร์ส

การใช้ XRSETs ด้วย Bindprocessor และ `_system_configuration.ncpus`

คุณไม่สามารถใช้ `bindprocessor` เพื่อทำให้งาน รันบนตัวประมวลผลที่ใช้เฉพาะตัว เฉพาะสิ่งที่แนบมาบนชุดรีซอร์สเท่านั้นสามารถทำให้งานรันบนตัวประมวลผลที่ใช้เฉพาะตัว

จำนวนของ CPUs ในคอนฟิกูเรชันระบบ (ฟิลด์ `_system_configuration.ncpus`) ไม่เปลี่ยนเมื่อสร้าง XRSETs ยังคงมี NCPUs ในระบบ

เมื่อ โปรแกรมใช้การเรียกระบบ `bindprocessor` ไปยัง NCPUs CPUs ใน XRSETs จะล้มเหลวด้วยข้อผิดพลาด EINVAL คุณสามารถโยง เข้ากับ ID ที่ส่งคืนใดๆ โดยใช้ตัวเลือกเดียวรีซอร์สของคำสั่ง `bindprocessor` ตัวเลือกเดียวรี (`bindprocessor -q`) จะส่งคืนเฉพาะ IDs โยงที่ถูกต้องเท่านั้น ไม่รวม ID ที่เชื่อมโยงกับ CPUs เฉพาะ

ตัวอย่าง เช่น ถ้ามี 10 CPUs ออนไลน์ในระบบและสามจากจำนวนนั้นอยู่ใน XRSETs `bindprocessor` ไปยัง CPUs ที่มี IDs โยงในช่วงตั้งแต่ 0 ถึง 6 จะสำเร็จ `bindprocessor` ไปยัง CPUs ที่มี IDs โยงในช่วง 7 ถึง 9 จะได้รับข้อผิดพลาด EINVAL

การใช้ XRSETs ด้วยการดำเนินงานรีคอนฟิกูเรชัน CPU แบบไดนามิก

โดยทั่วไป รีคอนฟิกูเรชัน CPU แบบไดนามิกไม่ได้ผลกระทบจาก ตัวประมวลผลที่ใช้เฉพาะตัว อย่างไรก็ตาม การสร้าง XRSETs และการกำหนดงานให้กับ ตัวประมวลผลดังกล่าวอาจทำให้ไม่สามารถลบ CPU ได้ CPUs ที่มีการเพิ่มลงในระบบแบบไดนามิก อาจเข้าสู่ระบบเป็นตัวประมวลผลที่ใช้ทั่วไปหรือตัวประมวลผลที่ใช้เฉพาะตัว อย่างไรก็ตาม หนึ่ง ตัวประมวลผลจะเข้าสู่ระบบเป็นตัวประมวลผลที่ใช้เฉพาะตัวถ้ามี XRSET ซึ่งมีโลจิคัล CPU ID เมื่อเข้าสู่ระบบ

ชุดรีซอร์สใน Workload Manager:

WLM ใช้ชุดรีซอร์ส (หรือ `rsets`) เพื่อจำกัดโปรเซส ในคลาสที่กำหนดให้สามารถใช้ได้เฉพาะชุดย่อยของฟิลิคลีซอร์สของระบบ ใน WLM ฟิลิคลีซอร์สที่จัดการคือหน่วยความจำและตัวประมวลผล ชุดรีซอร์สที่ถูกต้อง ประกอบด้วยหน่วยความจำและตัวประมวลผลอย่างน้อยหนึ่งตัว

เมื่อใช้ SMIT ผู้ดูแลระบบสามารถกำหนดและตั้งชื่อชุดรีซอร์สซึ่งประกอบด้วยชุดย่อย ของรีซอร์สที่มีอยู่บนระบบ จากนั้น โดยใช้อินเตอร์เฟซการจัดการ WLM ผู้ใช้รากหรือผู้ดูแลซูเปอร์คลาสที่ได้รับแต่งตั้งสามารถใช้ ชื่อของชุดรีซอร์สเป็นแอตทริบิวต์ `rset` ของ คลาส WLM ได้ หลังจากนั้น ทุกโปรเซสที่กำหนดให้กับคลาส WLM นี้จะถูกจัดส่ง บนตัวประมวลผลตัวใดตัวหนึ่งในชุดรีซอร์สเท่านั้น ซึ่งเป็นการแยกเวิร์กโหลด สำหรับรีซอร์ส CPU ได้อย่างมีประสิทธิภาพ

ระบบปัจจุบันทั้งหมดมีโดเมนหน่วยความจำเพียงรายการเดียวซึ่งแบ่งใช้โดย ชุดรีซอร์สทั้งหมด ดังนั้นเมธอดนี้จึงไม่ได้แบ่งแยกเวิร์กโหลดในหน่วยความจำทางฟิลิคล

รีจิสตรีซุตรีซอร์สใน Workload Manager:

เซอวิล์รีจิสตรี rset ช่วยให้ผู้ใช้และระบบ สามารถกำหนดและตั้งชื่อซุตรีซอร์ส เพื่อให้ผู้ใช้หรือแอปพลิเคชันอื่นสามารถใช้ซุตรีซอร์สนั้นได้

เพื่อลดความเสี่ยงของชื่อที่ซ้ำกัน รีจิสตรีจึงสนับสนุนแบบแผนการตั้งชื่อ สองระดับ ชื่อของซุตรีซอร์สอยู่ในรูปแบบ `name_space/rset_name` ทั้ง `name_space` และ `rset_name` สามารถ ยาวได้ 255 อักขระ เป็นแบบตรงตามตัวพิมพ์ และอาจประกอบด้วย ตัวอักษรพิมพ์ใหญ่และพิมพ์เล็ก ตัวเลข ชิดเส้นใต้ และเครื่องหมายมหัพภาค (.) เท่านั้น. `name_space` ของ `sys` ถูกสงวนไว้โดยระบบปฏิบัติการและใช้สำหรับนิยาม rset ที่แสดงถึงรีซอร์สของระบบ

ชื่อนิยาม rset ต้องไม่ซ้ำกันภายใน พื้นที่ชื่อรีจิสตรี การเพิ่มนิยาม rset ใหม่ลงใน รีจิสตรีโดยใช้ชื่อเดียวกันกับนิยาม rset ที่มีอยู่ ส่งผลให้นิยามที่มีอยู่ถูกแทนที่ด้วยนิยามใหม่ หากมีสิทธิอนุญาตและ เอกสิทธิ์ที่เหมาะสม เฉพาะผู้ใช้รากเท่านั้นที่สามารถสร้าง แก้ไข และลบรีซอร์ส และอัปเดตฐานข้อมูล in-core rset โดยใช้ SMIT

นิยาม rset แต่ละรายการมีเจ้าของ (ID ผู้ใช้) กลุ่ม (ID กลุ่ม) และสิทธิเข้าถึงที่เชื่อมโยง ข้อมูลเหล่านี้มีการระบุ ในเวลาที่สร้างนิยาม rset และมีไว้สำหรับ วัตถุประสงค์ในการควบคุมการเข้าถึง ในกรณีของไฟล์ มีสิทธิเข้าถึงที่แยกต่างหากสำหรับเจ้าของกลุ่ม และบุคคลอื่น ซึ่งกำหนดว่า มีการให้สิทธิการอ่านและ/หรือการเขียน สิทธิการอ่านอนุญาตให้ดึงข้อมูลนิยาม rset ในขณะที่สิทธิการเขียนอนุญาตให้แก้ไขหรือลบนิยาม rset ได้

นิยาม rset ที่กำหนดไว้โดยผู้ดูแลระบบ ถูกเก็บไว้ใน `/etc/rsets` ไฟล์ stanza ไม่มีมีการอธิบายรูปแบบของ ไฟล์นี้ และผู้ใช้ต้องจัดการ rset ผ่านทางอินเตอร์เฟซ SMIT เพื่อ ป้องกันปัญหาความเข้ากันได้ที่อาจเกิดขึ้นในอนาคตถ้ามีการแก้ไขรูปแบบไฟล์ ในกรณีของนิยามคลาส WLM ต้องโหลดนิยาม rset เข้าใน โครงสร้างข้อมูลเคอร์เนลก่อนที่ WLM สามารถใช้นิยามนั้นได้ งานที่เกี่ยวข้อง:

“การสร้างซุตรีซอร์ส” ในหน้า 537

การใช้ซุตรีซอร์ส (rsets) นับเป็นวิธีที่มีประสิทธิภาพในการแยกเวิร์กโหลด ที่เกี่ยวข้องกับ CPU ออกจากกัน โดยการแยกสองเวิร์กโหลดที่แตกต่างกัน ออกเป็นสองคลาสและให้ชุดย่อยที่แตกต่างกันของ CPU กับแต่ละคลาส ทำให้คุณมั่นใจได้ว่าสองเวิร์กโหลดจะไม่มีทางช่วงชิงรีซอร์ส CPU ระหว่างกัน แม้ว่าทั้งสองเวิร์กโหลดยังคงช่วงชิงหน่วยความจำฟิสิคัลและ I/O แบนด์วิธ

การตั้งค่า Workload Manager

นิยามคลาส แอ็ททริบิวต์คลาส ส่วนแบ่งใช้และขีดจำกัด และกฎการกำหนดคลาสอัตโนมัติ สามารถป้อนได้โดยใช้ SMIT หรืออินเตอร์เฟซบรรทัดคำสั่ง WLM นิยามและกฎเหล่านี้มีการเก็บไว้ในไฟล์ ข้อความปกติ ซึ่งยังสามารถสร้างหรือแก้ไขโดยใช้โปรแกรมแก้ไขข้อความได้ด้วย

ไฟล์เหล่านี้ (ที่เรียกว่า *ไฟล์คุณสมบัติ WLM*) มีการเก็บไว้ในไดเรกทอรีย่อย `/etc/wlm` ชุดของไฟล์ที่อธิบายซูเปอร์คลาสและคลาสย่อยที่เชื่อมโยงกำหนดคอนฟิกูเรชัน WLM ไฟล์สำหรับคอนฟิกูเรชัน WLM Config อยู่ใน `/etc/wlm/Config` ไดเรกทอรีนี้มีนิยามของพารามิเตอร์ WLM สำหรับซูเปอร์คลาส ไฟล์มีการตั้งชื่อว่า `description, classes, shares, limits,` และ `rules` ไดเรกทอรีนี้ยังอาจมีไดเรกทอรีย่อย ที่มีชื่อเป็นชื่อของซูเปอร์คลาสซึ่งเก็บนิยามคลาสย่อย ตัวอย่างเช่น สำหรับซูเปอร์คลาส `Super` ของคอนฟิกูเรชัน WLM Config ไดเรกทอรี `/etc/wlm/Config/Super` มีไฟล์คุณสมบัติ สำหรับคลาสย่อยของซูเปอร์คลาส `Super` ไฟล์มีการตั้งชื่อว่า `description, classes, shares, limits,` และ `rules`

หลังจากที่ผู้ดูแลระบบกำหนดคอนฟิกูเรชัน WLM แล้ว สามารถทำให้เป็นคอนฟิกูเรชันที่ใช้งานอยู่ได้โดยใช้พาด่วน `smit wlmmanage` หรือคำสั่ง `wlmcntrl`

คุณสามารถกำหนดชุดของไฟล์คุณสมบัติได้หลายชุด เพื่อกำหนดคอนฟิกูเรชันที่แตกต่างกันของการจัดการเวิร์กโหลด โดยปกติ คอนฟิกูเรชันเหล่านี้ตั้งอยู่ในไดเรกทอรีย่อย /etc/wlm ลิงก์สัญลักษณ์ /etc/wlm/current ซึ่งไปยังไดเรกทอรีที่มีไฟล์คอนฟิกูเรชันปัจจุบัน ลิงก์นี้มีการ อัปเดตโดยคำสั่ง `wlmcntrl` เมื่อ WLM เริ่มต้นด้วยชุดของ ไฟล์คอนฟิกูเรชันที่ระบุ

ข้อมูลที่เกี่ยวข้อง:

“การกำหนดคอนฟิก Workload Manager เพื่อรวม เวิร์กโหลด” ในหน้า 539
Workload Manager (WLM) ช่วยให้คุณสามารถควบคุมรีซอร์สที่ใช้โดยงานบนระบบของคุณได้

ข้อกำหนดแอ็พพลิเคชันสำหรับคอนฟิกูเรชัน Workload Manager:

ระยะแรกของการกำหนดคอนฟิกูเรชันต้องการความเข้าใจใน ผู้ใช้ของคุณและความต้องการในการคำนวณของผู้ใช้ ตลอดจนแอ็พพลิเคชันบน ระบบ ความต้องการรีซอร์สและความต้องการของธุรกิจ (ตัวอย่างเช่น ภารกิจใดที่สำคัญมากและภารกิจใดที่สามารถกำหนดลำดับความสำคัญต่ำลง) จากข้อมูล ความเข้าใจนี้ คุณกำหนดซูเปอร์คลาส ตามด้วยคลาสย่อยของคุณ

การตั้งค่าลำดับความสำคัญขึ้นอยู่กับฟังก์ชันของ WLM ในองค์กรของคุณ ในกรณีของการรวมเซิร์ฟเวอร์ คุณอาจทราบแอ็พพลิเคชัน ผู้ใช้ และความต้องการรีซอร์สของผู้ใช้อยู่แล้ว และคุณอาจสามารถข้ามหรือ ทำบางขั้นตอนให้สั้นลงได้

WLM อนุญาตให้คุณจัดประเภทโปรเซสตามผู้ใช้หรือกลุ่ม แอ็พพลิเคชัน ชนิด แท็ก หรือชุดของแอ็พทริบิวต์เหล่านี้ เนื่องจาก WLM ควบคุมการใช้รีซอร์ส ระหว่างคลาส ผู้ดูแลระบบจึงควรจัดกลุ่มแอ็พพลิเคชัน และผู้ใช้ที่มีรูปแบบการใช้รีซอร์สเหมือนกันเข้าไว้ในคลาสเดียวกัน ตัวอย่างเช่น คุณอาจต้องการแยกงานแบบโต้ตอบซึ่งโดยปกติ ใช้เวลา CPU น้อยมากแต่ต้องการเวลาตอบกลับที่รวดเร็วออกจากงานชนิดแบตช์ ซึ่งโดยปกติ เป็นงานที่ใช้ CPU และหน่วยความจำมาก เช่นเดียวกับใน สภาวะแวดล้อมฐานข้อมูลซึ่งคุณต้องการแยกการจราจรชนิด OLTP ออกจากเคียวรีของ data mining ที่เป็นภาระหนัก

ขั้นตอนนี้ทำได้โดยใช้ SMIT หรืออินเทอร์เฟซบรรทัดคำสั่ง สำหรับสองสามครั้งแรก อาจเป็นแนวคิดที่ดีที่จะใช้ SMIT เพื่อนำคุณผ่านขั้นตอนการสร้างคอนฟิกูเรชัน WLM แรกของคุณ รวมถึง การกำหนดซูเปอร์คลาสและการตั้งค่าแอ็พทริบิวต์ สำหรับการส่งผ่านครั้งแรก คุณสามารถตั้งค่าบางแอ็พทริบิวต์และปล่อยค่าอื่นเป็นค่า ดีฟอลต์ และทำแบบเดียวกันสำหรับส่วนแบ่งใช้และขีดจำกัดรีซอร์ส ลักษณะคลาสทั้งหมด เหล่านี้สามารถแก้ไขแบบไดนามิกได้ในภายหลัง

จากนั้น คุณสามารถเริ่มต้น WLM ในโหมด `passive` ตรวจสอบการจัดประเภท และเริ่มต้นตรวจทานรูปแบบการใช้รีซอร์สของแอ็พพลิเคชัน

ตรวจสอบคอนฟิกูเรชันโดยใช้คำสั่ง `wlmcntrl` หรือเมนู SMIT ที่สอดคล้องกัน จากนั้น เริ่มต้น WLM ในโหมด `passive` บนคอนฟิกูเรชันที่กำหนดใหม่ WLM จะ จัดประเภทโปรเซสที่มีอยู่ทั้งหมด (และโปรเซสทั้งหมดที่สร้างขึ้นจากจุดนั้น) และเริ่มต้นการคอมไพล์สถิติเกี่ยวกับการใช้ CPU, หน่วยความจำ และดิสก์ I/O ของคลาสต่างๆ WLM จะไม่พยายามควบคุมการใช้รีซอร์สนี้

ตรวจสอบว่าโปรเซสต่างๆ ถูกจัดประเภทในคลาสที่เหมาะสม ดังที่คาดไว้โดยผู้ดูแลระบบ (โดยใช้แฟล็ก `-o` ของคำสั่ง `ps`) หากบางโปรเซสไม่ได้ถูกจัดประเภทดังที่คุณคาดไว้ คุณสามารถแก้ไข กฎการกำหนดของคุณ หรือตั้งค่าบิตการสืบทอดสำหรับบางคลาส (ถ้าคุณต้องการ ให้โปรเซสใหม่ยังคงอยู่ในคลาสเดียวกันกับพารেন্ট) และ อัปเดต WLM คุณสามารถทำซ้ำโปรเซสจนกว่าคุณพอใจกับการจัดประเภท ระดับแรกนี้ (ซูเปอร์คลาส)

การรัน WLM ในโหมด `passive` และการรีเฟรช WLM (ในโหมด `passive` เสมอ) มีความเสี่ยงต่ำ มีค่าใช้จ่ายในการดำเนินงานต่ำ และสามารถทำได้อย่าง ปลอดภัยบนระบบการผลิตโดยไม่รบกวนการดำเนินงานระบบปกติ เมื่อต้องการเรียกใช้และรีเฟรช WLM ให้ใช้คำสั่ง `wlmcntrl` ซึ่งเรียกใช้จากบรรทัดคำสั่ง หรือจาก SMIT

รัน WLM ในโหมด passive เพื่อรวบรวมสถิติโดยใช้คำสั่ง `wlmstat` คำสั่ง `wlmstat` สามารถใช้ที่ช่วงเวลาสม่ำเสมอ เพื่อแสดงการใช้รีซอร์สต่อคลาสเป็นเปอร์เซ็นต์ของรีซอร์สทั้งหมด ที่มีอยู่สำหรับซูเปอร์คลาส ซึ่งช่วยให้คุณสามารถมอนิเตอร์ระบบในรอบระยะเวลาที่ยาวนานขึ้นเพื่อตรวจทานการใช้รีซอร์สของ แอปพลิเคชันหลัก

ระดับ ส่วนแบ่งใช้ และขีดจำกัดใน Workload Manager:

โดยใช้ข้อมูลที่รวบรวมจากการรัน WLM ในโหมด passive และเป้าหมาย ทางธุรกิจ ให้ตัดสินใจเลือกหมายเลขระดับซึ่งจะกำหนดให้กับทุกซูเปอร์คลาส และส่วนแบ่งใช้ของรีซอร์สแต่ละรายการที่ควรกำหนดให้กับคลาสต่างๆ

สำหรับบางคลาส คุณอาจต้องการกำหนดขีดจำกัดต่ำสุดหรือสูงสุด ปรับ ส่วนแบ่งใช้และหมายเลขระดับเพื่อให้ตรงกับเป้าหมายการจัดสรรรีซอร์ส สแกน ขีดจำกัดสำหรับกรณีที่ไม่สามารถแก้ไขด้วยส่วนแบ่งใช้เพียงอย่างเดียว คุณยังอาจต้องการ เพิ่มคลาสย่อยด้วย

- ใช้ขีดจำกัดต่ำสุดสำหรับแอปพลิเคชันซึ่งโดยปกติ มีการใช้รีซอร์สต่ำ แต่ต้องการเวลาตอบกลับเร็วเมื่อเรียกใช้โดยเหตุการณ์ภายนอก ปัญหาหนึ่งที่พบโดยงานแบบโต้ตอบในสถานการณ์ซึ่งหน่วยความจำมีจำกัดคือ หน้าถูกขโมยในระหว่างเวลาที่ไม่มิกิจกรรม สามารถใช้ขีดจำกัด ต่ำสุดของหน่วยความจำเพื่อป้องกันหน้าบางส่วนของงานแบบโต้ตอบ ถ้าคลาสอยู่ในระดับ 0
- ใช้ขีดจำกัดสูงสุดสำหรับงานลำดับความสำคัญต่ำที่ใช้รีซอร์สมาก ยกเว้นว่าคุณแบ่งพาร์ติชันรีซอร์สระบบด้วยเหตุผลอื่น ค่าสูงสุดฮาร์ด จะเหมาะสมที่สุดสำหรับรีซอร์สที่ไม่สามารถนำมาใช้ใหม่ เช่น หน่วยความจำ ที่เป็นเช่นนี้ เนื่องจากเวลาที่ใช้ในการเขียนข้อมูลไปยังพื้นที่การเลื่อนหน้า ถ้าคลาสระดับความ สำคัญสูงกว่าต้องการหน้าที่คลาสแรกควรจะได้ใช้สำหรับการใช้ CPU คุณสามารถใช้ระดับหรือค่าสูงสุดซอฟต์แวร์เพื่อให้แน่ใจว่าคลาสลำดับความสำคัญ สูงกว่าได้รับการกำหนดเวลา CPU ในทันที

เมื่อสร้างและปรับพารามิเตอร์ของคลาสย่อย คุณสามารถรีเฟรช WLM เฉพาะสำหรับคลาสย่อยของซูเปอร์คลาสที่กำหนดซึ่งไม่กระทบต่อผู้ใช้ และแอปพลิเคชันในซูเปอร์คลาสอื่น จนกว่าคุณจะพอใจกับลักษณะ การทำงานของระบบ

คุณยังสามารถกำหนดคอนฟิกูเรชันอื่นด้วยพารามิเตอร์ที่แตกต่างอื่น ตามความต้องการของธุรกิจ เมื่อทำเช่นนั้น คุณสามารถประหยัดเวลาโดยการคัดลอก และแก้ไขคอนฟิกูเรชันที่มีอยู่

การปรับคอนฟิกูเรชัน Workload Manager โดยละเอียด:

มอนิเตอร์ระบบโดยใช้คำสั่ง `wlmstat` และตรวจสอบว่า การควบคุมที่ทำโดย WLM เหมาะสมกับเป้าหมายของคุณ และไม่ได้ทำให้บาง แอปพลิเคชันขาดรีซอร์สอย่างไม่เหมาะสมในขณะที่แอปพลิเคชันอื่นได้รับรีซอร์สมากกว่าที่ควรจะได้รับ หากเป็นกรณีนี้ให้ปรับส่วนแบ่งใช้และรีเฟรช WLM

เมื่อคุณมอนิเตอร์และปรับส่วนแบ่งใช้ ขีดจำกัด และหมายเลขระดับ ให้ตัดสินใจว่าคุณต้องการมอบหมายการจัดการคลาasy่อยของบางซูเปอร์คลาสหรือ ทุกซูเปอร์คลาส จากนั้น ผู้ดูแลระบบสามารถมอนิเตอร์และตั้งค่า ส่วนแบ่งใช้คลาสย่อย ขีดจำกัด และหมายเลขระดับ

ผู้ดูแลระบบของแต่ละซูเปอร์คลาสสามารถทำซ้ำโปรเซสนี้สำหรับ คลาสย่อยของแต่ละซูเปอร์คลาสได้ ความแตกต่างเพียงอย่างเดียวคือ WLM ไม่สามารถรัน ในโหมด passive ที่ระดับคลาสย่อยเท่านั้น คอนฟิกูเรชันคลาสย่อยและ การปรับต้องทำด้วย WLM ในโหมด active วิธีหนึ่งที่ไม่กระทบต่อผู้ใช้ และแอปพลิเคชันในซูเปอร์คลาสคือ การเริ่มต้นหมายเลขระดับ และส่วนแบ่งใช้ และขีดจำกัดสำหรับคลาสย่อยที่ค่าฟอลต์ ('-') (ยติภังค์) สำหรับส่วนแบ่งใช้, 0% สำหรับค่าต่ำสุด, และ 100% สำหรับค่าสูงสุดซอฟต์แวร์และฮาร์ด) ด้วยค่าติดตั้งเหล่านี้ WLM จะไม่ควบคุมการจัดสรรรีซอร์สระหว่างคลาสย่อย

การแก้ไขปัญหา Workload Manager

หากคุณไม่เห็นลักษณะการทำงานที่ต้องการเมื่อใช้คอนฟิกูเรชันปัจจุบัน คุณอาจต้องปรับคอนฟิกูเรชัน WLM

ค่าปริมาณการใช้สำหรับแต่ละคลาสสามารถมอนิเตอร์โดยใช้คำสั่ง `wlmstat` ข้อมูลนี้สามารถรวบรวม และวิเคราะห์เพื่อช่วยกำหนดการเปลี่ยนแปลงที่อาจต้องทำในคอนฟิกูเรชัน หลังจากคุณอัปเดตคอนฟิกูเรชันแล้ว ให้อัปเดตคอนฟิกูเรชัน WLM ที่ใช้งานอยู่โดยใช้คำสั่ง `wlmcntrl -u`

แนวทางต่อไปนี้อาจช่วยให้คุณตัดสินใจเกี่ยวกับวิธีการเปลี่ยนคอนฟิกูเรชันของคุณ:

- หากจำนวนของส่วนแบ่งใช้ที่ใช้งานอยู่ในระดับแตกต่างกันมากเมื่อเวลาผ่านไป คุณสามารถกำหนดคลาสที่ไม่มีส่วนแบ่งใช้รีซอร์ส เพื่อให้มีเป้าหมายปริมาณการใช้ที่เป็นอิสระจากจำนวนของส่วนแบ่งใช้ที่ใช้งานอยู่ เทคนิคนี้มีประโยชน์สำหรับคลาสที่สำคัญซึ่งต้องการการเข้าถึงลำดับความสำคัญสูงในรีซอร์ส
- หากคุณต้องการรับประกันการเข้าถึงจำนวนรีซอร์สที่แน่นอน ให้ระบุขีดจำกัดต่ำสุด เทคนิคนี้ประโยชน์สำหรับงานแบบโต้ตอบที่ไม่ได้ใช้รีซอร์สมากนัก แต่ต้องตอบสนองต่อเหตุการณ์ภายนอกอย่างรวดเร็ว
- หากคุณต้องการจำกัดการเข้าถึงรีซอร์สแต่ส่วนแบ่งใช้ไม่ได้ให้อำนาจการควบคุม ที่เพียงพอ ให้ระบุขีดจำกัดสูงสุด ในกรณีส่วนใหญ่ แนะนำให้ใช้ขีดจำกัดสูงสุดซอฟต์แวร์ แต่สามารถใช้ค่าสูงสุดฮาร์ดแวร์ที่ต้องการการบังคับใช้ที่เข้มงวด เนื่องจากขีดจำกัดสูงสุดฮาร์ดแวร์ อาจส่งผลให้รีซอร์สระบบสูญเปล่า และอาจเพิ่มกิจกรรมการเลื่อนหน้า เมื่อใช้สำหรับการควบคุมหน่วยความจำ คุณจึงควรใช้ขีดจำกัดต่ำสุด สำหรับคลาสอื่นก่อนการใช้ขีดจำกัดฮาร์ดแวร์
- หากงานที่มีความสำคัญน้อยรบกวนงานที่มีความสำคัญมากกว่า ให้วางงานที่มีความสำคัญน้อยไว้ในระดับที่ต่ำกว่า เทคนิคนี้ช่วยให้มั่นใจว่างานที่มีความสำคัญน้อยกว่า มีลำดับความสำคัญต่ำกว่า และไม่สามารถช่วงชิงรีซอร์สที่มีอยู่ในขณะที่งานที่สำคัญมากกว่ากำลังรันอยู่
- หากคลาสไม่สามารถใช้งานจนถึงเป้าหมายปริมาณการใช้ของรีซอร์ส ให้ตรวจสอบ ว่าสภาพนี้เกิดจากการช่วงชิงรีซอร์สอื่นหรือไม่ ถ้าใช่ ให้เปลี่ยน การจัดสรรคลาสสำหรับรีซอร์สที่มีการช่วงชิง
- หากโปรเซสต่างๆ ภายในคลาสหนึ่งมีลักษณะการทำงานหรือปริมาณการใช้รีซอร์ส แตกต่างกันอย่างมาก ให้สร้างคลาสเพิ่มเติม เพื่อให้สามารถควบคุมได้ละเอียดมากขึ้น นอกจากนี้ อาจต้อง สร้างคลาสแยกต่างหากสำหรับแอพลิเคชันที่สำคัญแต่ละรายการ
- หากการวิเคราะห์แสดงว่ารีซอร์สที่ต้องการโดยคลาสหนึ่งขึ้นอยู่กับ ปริมาณการใช้ของคลาสอื่น ให้จัดสรรรีซอร์สอีกครั้งตามข้อมูลนั้น ตัวอย่างเช่น ถ้าจำนวนรีซอร์สที่ต้องการโดย ClassZ ขึ้นอยู่กับ จำนวนของคำร้องขอของงานที่สามารถจัดการได้โดย ClassA ต้องทำให้มั่นใจว่า ClassA มีสิทธิเข้าถึงรีซอร์สที่เพียงพอเพื่อนำเสนอสิ่งที่ ClassZ ต้องการ
- ถ้าหนึ่งแอพลิเคชันขึ้นไปไม่ได้รับรีซอร์สที่เพียงพอสำหรับการทำงาน ให้ถูกต้องอย่างต่อเนื่อง คุณอาจเลือกที่จะลดเวิร์กโหลด บนระบบ

หมายเหตุ: คุณสามารถกำหนด `adminuser` สำหรับซูเปอร์คลาสเพื่อลดจำนวน งานที่ต้องการของผู้ดูแลระบบ WLM หลังจากทดสอบและปรับคอนฟิกูเรชัน ระดับบนสุดแล้ว `adminusers` ซูเปอร์คลาสสามารถทำการเปลี่ยนแปลงในลำดับ ต่อมา (รวมถึงการสร้างและการกำหนดคอนฟิกูเรชัน) เพื่อให้เหมาะสมกับ ความต้องการเฉพาะ

Workload Manager API

แอพลิเคชันสามารถใช้ WLM APIs ซึ่งเป็นชุดของรูทีนในไลบรารี `/usr/lib/libwlm.a` เพื่อทำการกิจทั้งหมดที่ผู้ดูแลระบบ WLM สามารถทำได้โดยใช้อินเตอร์เฟส บรรทัดคำสั่ง WLM

ซึ่งรวมถึง:

- สร้าง แก้ไข หรือลบคลาส

- เปลี่ยนแอ็ททริบิวต์คลาส หรือส่วนแบ่งใช้และขีดจำกัดรีซอร์ส
- การลบคลาส
- กำหนดโปรเซสให้กับคลาสด้วยตนเอง
- การดึงข้อมูลสถิติ WLM

API ช่วยให้แอ็พพลิเคชั่นสามารถตั้งค่าแอ็ททริบิวต์การจัดประเภทที่แอ็พพลิเคชั่น กำหนดซึ่งเรียกว่าแท็ก การตั้งค่าแท็กนี้ โดยใช้ชุดของค่าที่ระบุโดย ผู้ดูแลระบบ (ในเอกสารคู่มือผู้ใช้ของแอ็พพลิเคชั่น) ช่วยให้สามารถ แยกความแตกต่างระหว่าง อินสแตนซ์ต่างๆ ของแอ็พพลิเคชั่นเดียวกัน ด้วยเหตุนี้ คลาส ที่แตกต่างกันจึงอาจถูกจัดประเภทด้วยลิสทรีซอร์สที่แตกต่างกัน ได้

นอกจากนี้ รูทีน `wlm_set_tag` ช่วยให้แอ็พพลิเคชั่นสามารถตั้งค่า แอ็พพลิเคชั่นแท็ก และระบุว่าแท็กนี้ควรสืบทอดโดย โปรเซส ภายนอกที่ `fork` หรือ `exec` หรือไม่ เเรตยังสามารถได้รับการกำหนด แอ็พพลิเคชั่นแท็กด้วยแท็ก `wlm_set_thread` ได้ด้วย แอ็พพลิเคชั่นแท็กของ เเรตสามารถได้รับสืบทอดบนรูทีนย่อย `fork`, `exec` หรือ `pthread_create` ไลบรารีให้การสนับสนุนแอ็พพลิเคชั่นมัลติเเรต 32-บิตหรือ 64-บิต

แอ็พพลิเคชั่นแท็ก:

แอ็พพลิเคชั่นแท็กคือสตริงของอักขระ และใช้เป็น เกณฑ์การจัดประเภทอย่างหนึ่งสำหรับการจัดประเภทอัตโนมัติของโปรเซส หรือเเรต (โดยใช้ไฟล์ `rules`) โดยพื้นฐาน แท็กนี้ นำเสนอเกณฑ์การจัดประเภทที่แอ็พพลิเคชั่นกำหนดเพิ่มเติมจาก เกณฑ์ที่ระบบกำหนด เช่น `user`, `group`, `application` และ `type`

เมื่อแอ็พพลิเคชั่นโปรเซสหรือเเรตตั้งค่าแท็ก แท็กจะถูก จัดประเภทใหม่ในทันทีโดยใช้กฎซูเปอร์คลาสและคลาสย่อยที่บังคับใช้สำหรับ คอนฟิกูเรชัน WLM ที่ใช้อยู่ในปัจจุบัน จากนั้น WLM ทบทวนกฎการกำหนด เพื่อค้นหารายการที่ตรงกัน โดยใช้แอ็ททริบิวต์โปรเซสทั้งหมด รวมถึง แท็กใหม่

เพื่อให้มีผล แท็กนี้ต้องปรากฏขึ้นในกฎการกำหนดตั้งแต่หนึ่งกฎ ขึ้นไป รูปแบบและการใช้งานของแท็กต่างๆ โดยแต่ละแอ็พพลิเคชั่นต้องมีการระบุอย่างชัดเจนในเอกสารคู่มือการจัดการของแอ็พพลิเคชั่นและผู้ดูแลระบบ WLM รับทราบข้อมูลนั้น เป็นอย่างดี เพื่อให้พวกเขาสามารถใช้กฎต่างๆ ของ แท็กในกฎการกำหนดเพื่อแยกความแตกต่างระหว่างอินสแตนซ์ที่แตกต่างกัน ของแอ็พพลิเคชั่นเดียวกัน

เนื่องจากผู้ใช้ที่แตกต่างกันอาจมีความต้องการที่แตกต่างกันเกี่ยวกับชุดของ ลักษณะของแอ็พพลิเคชั่นโปรเซสซึ่งพวกเขา ต้องการใช้เพื่อจัดประเภท โปรเซสเหล่านั้น ขอแนะนำว่าแอ็พพลิเคชั่นควรนำเสนอชุดของคอนฟิกูเรชัน หรือแอ็ททริบิวต์รีน โทม์ซึ่งสามารถนำไปใช้เพื่อสร้างแท็กได้ ผู้ดูแลระบบ แอ็พพลิเคชั่นสามารถระบุรูปแบบของแท็กนี้ในแอ็พพลิเคชั่น แอ็ททริบิวต์ ที่สามารถใช้สำหรับแท็กและไวยากรณ์เพื่อระบุรูปแบบของแท็ก WLM ขึ้นอยู่กับแอ็พพลิเคชั่น และถือเป็นความรับผิดชอบของผู้แนะนำเสนอ แอ็พพลิเคชั่น

ตัวอย่างเช่น อินสแตนซ์ของเซิร์ฟเวอร์ฐานข้อมูลสามารถกำหนด ฐานข้อมูลที่กำลังทำงานอยู่ (`db_name`) และ พอร์ต TCP ซึ่งผู้ใช้ที่กำหนดเชื่อมต่อ (`port_num`) ผู้ดูแลระบบอาจมีลำดับความสำคัญต่อไปนี้:

- สร้างคลาสที่แตกต่างกันสำหรับโปรเซสที่เข้าถึงฐานข้อมูลที่แตกต่างกัน เพื่อให้แต่ละคลาสมีสิทธิในรีซอร์สที่แตกต่างกัน
- แบ่งแยกโปรเซสที่ให้บริการคำร้องขอรีโมตออกจากต้นกำเนิดอื่น และใช้หมายเลขพอร์ตเป็นแอ็ททริบิวต์การจัดประเภท
- สร้างหนึ่งซูเปอร์คลาสสำหรับแต่ละฐานข้อมูล และคลาสย่อยสำหรับแต่ละ หมายเลขพอร์ตในแต่ละซูเปอร์คลาส

วิธีหนึ่งในการตอบสนองความต้องการที่แตกต่างกันเหล่านี้คือ การระบุเนื้อหา และรูปแบบของแท็ก ในตัวอย่างนี้ สมมติว่า สามารถส่งผ่านแท็กไปยัง แอ็พพลิเคชันในไฟล์คอนฟิกูเรชันหรือพารามิเตอร์รันไทม์ เช่น `WLM_TAG=$db_name` หรือ `WLM_TAG=$db_name_$port_num`

เมื่อตั้งค่าแท็ก แอ็พพลิเคชันสามารถระบุแท็กนี้ สืบทอดได้โดยขายนหรือไม เพื่อให้โปรเซสทั้งหมดที่สร้างโดย อินสแตนซ์ของแอ็พพลิเคชันที่ระบุสามารถจัดกลุ่มอยู่ในคลาสเดียวกันได้ การตั้งค่าการสืบทอด แท็กเป็นวิธีการใช้แอ็พพลิเคชันแท็กที่ใช้กันมากที่สุด

ข้อมูลต่อไปนี้เป็นตัวอย่างวิธีการใช้แอ็พพลิเคชันแท็ก ในตัวอย่างนี้ แท็กของฐานข้อมูลเหมือนกับชื่อฐานข้อมูล จากนั้น สองอินสแตนซ์ของเซิร์ฟเวอร์ที่กำลังทำงานบนสองฐานข้อมูลที่แตกต่างกันจะตั้งค่า สองแท็กที่แตกต่างกัน ตัวอย่างเช่น db1 และ db2

ผู้ดูแลระบบสามารถสร้างคลาสที่แตกต่างกันสองคลาสคือ dbserv1 และ dbserv2 และ จัดประเภทสองเซิร์ฟเวอร์ฐานข้อมูล (และขายนทั้งหมดถ้าใช้การสืบทอด แท็ก) ในคลาสเหล่านี้โดยใช้แท็ก จากนั้น อาจให้สิทธิ์ในรีซอร์สที่แตกต่างกันกับแต่ละคลาสตามเป้าหมายธุรกิจเฉพาะ

กฎการกำหนดที่สอดคล้องกันมีลักษณะคล้ายต่อไปนี้:

```
* class resvd user group application type tag
*
dbserv1 - - dbadm /usr/sbin/dbserv - db1
dbserv2 - - dbadm /usr/sbin/dbserv - db2
```

ชนิดอินเตอร์เฟซการเขียนโปรแกรมแอ็พพลิเคชัน:

ข้อมูลต่อไปนี้เป็นชนิดอินเตอร์เฟซการเขียนโปรแกรมแอ็พพลิเคชัน Workload Manager (WLM)

APIs การจัดการคลาส

WLM API ทำให้แอ็พพลิเคชันมีความสามารถ:

- เคียวยริชื่อและลักษณะของคลาสที่มีอยู่ของคอนฟิกูเรชัน WLM ที่กำหนด (`wlm_read_classes`)
- สร้างคลาสใหม่สำหรับคอนฟิกูเรชัน WLM ที่กำหนด กำหนดค่าของ แอ็ตทริบิวต์ต่างๆ ของคลาส (เช่น ระดับและการสืบทอด) และ ส่วนแบ่งใช้และขีดจำกัดสำหรับรีซอร์สที่จัดการโดย WLM เช่น CPU, หน่วยความจำฟิสิคัล และบล็อก I/O (`wlm_create_class`)
- เปลี่ยนลักษณะของคลาสที่มีอยู่ของคอนฟิกูเรชัน WLM ที่กำหนด รวมถึงแอ็ตทริบิวต์คลาส และส่วนแบ่งใช้และขีดจำกัดรีซอร์ส (`wlm_change_class`)
- ลบคลาสที่มีอยู่ของคอนฟิกูเรชันที่กำหนด (`wlm_delete_class`)

การเปลี่ยนแปลงจะใช้กับไฟล์คุณสมบัติของคอนฟิกูเรชัน WLM ที่ระบุเท่านั้น ทางเลือกคือ โดยการระบุสตริงว่าง เป็นชื่อคอนฟิกูเรชัน ทำให้สามารถใช้การเปลี่ยนแปลงกับคลาส in-core เท่านั้น ส่งผลให้เกิดการ อัปเดตสถานะของคอนฟิกูเรชันที่ใช้งานอยู่ในทันที

การเรียก API จากผู้เรียก ต้องการเอกสิทธิ์ระดับเดียวกันกับที่อาจต้องการ สำหรับบรรทัดคำสั่ง หรือสำหรับ SMIT หรืออินเตอร์เฟซดังต่อไปนี้:

- ผู้ใช้ทุกราชสามารถอ่านชื่อคลาสและลักษณะได้
- เฉพาะผู้ไครกสามารถสร้าง แก้ไข หรือลบซูเปอร์คลาสได้

- เฉพาะผู้ใช้งานหรือผู้ดูแลซูเปอร์คลาสที่แต่งตั้งเท่านั้น (แอดมินิวิวด์ ซูเปอร์คลาส `adminuser` หรือ `admingroup`) สามารถสร้าง แก้ไข หรือลบคลาสย่อยของซูเปอร์คลาสที่กำหนดได้

ในกรณีที่ทำการจัดการ WLM ทั้งผ่านทาง บรรทัดคำสั่งและเครื่องมือการจัดการโดยผู้ดูแลระบบ WLM และโดยแอ็พพลิเคชัน ผ่านทาง API มีข้อควรระวังบางอย่าง ทั้งสองอินเตอร์เฟซแบ่งใช้พื้นที่ชื่อเดียวกัน สำหรับชื่อซูเปอร์คลาสและคลาสย่อย และจำนวนทั้งหมดของ ซูเปอร์คลาสและคลาสย่อย

นอกจากนี้ เมื่อ API แก้ไขข้อมูล in-core WLM โดยตรง (ตัวอย่างเช่น สร้างคลาสใหม่) ผู้ดูแลระบบ WLM ไม่ทราบถึงการดำเนินการนี้จนกว่าคลาสที่ API ดำเนินการไม่ปรากฏขึ้นบนเอาต์พุต ของคำสั่ง เช่น คำสั่ง `wlmstat` เพื่อหลีกเลี่ยง ความขัดแย้งที่อาจทำให้เกิดความสับสนแก่แอ็พพลิเคชันที่ใช้ API นี้เมื่อผู้ดูแลระบบอัปเดต WLM คลาสที่สร้างขึ้นผ่านทาง API ซึ่งไม่ได้กำหนดไว้ในไฟล์คุณสมบัติ WLM จะไม่ถูกลบออกจากข้อมูล in-core โดย อัตโนมัติ คลาสเหล่านี้จะยังคงใช้งานได้จนกว่าจะถูกลบออกอย่างชัดแจ้งผ่านทางรูทีน `wlm_delete_class` หรือผ่านทาง การเรียกใช้คำสั่ง `rmclass` (เรียกโดยตรงหรือผ่าน SMIT โดยผู้ดูแลระบบ)

WLM API ยังทำให้แอ็พพลิเคชันมีความสามารถ:

- เคียวีรีหรือเปลี่ยนโหมดการดำเนินงานของ WLM โดยใช้ฟังก์ชัน `wlm_set`
- เคียวีรีสถานะปัจจุบันของ WLM
- หยุด WLM
- สลับระหว่างโหมด `active` และ `passive`
- เปิดและปิดการผูก `rset`
- เริ่มต้นหรืออัปเดต WLM ด้วยคอนฟิกูเรชันปัจจุบันหรือคอนฟิกูเรชันอื่น โดยใช้รูทีน `wlm_load`
- กำหนดโปรเซสหรือกลุ่มของโปรเซสให้กับคลาสโดยใช้รูทีน `wlm_assign`

API ต้องการเอกสิทธิ์ระดับเดียวกับคำสั่ง `wlmcntrl` และ `wlmassign` ที่สอดคล้องกัน:

- ผู้ใช้ทุกรายสามารถเคียวีรีสถานะของ WLM ได้
- เฉพาะผู้ใช้งานเท่านั้นสามารถเปลี่ยนโหมดการดำเนินงานของ WLM ได้
- เฉพาะผู้ใช้งานเท่านั้นสามารถอัปเดตหรือรีเฟรชคอนฟิกูเรชันทั้งหมดได้
- ผู้ดูแลระบบรากหรือซูเปอร์คลาสที่ได้รับอนุญาต (`adminuser` หรือ `admingroup`) สามารถอัปเดต WLM สำหรับ คลาสย่อยของซูเปอร์คลาสที่กำหนด
- ผู้ใช้งานหรือผู้ใช้ที่ได้รับอนุญาต (ที่ระบุโดย `authuser` หรือ `authgroup`) หรือผู้ดูแลซูเปอร์คลาสที่ได้รับอนุญาต (`adminuser` หรือ `admingroup`) สามารถกำหนดโปรเซสให้กับซูเปอร์คลาสหรือคลาสย่อย

API สถิติ WLM

รูทีน WLM API และ `wlm_get_bio_stats` ช่วยให้ แอ็พพลิเคชันเข้าถึงสถิติ WLM ที่แสดงขึ้นโดยคำสั่ง `wlmstat`

API การจัดประเภท WLM

รูทีน `wlm_check` ช่วยให้ผู้ใช้สามารถ ตรวจสอบนิยามคลาสและกฎการกำหนดสำหรับคอนฟิกูเรชัน WLM ที่กำหนด รูทีน API `wlm_classify` ช่วยให้ แอ็พพลิเคชันสามารถกำหนดคลาสซึ่งควรจัดประเภทโปรเซสที่มี ชุดของแอ็พทริ บิวต์ที่ระบุ

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง wlmassign

ความเข้ากันได้ของไบนารี:

เพื่อให้มีความเข้ากันได้ของไบนารีถ้ามีการเปลี่ยนแปลงในโครงสร้างข้อมูลในอนาคต การเรียก API แต่ละรายการจึงมีการส่งผ่านหมายเลขเวอร์ชันเป็นพารามิเตอร์

การทำเช่นนี้ช่วยให้ไลบรารีสามารถทราบถึงเวอร์ชันของโครงสร้างข้อมูลที่ใช้ในการสร้างแอปพลิเคชันได้

ตัวอย่างของการจัดประเภท Workload Manager กฎ และขีดจำกัด

มีหลายเมธอดสำหรับการจัดประเภทโปรเซส และเมธอดทั้งหมดดำเนินงานพร้อมกัน

ใช้ขั้นตอนวิธีการจับคู่รายการแรกตรงกันจากบนลงล่างเพื่อให้ คอนฟิกูเรชันมีความยืดหยุ่นสูงสุด คุณสามารถจัดระเบียบการจัดกลุ่มโปรเซส โดยผู้ใช้ที่มีตัวพิมพ์พิเศษสำหรับโปรแกรมที่มีชื่อ หรือโดยชื่อพาธที่มี ตัวพิมพ์พิเศษสำหรับผู้ใช้เฉพาะ หรือการจัดเรียงอื่นใด

ตัวอย่างของกฎการกำหนด Workload Manager:

ตัวอย่างนี้แสดงไฟล์ rules ระดับบนสุดสำหรับคอนฟิกูเรชัน Config (ไฟล์ /etc/wlm/Config/rules)

```
* This file contains the rules used by WLM to
* assign a process to a superclass
*
* class resvd user      group  application      type  tag
db1      -      -      -      /usr/bin/oracle*  _DB1
db2      -      -      -      /usr/bin/oracle*  _DB2
devlt    -      -      dev    -                -     -
VPs      -      bob,ted -      -                -     -
acctg    -      -      acct*  -                -     -
System  -      root   -      -                -     -
Default -      -      -      -                -     -
```

ข้อมูลต่อไปนี้เป็นตัวอย่างของไฟล์ rules สำหรับซูเปอร์คลาส devlt ในไฟล์ /etc/wlm/Config/devlt/rules:

```
* This file contains the rules used by WLM to
* assign a process to a subclass of the
* superclass devlt
*
* class resvd user      group  application      type  tag
hackers  -      jim,liz -      -                -     -
hogs     -      -      -      -                64bit+plock -
editors  -      !sue   -      /bin/vi,/bin/emacs -     -
build    -      -      -      /bin/make,/bin/cc -     -
Default  -      -      -      -                -     -
```

หมายเหตุ: เครื่องหมายดอกจัน (*) เป็นอักขระข้อคิดเห็นที่ใช้ในไฟล์ rules

ข้อมูลต่อไปนี้เป็นตัวอย่างการใช้ไฟล์กฎนี้ ตัวอย่างต่อไปนี้จะสมมติว่าซูเปอร์คลาสและคลาสย่อยที่อธิบายไม่ได้มีการตั้งค่าแอตทริบิวต์การสืบทอดเป็นใช่ หากเปิดใช้งานการสืบทอด โปรเซสใหม่ จะได้รับสืบทอดซูเปอร์คลาสและคลาสย่อยจากโปรเซสพารেন্ট

- หากผู้ใช้joe จากกลุ่ม acct3 ดำเนินการ /bin/vi โปรเซสจะถูกวางไว้ในซูเปอร์คลาส acctg

- ถ้าผู้ใช้ sue จากกลุ่ม dev ดำเนินการ `/bin/emacs` โพรเซสจะถูกวางไว้ในซูเปอร์คลาส devIt (ID กลุ่มตรงกัน) แต่จะไม่ถูกจัดประเภทเข้าในคลาสย่อย editors เนื่องจากผู้ใช้ถูกแยกออกจาก คลาสนั้น โพรเซสจะอยู่ใน devIt โดยค่าดีฟอลต์
- เมื่อผู้ดูแลฐานข้อมูลที่มี ID ผู้ใช้เป็น oracle และ ID กลุ่ม เป็น dbm เริ่มต้น `/usr/sbin/oracle` เพื่อให้บริการฐานข้อมูล DB1 โพรเซสจะถูกจัดประเภทเข้าในซูเปอร์คลาส Default เฉพาะเมื่อโพรเซส ตั้งค่าแท็กเป็น `_DB1` เท่านั้น โพรเซสจะถูกกำหนดให้กับซูเปอร์คลาส db1

ตัวอย่างของคลาส Workload Manager ที่มี ส่วนแบ่งใช้และขีดจำกัด:

สำหรับตัวอย่างนี้ สมมติว่าคลาส A, B, C, และ D มี ส่วนแบ่งใช้ 3, 2, 1, และ 1 ตามลำดับ

ถ้าคลาส A, C, และ D ใช้งานอยู่ เป้าหมายที่คำนวณจะเป็น:

$$\text{target(A)} = 3/5 = 60\%$$

$$\text{target(C)} = 1/5 = 20\%$$

$$\text{target(D)} = 1/5 = 20\%$$

หากในระหว่างการทดสอบ พบว่าแอพลิเคชันในคลาส A ทำงานได้ถูกต้องเมื่อสามารถใช้ 50% ของรีซอร์ส อาจต้องการทำให้ อีก 50% ที่เหลือของรีซอร์สมียอยู่ในคลาสอื่น ซึ่งสามารถทำได้โดยการกำหนดค่าสูงสุดซอฟต์แวร์ให้กับคลาส A เป็น 50% สำหรับ รีซอร์สนี้ เนื่องจากเป้าหมายที่คำนวณได้ปัจจุบันคือ 60% เกินกว่าขีดจำกัดนี้ ค่าที่คำนวณได้จะถูกปรับลงเป็นค่าสูงสุดซอฟต์แวร์ เมื่อเป็นเช่นนี้ ปริมาณการใช้ เป้าหมายหรือจริง (ค่าใดที่ต่ำกว่า) ของคลาส A จะถูกลบออกจากจำนวนของรีซอร์สที่มีอยู่ เนื่องจากขณะนี้ คลาสนี้มีเป้าหมายที่ถูกจำกัด โดยขีดจำกัด (ไม่ใช่ส่วนแบ่งใช้) ส่วนแบ่งใช้ของคลาสจะถูก ลบออกจากจำนวนของส่วนแบ่งใช้ที่ใช้งานอยู่ด้วย สมมติว่าคลาส A มี ปริมาณการใช้ปัจจุบันเป็น 48% ขณะนี้เป้าหมายจะเป็น:

$$\text{target(A)} = 3/5 = 60\%, \text{ softmax} = 50, = 50\%$$

$$\text{target(C)} = 1/2 * (100 - 48) = 26\%$$

$$\text{target(D)} = 1/2 * (100 - 48) = 26\%$$

บางครั้งในเวลาต่อมา คลาสทั้งหมด อาจใช้งานอยู่ และเป้าหมายจะถูกปรับโดยอัตโนมัติอีกครั้ง:

$$\text{target(A)} = 3/7 = 42\%$$

$$\text{target(B)} = 2/7 = 28\%$$

$$\text{target(C)} = 1/7 = 14\%$$

$$\text{target(D)} = 1/7 = 14\%$$

ตัวอย่างของคลาส Workload Manager ที่มี ขีดจำกัด CPU:

ตัวอย่างนี้ตรวจสอบการจัดสรร CPU โดยสมมติว่า แต่ละคลาสสามารถใช้ CPU ทั้งหมดที่กำหนดให้

สองคลาสคือ A และ B อยู่ในระดับเดียวกัน ขีดจำกัด CPU สำหรับ A คือ [30% - 100%] ขีดจำกัด CPU สำหรับ B คือ [20% - 100%] เมื่อทั้งสองคลาสกำลังรัน และกำลังใช้ CPU อย่างเพียงพอ อันดับแรก WLM ตรวจสอบให้แน่ใจว่าทั้งสอง คลาสได้รับเปอร์เซ็นต์ต่ำสุดของแต่ละวินาที (เฉลี่ยจากหลายวินาที) จากนั้น WLM แจกจ่ายรอบ CPU ที่เหลือตามค่าส่วนแบ่งใช้เป้าหมายของ CPU ใดๆ

หากส่วนแบ่งใช้เป้าหมาย CPU สำหรับ A และ B เป็น 60% และ 40% ตามลำดับ การใช้ CPU สำหรับ A และ B จะคงที่ในระดับ 60% และ 40% ตามลำดับ

มีการเพิ่มคลาสที่สามคือ C คลาสนี้เป็นกลุ่มของงานที่โยงกับ CPU และควรรันด้วย CPU ที่มีอยู่จำนวนครึ่งหนึ่ง (หรือมากกว่า) คลาส C มีขีดจำกัดเป็น [20% - 100%] และส่วนแบ่งใช้เป้าหมาย CPU เป็น 100% ถ้า C อยู่ในระดับเดียวกันกับ A และ B ผลคือเมื่อ C กำลังเริ่มต้น A และ B จะเห็นว่าการจัดสรร CPU ของตนลดลงอย่างมาก และสามคลาสจะคงที่ในระดับ 30%, 20% และ 50% ตามลำดับ เป้าหมายในกรณีนี้เป็นค่าต่ำสุดสำหรับ A และ B ด้วย

ผู้ดูแลระบบอาจไม่ต้องการให้งานแบตช์ใช้ 50% ของ CPU เมื่องานอื่นที่อาจมีลำดับความสำคัญสูงกว่ากำลังรันอยู่เช่นกัน ในสถานการณ์เช่นตัวอย่างก่อนหน้านี้ C จะถูกวางไว้ในระดับที่มีระดับความสำคัญต่ำกว่า ผลคือ C จะได้รับ CPU จำนวนใดก็ตามที่เหลืออยู่หลังจาก A และ B ได้รับ CPU ตามจำนวนที่ต้องการแล้ว ในตัวอย่างข้างบน C ไม่ได้รับเวลา CPU เนื่องจาก A และ B ต่างก็สามารถใช้ได้มากถึง 100% ของ CPU อย่างไรก็ตาม ในสถานการณ์ส่วนใหญ่ A และ B ในระดับที่มีลำดับความสำคัญสูงประกอบด้วยงานแบบโต้ตอบหรือที่มุ่งเน้นธุรกรรม ซึ่งไม่ได้ใช้ CPU ทั้งหมดตลอดเวลา ผลคือ C ได้รับ CPU บางส่วน ซึ่งช่วงชิงกับคลาสอื่นในระดับเดียวกันหรือระดับที่ต่ำกว่า

ตัวอย่างของคลาส Workload Manager ที่มี ขีดจำกัดหน่วยความจำ:

ตัวอย่างนี้ตรวจสอบการจัดสรรหน่วยความจำให้กับกลุ่ม ของโปรเซสที่มีเป้าหมายหน่วยความจำแตกต่างกัน

ต้องรันโปรเซสสามกลุ่ม: กลุ่มของโปรเซสแบบโต้ตอบ ที่ต้องรันในทุกเมื่อที่มีการใช้งาน (PEOPLE), งานแบตช์ที่รันใน พื้นหลังเสมอ (BATCH1) และงานแบตช์ตัวที่สองที่มีความสำคัญมากกว่า ซึ่งรันทุกคืน (BATCH0)

PEOPLE มีค่าต่ำสุดหน่วยความจำที่ระบุเป็น 20%, เป้าหมายหน่วยความจำ เป็น 50 ส่วนแบ่งใช้ และค่าระดับคลาสเป็น 1 ขีดจำกัดต่ำสุด 20% ทำให้มั่นใจว่า เดสก์ท็อปแอ็พพลิเคชันในคลาสนี้ดำเนินการต่อค่อนข้างรวดเร็วเมื่อผู้ใช้สัมผัส คีย์บอร์ด

BATCH1 มีค่าต่ำสุดหน่วยความจำเป็น 50%, เป้าหมายหน่วยความจำเป็น 50 ส่วนแบ่งใช้ และค่าระดับเป็น 3

BATCH0 มีค่าต่ำสุดหน่วยความจำเป็น 80%, เป้าหมายหน่วยความจำเป็น 50 ส่วนแบ่งใช้ และค่าระดับเป็น 2

คลาส PEOPLE และ BATCH1 มีขีดจำกัดค่าต่ำสุดหน่วยความจำทั้งหมด เป็น 70 ภายใต้การดำเนินงานปกติ (เมื่อ BATCH0 ไม่ได้กำลังรัน) ทั้งสองคลาสเหล่านี้ได้รับอนุญาตให้ใช้หน่วยความจำที่สงวนไว้ทั้งหมดได้ ทั้งสองคลาสแบ่งใช้หน่วยความจำ ส่วน ที่เหลือในเครื่องประมาณครึ่งต่อครึ่ง แม้ว่าอยู่ในระดับที่แตกต่างกัน ตอนเที่ยงคืนเมื่อ BATCH0 เริ่มต้นขึ้น ค่าต่ำสุด หน่วยความจำทั้งหมดขึ้นถึง 150 WLM ละเว้นข้อกำหนดค่าต่ำสุดสำหรับระดับต่ำสุดจนกว่ามีโปรเซสในระดับที่สูงกว่า BATCH0 ใช้หน่วยความจำจากส่วนสำรอง BATCH1 50% แต่ไม่ใช่จากส่วนสำรอง PEOPLE 20% หลังจาก BATCH0 เสร็จ แล้ว จะมีการสำรองหน่วยความจำ สำหรับโปรเซสระดับ 3 อีกครั้งและระบบกลับเข้าสู่ภาวะสมดุลของหน่วยความจำ ปกติ

คำสั่ง Workload Manager

WLM นำเสนอคำสั่งที่ช่วยให้ผู้ดูแลระบบสามารถทำฟังก์ชัน ได้จำนวนมาก

ซึ่งรวมถึง:

- สร้าง แก้ไข และลบซูเปอร์คลาสและคลาสย่อยโดยใช้คำสั่ง `mkclass`, `chclass`, และ `rmclass` คำสั่งเหล่านี้อัปเดต *คลาส, ส่วนแบ่งใช้ และ ขีดจำกัด* ของไฟล์
- เริ่มต้น หยุด และอัปเดต WLM โดยใช้คำสั่ง `wlmentrl`
- ตรวจสอบไฟล์คุณสมบัติ WLM สำหรับคอนฟิกูเรชันที่กำหนด และพิจารณาว่า โปรเซสที่มีชุดของแอ็ททริบิวต์ที่กำหนดถูก กำหนดให้กับคลาสใด (ซูเปอร์คลาสและ คลาสย่อย) โดยใช้คำสั่ง `wlmcheck`

- มอนิเตอร์การใช้ซอร์สต่อคลาสโดยใช้คำสั่ง `wlmsat` (ASCII) เครื่องมือประสิทธิภาพส่วนใหญ่ เช่น เครื่องมือที่เริ่มต้นโดยคำสั่ง `svmon` และ `topas` มีส่วนขยายที่พิจารณาถึงคลาส WLM และนำเสนอข้อมูลสถิติต่อคลาส และต่อระดับโดยใช้ชื่อชั้นบรรทัดคำสั่งใหม่
- แฟล็กในคำสั่ง `ps` ช่วยให้ผู้ใช้สามารถแสดง คลาสซึ่งมีโปรเซสและแอ็พพลิเคชันแท็ก ผู้ใช้ยังสามารถใช้คำสั่ง `ps` เพื่อแสดงรายการโปรเซสทั้งหมดที่เป็นสมาชิกของซูเปอร์คลาส หรือคลาสย่อยที่กำหนดได้ด้วย
- ไม่มีอินเตอร์เฟซบรรทัดคำสั่งในการจัดการกฎการกำหนด คุณต้องใช้เครื่องมือดูแลระบบ SMIT หรือโปรแกรมแก้ไขข้อความ

การติดตั้งอุปกรณ์ IDE

คุณสามารถติดตั้งอุปกรณ์ IDE บนระบบของคุณได้ โพรซีเจอร์การติดตั้ง อุปกรณ์ IDE แบ่งออกเป็นหลายภารกิจซึ่งต้องทำตามลำดับ

ข้อกำหนดเบื้องต้นสำหรับการติดตั้งอุปกรณ์ IDE คือ:

- คุณต้องสามารถเข้าถึงคู่มือของผู้ใช้งานสำหรับยูนิตรระบบของคุณและ คู่มือการติดตั้งสำหรับอุปกรณ์ที่จะติดตั้ง เอกสารคู่มือต้อง ระบุวิธีการตั้งค่าจัมเปอร์อุปกรณ์ IDE เพื่อกำหนดคอนฟิกอุปกรณ์เป็นค่าติดตั้ง master หรือ slave อย่างใดอย่างหนึ่ง
- ต้องมี ID อุปกรณ์ IDE ที่ไม่ได้ใช้อย่างน้อยหนึ่ง ID บนอะแดปเตอร์ IDE บน ระบบ
- หากคุณกำลังอัปเดตดิสเก็ตทอพอโลยีผลิตภัณฑ์ คุณต้องใช้ดิสเก็ต ระบบทอพอโลยีผลิตภัณฑ์ ซึ่งเก็บไว้พร้อมกับเร็กคอร์ดที่สำคัญของระบบ และดิสเก็ต อัปเดตทอพอโลยีระบบ ซึ่งจัดส่งมาพร้อมกับอุปกรณ์
- ตรวจสอบว่าอินเตอร์เฟซของอุปกรณ์เข้ากันได้กับอินเตอร์เฟซ ของ IDE คอนโทรลเลอร์บนยูนิตรระบบ
- มีสองประเภทสำหรับอุปกรณ์ IDE, ATA และ ATAPI ATA คือ อุปกรณ์ดิสก์ และ ATAPI คืออุปกรณ์ซีดีรอมหรือเทปอนุญาตให้เชื่อมต่อได้สูงสุดสองอุปกรณ์ สำหรับแต่ละ IDE คอนโทรลเลอร์ หนึ่ง master และหนึ่ง slave โดยปกติ อะแดปเตอร์ IDE หนึ่งตัวมีคอนโทรลเลอร์สองตัว ซึ่งทำให้สามารถต่อพ่วงอุปกรณ์ IDE ได้สูงสุดสี่ตัวด้วยการเดินสายเคเบิลที่เหมาะสม คุณสามารถต่อพ่วงชุดอุปกรณ์ใดๆ ต่อ ไปนี้เข้ากับคอนโทรลเลอร์เดียว:

- 1 ATA อุปกรณ์เป็น master
- 1 ATAPI อุปกรณ์เป็น master
- 2 ATA อุปกรณ์เป็น master และ slave
- 1 ATA อุปกรณ์เป็น master และ 1 ATAPI อุปกรณ์เป็น slave
- 2 ATAPI อุปกรณ์เป็น master และ slave

คุณไม่สามารถต่อพ่วงดังต่อไปนี้:

- 1 ATA อุปกรณ์เป็น slave อย่างเดียว
- 1 ATAPI อุปกรณ์เป็น slave อย่างเดียว
- 1 ATAPI อุปกรณ์เป็น master และ 1 ATA อุปกรณ์เป็น slave

ทำภารกิจต่อไปนี้ให้สมบูรณ์เพื่อติดตั้งอุปกรณ์ IDE

การกำหนดหมายเลขและที่ตั้งของ IDE คอนโทรลเลอร์

คุณสามารถกำหนดจำนวน IDE คอนโทรลเลอร์ที่จะต่อพ่วง เข้ากับยูนิตรระบบ และตำแหน่งของ IDE คอนโทรลเลอร์ได้อะแด็ปเตอร์ IDE อาจอยู่ในสล็อตอะแด็ปเตอร์หรืออยู่ในตัว planar ระบบ โปรดจำไว้ว่าอะแด็ปเตอร์ IDE มีสอง IDE คอนโทรลเลอร์ (IDE บัส) ดังนั้นจึงพบ IDE คอนโทรลเลอร์สองตัวในสล็อตอะแด็ปเตอร์หรือในตัว planar ระบบ

คุณสามารถได้รับหมายเลขและที่ตั้งของ IDE คอนโทรลเลอร์ โดยใช้เมธอดอย่างใดอย่างหนึ่งจากสองเมธอดต่อไปนี้:

- การใช้คำสั่งคอนฟิกูเรชันซอฟต์แวร์ เมธอดนี้มีอยู่เฉพาะถ้า ระบบปฏิบัติการมีการติดตั้งบนยูนิตรระบบแล้วเท่านั้น
- การใช้เอกสาร *เกี่ยวกับเครื่องของคุณ* ที่จัดส่งมาพร้อมกับยูนิตรระบบ ของคุณ เมธอดนี้ใช้ได้สำหรับเซิร์ฟเวอร์และการติดตั้งครั้งแรกของยูนิตรระบบใหม่เท่านั้น

การใช้คำสั่งคอนฟิกูเรชันซอฟต์แวร์

เมธอดนี้ใช้กับระบบที่ติดตั้งระบบปฏิบัติการแล้ว

เมื่อต้องการแสดง รายการ IDE I/O คอนโทรลเลอร์บนระบบ ให้พิมพ์คำสั่งต่อไปนี้:

```
lscfg -l ide*
```

ตรวจสอบ รายการของ IDE คอนโทรลเลอร์ที่แสดงขึ้น การแสดงตัวอย่างต่อไปนี้ จากคำสั่ง `lscfg -l ide` แสดงสอง IDE I/O คอนโทรลเลอร์ คอนโทรลเลอร์ `ide0` และ `ide1` อยู่บน planar ระบบ ตัวบ่งชี้ planar เป็นตำแหน่งที่สอง ในค่าที่ตั้งซึ่งมีค่าเป็น 1

DEVICE	LOCATION	DESCRIPTION
ide0	01-00-00	ATA/IDE Controller Device
ide1	01-00-01	ATA/IDE Controller Device

2nd digit is the adapter slot number
6th digit indicates the controller number.

เซิร์ฟเวอร์แรกเริ่ม

ใช้เอกสาร *เกี่ยวกับเครื่องของคุณ* เพื่อกำหนด IDE I/O คอนโทรลเลอร์บน ระบบถ้ากำลังติดตั้งอุปกรณ์ในระหว่างเซิร์ฟเวอร์แรกเริ่ม

หมายเหตุ: ผลลัพธ์ ไม่ถูกต้องถ้าคอนโทรลเลอร์ถูกเพิ่มหลังจากจัดระบบ จากโรงงานแล้ว

พิจารณาว่ายูนิตรระบบ มี IDE คอนโทรลเลอร์ในตัวบอร์ด planar หรือไม่ IDE I/O คอนโทรลเลอร์ในตัวเป็นอุปกรณ์มาตรฐาน บนบางยูนิตรระบบ ยูนิตรระบบของคุณ มี IDE คอนโทรลเลอร์ในตัวถ้าเอกสาร *เกี่ยวกับเครื่องของคุณ* แสดงอุปกรณ์ IDE สล็อตภายในที่มีหมายเลขสล็อตวาง

การเลือก IDE คอนโทรลเลอร์และ IDE แอดเดรสบนคอนโทรลเลอร์

คุณสามารถเลือก IDE คอนโทรลเลอร์และ IDE แอดเดรสบนคอนโทรลเลอร์

หลังจากระบุ IDE คอนโทรลเลอร์ที่ต่อพ่วงเข้ากับ ยูนิตรระบบแล้ว ให้เลือก IDE I/O คอนโทรลเลอร์ซึ่งคุณต้องการเชื่อมต่อ อุปกรณ์ IDE I/O คอนโทรลเลอร์นี้ต้องมีหนึ่งค่าที่ตั้ง IDE เป็นอย่างน้อยที่ยังไม่ได้กำหนดให้กับอุปกรณ์อื่น

กำหนดว่าต้องจัดจัมเปอร์ค่าติดตั้งอุปกรณ์ IDE เป็น master หรือ slave หากไม่มีอุปกรณ์ที่ต่อพ่วงกับคอนโทรลเลอร์ในปัจจุบัน ต้องตั้งค่าจัมเปอร์อุปกรณ์ IDE เป็น master (บางอุปกรณ์ไม่ต้องการค่าติดตั้ง ID อุปกรณ์ในสถานการณ์นี้) หากอุปกรณ์ IDE ต่อพ่วงแล้ว ต้องกำหนดชนิดของอุปกรณ์ ดิสก์เป็นอุปกรณ์ ATA ซีดีรอม และเทปเป็นอุปกรณ์ ATAPI ถ้าทั้งอุปกรณ์ ATA และ ATAPI ต่อพ่วงกับ IDE คอนโทรลเลอร์ตัวเดียวกัน ต้องตั้งค่าอุปกรณ์ ATA เป็น master ID และ ต้องตั้งค่าอุปกรณ์ ATAPI เป็น slave ID

กำหนดว่าอุปกรณ์ IDE ใดต่อพ่วงกับ คอนโทรลเลอร์โดยการดูข้อมูลเกี่ยวกับอุปกรณ์ที่เชื่อมต่อกับ IDE คอนโทรลเลอร์แล้ว

คุณสามารถใช้สองเมธอดในการเลือก IDE I/O คอนโทรลเลอร์และ IDE แอดเดรสบนคอนโทรลเลอร์ซึ่งยังไม่ได้กำหนดให้กับ อุปกรณ์อื่น:

- การใช้คำสั่งคอนฟิกูเรชันซอฟต์แวร์ถ้าติดตั้งระบบปฏิบัติการ บนยูนิตรบบแล้ว
- การใช้เอกสาร *เกี่ยวกับเครื่องของคุณ* สำหรับเชื่อมต่อและการติดตั้งครั้งแรก ของยูนิตรบบใหม่

การใช้คำสั่งคอนฟิกูเรชันซอฟต์แวร์

เมธอดนี้ ใช้กับระบบที่ติดตั้งระบบปฏิบัติการแล้ว

1. พิมพ์คำสั่งต่อไปนี้เพื่อแสดงรายการอุปกรณ์ IDE ทั้งหมดที่กำหนดในปัจจุบัน:

```
lsdev -C -s ide -H
```

2. ตรวจสอบรายการของอุปกรณ์ที่กำหนดให้กับ IDE คอนโทรลเลอร์แต่ละตัวแล้ว แต่ละแถวในจอแสดงผลนี้แสดงชื่อโลจิคัล สถานะ ที่ตั้ง และ คำอธิบายของอุปกรณ์ IDE ที่ตั้งของแต่ละอุปกรณ์ขึ้นต้นด้วย ที่ตั้งของคอนโทรลเลอร์ซึ่งอุปกรณ์ เชื่อมต่อ ในตัวอย่าง ข้างล่าง IDE I/O คอนโทรลเลอร์ที่มีแอดเดรส 01-00-00 มีอุปกรณ์ IDE ที่ต่อพ่วงสองตัว IDE I/O คอนโทรลเลอร์ที่มีที่ตั้ง 01-00-01 มีอุปกรณ์ IDE ที่ต่อพ่วงหนึ่งตัว

name	status	location	description
hdisk0	Available	01-00-00-00	720 MB IDE Disk Drive
hdisk1	Available	01-00-00-01	540 MB IDE Disk Drive
cd0	Available	01-00-01-00	IDE CD-ROM Drive

|
IDE controller address (6th digit)

3. เลือกคอนโทรลเลอร์ที่ไม่มีอุปกรณ์ IDE เชื่อมต่อแล้ว สองตัว
4. หากอุปกรณ์หนึ่งต่อพ่วงกับคอนโทรลเลอร์แล้ว ให้พิจารณา ชนิดของอุปกรณ์ และกำหนดชนิดของอุปกรณ์ที่จะติดตั้ง อุปกรณ์ดิสก์มีการจัดประเภทเป็นอุปกรณ์ ATA อุปกรณ์ซีดีรอมและเทปมีการจัดประเภทเป็น อุปกรณ์ ATAPI
5. กำหนดค่าติดตั้งจัมเปอร์ IDE สำหรับอุปกรณ์ใหม่ ขึ้นอยู่กับชุด ของอุปกรณ์ที่จะเชื่อมต่อกับ IDE คอนโทรลเลอร์ หาก อุปกรณ์ใหม่ เป็นเพียงอุปกรณ์เดียวที่เชื่อมต่อกับคอนโทรลเลอร์ ต้องตั้งค่าติดตั้งจัมเปอร์ อุปกรณ์เป็นตำแหน่ง master (บางอุปกรณ์ไม่ต้องการค่าติดตั้งในกรณี นี้) ถ้าอุปกรณ์ทั้งสองตัวเป็นชนิดเดียวกัน สามารถตั้งค่าติดตั้งจัมเปอร์อุปกรณ์ใหม่ เป็นตำแหน่ง slave ถ้าใช้อุปกรณ์ผสมกัน (ATA และ ATAPI) ต้องตั้งค่าจัมเปอร์อุปกรณ์ ATA เป็น ตำแหน่ง master และต้องตั้งค่าจัมเปอร์ อุปกรณ์ ATAPI เป็นตำแหน่ง slave ถ้ามีชนิดอุปกรณ์ผสมกันและ อุปกรณ์ใหม่ เป็นอุปกรณ์ ATA (ดิสก์) ต้องเปลี่ยนจัมเปอร์อุปกรณ์สำหรับอุปกรณ์ ATAPI ที่มีอยู่ในปัจจุบันเป็นตำแหน่ง slave และ ต้องตั้งค่าจัมเปอร์อุปกรณ์ ATA ใหม่เป็น master ถ้ามีชนิดอุปกรณ์ผสมกันและ อุปกรณ์ใหม่เป็นอุปกรณ์ ATAPI (ซีดีรอมหรือเทป) ต้องตั้งค่าจัมเปอร์อุปกรณ์ สำหรับอุปกรณ์ ATAPI ใหม่เป็น slave และถ้าอุปกรณ์ ATA ไม่มีค่าติดตั้ง จัมเปอร์ในปัจจุบัน ต้องตั้งค่าอุปกรณ์ ATA เป็น master

เชื่อมต่อแรกเริ่ม

ใช้เอกสาร *เกี่ยวกับเครื่องของคุณ* เพื่อกำหนดอุปกรณ์ที่กำหนดให้กับ IDE I/O คอนโทรลเลอร์บน ระบบที่กำลังติดตั้ง อุปกรณ์ในระหว่างเซ็ทอัพแรกเริ่ม

หมายเหตุ: ผลลัพธ์ ไม่ถูกต้องถ้าคอนโทรลเลอร์ถูกเพิ่มหลังจากจัดส่งระบบ จากโรงงานแล้ว

1. เมื่อต้องการกำหนดอุปกรณ์ IDE ที่กำหนดให้กับแอดเดรสบน IDE คอนโทรลเลอร์ ให้ดูที่ "อุปกรณ์สื่อบันทึกภายใน" ใน *เกี่ยวกับเครื่องของคุณ*
2. เลือกคอนโทรลเลอร์ที่ไม่มีอุปกรณ์ IDE เชื่อมต่อแล้วสองตัว
3. หากอุปกรณ์หนึ่งต่อพ่วงกับคอนโทรลเลอร์แล้ว ให้พิจารณา ชนิดของอุปกรณ์ และกำหนดชนิดของอุปกรณ์ที่จะติดตั้ง อุปกรณ์ดิสก์มีการจัดประเภทเป็นอุปกรณ์ ATA อุปกรณ์ซีดีรอมและเทปมีการจัดประเภทเป็น อุปกรณ์ ATAPI
4. กำหนดค่าติดตั้งจัมเปอร์ IDE สำหรับอุปกรณ์ใหม่ ขึ้นอยู่กับชุด ของอุปกรณ์ที่จะเชื่อมต่อกับ IDE คอนโทรลเลอร์ หาก อุปกรณ์ใหม่ เป็นเพียงอุปกรณ์เดียวที่เชื่อมต่อกับคอนโทรลเลอร์ ต้องตั้งค่าติดตั้งจัมเปอร์ อุปกรณ์เป็นตำแหน่ง master (บางอุปกรณ์ไม่ต้องการค่าติดตั้งในกรณีนี้) ถ้าอุปกรณ์ทั้งสองตัวเป็นชนิดเดียวกัน สามารถตั้งค่าติดตั้งจัมเปอร์อุปกรณ์ใหม่ เป็นตำแหน่ง slave ถ้าใช้อุปกรณ์ผสมกัน (ATA และ ATAPI) ต้องตั้งค่าจัมเปอร์อุปกรณ์ ATA เป็น ตำแหน่ง master และต้องตั้งค่าจัมเปอร์ อุปกรณ์ ATAPI เป็นตำแหน่ง slave ถ้ามีชนิดอุปกรณ์ผสมกันและ อุปกรณ์ใหม่ เป็นอุปกรณ์ ATA (ดิสก์) ต้องเปลี่ยนจัมเปอร์อุปกรณ์สำหรับอุปกรณ์ ATAPI ที่มีอยู่ในปัจจุบันเป็นตำแหน่ง slave และ ต้องตั้งค่าจัมเปอร์อุปกรณ์ ATA ใหม่เป็น master ถ้ามีชนิดอุปกรณ์ผสมกันและ อุปกรณ์ใหม่เป็นอุปกรณ์ ATAPI (ซีดีรอมหรือเทป) ต้องตั้งค่าจัมเปอร์อุปกรณ์ สำหรับอุปกรณ์ ATAPI ใหม่เป็น slave และถ้าอุปกรณ์ ATA ไม่มีค่าติดตั้งจัมเปอร์ในปัจจุบัน ต้องตั้งค่าอุปกรณ์ ATA เป็น master

การตั้งค่าฮาร์ดแวร์สำหรับการติดตั้งอุปกรณ์ IDE

คุณต้องตั้งค่าฮาร์ดแวร์เพื่อติดตั้งอุปกรณ์ IDE

- อย่าเริ่มภารกิจนี้จนกว่าคุณได้เลือกและบันทึกข้อมูลต่อไปนี้แล้ว:
 - ตำแหน่งของ IDE I/O คอนโทรลเลอร์ซึ่งจะเชื่อมต่อกับอุปกรณ์ (ในตัวหรือที่ระบุโดยหมายเลขสล๊อตอะแดปเตอร์อย่างใดอย่างหนึ่ง)
 - IDE แอดเดรสของอุปกรณ์
 - กำหนดตำแหน่งทางกายภาพบนยูนิตระบบที่จะเชื่อมต่อกับ IDE คอนโทรลเลอร์ที่เลือก ตัวอย่างเช่น ระบุตำแหน่งของ IDE คอนโทรลเลอร์ในตัว โปรดอ้างอิงคู่มือของผู้ใช้งานเพื่อดูวิธีใช้
1. ปิดยูนิตระบบโดยใช้คำสั่ง **shutdown** หลังจากหยุดแอ็พพลิเคชันทั้งหมดที่กำลังรันอยู่ในปัจจุบัน พิมพ์ **shutdown -F** เพื่อ หยุดระบบทันทีโดยไม่แจ้งผู้ใ้รายอื่น
 2. รอให้ข้อความ **Halt Completed** หรือข้อความที่ คล้ายกันแสดงขึ้น
 3. ปิดยูนิตระบบและอุปกรณ์ที่ต่อพ่วงทั้งหมด
 4. ถอดปลั๊กยูนิตระบบและอุปกรณ์ที่ต่อพ่วงทั้งหมด
 5. ทำการเชื่อมต่อทางกายภาพตามโพธิ์เตอร์ที่อธิบายไว้ในคู่มือเซ็ทอัพและผู้ใช้งาน

หมายเหตุ: อย่าปิดยูนิต ระบบ; ให้ทำ "การเพิ่มอุปกรณ์ IDE ลงในฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเอง" ในหน้า 578 ต่อไป

การเพิ่มอุปกรณ์ IDE ลงในฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเอง

ภารกิจนี้ทำให้ระบบรู้จักอุปกรณ์ในระหว่างสตาร์ทอัปยูนิตรระบบ ระบบปฏิบัติการจะอ่านคอนฟิกูเรชันปัจจุบันและตรวจพบอุปกรณ์ใหม่ เรียกร้องของอุปกรณ์ใหม่แต่ละรายการจะถูกเพิ่มลงในฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเอง และได้รับแอตทริบิวต์ที่พอลต์

หากกำลังติดตั้งอุปกรณ์บนยูนิตรระบบใหม่ ต้องติดตั้ง ระบบปฏิบัติการ คำแนะนำสำหรับการติดตั้งระบบปฏิบัติการ มีการรวมไว้ในคู่มือการติดตั้งของระบบปฏิบัติการ

ปฏิบัติตามโพรซีเจอร์นี้เพื่อเพิ่มอุปกรณ์ลงในฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเอง:

1. เสียบปลั๊กยูนิตรระบบและอุปกรณ์ที่ต่อพ่วงทั้งหมด
2. เปิดอุปกรณ์ทั้งหมด แต่ปล่อยให้ยูนิตรระบบปิดอยู่
3. เปิดยูนิตรระบบเมื่ออุปกรณ์ที่ต่อพ่วงทั้งหมดทดสอบการเปิดด้วยตนเอง (POSTs) เสร็จแล้ว

หมายเหตุ: โพรเซสสตาร์ทอัปตรวจพบและบันทึกอุปกรณ์ไว้ใน ฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเองโดยอัตโนมัติ

4. ยืนยันว่า อุปกรณ์ถูกเพิ่มให้แก่ฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเองแล้วโดยใช้พารามิเตอร์ `smit lsidea` รายการของอุปกรณ์ที่กำหนดไว้ ทั้งหมดแสดงขึ้น ดูที่ฟิลด์ที่ตั้งของอะแดปเตอร์ IDE และค่าแอดเดรส IDE ของอุปกรณ์ซึ่งคุณเพิ่งติดตั้ง

การกำหนดแอตทริบิวต์สำหรับอุปกรณ์ IDE เอง

มีการกำหนดแอตทริบิวต์ที่พอลต์ให้กับอุปกรณ์ที่สนับสนุนเมื่ออุปกรณ์ ถูกเพิ่มลงในฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเอง แอตทริบิวต์เหล่านี้เหมาะสม สำหรับการใช้อุปกรณ์ตามปกติ

เปลี่ยนแอตทริบิวต์อุปกรณ์เมื่ออุปกรณ์ซึ่งคุณกำลังติดตั้ง ไม่ได้รับการสนับสนุน หรือเมื่อคุณต้องการกำหนดบางส่วนของการทำงานของ อุปกรณ์เอง ตัวอย่างเช่น คุณอาจต้องการเปลี่ยนเทปไดรฟ์ เพื่อเขียนเทปในรูปแบบความหนาแน่นน้อยลง

เมื่อต้องการกำหนดแอตทริบิวต์ของอุปกรณ์เอง ให้ใช้พารามิเตอร์ `SMIT, smit dev`

โหนดอุปกรณ์

อุปกรณ์มีการจัดระเบียบเป็นคลัสเตอร์ที่เรียกว่า โหนด แต่ละโหนด เป็นระบบย่อยโลจิคัลของอุปกรณ์ โดยที่อุปกรณ์ระดับต่ำกว่ามีการพึ่งพา อุปกรณ์ระดับสูงกว่าในความสัมพันธ์ระหว่างชายน์-พารেন্ট

ตัวอย่างเช่น โหนดระบบคือส่วนสูงสุดของโหนดทั้งหมดและประกอบด้วย อุปกรณ์ฟิสิคัลทั้งหมดในระบบ อุปกรณ์ระบบอยู่ที่ตำแหน่งบนสุดของโหนด และสิ่งที่อยู่ข้างใต้ อุปกรณ์ระบบคือบัสและอะแดปเตอร์ ซึ่งมีการพึ่งพา อุปกรณ์ระบบระดับที่สูงกว่าที่ด้านล่างของลำดับชั้นคืออุปกรณ์ ซึ่งไม่มีอุปกรณ์อื่นเชื่อมต่อ อุปกรณ์เหล่านี้พึ่งพาอุปกรณ์ทั้งหมด ที่อยู่สูงกว่าในลำดับชั้น

ในเวลาสตาร์ทอัป มีการใช้การพึ่งพาแบบพารেন্ট-ชายน์เพื่อกำหนดคอนฟิก อุปกรณ์ทั้งหมดที่ประกอบขึ้นเป็นโหนด คอนฟิกูเรชันเกิดขึ้นจากโหนดบนสุดไล่ลงมาข้างล่าง และอุปกรณ์ ใดๆ ที่มีการพึ่งพาอุปกรณ์ระดับสูงกว่าจะไม่ถูกกำหนดคอนฟิกจนกว่ากำหนดคอนฟิก อุปกรณ์ระดับสูงกว่าแล้ว

ระบบปฏิบัติการ AIX สนับสนุนคุณลักษณะ I/O แบบหลายพาร (MPIO) หากอุปกรณ์มีไดรเวอร์อุปกรณ์ที่มีความสามารถ MPIO อุปกรณ์นั้นสามารถมี มากกว่าหนึ่งพารেন্টภายในลำดับชั้นนี้ ซึ่งอนุญาตให้มีหลายพารการสื่อสารในเวลาเดียวกัน ระหว่างอุปกรณ์และเครื่องที่กำหนดหรือพาร์ติชันโลจิคัลภายใน เครื่อง

คลาสอุปกรณ์

ในการจัดการกับอุปกรณ์ ระบบปฏิบัติการต้องเข้าใจเป็นอย่างดีว่า สามารถเชื่อมต่ออุปกรณ์ใดได้บ้าง ระบบปฏิบัติการจัดหมวดหมู่อุปกรณ์ตามลำดับชั้น ออกเป็นสามกลุ่ม

กลุ่มเหล่านี้ได้แก่:

- คลาสฟังก์ชัน
- คลาสย่อยฟังก์ชัน
- ชนิดอุปกรณ์

คลาสฟังก์ชัน ประกอบด้วยอุปกรณ์ที่ทำฟังก์ชันเดียวกัน ตัวอย่างเช่น เครื่องพิมพ์ประกอบเป็นคลาสฟังก์ชัน คลาสฟังก์ชันมีการจัดกลุ่ม เป็นคลาสย่อยตามความคล้ายคลึงของอุปกรณ์บางอย่าง ตัวอย่างเช่น เครื่องพิมพ์ที่มีอินเทอร์เฟซอนุกรมหรือขนาน เครื่องพิมพ์อนุกรมถือเป็นคลาสย่อยหนึ่ง และเครื่องพิมพ์ขนานเป็นอีกคลาสย่อยหนึ่ง ชนิดอุปกรณ์มีการจัดประเภทตาม รุ่นของอุปกรณ์และผู้ผลิต

คลาสอุปกรณ์ กำหนดการเชื่อมต่อระหว่างพารานต์และชาเยนที่ถูกต้องสำหรับ ระบบปฏิบัติการ ลำดับชั้นกำหนดคลาสย่อยที่เป็นไปได้ซึ่งสามารถเชื่อมต่อได้ สำหรับแต่ละที่ต้งการเชื่อมต่อชาเยนที่เป็นไปได้ ตัวอย่างเช่น คำว่าอะแด็ปเตอร์ RS-232 8-พอร์ตรับว่า เฉพาะอุปกรณ์ที่เป็นสมาชิกของคลาสย่อย RS-232 เท่านั้นสามารถเชื่อมต่อกับพอร์ตใดๆ ในแปดพอร์ตของอะแด็ปเตอร์ได้

คลาสอุปกรณ์และการฟังหาแบบลำดับชั้นมีการเก็บรักษาไว้ในฐานข้อมูล คอนฟิกูเรชันอุปกรณ์ Object Data Manager (ODM)

ฐานข้อมูลคอนฟิกูเรชันอุปกรณ์และการจัดการอุปกรณ์

ข้อมูลอุปกรณ์มีอยู่ในฐานข้อมูลที่กำหนดไว้ล่วงหน้า หรือฐานข้อมูล แบบกำหนดเองซึ่งประกอบเป็นฐานข้อมูลคอนฟิกูเรชันอุปกรณ์

ฐานข้อมูลที่กำหนดไว้ล่วงหน้ามีข้อมูลคอนฟิกูเรชันสำหรับอุปกรณ์ที่เป็นไปได้ ทั้งหมดซึ่งได้รับการสนับสนุนจากระบบ ข้อมูลคลาสอุปกรณ์ตามลำดับชั้นมีอยู่ในฐานข้อมูลนี้ ฐานข้อมูลแบบกำหนดเองมีข้อมูลคอนฟิกูเรชันสำหรับอุปกรณ์ ที่กำหนดและกำหนดคอนฟิกูเรชันในปัจจุบันทั้งหมดในระบบ มีการเก็บรักษาเร็กคอร์ด ของแต่ละอุปกรณ์ที่เชื่อมต่อกับระบบของคุณอยู่ในปัจจุบัน

Configuration Manager คือโปรแกรมที่กำหนดคอนฟิกูเรชันบนระบบ ของคุณโดยอัตโนมัติในระหว่างสตาร์ทอัพระบบและรันไทม์ Configuration Manger ใช้ข้อมูลจากฐานข้อมูลที่กำหนดไว้ล่วงหน้าและแบบกำหนดเองในระหว่าง โปรเซสส์ และอัปเดตฐานข้อมูลแบบกำหนดเองหลังจากนั้น

ฟังก์ชัน I/O หลายพาทใช้ตัวระบุอุปกรณ์จำเพาะ (UDID) เพื่อจำแนกอุปกรณ์ที่มีความสามารถ MPIO แต่ละตัว โดยไม่พิจารณาถึง พาทที่ค้นพบ UDID มีการบันทึกไว้ในฐานข้อมูลคอนฟิกูเรชันอุปกรณ์ เมื่อค้นพบ อุปกรณ์ จะมีการตรวจสอบ UDID ในฐานข้อมูลเพื่อกำหนดว่า อุปกรณ์เป็นอุปกรณ์ใหม่ หรือการค้นพบเป็นอีกพาทหนึ่งไปยังอุปกรณ์ ที่มีอยู่ เมื่อตรวจพบหลายพาทไปยังอุปกรณ์เดียวกัน ไดรเวอร์อุปกรณ์หรือ ส่วนขยายเคอร์เนล Path Control Manager จะเลือกพาทที่จะใช้สำหรับ คำร้องขอเฉพาะ

คุณสามารถใช้ SMIT หรือคำสั่งระบบปฏิบัติการในการจัดการงานการจัดการของอุปกรณ์ เช่น การลบ เพิ่ม หรือกำหนดคอนฟิกูเรชัน

หลักการที่เกี่ยวข้อง:

“การจัดการอุปกรณ์ที่มีความสามารถ MPIO” ในหน้า 593

สามารถใช้คุณลักษณะ Multiple Path I/O (MPIO) เพื่อกำหนดพารามิเตอร์ไปยังอุปกรณ์สำหรับวัตถุประสงค์ failover งานที่เกี่ยวข้อง:

“การจัดเตรียมเพื่อติดตั้งอุปกรณ์” ในหน้า 422

การติดตั้งอุปกรณ์บนระบบของคุณประกอบด้วยการระบุตำแหน่งที่จะต่อพ่วงอุปกรณ์ การเชื่อมต่ออุปกรณ์ทางกายภาพ และการกำหนดคอนฟิก อุปกรณ์ด้วย Configuration Manager หรือ SMIT

สถานะอุปกรณ์

อุปกรณ์ที่เชื่อมต่อกับระบบสามารถอยู่ในสถานะอย่างใดอย่างหนึ่งจาก สี่สถานะ

อุปกรณ์ที่เชื่อมต่อกับระบบสามารถอยู่ในสถานะอย่างใดอย่างหนึ่งจาก สี่สถานะต่อไปนี้:

ไอเท็ม	คำอธิบาย
ไม่ได้กำหนด	ระบบไม่รู้จักอุปกรณ์
กำหนดไว้	ข้อมูลเฉพาะเกี่ยวกับอุปกรณ์มีการบันทึกไว้ในฐานข้อมูลแบบกำหนดเอง แต่ไม่มีอยู่บนระบบ
มีอยู่	อุปกรณ์ที่กำหนดต่อคู่กับระบบปฏิบัติการ หรือมีการกำหนดคอนฟิก อุปกรณ์ที่กำหนด
หยุด	อุปกรณ์ไม่มีอยู่แต่ยังคงเป็นที่รู้จักของไดรเวอร์อุปกรณ์

หากอุปกรณ์ tty และเครื่องพิมพ์สลับกันใช้ตัวเชื่อมต่อ tty เดียวกัน ทั้งอุปกรณ์ tty และเครื่องพิมพ์จะถูกกำหนดไว้บนพาเรนต์ และพอร์ตเดียวกันใน ฐานข้อมูลคอนฟิกูเรชันอุปกรณ์ สามารถกำหนดคอนฟิกูเรชันเหล่านี้ได้ที่ละ หนึ่งรายการเท่านั้น เมื่อกำหนดคอนฟิกูเรชันตัวเชื่อมต่อ tty จะมีการรักษาข้อมูลเชื่อมต่อเฉพาะ เครื่องพิมพ์ไว้จนกว่าจะถูกกำหนดคอนฟิกูเรชันอีกครั้ง อุปกรณ์ไม่ได้ถูกลบออก แต่อยู่ในสถานะที่กำหนด การรักษาอุปกรณ์ในสถานะที่กำหนดจะคง ข้อมูลแบบกำหนดเองของอุปกรณ์ซึ่งไม่ได้ใช้งานอยู่ในปัจจุบันไว้ ก่อนทำให้มีอยู่ เป็นครั้งแรก หรือในขณะที่ถูกลบออกจากระบบเป็นการชั่วคราว

หากมีไดรเวอร์อุปกรณ์สำหรับอุปกรณ์ สามารถทำให้อุปกรณ์พร้อมใช้งานได้ ผ่านทางไดรเวอร์อุปกรณ์

บางอุปกรณ์ โดยเฉพาะอุปกรณ์เทียม TCP/IP ต้องมีสถานะเป็น หยุด

งานที่เกี่ยวข้อง:

“การยกเลิกกำหนดคอนฟิกูเรชันอะแดปเตอร์อะซิงค์” ในหน้า 611

คุณสามารถยกเลิกกำหนดคอนฟิกูเรชันอะแดปเตอร์อะซิงค์ได้

โค้ดที่ตั้งอุปกรณ์

โค้ดที่ตั้ง คือพารามิเตอร์ที่เก็บ CPU หรือยูนิตรระบบ ผ่านทางอะแดปเตอร์ สายเคเบิลสัญญาณ และกล่องตัวรับแบบอะซิงโครนัส (ถ้ามีอยู่) ไปยังอุปกรณ์หรือเวิร์กสเตชัน โค้ดนี้นับเป็นอีกวิธีหนึ่ง ในการระบุอุปกรณ์ฟิสิคัล

โค้ดที่ตั้งประกอบด้วยฟิลด์ข้อมูลที่สูงที่สุดสี่ฟิลด์ ขึ้นอยู่กับ ชนิดของอุปกรณ์ ฟิลด์เหล่านี้แสดงถึงตัวเก็บ สล็อต ตัวเชื่อมต่อ และพอร์ต แต่ละฟิลด์เหล่านี้ประกอบด้วยอักขระสองตัว

โค้ดที่ตั้งของตัวเก็บประกอบด้วยฟิลด์ตัวเก็บอย่างเดี่ยว และเป็น โค้ดสองอักขระแบบง่าย โค้ดที่ตั้งของอะแดปเตอร์ประกอบด้วย ฟิลด์ตัวเก็บและสล็อต และมีรูปแบบ AA-BB โดยที่ AA สอดคล้องกับที่ตั้งของตัวเก็บ และ BB บ่งชี้บัสและสล็อตที่มีอะแดปเตอร์ อุปกรณ์อื่นมีโค้ดที่ตั้ง รูปแบบ AA-BB-CC หรือ AA-BB-CC-DD โดยที่ AA-BB คือโค้ดที่ตั้ง ของอะแดปเตอร์ซึ่งอุปกรณ์เชื่อมต่อ และ CC สอดคล้อง กับตัวเชื่อมต่อบนอะแดปเตอร์ซึ่งอุปกรณ์เชื่อมต่อ และ DD สอดคล้องกับหมายเลขพอร์ตหรือ แอดเดรสอุปกรณ์ SCSI

ข้อมูลที่เกี่ยวข้อง:

ตำแหน่งชิ้นส่วนและโค้ดตำแหน่ง

โค้ดที่ตั้งอะแดปเตอร์

โค้ดที่ตั้งของอะแดปเตอร์ประกอบด้วยคู่ตัวเลขสองคู่ในรูปแบบ AA-BB โดยที่ AA ระบุโค้ดที่ตั้งของตู้เก็บ ที่มีอะแดปเตอร์ และ BB ระบุบัส I/O และสล็อตที่มีการ์ด

ค่า 00 สำหรับฟิลด์ AA หมายความว่า อะแดปเตอร์ตั้งอยู่ในตู้เก็บ CPU หรือยูนิตรระบบ ทั้งนี้ขึ้นอยู่กับ ชนิดของระบบ ค่าอื่นใด ๆ สำหรับฟิลด์ AA บ่งชี้ว่าการ์ดตั้งอยู่ในตู้เก็บส่วนขยาย I/O ในกรณีนี้ ค่า AA ระบุบัส I/O และหมายเลขสล็อต ในตู้เก็บ CPU ซึ่งมีอะแดปเตอร์ส่วนขยายแบบอะซิงโครนัส เลขตัวแรก ระบุบัส I/O โดยที่ 0 หมายถึง บัส I/O มาตรฐาน และ 1 หมายถึง บัส I/O ที่เป็นทางเลือก เลขตัวที่สองระบุหมายเลขสล็อตบนบัส I/O ที่บ่งชี้

เลขตัวแรกของฟิลด์ BB ระบุบอร์ด I/O ที่มีอะแดปเตอร์การ์ด หากการ์ดอยู่ในตู้เก็บ CPU หรือยูนิตรระบบ เลขตัวนี้จะเป็น 0 สำหรับบัส I/O มาตรฐาน และ 1 สำหรับ บัส I/O ที่เป็นทางเลือก หากการ์ดอยู่ในตู้เก็บส่วนขยาย I/O เลขตัวนี้จะเป็น 0 เลขตัวที่สองระบุหมายเลขสล็อตบนบัส I/O ที่บ่งชี้ (หรือหมายเลขสล็อตในตู้เก็บส่วนขยาย I/O) ที่มี การ์ด

โค้ดที่ตั้ง 00-00 ใช้เพื่อระบุบอร์ด I/O มาตรฐาน

ตัวอย่าง:

ไอเอ็ม	คำอธิบาย
00-05	ระบุอะแดปเตอร์การ์ดในสล็อต 5 ของบอร์ด I/O มาตรฐาน และ ตั้งอยู่ในตู้เก็บ CPU หรือยูนิตรระบบ ขึ้นอยู่กับชนิด ของระบบ
00-12	ระบุอะแดปเตอร์ในสล็อต 2 ของบัส I/O ที่เป็นทางเลือก และ ตั้งอยู่ในตู้เก็บ CPU
18-05	ระบุอะแดปเตอร์การ์ดที่ตั้งอยู่ในสล็อต 5 ของตู้เก็บส่วนขยาย I/O ตู้เก็บมีการเชื่อมต่อกับอะแดปเตอร์ส่วนขยายแบบอะซิงโครนัสที่ตั้งอยู่ในสล็อต 8 ของบัส I/O ที่เป็นทางเลือกในตู้เก็บ CPU

ข้อมูลที่เกี่ยวข้อง:

ตำแหน่งชิ้นส่วนและโค้ดตำแหน่ง

โค้ดที่ตั้งเครื่องพิมพ์และพล็อตเตอร์

โค้ดที่ตั้ง 00-00-S1-00 หรือ 00-00-S2-00 บ่งชี้ว่าเครื่องพิมพ์ พล็อตเตอร์ หรืออุปกรณ์ tty เชื่อมต่ออยู่กับพอร์ตอนุกรมของ บอร์ด I/O มาตรฐาน s1 หรือ s2 โค้ดที่ตั้ง 00-00-0P-00 บ่งชี้ว่า เครื่องพิมพ์แบบขนานเชื่อมต่อกับพอร์ตขนานของบอร์ด I/O มาตรฐาน

โค้ดที่ตั้งอื่นบ่งชี้ว่าเครื่องพิมพ์ พล็อตเตอร์ หรืออุปกรณ์ tty เชื่อมต่อ อยู่กับอะแดปเตอร์การ์ดอื่นที่ไม่ใช่บอร์ด I/O มาตรฐาน สำหรับเครื่องพิมพ์ พล็อตเตอร์ และอุปกรณ์ tty เหล่านี้ รูปแบบโค้ดที่ตั้งคือ AA-BB-CC-DD โดยที่ AA-BB บ่งชี้ โค้ดที่ตั้งของอะแดปเตอร์ควบคุม

ไอเอ็ม คำอธิบาย

- AA** ค่า 00 สำหรับฟิลด์ AA บ่งชี้ว่าจะติดตั้งอะแดปเตอร์การ์ดตั้งอยู่ในตู้เก็บ CPU หรือยูนิตระบบ ทั้งนี้ขึ้นอยู่กับชนิดของระบบ ค่าอื่นสำหรับฟิลด์ AA บ่งชี้ว่าการติดตั้งอยู่ในตู้เก็บส่วนขยาย I/O ในกรณีนี้ เลขตัวแรก ระบุบัส I/O และเลขตัวที่สองระบุหมายเลขสล็อตบนบัสในตู้เก็บ CPU ซึ่งมีอะแดปเตอร์ส่วนขยายแบบอะซิงโครนัสซึ่งตู้เก็บส่วนขยาย I/O เชื่อมต่ออยู่
- BB** เลขตัวแรกของฟิลด์ BB ระบุบัส I/O ที่มีอะแดปเตอร์การ์ด หากการติดตั้งอยู่ในตู้เก็บ CPU หรือยูนิตระบบ เลขตัวนี้จะเป็น 0 สำหรับบัส I/O มาตรฐาน และ 1 สำหรับ บัส I/O ที่เป็นทางเลือก หากการติดตั้งอยู่ในตู้เก็บส่วนขยาย I/O เลขตัวนี้จะเป็น 0 เลขตัวที่สองระบุหมายเลขสล็อตบนบัส I/O (หรือหมายเลขสล็อตในตู้เก็บส่วนขยาย I/O) ที่มีการ์ด
- CC** ฟิลด์ CC ระบุตัวเชื่อมต่อนบน อะแดปเตอร์การ์ดซึ่งกล่องตัวรับแบบอะซิงโครนัสเชื่อมต่อ ค่าที่เป็นไปได้คือ 01, 02, 03, และ 04
- DD** ฟิลด์ DD ระบุหมายเลขพอร์ตบน กล่องตัวรับแบบอะซิงโครนัสซึ่งเครื่องพิมพ์ สล็อตเตอร์ หรืออุปกรณ์ tty ต่อพวง

ข้อมูลที่เกี่ยวข้อง:

ตำแหน่งชิ้นส่วนและโค้ดตำแหน่ง

โค้ดที่ตั้ง tty

โค้ดที่ตั้ง 00-00-S1-00 หรือ 00-00-S2-00 บ่งชี้ว่าอุปกรณ์ tty เชื่อมต่ออยู่กับพอร์ตอนุกรม I/O มาตรฐาน s1 หรือ s2

โค้ดที่ตั้งอื่นบ่งชี้ว่าอุปกรณ์ tty เชื่อมต่อ อยู่กับอะแดปเตอร์การ์ดอื่นที่ไม่ใช่บอร์ด I/O มาตรฐาน สำหรับอุปกรณ์เหล่านี้ รูปแบบโค้ดที่ตั้งคือ AA-BB-CC-DD โดยที่ AA-BB บ่งชี้โค้ดที่ตั้งของอะแดปเตอร์การ์ดควบคุม

ไอเอ็ม คำอธิบาย

- AA** ค่า 00 สำหรับฟิลด์ AA บ่งชี้ว่าจะติดตั้งอะแดปเตอร์การ์ดตั้งอยู่ในตู้เก็บ CPU หรือยูนิตระบบ ทั้งนี้ขึ้นอยู่กับชนิดของระบบ ค่าอื่นใดๆ สำหรับฟิลด์ AA บ่งชี้ว่าการติดตั้งอยู่ในตู้เก็บส่วนขยาย I/O ในกรณีนี้ เลขตัวแรก ระบุบัส I/O และเลขตัวที่สองระบุหมายเลขสล็อต บนบัสในตู้เก็บ CPU ซึ่งมีอะแดปเตอร์ส่วนขยายแบบอะซิงโครนัสซึ่งตู้เก็บส่วนขยาย I/O เชื่อมต่ออยู่
- BB** เลขตัวแรกของฟิลด์ BB ระบุบัส I/O ที่มีอะแดปเตอร์การ์ด หากการติดตั้งอยู่ในตู้เก็บ CPU หรือยูนิตระบบ เลขตัวนี้จะเป็น 0 สำหรับบัส I/O มาตรฐาน และ 1 สำหรับ บัส I/O ที่เป็นทางเลือก หากการติดตั้งอยู่ในตู้เก็บส่วนขยาย I/O เลขตัวนี้จะเป็น 0 เลขตัวที่สองระบุหมายเลขสล็อตบนบัส I/O (หรือหมายเลขสล็อตในตู้เก็บส่วนขยาย I/O) ที่มีการ์ด
- CC** ฟิลด์ CC ระบุตัวเชื่อมต่อนบน อะแดปเตอร์การ์ดซึ่งกล่องตัวรับแบบอะซิงโครนัสเชื่อมต่อ ค่าที่เป็นไปได้คือ 01, 02, 03, และ 04
- DD** ฟิลด์ DD ระบุหมายเลขพอร์ตบน กล่องตัวรับแบบอะซิงโครนัสซึ่งอุปกรณ์ tty ต่อพวง

ข้อมูลที่เกี่ยวข้อง:

ตำแหน่งชิ้นส่วนและโค้ดตำแหน่ง

โค้ดที่ตั้งอุปกรณ์ SCSI

ข้อมูลต่อไปนี้เป็นโค้ดที่ตั้งสำหรับอุปกรณ์ SCSI

โค้ดที่ตั้งเหล่านี้ใช้กับอุปกรณ์ SCSI ทั้งหมดรวมถึง:

- ซีดีรอม
- ดิสก์
- อุปกรณ์ตัวเริ่มต้น
- ออปติคัลไดรฟ์อ่าน/เขียน
- เทป
- โหมดเป้าหมาย

รูปแบบโค้ดที่ตั้งคือ AA-BB-CC-S, L ฟิลด์ AA-BB ระบุโค้ดที่ตั้งของอะแดปเตอร์ SCSI ที่ควบคุมอุปกรณ์ SCSI

ไอเอ็ม	คำอธิบาย
AA	ค่า 00 สำหรับฟิลต์ AA บ่งชี้ว่าอะแดปเตอร์การ์ดควบคุมตั้งอยู่ในตู้เก็บ CPU หรือยูนิตรระบบ ทั้งนี้ขึ้นอยู่กับชนิดของระบบ
BB	ฟิลต์ BB ระบุบัส I/O และสล็อต ที่มีการ์ด เลขตัวแรกระบุบัส I/O ซึ่งเป็น 0 สำหรับ บัส I/O มาตรฐานและ 1 สำหรับบัส I/O ที่เป็นทางเลือก เลขตัวที่สองคือสล็อตบนบัส I/O ที่บ่งชี้ซึ่งมีการ์ด ค่า 00 สำหรับฟิลต์ BB บ่งชี้ SCSI คอนโทรลเลอร์มาตรฐาน
CC	ฟิลต์ CC ระบุบัส SCSI ของการ์ด ที่อุปกรณ์ต่อพ่วง สำหรับการ์ดที่มีบัส SCSI ตัวเดียวเท่านั้น ฟิลต์นี้มีการตั้งค่าเป็น 00 มิฉะนั้น ค่า 00 บ่งชี้ อุปกรณ์ที่ต่อพ่วงกับบัส SCSI ภายในของการ์ด และค่า 01 บ่งชี้ อุปกรณ์ที่ต่อพ่วงกับบัส SCSI ภายนอกของการ์ด
S,L	ฟิลต์ S,L ระบุ SCSI ID และหมายเลข โลจิคัลยูนิท (LUN) ของอุปกรณ์ SCSI ค่า S บ่งชี้ SCSI ID และค่า L บ่งชี้ LUN

ข้อมูลที่เกี่ยวข้อง:

ตำแหน่งชิ้นส่วนและโค้ดตำแหน่ง

โค้ดที่ตั้ง direct-bus-attached

สำหรับอุปกรณ์ดิสก์ที่ต่อพ่วงโดยตรง รูปแบบโค้ดที่ตั้งคือ AA-BB

ฟิลต์ AA มีค่าเป็น 00 ซึ่งบ่งชี้ว่าดิสก์ตั้งอยู่ในยูนิตรระบบ ฟิลต์ BB บ่งชี้บัส I/O และหมายเลขสล็อตซึ่งต่อพ่วงดิสก์ เลขตัวแรกเป็น 0 เสมอ ซึ่งบ่งชี้ว่าดิสก์ต่อพ่วงกับ บัส I/O มาตรฐาน เลขตัวที่สองระบุหมายเลขสล็อตบนบัส I/O มาตรฐานซึ่งดิสก์ต่อพ่วง

ข้อมูลที่เกี่ยวข้อง:

ตำแหน่งชิ้นส่วนและโค้ดตำแหน่ง

โค้ดที่ตั้งดิสก์ที่ลิงก์แบบอนุกรม

โค้ดที่ตั้งของดิสก์ไดรฟ์ที่ลิงก์แบบอนุกรมอยู่ในรูปแบบ AA-BB-CC-DD โดยที่ AA-BB บ่งชี้ โค้ดที่ตั้งของอะแดปเตอร์การ์ดควบคุม

แต่ละฟิลต์มีการตีความดังนี้:

ไอเอ็ม	คำอธิบาย
AA	ค่า 00 สำหรับฟิลต์ AA บ่งชี้ว่าอะแดปเตอร์การ์ดควบคุมตั้งอยู่ในตู้เก็บ CPU หรือยูนิตรระบบ ทั้งนี้ขึ้นอยู่กับชนิดของระบบ
BB	ฟิลต์ BB ระบุบัส I/O และสล็อต ที่มีการ์ด เลขตัวแรกระบุบัส I/O ซึ่งเป็น 0 สำหรับ บัส I/O มาตรฐานและ 1 สำหรับบัส I/O ที่เป็นทางเลือก เลขตัวที่สองคือสล็อตบนบัส I/O ที่บ่งชี้ซึ่งมีการ์ด
CC	ฟิลต์ CC ระบุตัวเชื่อมต่อบน อะแดปเตอร์การ์ดซึ่งตู้เก็บคอนโทรลเลอร์เชื่อมต่อ ค่าที่เป็นไปได้คือ 00, 01, 02, และ 03
DD	ฟิลต์ DD ระบุหมายเลขโลจิคัลยูนิท (LUN) ของดิสก์ ค่านี้สอดคล้องกับสล็อตในตู้เก็บซึ่งดิสก์ ตั้งอยู่

ข้อมูลที่เกี่ยวข้อง:

ตำแหน่งชิ้นส่วนและโค้ดตำแหน่ง

โค้ดที่ตั้งดิสก์ไดรฟ์

ดิสก์ไดรฟ์ได้รับการกำหนดโค้ดที่ตั้ง

ดิสก์ไดรฟ์มีโค้ดที่ตั้งเป็น 00-00-0D-01 หรือ 00-00-0D-02 อย่างใดอย่างหนึ่ง ซึ่งบ่งชี้ว่าดิสก์ไดรฟ์มีการต่อพ่วงเข้ากับ I/O planar ดิสก์ไดรฟ์พอร์ตมาตรฐาน 0 หรือ 1

ข้อมูลที่เกี่ยวข้อง:

ตำแหน่งชิ้นส่วนและโค้ดตำแหน่ง

โค้ดที่ตั้ง Dials/LPFKeys

สำหรับอุปกรณ์ Dials/LPFKeys ที่แนบเข้ากับกราฟิกอินพุตอะแดปเตอร์ รูปแบบโค้ดที่ตั้งคือ AA-BB-CC

แต่ละฟิลด์มีการตีความดังนี้:

ไอเท็ม คำอธิบาย

- AA ค่า 00 สำหรับฟิลด์ AA บ่งชี้ว่าอะแดปเตอร์การ์ดควบคุมตั้งอยู่ในตู้เก็บ CPU หรือยูนิตระบบ ทั้งนี้ขึ้นอยู่กับชนิดของระบบ
- BB ฟิลด์ BB ระบุบัส I/O และสล็อตที่มีการ์ด เลขตัวแรกระบุบัส I/O ซึ่งเป็น 0 สำหรับ บัส I/O มาตรฐานและ 1 สำหรับบัส I/O ที่เป็นทางเลือก เลข ตัวที่สองคือสล็อตบนบัส I/O ที่บ่งชี้ซึ่งมีการ์ด
- CC ฟิลด์ CC บ่งชี้ตัวเชื่อมต่อการ์ดซึ่ง ต่อพ่วงอุปกรณ์ ค่าเป็น 01 หรือ 02 ขึ้นอยู่กับว่าอุปกรณ์ที่ต่อพ่วงเป็นพอร์ต 1 หรือพอร์ต 2 บน การ์ด

หมายเหตุ: อุปกรณ์ Dials/LPFKeys ที่ต่อพ่วงแบบอนุกรมไม่ได้บ่งชี้โค้ดที่ตั้ง ที่เป็นเช่นนี้เนื่องจากอุปกรณ์เหล่านี้มีการพิจารณาว่าต่อพ่วงเข้ากับอุปกรณ์ tty อุปกรณ์ tty มีการระบุโดยผู้ใช้ในระหว่างคำนิยาม Dials/LPFKeys

ข้อมูลที่เกี่ยวข้อง:

ตำแหน่งชิ้นส่วนและโค้ดตำแหน่ง

โค้ดที่ตั้งพอร์ตหลายโปรโตคอล

โค้ดที่ตั้งของพอร์ตหลายโปรโตคอลอยู่ในรูปแบบ AA-BB-CC-DD โดยที่ AA-BB บ่งชี้โค้ดที่ตั้งของอะแดปเตอร์การ์ดหลายโปรโตคอล

แต่ละฟิลด์มีการตีความดังนี้:

ไอเท็ม คำอธิบาย

- AA ค่า 00 สำหรับฟิลด์ AA บ่งชี้ว่าอะแดปเตอร์การ์ดหลายโปรโตคอลตั้งอยู่ในตู้เก็บ CPU หรือยูนิตระบบ ทั้งนี้ขึ้นอยู่กับชนิดของระบบ
- BB ฟิลด์ BB ระบุบัส I/O และสล็อตที่มีการ์ด เลขตัวแรกระบุบัส I/O ซึ่งเป็น 0 สำหรับ บัส I/O มาตรฐานและ 1 สำหรับบัส I/O ที่เป็นทางเลือก เลข ตัวที่สองคือสล็อตบนบัส I/O ที่บ่งชี้ซึ่งมีการ์ด
- CC ฟิลด์ CC ระบุตัวเชื่อมต่อบน อะแดปเตอร์การ์ดซึ่งกล่องตัวรับหลายโปรโตคอลเชื่อมต่อ ค่า เป็น 01 เสมอ
- DD ฟิลด์ DD ระบุหมายเลขพอร์ตฟิสิคัล บนกล่องตัวรับหลายโปรโตคอล ค่าที่เป็นไปได้คือ 00, 01, 02, และ 03

ข้อมูลที่เกี่ยวข้อง:

ตำแหน่งชิ้นส่วนและโค้ดตำแหน่ง

การตั้งค่า iSCSI

การตั้งค่า iSCSI เกี่ยวข้องกับการกำหนดคอนฟิกอะแดปเตอร์และการเพิ่มหรือ การอัปเดตเป้าหมาย

การกำหนดคอนฟิกอะแดปเตอร์ iSCSI ใน AIX

คอนฟิกูเรชันอะแดปเตอร์ iSCSI เป็นภารกิจที่ง่ายมากและ ตรงไปตรงมา

1. บ้อน smit iscsi ที่พร้อมคำสั่ง AIX หน้าจอ iSCSI แสดงขึ้น
2. เลือก อะแดปเตอร์ iSCSI จากหน้าจอ iSCSI หน้าจอ อะแดปเตอร์ iSCSI แสดงขึ้น
3. เลือก เปลี่ยน / แสดงลักษณะของอะแดปเตอร์ iSCSI จาก หน้าจอ อะแดปเตอร์ iSCSI หน้าจอ เปลี่ยน / แสดงลักษณะของอะแดปเตอร์ iSCSI แสดงขึ้น
4. เลือกอะแดปเตอร์ iSCSI ซึ่งคุณต้องการกำหนดคอนฟิกจากรายการ หน้าจอคอนฟิกูเรชันแสดงขึ้นคล้ายกับตัวอย่างต่อไปนี้

[Entry Fields]

iSCSI Adapter	ics0
Description	iSCSI Adapter

Status	Available
Location	10-60
Max. number of commands to queue to adapter	[200] + + #
Maximum Transfer Size	[0x100000] +
Discovery Filename	[/etc/iscsi/targetshw]
Discovery Policy	[file]
Automatic Discovery Secrets Filename	[/etc/iscsi/autosecret]
Adapter IP Address	[10.1.4.187]
Adapter Subnet Mask	[255.255.255.0]
Adapter Gateway Address	[10.1.4.1]
Apply change to DATABASE only	no +

หมายเหตุ: หากคุณมีคำถามเกี่ยวกับวัตถุประสงค์ของฟิลด์เฉพาะให้วางเคอร์เซอร์ในฟิลด์ และกด **F1** สำหรับวิธีใช้

เมื่อต้องการใช้การค้นหาไฟล์ flat ให้พิมพ์ file ในฟิลด์ นโยบายการค้นหา เมื่อต้องการใช้การค้นหา ODM ให้พิมพ์ odm ในฟิลด์ นโยบายการค้นหา หากต้องการเป้าหมาย iSCSI ที่ค้นหา DHCP ให้พิมพ์ slp ในฟิลด์ นโยบายการค้นหา

การอัปเดตไฟล์ flat ของเป้าหมาย iSCSI

ไฟล์ flat คือไฟล์คอนฟิกูเรชันสแตติกที่ใช้เพื่อกำหนดคอนฟิกเป้าหมาย iSCSI ชื่อไฟล์ที่พอลต์ของไฟล์นี้คือ /etc/iscsi/targetshw

คุณต้องระบุคุณสมบัติการค้นหาเป้าหมาย iSCSI ที่เกี่ยวข้องทั้งหมดอย่าง ชัดแจ้งในไฟล์ flat

ข้อมูลที่เกี่ยวข้อง:

ไฟล์เป้าหมาย

การเพิ่มเป้าหมาย iSCSI ที่ค้นพบแบบสแตติกเข้าใน ODM

เมื่อไม่ได้ใช้การค้นหาอัตโนมัติ อะแดปเตอร์ iSCSI ได้รับคำอธิบายเป้าหมาย iSCSI จากไฟล์ flat หรือ ODM อย่างไม่ได้อ่างหนึ่ง

คุณสามารถใช้คำสั่ง AIX หรือ SMIT เพื่อจัดวางข้อมูลเป้าหมาย iSCSI ใน ODM คำสั่ง **chiscsi**, **lsiscsi**, **mkiscsi**, และ **rmiscsi** เปลี่ยน แสดง เพิ่ม และลบข้อมูลเป้าหมาย iSCSI ออกจาก ODM

เมื่อต้องการเพิ่มเป้าหมาย iSCSI ที่ค้นพบแบบสแตติกหนึ่งรายการ เข้าใน ODM โดยใช้ SMIT ให้ทำดังต่อไปนี้:

1. ป้อน **smit iscsi** ที่พร้อมคำสั่ง AIX หน้าจอ iSCSI แสดงขึ้น
2. เลือก พารามิเตอร์อุปกรณ์เป้าหมาย iSCSI ใน ODM จากหน้าจอ iSCSI หน้าจอ พารามิเตอร์อุปกรณ์เป้าหมาย iSCSI ใน ODM แสดงขึ้น
3. เลือก เพิ่มอุปกรณ์เป้าหมาย iSCSI ใน ODM จากหน้าจอ iSCSI หน้าจอ เพิ่มอุปกรณ์เป้าหมาย iSCSI ใน ODM แสดงขึ้น
4. เลือก เพิ่มอุปกรณ์เป้าหมาย iSCSI ที่ค้นพบแบบสแตติกใน ODM จากหน้าจอ iSCSI หน้าจอ เพิ่มอุปกรณ์เป้าหมาย iSCSI ที่ค้นพบแบบสแตติกใน ODM แสดงขึ้น
5. เลือกอะแดปเตอร์ iSCSI ซึ่งคุณต้องการกำหนดคอนฟิกจากรายการ หน้าจอ เพิ่มอุปกรณ์เป้าหมาย iSCSI ที่ค้นพบแบบสแตติก สำหรับอะแดปเตอร์ iSCSI ที่คุณเลือกไว้แสดงขึ้น

6. ป้อนข้อมูลที่เหมาะสมลงในฟิลด์ข้างล่างนี้เป็นตัวอย่าง

[Entry Fields]

iSCSI Adapter	ics0	
iSCSI Target Name	[iqn.mds9216.emc.sym]	+
iSCSI Group	static	+
IP Address of iSCSI Target	[10.1.4.25]	+
Port Number of iSCSI Target	[3260]	+
Password	[my password]	+

หมายเหตุ: หาก你有คำถามเกี่ยวกับวัตถุประสงค์ของฟิลด์เฉพาะให้วางเคอร์เซอร์ในฟิลด์ และกด F1 สำหรับวิธีใช้

การเพิ่มเป้าหมาย iSCSI ที่ค้นพบแบบอัตโนมัติจากไฟล์ flat เข้าใน ODM

คุณสามารถใช้ SMIT เพื่อนำเข้าข้อมูลของไฟล์ flat ไปใน ODM

1. ป้อน smit iscsi ที่พร้อมคำสั่ง AIX หน้าจอ iSCSI แสดงขึ้น
2. เลือก พารามิเตอร์อุปกรณ์เป้าหมาย iSCSI ใน ODM จากหน้าจอ iSCSI หน้าจอ พารามิเตอร์อุปกรณ์เป้าหมาย iSCSI ใน ODM แสดงขึ้น
3. เลือก เพิ่มอุปกรณ์เป้าหมาย iSCSI ใน ODM จากหน้าจอ iSCSI หน้าจอ เพิ่มอุปกรณ์เป้าหมาย iSCSI ใน ODM แสดงขึ้น
4. เลือก เพิ่มข้อมูลอุปกรณ์เป้าหมาย iSCSI ใน ODM จากไฟล์ จากหน้าจอ iSCSI หน้าจอ เพิ่มข้อมูลอุปกรณ์เป้าหมาย iSCSI ใน ODM จากไฟล์ แสดงขึ้น
5. เลือกอะแดปเตอร์ iSCSI ซึ่งคุณต้องการกำหนดคอนฟิกจากรายการ หน้าจอ เพิ่มข้อมูลอุปกรณ์เป้าหมาย iSCSI ใน ODM จากไฟล์ สำหรับอะแดปเตอร์ iSCSI ที่คุณเลือกไว้แสดงขึ้น
6. ป้อนข้อมูลที่เหมาะสมลงในฟิลด์ข้างล่างนี้เป็นตัวอย่าง

[Entry Fields]

iSCSI Protocol Device	iscsi3	
iSCSI Group	[static]	+
Filename of iSCSI Targets	[/etc/iscsi/targetshw]	+

หมายเหตุ: หาก你有คำถามเกี่ยวกับวัตถุประสงค์ของฟิลด์เฉพาะให้วางเคอร์เซอร์ในฟิลด์ และกด F1 สำหรับวิธีใช้

การจัดการฮอตปลั๊ก PCI

คุณสามารถเสียบอะแดปเตอร์ฮอตปลั๊ก PCI ใหม่เข้าในสล็อต PCI ที่มีอยู่ในขณะที่ระบบปฏิบัติการกำลังรัน

อะแดปเตอร์นี้สามารถเป็นอะแดปเตอร์อีกตัวหนึ่งของชนิดเดียวกันกับที่ติดตั้ง อยู่ในปัจจุบัน หรืออาจเป็นอะแดปเตอร์ PCI ชนิดอื่นก็ได้ รีซอร์สใหม่จะถูกทำให้พร้อมใช้งานสำหรับ ระบบปฏิบัติการและแอปพลิเคชันโดยไม่ต้องรีสตาร์ท ระบบปฏิบัติการ เหตุผลบางอย่างสำหรับการเพิ่มอะแดปเตอร์ฮอตปลั๊กอาจรวมถึง:

- การเพิ่มฟังก์ชันหรือความสามารถเพิ่มเติมลงในฮาร์ดแวร์และเฟิร์มแวร์ที่มีอยู่ของคุณ
- การย้ายอะแดปเตอร์ PCI จากระบบซึ่งไม่ต้องการการทำงานของอะแดปเตอร์ดังกล่าวอีกต่อไป

- การติดตั้งระบบใหม่ซึ่งอะแดปเตอร์การ์ดจะมีอยู่หลังจากคอนฟิกูเรชัน แรกเริ่มของระบบย่อยฮาร์ดแวร์ที่เป็นทางเลือก รวมถึงอะแดปเตอร์ PCI และการติดตั้งและการเริ่มต้นของระบบปฏิบัติการ

หมายเหตุ: หากคุณเพิ่มอะแดปเตอร์โดยใช้การดำเนินงานแทนที่หรือเพิ่มฮอตปลั๊ก PCI หรือใช้ Dynamic Logical Partitioning อะแดปเตอร์และอุปกรณ์ภายนอก อาจไม่พร้อมใช้งานสำหรับข้อกำหนดคุณลักษณะที่จะใช้เป็นอุปกรณ์บูตเมื่อใช้คำสั่ง `bootlist` คุณอาจต้องรีสตาร์ทเครื่องเพื่อทำให้ระบบปฏิบัติการรู้จักอุปกรณ์บูตที่เป็นไปได้ทั้งหมด อะแดปเตอร์ที่แสดงในรายการบูตอยู่แล้ว ซึ่งจากนั้นจะถูก แทนที่โดยชนิดอะแดปเตอร์ที่ระบุ ยังคงเป็นอุปกรณ์บูตที่ถูกต้อง

คุณยังสามารถลบอะแดปเตอร์ฮอตปลั๊ก PCI ที่บกพร่องหรือล้มเหลว หรือแลกเปลี่ยนกับอะแดปเตอร์ตัวอื่นที่เป็นชนิดเดียวกันได้โดยไม่ต้องปิด ระบบ เมื่อคุณแลกเปลี่ยนอะแดปเตอร์ ไดรเวอร์อุปกรณ์ที่มีอยู่สนับสนุนอะแดปเตอร์ เนื่องจากเป็นชนิดเดียวกัน คอนฟิกูเรชันอุปกรณ์และข้อมูลคอนฟิกูเรชัน เกี่ยวกับอุปกรณ์ที่ต่ำกว่าคอนฟิกูเรชันมีการคงไว้สำหรับอุปกรณ์ที่เปลี่ยน เหตุผลบางอย่างสำหรับการเปลี่ยนอะแดปเตอร์อาจรวมถึง:

- การเปลี่ยนการ์ดเป็นการชั่วคราวเพื่อช่วยในการระบุปัญหา หรือเพื่อแยกแยะ FRU ที่ล้มเหลว
- การเปลี่ยนอะแดปเตอร์ที่มีตำหนิ ล้มเหลว หรือล้มเหลวเป็นครั้งคราวเนื่องจาก การ์ดการทำงาน
- การเปลี่ยนอะแดปเตอร์ทำซ้ำที่ล้มเหลวใน HACMP™ หรือคอนฟิกูเรชัน I/O หลายพาส

เมื่อคุณเอาอะแดปเตอร์ฮอตปลั๊กออก ริชอร์สที่นำเสนอโดยอะแดปเตอร์นั้น จะไม่มีอยู่สำหรับระบบปฏิบัติการและแอปพลิเคชันต่างๆ เหตุผลบางอย่าง สำหรับการเอาอะแดปเตอร์ออกอาจรวมถึง:

- การลบระบบย่อย I/O ที่มีอยู่
- การลบอะแดปเตอร์ที่ไม่ต้องการอีกต่อไป หรือที่ล้มเหลวและไม่มีการ์ด สำหรับเปลี่ยน
- การย้ายอะแดปเตอร์ไปยังระบบอื่นเมื่อไม่ต้องการการทำงานอีกต่อไปบนระบบ ซึ่งกำลังจะย้ายไป

ก่อนคุณสามารถลบหรือเปลี่ยนอุปกรณ์ฮอตปลั๊ก ต้องยกเลิกกำหนดคอนฟิกูเรชันนั้น ไดรเวอร์อุปกรณ์ที่เชื่อมโยงต้องว่างจากริชอร์สระบบใดๆ ที่มีการจัดสรร สำหรับอุปกรณ์ ซึ่งรวมถึง unpinning และการทำให้หน่วยความจำว่าง การยกเลิกการกำหนด ตัวจัดการการขัดจังหวะและ EPOW, การรีเซ็ต DMA และริชอร์สตัวจับเวลา และขั้นตอนที่จำเป็น อื่นๆ ไดรเวอร์ยังต้องแน่ใจว่าการขัดจังหวะ หน่วยความจำบัส และบัส I/O ถูกปิดใช้งานบนอุปกรณ์

ผู้ดูแลระบบต้องทำภารกิจต่อไปนี้ทั้งก่อนและหลังจาก การลบอะแดปเตอร์:

- ยุติและเรียกคืนแอปพลิเคชัน, daemons, หรือโปรเซสที่ใช้อุปกรณ์
- ยกเลิกการติดตั้งและติดตั้งระบบไฟล์อีกครั้ง
- ลบและสร้างนิยามอุปกรณ์ขึ้นใหม่ และทำการดำเนินงานอื่นที่จำเป็น เพื่อให้ให้อุปกรณ์ที่ใช้งานอยู่ว่าง
- วางระบบไว้ในสภาพที่ปลอดภัยสำหรับการรับเซอร์วิส
- จัดหาและติดตั้งไดรเวอร์อุปกรณ์ที่จำเป็น

การดำเนินการลบและแทนที่ล้มเหลว นอกจากอุปกรณ์ที่เชื่อมต่อ กับสล็อตที่ระบุถูกยกเลิกกำหนดคอนฟิกและอยู่ในสถานะที่กำหนด คุณสามารถทำเช่นนี้โดยใช้คำสั่ง `rmdev` ก่อนการวางอะแดปเตอร์ ในสถานะที่กำหนด ให้ปิดแอปพลิเคชันทั้งหมดที่กำลังใช้อะแดปเตอร์ มิฉะนั้น คำสั่งจะไม่สำเร็จ สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง `rmdev` ให้ดูที่ `rmdev`

ในบางกรณี คุณยังสามารถทำภารกิจต่อไปนี้:

- จัดเตรียมอะแดปเตอร์ฮอตปลั๊ก PCI ที่จะเสียบ เอาออก หรือเปลี่ยน
- ระบุสล็อตหรืออะแดปเตอร์ PCI ที่เกี่ยวข้องในการดำเนินงานฮอตปลั๊ก

- เอาออกหรือเสียบอะแดปเตอร์ฮอตปลั๊ก PCI

หมายเหตุ: ในระหว่างการดำเนินการฮอตปลั๊ก Object Data Manager (ODM) จะยังคงถูกล็อกอยู่ ดังนั้น ภารกิจอื่นๆ ที่จำเป็นต้องใช้ ODM อาจแฉกหรือล้มเหลว การกำหนดคอนฟิกแบบ cluster-wide เปลี่ยนแปลง ซึ่งเริ่มต้นโดยโหนดอื่นอาจแฉกหรือล้มเหลวในคลัสเตอร์ ดังนั้น ให้แน่ใจว่าภารกิจไม่ถูกดำเนินการจนกว่าการดำเนินการ ฮอตปลั๊กจะเสร็จสมบูรณ์

ข้อควรสนใจ: แม้ว่าการจัดการฮอตปลั๊ก PCI มีความสามารถในการเพิ่ม การเอาออก และการเปลี่ยนอะแดปเตอร์ PCI ได้โดยไม่ต้องปิดระบบ หรือรีสตาร์ทระบบปฏิบัติการ แต่บางอุปกรณ์ในฮอตปลั๊กสล็อตไม่สามารถจัดการ ในลักษณะนี้ได้ ตัวอย่างเช่น ฮาร์ดดิสก์ที่ประกอบขึ้นเป็นกลุ่มวอลุ่ม rootvg หรือ I/O คอนโทรลเลอร์ซึ่งมีการต่อพ่วงไม่สามารถเอาออก หรือเปลี่ยนโดยไม่ต้องปิดระบบ เนื่องจากเป็นอุปกรณ์ที่จำเป็นสำหรับการรัน ระบบปฏิบัติการ หากมีการมีเรอร์กลุ่มวอลุ่ม rootvg คุณสามารถใช้ คำสั่ง `chpv` เพื่อนำดิสก์มาออฟไลน์ หากกลุ่มวอลุ่ม rootvg ตั้งอยู่บนดิสก์ตั้งแต่หนึ่งรายการขึ้นไปซึ่งมีการเปิดใช้งาน Multi-Path I/O (MPIO) และเชื่อมต่อกับ I/O คอนโทรลเลอร์หลายตัว I/O คอนโทรลเลอร์ตัวใดตัวหนึ่ง เหล่านี้สามารถเอาออก (หรือเปลี่ยน) ได้โดยไม่ต้องรีบูตระบบ ในสถานการณ์นี้ พาททั้งหมดจาก I/O คอนโทรลเลอร์ที่จะเอาออก (หรือเปลี่ยน) ควรมีการยกเลิก กำหนดคอนฟิกโดยใช้คำสั่ง `rmdev -R` บนอะแดปเตอร์ คำสั่งนี้จะยกเลิกกำหนดคอนฟิก พาท และอะแดปเตอร์ จากนั้น คุณสามารถทำการจัดการฮอตปลั๊กต่อไป ก่อนคุณพยายามเอาออกหรือเสียบอะแดปเตอร์ฮอตปลั๊ก PCI ให้อ้างอิง *การอ้างอิงการวางตำแหน่งอะแดปเตอร์ PCI* (ซึ่งจัดส่งมาพร้อมกับยูนิตรบบที่สนับสนุน ฮอตปลั๊ก) เพื่อกำหนดว่าอะแดปเตอร์ของคุณสามารถ hot-swapped ได้หรือไม่ โปรดอ้างอิงเอกสารคู่มือ ยูนิตรบบของคุณเพื่อดูคำแนะนำ การติดตั้งหรือการเอาอะแดปเตอร์ออก

หลักการที่เกี่ยวข้อง:

“การเอาอะแดปเตอร์การสื่อสารออก” ในหน้า 603

ก่อนคุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก คุณต้องยกเลิก กำหนดคอนฟิกอะแดปเตอร์นั้น งานที่เกี่ยวข้อง:

“การยกเลิกกำหนดคอนฟิกอะแดปเตอร์หน่วยเก็บ” ในหน้า 610

ก่อนคุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์หน่วยเก็บ คุณต้องยกเลิก กำหนดคอนฟิกอะแดปเตอร์นั้น

“การยกเลิกกำหนดคอนฟิกอะแดปเตอร์อะซิงค์” ในหน้า 611

คุณสามารถยกเลิกกำหนดคอนฟิกอะแดปเตอร์อะซิงค์ได้

การแสดงผลข้อมูลสล็อตฮอตปลั๊ก PCI

ก่อนคุณเพิ่ม เอาออก หรือเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก คุณสามารถ แสดงข้อมูลเกี่ยวกับสล็อตฮอตปลั๊ก PCI ในเครื่อง

คุณสามารถแสดงข้อมูลต่อไปนี้:

- รายการของสล็อตฮอตปลั๊ก PCI ทั้งหมดในเครื่อง
- สล็อตที่มีอยู่หรือว่าง
- สล็อตที่ใช้อยู่ในปัจจุบัน
- ลักษณะของสล็อตเฉพาะ เช่น ชื่อสล็อต คำอธิบาย ชนิดตัวเชื่อมต่อ และชื่ออุปกรณ์ที่ต่อพ่วง

คุณสามารถใช้ SMIT หรือคำสั่งระบบ เพื่อทำภารกิจเหล่านี้ คุณต้อง ล็อกอินเป็นผู้ใช้ราก

โพธิ์เตอร์พาด่วน SMIT

1. พิมพ์ `smit devdrpci` ที่พร้อมระบบ แล้วกด Enter
2. ใช้ไดอะล็อก SMIT เพื่อทำภารกิจให้เสร็จสมบูรณ์

เมื่อต้องการข้อมูลเพิ่มเติมเกี่ยวกับการทำภารกิจให้สมบูรณ์ คุณสามารถเลือกปุ่มวิธีใช้ F1 ในไดอะล็อก SMIT

โพรซีเจอร์คำสั่ง

คุณสามารถใช้คำสั่งต่อไปนี้เพื่อแสดงข้อมูลเกี่ยวกับสล็อตฮอตพลา๊ก์ และอุปกรณ์ที่เชื่อมต่อ:

- คำสั่ง `Isslot` แสดงรายการของสล็อตฮอตพลา๊ก์ PCI ทั้งหมด และลักษณะของสล็อตนั้น
- คำสั่ง `Isdev` แสดงสถานะปัจจุบันของอุปกรณ์ทั้งหมด ที่ติดตั้งในระบบของคุณ

การยกเลิกกำหนดคอนฟิกอะแดปเตอร์การสื่อสาร PCI

ข้อมูลต่อไปนี้ เป็นภาพรวมของโปรเซสการยกเลิกกำหนดคอนฟิก อะแดปเตอร์การสื่อสาร PCI ซึ่งรวมถึงอะแดปเตอร์อีเทอร์เน็ต, Token-ring, FDDI, และ ATM

หากแอปพลิเคชันของคุณใช้โปรโตคอล TCP/IP อยู่ คุณต้อง ลบอินเตอร์เฟซ TCP/IP สำหรับอะแดปเตอร์ออกจากรายการอินเตอร์เฟซ เครือข่ายก่อนคุณสามารถวางอะแดปเตอร์ในสถานะที่กำหนด ใช้คำสั่ง `netstat` เพื่อกำหนดว่า อะแดปเตอร์ของคุณมีการกำหนดคอนฟิกสำหรับ TCP/IP หรือไม่ และตรวจสอบ อินเตอร์เฟซเครือข่ายที่ใช้งานอยู่บนอะแดปเตอร์ของคุณ สำหรับข้อมูลเกี่ยวกับคำสั่ง `netstat` ให้ดูที่ `netstat`

อะแดปเตอร์อีเทอร์เน็ต สามารถมีสองอินเตอร์เฟซคือ: อีเทอร์เน็ตมาตรฐาน (enX) หรือ IEEE 802.3 (etX) X เป็น ตัวเลขเดียวกันกับในชื่ออะแดปเตอร์ entX มีเพียงอินเตอร์เฟซเดียวเท่านั้นสามารถใช้ TCP/IP ในแต่ละครั้ง ตัวอย่างเช่น อะแดปเตอร์อีเทอร์เน็ต ent0 สามารถมีอินเตอร์เฟซ en0 และ et0

อะแดปเตอร์ Token ring สามารถมีได้เพียงอินเตอร์เฟซเดียวคือ: Token-ring (trX) X เป็น ตัวเลขเดียวกันกับในชื่ออะแดปเตอร์ tokX ตัวอย่างเช่น อะแดปเตอร์ Token-ring tok0 มีอินเตอร์เฟซ tr0

อะแดปเตอร์ ATM สามารถมีอินเตอร์เฟซ atm หนึ่งรายการเท่านั้น: ATM (atX) X เป็น ตัวเลขเดียวกันกับในชื่ออะแดปเตอร์ atmX ตัวอย่างเช่น อะแดปเตอร์ ATM atm0 มีอินเตอร์เฟซ at0 อย่างไรก็ตาม อะแดปเตอร์ ATM สามารถมีโคลเอ็นต์ที่เลียนแบบหลายรายการซึ่งกำลังรันบนอะแดปเตอร์เดียว

คำสั่ง `ifconfig` ลบอินเตอร์เฟซ ออกจากเครือข่าย คำสั่ง `rmdev` ยกเลิกกำหนดคอนฟิกอุปกรณ์ PCI ในขณะที่คณนิยามอุปกรณ์ไว้ในคลาสอ็อบเจกต์อุปกรณ์ แบบกำหนดเอง หลังจากอะแดปเตอร์อยู่ในสถานะที่กำหนดแล้ว คุณสามารถใช้คำสั่ง `drslot` เพื่อลบอะแดปเตอร์

เมื่อต้องการยกเลิกกำหนดคอนฟิกขายน์ของ PCI บัส pci1 และอุปกรณ์อื่นทั้งหมดภายใต้มันในขณะที่ยังนิยามอุปกรณ์ไว้ในคลาสอ็อบเจกต์ อุปกรณ์แบบกำหนดเอง ให้พิมพ์:

```
rmdev -p pci1
```

ระบบ แสดงข้อความคล้ายกับต่อไปนี้:

```
rmt0 Defined
hdisk1 Defined
scsil Defined
ent0 Defined
```

หลักการที่เกี่ยวข้อง:

“การเอาอะแดปเตอร์การสื่อสารออก” ในหน้า 603

ก่อนคุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์ฮอตพลา๊ก์ คุณต้องยกเลิก กำหนดคอนฟิกอะแดปเตอร์นั้น

การเอาออกหรือการเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก PCI

คุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก PCI จากยูนิตรระบบ ได้โดยไม่ต้องปิดระบบปฏิบัติการหรือปิดระบบการเอาอะแดปเตอร์ออกทำให้รีซอร์สที่นำเสนอโดยอะแดปเตอร์นั้น ไม่มีอยู่สำหรับระบบปฏิบัติการและแอปพลิเคชันต่างๆ

ก่อนคุณสามารถเอาอะแดปเตอร์ออก คุณต้องยกเลิกกำหนดคอนฟิกอะแดปเตอร์นั้น

ข้อมูลต่อไปนี้เป็นโพรซีเจอร์สำหรับการเอาอะแดปเตอร์ฮอตปลั๊ก PCI ออก คุณสามารถทำงานเหล่านี้ให้เสร็จสิ้นด้วย SMIT หรือคำสั่งระบบ เพื่อทำภารกิจเหล่านี้ คุณต้อง ล็อกอินเป็นผู้ใช้ราก

การเปลี่ยนอะแดปเตอร์ด้วยอะแดปเตอร์ตัวอื่นที่เป็นชนิดเดียวกัน คงข้อมูลคอนฟิกูเรชันของอะแดปเตอร์ที่ถูกเปลี่ยนไว้ และเปรียบเทียบข้อมูลนั้น กับการ์ดที่นำมาเปลี่ยน ไดรเวอร์อุปกรณ์ที่มีอยู่ของ อะแดปเตอร์ที่ถูกเปลี่ยนต้องสามารถสนับสนุนอะแดปเตอร์ที่นำมาเปลี่ยนได้

โพรซีเจอร์พาด่วน SMIT

1. พิมพ์ `smit devdrpci` ที่พร้อมระบบ แล้วกด Enter
2. ใช้ไดอะล็อก SMIT เพื่อทำภารกิจให้เสร็จสมบูรณ์

เมื่อต้องการข้อมูลเพิ่มเติมเกี่ยวกับการทำภารกิจให้สมบูรณ์ คุณสามารถ เลือกปุ่มวิธีใช้ F1 ในไดอะล็อก SMIT

โพรซีเจอร์คำสั่ง

คุณสามารถใช้คำสั่งต่อไปนี้เพื่อแสดงข้อมูลเกี่ยวกับสล็อตฮอตปลั๊ก และอุปกรณ์ที่เชื่อมต่อ และเพื่อเอาอะแดปเตอร์ฮอตปลั๊ก PCI ออก:

- คำสั่ง `lsslot` แสดงรายการของสล็อตฮอตปลั๊ก PCI ทั้งหมด และลักษณะของสล็อตนั้น สำหรับข้อมูลเกี่ยวกับการใช้คำสั่งนี้ ให้ดูที่ `lsslot` ใน *Commands Reference, Volume 3*
- คำสั่ง `lsdev` แสดงสถานะปัจจุบันของอุปกรณ์ทั้งหมด ที่ติดตั้งในระบบของคุณ สำหรับข้อมูลเกี่ยวกับการใช้คำสั่งนี้ ให้ดูที่ `lsdev` ใน *Commands Reference, Volume 3*
- คำสั่ง `drslot` จัดเตรียมสล็อตฮอตปลั๊กสำหรับการเอาอะแดปเตอร์ฮอตปลั๊กออก สำหรับข้อมูลเกี่ยวกับการใช้คำสั่งนี้ ให้ดูที่ `drslot` ใน *Commands Reference, Volume 2*

หลักการที่เกี่ยวข้อง:

“การเอาอะแดปเตอร์การสื่อสารออก” ในหน้า 603

ก่อนคุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก คุณต้องยกเลิก กำหนดคอนฟิกอะแดปเตอร์นั้น

งานที่เกี่ยวข้อง:

“การยกเลิกกำหนดคอนฟิกอะแดปเตอร์หน่วยเก็บ” ในหน้า 610

ก่อนคุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์หน่วยเก็บ คุณต้องยกเลิก กำหนดคอนฟิกอะแดปเตอร์นั้น

“การยกเลิกกำหนดคอนฟิกอะแดปเตอร์อะซิงค์” ในหน้า 611

คุณสามารถยกเลิกกำหนดคอนฟิกอะแดปเตอร์อะซิงค์ได้

การเพิ่มอะแดปเตอร์ฮอตปลั๊ก PCI

คุณสามารถเพิ่มอะแดปเตอร์ฮอตปลั๊ก PCI เข้าในสล็อตที่มีอยู่ในยูนิตรระบบ และทำให้รีซอร์สใหม่พร้อมใช้งานสำหรับระบบปฏิบัติการและแอปพลิเคชัน ได้โดยไม่ต้องรีบูตระบบปฏิบัติการ อะแดปเตอร์นี้สามารถเป็นอะแดปเตอร์อีกตัวหนึ่งของชนิดเดียวกันกับที่ติดตั้งอยู่ในปัจจุบัน หรืออาจเป็นอะแดปเตอร์ชนิดอื่นก็ได้

ข้อมูลต่อไปนี้เป็นโพธิ์เตอร์สำหรับการเพิ่มอะแดปเตอร์ฮอตปลั๊ก PCI ใหม่

ข้อควรสนใจ: ก่อนคุณพยายามเพิ่มอะแดปเตอร์ฮอตปลั๊ก PCI ให้อ้างอิง *การอ้างอิงการวางตำแหน่งอะแดปเตอร์ PCI* ซึ่งจัดส่งมาพร้อมกับยูนิตรบบที่สนับสนุน ฮอตปลั๊ก เพื่อกำหนดว่าอะแดปเตอร์ของคุณสามารถเป็นฮอตปลั๊กได้หรือไม่ โปรดอ้างอิงเอกสารคู่มือ ยูนิตรบบของคุณเพื่อดูคำแนะนำการติดตั้งหรือการเอาอะแดปเตอร์ออก

การเพิ่มอะแดปเตอร์ฮอตปลั๊ก PCI ใหม่ เกี่ยวข้องกับภารกิจต่อไปนี้:

- การค้นหาและการระบุสล็อตที่มีอยู่ในเครื่อง
- การจัดเตรียมสล็อตสำหรับการกำหนดคอนฟิเกอะแดปเตอร์
- การติดตั้งไดรเวอร์อุปกรณ์ ถ้าจำเป็น
- การกำหนดคอนฟิเกอะแดปเตอร์ใหม่

คุณสามารถใช้ SMIT หรือคำสั่งระบบ เพื่อทำภารกิจเหล่านี้ คุณต้อง ล็อกอินเป็นผู้ใช้ราก

หมายเหตุ: เมื่อคุณเพิ่มอะแดปเตอร์ฮอตปลั๊กในระบบ อะแดปเตอร์นั้น พร้อมอุปกรณ์ชายน่าอาจไม่พร้อมใช้งานสำหรับข้อกำหนดคุณลักษณะที่จะใช้ เป็น อุปกรณ์บูตเมื่อคุณใช้คำสั่ง `bootlist` คุณอาจจำเป็นต้อง รีบูตระบบของคุณ เพื่อให้ระบบปฏิบัติการรู้จักอุปกรณ์บูตที่เป็นไปได้ทั้งหมด

โพธิ์เตอร์พาด่วน SMIT

1. พิมพ์ `smit devdrpci` ที่พร้อมระบบ แล้วกด Enter
2. ใช้ไดอะล็อก SMIT เพื่อทำภารกิจให้เสร็จสมบูรณ์

เมื่อต้องการข้อมูลเพิ่มเติมเกี่ยวกับการทำภารกิจให้สมบูรณ์ คุณสามารถ เลือกปุ่มวิธีใช้ F1 ในไดอะล็อก SMIT

โพธิ์เตอร์คำสั่ง

คุณ สามารถใช้คำสั่งต่อไปนี้เพื่อแสดงข้อมูลเกี่ยวกับสล็อตฮอตปลั๊ก PCI และอุปกรณ์ที่เชื่อมต่อ และเพื่อเพิ่มอะแดปเตอร์ฮอตปลั๊ก PCI:

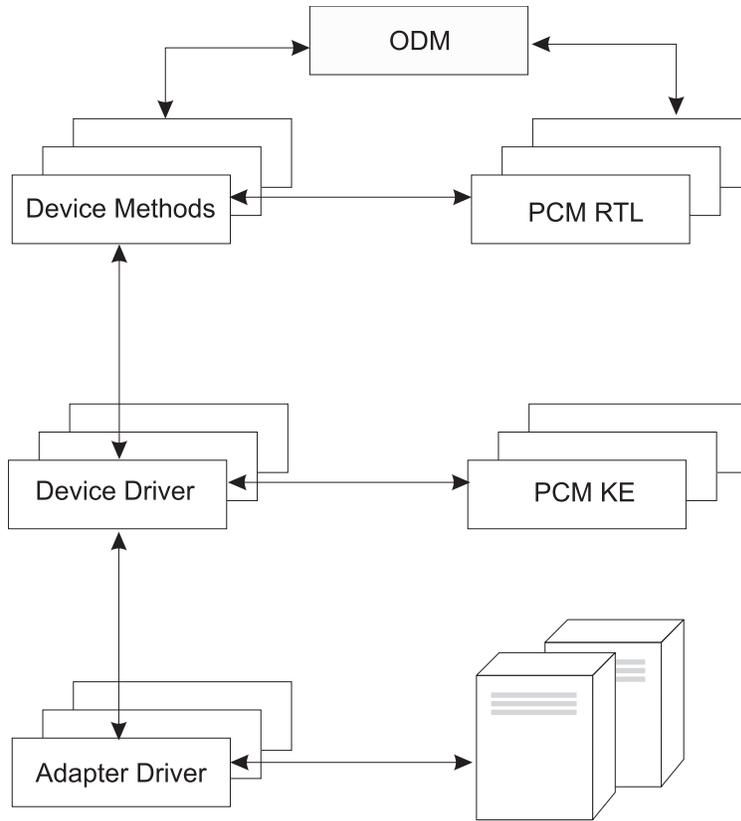
- คำสั่ง `lsslot` แสดงรายการของสล็อตฮอตปลั๊กทั้งหมด และลักษณะของสล็อตนั้น สำหรับข้อมูลเกี่ยวกับการใช้คำสั่งนี้ ให้ดูที่ `lsslot` ใน *Commands Reference, Volume 3*
- คำสั่ง `lsdev` แสดงสถานะปัจจุบันของอุปกรณ์ทั้งหมด ที่ติดตั้งในระบบของคุณ สำหรับข้อมูลเกี่ยวกับการใช้คำสั่งนี้ ให้ดูที่ `lsdev` ใน *Commands Reference, Volume 3*
- คำสั่ง `drslot` จัดเตรียมสล็อตฮอตปลั๊กสำหรับการเพิ่ม หรือการเอาอะแดปเตอร์ฮอตปลั๊กออก สำหรับข้อมูลเกี่ยวกับการใช้คำสั่งนี้ ให้ดูที่ `drslot` ใน *Commands Reference, Volume 2*

Multiple Path I/O

ด้วย Multiple Path I/O (MPIO) อุปกรณ์สามารถตรวจพบได้ไม่ซ้ำกัน ผ่านทางการเชื่อมต่อฟิสิคัลตั้งแต่หนึ่งรายการขึ้นไป หรือ *พาท*

โมดูลควบคุมพาท (PCM) นำเสนอฟังก์ชันการจัดการพาท

ไดรเวอร์อุปกรณ์ที่มีความสามารถ MPIO สามารถควบคุมอุปกรณ์เป้าหมายได้มากกว่าหนึ่งชนิด PCM สามารถสนับสนุนอุปกรณ์เฉพาะตั้งแต่หนึ่งรายการขึ้นไป ด้วยเหตุนี้ไดรเวอร์อุปกรณ์หนึ่งจึงสามารถอินเตอร์เฟสกับหลาย PCMs ที่ควบคุม I/O บนพาร์ต ไปยังแต่ละอุปกรณ์เป้าหมาย



รูปที่ 12. การโต้ตอบระหว่างคอมพิวเตอร์ MPIO. ภาพสาธิตนี้แสดง การโต้ตอบระหว่างคอมพิวเตอร์ต่างๆ ที่ประกอบขึ้นเป็นโซลูชัน MPIO ในรูปภาพนี้ไดรเวอร์อุปกรณ์ MPIO ควบคุมอุปกรณ์เป้าหมายหลายชนิด แต่ละชนิดต้องการ PCM ที่แตกต่างกัน (KE=Kernel Extension, RTL=Run-time Loadable)

ก่อนอุปกรณ์สามารถใช้ MPIO ได้ไดรเวอร์ของอุปกรณ์ เมธอด และแอ็ททริบิวต์ที่กำหนดไว้ล่วงหน้าใน Object Data Manager (ODM) ต้องได้รับการแก้ไขเพื่อสนับสนุนการตรวจพบ คอนฟิกูเรชัน และการจัดการของหลายพาร์ต ไดรเวอร์ อุปกรณ์ดิสก์ SCSI ขนานและ Fibre Channel และเมธอดอุปกรณ์ของไดรเวอร์ ดังกล่าวสนับสนุนอุปกรณ์ดิสก์ MPIO อุปกรณ์ดิสก์ iSCSI ได้รับการสนับสนุนเป็นอุปกรณ์ MPIO ไดรเวอร์อุปกรณ์เทป Fibre Channel และเมธอดของอุปกรณ์สนับสนุนอุปกรณ์เทป MPIO นอกจากนี้ แอ็ททริบิวต์ที่กำหนดไว้ล่วงหน้าสำหรับบางอุปกรณ์ ใน ODM ยังได้รับการแก้ไขสำหรับ MPIO ด้วย

AIX PCM ประกอบด้วย: โมดูลคอนฟิกูเรชัน PCM RTL และส่วนขยายเคอร์เนล PCM KE PCM RTL คือโมดูลรันไทม์ที่โหลดได้ซึ่งเปิดใช้งานเมธอดอุปกรณ์เพื่อตรวจพบ PCM KE เพิ่มเติมเฉพาะอุปกรณ์ หรือพาร์ตแอ็ททริบิวต์ ODM ที่ PCM KE ต้องการ PCM RTL ถูกโหลดโดยเมธอดอุปกรณ์ จากนั้น มีการเข้าถึงรูทีนตั้งแต่หนึ่งรายการขึ้นไปภายใน PCM RTL เพื่อทำการดำเนินงานเฉพาะที่เริ่มต้นหรือ แก้ไขตัวแปร PM KE

PCM KE ให้ความสามารถในการจัดการควบคุมพาร์ตแก่ไดรเวอร์อุปกรณ์ใดๆ ที่สนับสนุนอินเตอร์เฟส MPIO PCM KE ตรวจพบพาร์ตและสื่อสารข้อมูลนั้นกับ ไดรเวอร์อุปกรณ์ โดยขึ้นอยู่กับคอนฟิกูเรชันอุปกรณ์ ไดรเวอร์อุปกรณ์ที่มีความสามารถ MPIO แต่ละรายการเพิ่มพาร์ตลงในอุปกรณ์จากพาร์ตระหว่างกลางหรือ พาร์ตใดๆ การบำรุงรักษาและการจัดตารางเวลาของ I/O บนพาร์ตอื่นมีการนำเสนอ โดย PCM KE และไม่ปรากฏในไดรเวอร์อุปกรณ์ที่มีความสามารถ MPIO

PCM KE สามารถนำเสนอมากกว่าหนึ่งขั้นตอนวิธีการเราต์ซึ่งผู้ใช้สามารถเลือกได้ PCM KE ยังช่วยรวบรวมข้อมูลที่สามารถใช้เพื่อ กำหนดและเลือกพาร์ตที่ดีที่สุดสำหรับคำร้องขอ I/O ต่างๆ PCM KE สามารถเลือก พาร์ตที่ดีที่สุดโดยพิจารณาจากหลักเกณฑ์หลายอย่าง รวมถึงความสมดุลในการโหลด ความเร็วในการเชื่อมต่อ ความล้มเหลวในการเชื่อมต่อ และอื่นๆ

AIX PCM มีความสามารถในการตรวจสอบ ความสมบูรณ์ซึ่งสามารถใช้เพื่อทำดังต่อไปนี้:

- ตรวจสอบพาร์ตและกำหนดว่าพาร์ตใดสามารถใช้งานได้ในปัจจุบันสำหรับการส่ง I/O
- เปิดใช้งานพาร์ตที่ถูกทำเครื่องหมายก่อนหน้านี้ว่าล้มเหลวเนื่องจากข้อบกพร่อง ของพาร์ตชั่วคราว (ตัวอย่างเช่น เมื่อสายเคเบิลของอุปกรณ์ถูกถอดออก แล้วเชื่อมต่อใหม่)
- ตรวจสอบพาร์ตที่ไม่ได้ใช้งานในปัจจุบันซึ่งอาจจะใช้ถ้า failover เกิดขึ้น (ตัวอย่างเช่น เมื่อค่าแอ็ททริบิวต์ขั้นตอนวิธีเป็น failover การตรวจสอบความสมบูรณ์สามารถทดสอบพาร์ตอื่นได้)

อุปกรณ์ดิสก์และอุปกรณ์เทปบางรายการไม่สามารถตรวจพบและกำหนดคอนฟิก โดยใช้ AIX ดีพอลต์ PCMs AIX ดีพอลต์ PCMs ประกอบด้วย โมดูลควบคุมพาร์ตสองโมดูล โดยโมดูลหนึ่งจัดการกับอุปกรณ์ดิสก์และอีกโมดูลหนึ่ง จัดการกับอุปกรณ์เทป หากตรวจไม่พบอุปกรณ์ของคุณ ให้สอบถามจากผู้ขายอุปกรณ์เพื่อให้ทราบว่ามี PCM สำหรับอุปกรณ์ของคุณหรือไม่ **สิ่งอ้างอิงที่เกี่ยวข้อง:**

“แอ็ททริบิวต์สำหรับเทป SCSI อื่น (ชนิด ost)” ในหน้า 629
ข้อมูลต่อไปนี้เป็นแอ็ททริบิวต์สำหรับเทป SCSI อื่น (ชนิด ost)

การจัดการอุปกรณ์ที่มีความสามารถ MPIO

สามารถใช้คุณลักษณะ Multiple Path I/O (MPIO) เพื่อกำหนดพาร์ตอื่น ไปยังอุปกรณ์สำหรับวัตถุประสงค์ failover

Failover คือขั้นตอนวิธีการจัดการพาร์ตที่ปรับปรุงความน่าเชื่อถือ และการมีอยู่ของอุปกรณ์ เนื่องจากระบบจะตรวจพบโดยอัตโนมัติเมื่อ พาร์ต I/O หนึ่งล้มเหลวและรีเราต์ I/O ผ่านทางพาร์ตอื่น All SCSI SCSI disk drives are automatically configured as MPIO devices and a select number of Fibre Channel disk drives can be configured as MPIO Other disk. อุปกรณ์อื่นสามารถได้รับการสนับสนุนได้ หากว่าไดรเวอร์อุปกรณ์เข้ากันได้กับการนำ MPIO ไปใช้ใน AIX

MPIO มีการติดตั้งและกำหนดคอนฟิกเป็นส่วนหนึ่งของการติดตั้ง BOS ไม่จำเป็นต้องทำ คอนฟิกเรชั่นเพิ่มเติม แต่คุณสามารถเพิ่ม ลบ กำหนดคอนฟิกอีกครั้ง เปิดใช้งาน และปิดใช้งานอุปกรณ์ (หรือพาร์ตอุปกรณ์) โดยใช้ SMI หรืออินเตอร์เฟซบรรทัดคำสั่ง คำสั่งต่อไปนี้จะช่วยจัดการพาร์ต MPIO:

mkpath

เพิ่มพาร์ตลงในอุปกรณ์เป้าหมาย

rmpath ลบพาร์ตออกจากอุปกรณ์เป้าหมาย

chpath เปลี่ยนแอ็ททริบิวต์หรือสถานะการดำเนินงานของพาร์ตไปยังอุปกรณ์เป้าหมาย

lspath แสดงข้อมูลเกี่ยวกับพาร์ตไปยังอุปกรณ์เป้าหมาย

หลักการที่เกี่ยวข้อง:

“ฐานข้อมูลคอนฟิกเรชั่นอุปกรณ์และการจัดการอุปกรณ์” ในหน้า 579

ข้อมูลอุปกรณ์มีอยู่ในฐานข้อมูลที่กำหนดไว้ล่วงหน้า หรือฐานข้อมูล แบบกำหนดเองซึ่งประกอบเป็นฐานข้อมูลคอนฟิกเรชั่นอุปกรณ์

“การกำหนดคอนฟิกอุปกรณ์ MPIO” ในหน้า 595

การกำหนดคอนฟิกอุปกรณ์ที่มีความสามารถ MPIO ใช้คำสั่งเดียวกันกับ อุปกรณ์ที่ไม่ใช่ MPIO

การเดินสายเคเบิลอุปกรณ์ SCSI เป็นอุปกรณ์ MPIO:

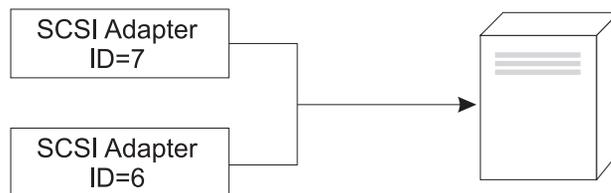
อุปกรณ์ SCSI สามารถได้รับการสนับสนุนจากอะแดปเตอร์สูงสุดสองตัว เมื่อกำหนดคอนฟิกเป็นอุปกรณ์ที่มีความสามารถ MPIO

เมื่อต้องการเดินสายเคเบิลอุปกรณ์ SCSI ขนานเป็นอุปกรณ์ MPIO ให้ใช้ คอนฟิกูเรชันแบบง่ายเป็นตัวอย่าง ข้อมูลต่อไปนี้ เป็นคอนฟิกูเรชันต่ำสุด ที่ต้องทำ อย่างไรก็ตาม อุปกรณ์ของคุณอาจต้องการคอนฟิกูเรชันเพิ่มเติม

1. ในขณะที่ระบบปิดอยู่ให้ติดตั้งอะแดปเตอร์ SCSI สองตัว
2. เดินสายเคเบิลจากอุปกรณ์ไปยังอะแดปเตอร์ SCSI ทั้งสองตัว
3. เปิดระบบ
4. เปลี่ยนค่าติดตั้งบนอะแดปเตอร์ตัวใดตัวหนึ่งเป็น SCSI ID เฉพาะ โดยค่าดีฟอลต์ อะแดปเตอร์ SCSI มี SCSI ID เป็น 7 เนื่องจากแต่ละ ID ต้องไม่ซ้ำกัน ดังนั้นให้เปลี่ยนอะแดปเตอร์หนึ่งเป็นหมายเลขอื่น ตัวอย่างเช่น 6
5. รันคำสั่ง `cfgmgr`
6. เมื่อต้องการตรวจสอบคอนฟิกูเรชัน ให้พิมพ์ดังต่อไปนี้บนบรรทัด คำสั่ง:

```
lspath -l hdiskX
```

โดยที่ *X* คือ หมายเลขโลจิคัลของอุปกรณ์ที่กำหนดคอนฟิกใหม่ เอาต์พุตคำสั่งควร แสดงสองพาทและสถานะของพาทนั้น



รูปที่ 13. คอนฟิกูเรชันสายเคเบิลสำหรับอุปกรณ์ MPIO SCSI

ภาพสาธิตนี้แสดงการเดินสายเคเบิลอะแดปเตอร์ SCSI สองตัวไปยังอุปกรณ์เดียวกัน

การเดินสายเคเบิลอุปกรณ์ Fibre Channel เป็นอุปกรณ์ MPIO:

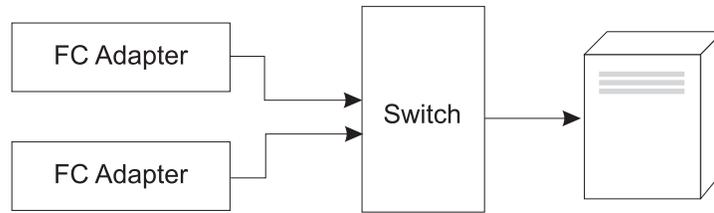
อุปกรณ์ Fibre Channel สามารถมีการเดินสายเคเบิลไปยังหลายอะแดปเตอร์ ไม่มี ซีดจำกัดภายในซอฟต์แวร์

เมื่อต้องการเดินสายเคเบิลอุปกรณ์ Fibre Channel เป็นอุปกรณ์ MPIO ให้ใช้คอนฟิกูเรชันแบบง่ายต่อไปนี้เป็นตัวอย่างเป็นตัวอย่าง ข้อมูลต่อไปนี้ เป็นคอนฟิกูเรชันต่ำสุด ที่ต้องทำ อย่างไรก็ตาม อุปกรณ์ของคุณอาจต้องการคอนฟิกูเรชันเพิ่มเติม

1. ในขณะที่ระบบปิดอยู่ให้ติดตั้งอะแดปเตอร์ Fibre Channel สองตัว
2. เดินสายเคเบิลจากอะแดปเตอร์ไปยังสวิตช์หรือฮับ
3. เดินสายเคเบิลจากอุปกรณ์ไปยังสวิตช์หรือฮับ
4. Power on the system.
5. เมื่อต้องการตรวจสอบคอนฟิกูเรชัน ให้พิมพ์ดังต่อไปนี้บนบรรทัด คำสั่ง:

```
lspath -l hdiskX
```

โดยที่ X คือ หมายเลขโลจิคัลของอุปกรณ์ที่กำหนดคอนฟิกใหม่ เอาต์พุตคำสั่งควร แสดงหนึ่งพารสำหรับแต่ละอะแดปเตอร์ซึ่งคุณติดตั้ง และสถานะของแต่ละพาร



รูปที่ 14. คอนฟิกูเรชันสายเคเบิลสำหรับอุปกรณ์ MPIO Fibre Channel. ภาพสาธิตนี้แสดงคอนฟิกูเรชันแบบง่ายของอะแดปเตอร์ Fibre Channel สองตัวไปยังสวิตช์ ซึ่งมีการเดินสายเคเบิลไปยังอุปกรณ์

การกำหนดคอนฟิกอุปกรณ์ MPIO

การกำหนดคอนฟิกอุปกรณ์ที่มีความสามารถ MPIO ใช้คำสั่งเดียวกันกับ อุปกรณ์ที่ไม่ใช่ MPIO

คำสั่ง `cfgmgr`, `mkdev`, `chdev`, `rmdev` และ `lsdev` สนับสนุนการจัดการอินสแตนซ์ของอุปกรณ์ MPIO และแสดงแอตทริบิวต์อินสแตนซ์อุปกรณ์ MPIO ยังมี พารอินสแตนซ์ที่เชื่อมโยงกับอินสแตนซ์อุปกรณ์ด้วย คำสั่ง `mkpath`, `chpath`, `rmpath`, และ `lspath` จัดการกับพารอินสแตนซ์ และแสดงแอตทริบิวต์

พารอินสแตนซ์สามารถเพิ่มหรือลบออกจากอุปกรณ์ MPIO ได้โดยไม่ต้องยกเลิก กำหนดคอนฟิกอุปกรณ์

อุปกรณ์ MPIO สามารถมีแอตทริบิวต์เพิ่มเติม แต่แอตทริบิวต์บังคับ ที่อุปกรณ์ MPIO ทั้งหมดต้องสนับสนุนคือ `reserve_policy` และ `algorithm` แอตทริบิวต์ `reserve_policy` กำหนดชนิดของระเบียบวิธี การสำรองที่ไดรเวอร์อุปกรณ์จะนำไปใช้เมื่อเปิดอุปกรณ์ และสามารถใช้เพื่อจำกัดสิทธิ์เข้าถึงอุปกรณ์จากอะแดปเตอร์อื่น ไม่ว่าบน ระบบเดียวกันหรือบนระบบอื่น แอตทริบิวต์ `algorithm` กำหนดระเบียบวิธีที่ PCM ใช้เพื่อจัดการ I/O บนพารที่กำหนดคอนฟิก สำหรับอุปกรณ์

หลักการที่เกี่ยวข้อง:

“การจัดการอุปกรณ์ที่มีความสามารถ MPIO” ในหน้า 593

สามารถใช้คุณลักษณะ Multiple Path I/O (MPIO) เพื่อกำหนดพารอื่น ไปยังอุปกรณ์สำหรับวัตถุประสงค์ failover

“แอตทริบิวต์อุปกรณ์ MPIO” ในหน้า 597

แอตทริบิวต์ต่อไปนี้ได้รับการสนับสนุนเฉพาะโดยอุปกรณ์หลายพาร แอตทริบิวต์สามารถแสดงหรือเปลี่ยนแปลงโดยใช้ SMIT, หรือคำสั่ง (โดยเฉพาะอย่างยิ่งคำสั่ง `lsattr` และ `chdev`)

“แอตทริบิวต์โมดูลควบคุมพาร” ในหน้า 598

นอกเหนือจากโมดูลควบคุมพาร (PCMs) ดีฟอลต์ของ AIX ดีฟอลต์แล้ว PCM เฉพาะอุปกรณ์อาจมีการจัดให้โดย ผู้ขาย อุปกรณ์ ชุดของแอตทริบิวต์ที่เปลี่ยนได้โดยผู้มีการกำหนดโดยผู้ขาย อุปกรณ์ PCM เฉพาะอุปกรณ์สามารถมีแอตทริบิวต์อุปกรณ์และพาร

อุปกรณ์หลายพารที่สนับสนุน

AIX ดีฟอลต์ PCMs สนับสนุนชุดของอุปกรณ์ดิสก์และอุปกรณ์เทปที่กำหนดไว้ในชุดไฟล์ `devices.common.IBM.mpio.rte`

อุปกรณ์ที่ไม่ได้รับการสนับสนุนจาก AIX ดิสก์ PCMs หรือเทป PCMs ต้องการให้ผู้ชายอุปกรณ์นำเสนอแอตทริบิวต์ที่กำหนดไว้ล่วงหน้าใน ODM, PCM, และโค้ดสนับสนุนอื่นใดที่จำเป็นในการรับรู้ว่าเป็นอุปกรณ์ที่มีความสามารถ MPIO

เมื่อต้องการกำหนดว่าอุปกรณ์ดิสก์ใดได้รับการสนับสนุนจาก AIX ดิสก์ PCM ให้รันสคริปต์ต่อไปนี้:

```
odmget -qDvDr=aixdiskpcmke PdDv | grep uniquetype | while read line
do
    utype=`echo $line | cut -d' ' -f2`
    dvc=`odmget -q"uniquetype=$utype AND attribute=dvc_support" PdAt`
    echo $dvc | grep values | cut -d' ' -f2
done
```

เมื่อต้องการกำหนดว่าอุปกรณ์เทปใดได้รับการสนับสนุนจาก AIX ดิสก์ PCM ให้รันสคริปต์ต่อไปนี้:

```
odmget -qDvDr=aixtapepcmke PdDv | grep uniquetype | while read line
do
    utype=`echo $line | cut -d' ' -f2`
    dvc=`odmget -q"uniquetype=$utype AND attribute=dvc_support" PdAt`
    echo $dvc | grep values | cut -d' ' -f2
done
```

เอาต์พุตสคริปต์แสดงรายการของชนิดอุปกรณ์เฉพาะที่สนับสนุน โดย AIX ดีพอลต์ PCMs อุปกรณ์สามชนิดที่ได้รับการสนับสนุนจาก AIX ดิสก์ PCM คือ *ดิสก์ SCSI ขนาดแบบกำหนดคอนฟิกเอง* (disk/scsi/scsd) และ *FC ดิสก์ MPIO อื่น* (disk/fcp/mpioosdisk) และ *iSCSI MPIO อื่น* (disk/iscsi/mpioosdisk) ชนิดอุปกรณ์ที่ได้รับการสนับสนุนจาก AIX เทป PCM คือ *FC เทป MPIO อื่น* (tape/fcp/mpioost)

อุปกรณ์ *FC ดิสก์ MPIO อื่น* และ *FC เทป MPIO อื่น* เป็นชุดย่อยของดิสก์ Fibre Channel อื่นและเทป Fibre Channel อื่น ตามลำดับ อุปกรณ์ได้รับการสนับสนุนเป็นอุปกรณ์ *FC MPIO อื่น* เฉพาะถ้า ไม่มีแอตทริบิวต์ที่กำหนดไว้ล่วงหน้า ODM ซึ่งกำหนดโดยผู้ชายและอุปกรณ์ได้รับการรับรองในการทำงานกับ AIX ดีพอลต์ PCMs อย่างไม่อย่างหนึ่ง ใรับรองไม่ได้รับประกันว่าความสามารถของอุปกรณ์ทั้งหมดได้รับการสนับสนุนเมื่อ อุปกรณ์มีการกำหนดคอนฟิกเป็นอุปกรณ์ *FC MPIO อื่น*

ดิสก์ *iSCSI MPIO อื่น* เป็นชุดย่อยของดิสก์ iSCSI อื่น อุปกรณ์ได้รับการสนับสนุนเป็นดิสก์ *iSCSI MPIO อื่น* เฉพาะถ้า ไม่มีแอตทริบิวต์ที่กำหนดไว้ล่วงหน้า ODM ซึ่งกำหนดโดยผู้ชายและอุปกรณ์ได้รับการรับรองในการทำงานกับ AIX PCM ใรับรองไม่ได้รับประกันว่าความสามารถของอุปกรณ์ทั้งหมดได้รับการสนับสนุนเมื่อ อุปกรณ์มีการกำหนดคอนฟิกเป็นดิสก์ *iSCSI MPIO อื่น*

เมื่อต้องการกำหนดว่าอุปกรณ์ใดได้รับการสนับสนุนเป็นดิสก์ *FC MPIO อื่น* ให้รันสคริปต์ต่อไปนี้:

```
odmget -quniquetype=disk/fcp/mpioosdisk PdAt | grep deflt | cut -d' ' -f2
```

เมื่อต้องการกำหนดว่าอุปกรณ์ใดได้รับการสนับสนุนเป็นเทป *FC MPIO อื่น* ให้รันสคริปต์ต่อไปนี้:

```
odmget -q "uniquetype=tape/fcp/mpioosdisk AND attribute=mpio_model_map PdAt | grep deflt | cut -d' ' -f2
```

เมื่อต้องการกำหนดว่าอุปกรณ์ใดได้รับการสนับสนุนเป็นดิสก์ *iSCSI MPIO อื่น* ให้รันสคริปต์ต่อไปนี้:

```
odmget -quniquetype=disk/iscsi/mpioosdisk PdAt | grep deflt | cut -d' ' -f2
```

เอาต์พุตสคริปต์แสดงรายการของข้อมูลการสอบถามที่มี ผู้ชายและรุ่นของอุปกรณ์

เมื่อต้องการแสดงอุปกรณ์ที่มีความสามารถ MPIO ทั้งหมดซึ่งมีการติดตั้งบนระบบ ให้รันสคริปต์ต่อไปนี้:

```
odmget -qattribute=unique_id PdAt | grep uniquetype | cut -d'"' -f2
```

เอาต์พุตสคริปต์แสดงรายการของชนิดอุปกรณ์ที่มีความสามารถ MPIO เฉพาะ ซึ่งได้รับการสนับสนุนจาก AIX ดีพอลต์ PCMs และ PCMs ที่นำเสนอโดยผู้ขาย

แอ็ตทริบิวต์อุปกรณ์ MPIO

แอ็ตทริบิวต์ต่อไปนี้ได้รับการสนับสนุนเฉพาะโดยอุปกรณ์หลายพาร แอ็ตทริบิวต์สามารถแสดงหรือเปลี่ยนแปลงโดยใช้ SMIT, หรือคำสั่ง (โดยเฉพาะอย่างยิ่งคำสั่ง `lsattr` และ `chdev`)

แอ็ตทริบิวต์บางอย่างของอุปกรณ์ Multiple Path I/O (MPIO) ถูกเปิดใช้งานสำหรับการอัปเดตพร้อมกันของแอ็ตทริบิวต์ คุณสามารถอัปเดต ค่าแอ็ตทริบิวต์ขณะที่ดิสก์เปิดและใช้งานอยู่ และค่าใหม่ มีผลใช้งานทันที สำหรับบางแอ็ตทริบิวต์ โดยเฉพาะแอ็ตทริบิวต์ `reserve_policy` ซึ่งอาจมีข้อจำกัด เกี่ยวกับเวลาที่สามารถเปลี่ยนแอ็ตทริบิวต์ หรือค่าใหม่ที่แอ็ตทริบิวต์สามารถรับได้ ตัวอย่างเช่น ถ้าดิสก์เปิดและกำลังใช้งานอยู่ เป็นดิสก์ที่เก็บคลัสเตอร์ ระบบปฏิบัติการ AIX จะบล็อกความพยายามตั้งค่านโยบายที่ส่งวนไบนดิสก์ เนื่องจากทำให้ไหนดิสก์อื่นสูญเสียการเข้าถึงที่เก็บ

แอ็ตทริบิวต์อุปกรณ์บังคับซึ่งอุปกรณ์ MPIO ทั้งหมดต้องสนับสนุน คือ `reserve_policy` ตามปกติแล้ว อุปกรณ์หลายพารยังมีแอ็ตทริบิวต์ `PR_key` ด้วย อุปกรณ์หลายพารสามารถมีแอ็ตทริบิวต์เฉพาะอุปกรณ์เป็นการเพิ่มเติมได้ แอ็ตทริบิวต์เฉพาะอุปกรณ์อื่นมีดังนี้:

FC3_REC

บ่งชี้ว่าอุปกรณ์ต้องเปิดการกักกันข้อผิดพลาด ที่ใช้ Fibre Channel class 3 หรือไม่ การเปิดใช้งานคุณลักษณะนี้จะเพิ่มการตรวจจับข้อผิดพลาดและการกักกันข้อผิดพลาดสำหรับข้อผิดพลาด fabric บางอย่างที่เกี่ยวข้องกับไฟเบอร์ แชนแนล แอ็ตทริบิวต์นี้พร้อมใช้งานเฉพาะ ชุดอุปกรณ์จำกัด แอ็ตทริบิวต์นี้สามารถมีค่า ต่อไปนี้:

true เปิดใช้งานการกักกันข้อผิดพลาดที่ใช้ Fibre Channel class 3

false ปิดใช้งานการกักกันข้อผิดพลาดที่ใช้การกักกันข้อผิดพลาด Fibre Channel class 3

reserve_policy

ระบุว่าจะใช้ระเบียบวิธีการสำรองหรือไม่เมื่อเปิด อุปกรณ์ ค่าเป็นดังนี้:

no_reserve

ไม่ใช้ระเบียบวิธีการสำรองสำหรับอุปกรณ์ อุปกรณ์อาจ มีการเข้าถึงโดยตัวเริ่มต้นอื่น และตัวเริ่มต้นเหล่านี้ อาจอยู่บนระบบไฮสตร์อื่น

single_path_reserve

ใช้ระเบียบวิธีการสำรอง SCSI2 สำหรับอุปกรณ์ ซึ่งหมายความว่าอุปกรณ์นั้น สามารถเข้าถึงได้โดยตัวเริ่มต้นที่ออกใช้การสำรองเท่านั้น นโยบายนี้ป้องกัน ไม่ให้ตัวเริ่มต้นอื่นในไฮสตร์เดียวกันหรือบนไฮสตร์อื่นเข้าถึง อุปกรณ์ นโยบายนี้ใช้การสำรอง SCSI2 เพื่อล็อกอุปกรณ์ไว้สำหรับตัวเริ่มต้นเดียว (พาร) และคำสั่งที่เราต์ ผ่านทางพารอื่นจะส่งผลให้เกิดข้อขัดแย้งในการ สำรอง

ขั้นตอนวิธีการเลือกพารที่สลับคำสั่งระหว่างหลายพารอาจส่งผล ให้เกิดการเรชเมื่อเลือกค่า

`single_path_reserve` ในตัวอย่าง สมมติว่า PCM เฉพาะอุปกรณ์มีแอ็ตทริบิวต์บังคับ ที่มีการตั้งค่าเป็นค่าที่ แจกจ่าย I/O บนหลายพาร เมื่อ `single_path_reserve` มีผลบังคับ ใช้ ดิสก์ไดรวเวอร์ต้องออกใช้รีเซ็ท อุปกรณ์บัส (BDR) จากนั้นออกใช้ การสำรองโดยใช้พารใหม่สำหรับการส่งคำสั่งถัดไปเพื่อพักการสำรอง

ก่อนหน้านี้ในแต่ละครั้งที่เลือกพาทอื่น การเรซจะเกิดขึ้นและประสิทธิภาพการทำงาน ต่ำลงเนื่องจากค่า
ใช้จ่ายในการส่ง BDR และการออกใช้ การสำรองไปยังอุปกรณ์เป้าหมาย (AIX PCM ไม่อนุญาตให้คุณเลือก
ขั้นตอนวิธีที่อาจส่งผลให้เกิดการเรซ)

PR_exclusive

ใช้ระเบียบวิธีการสำรองถาวร SCSI3 เฉพาะโฮสต์เมื่อเปิด อุปกรณ์ ค่าแอตทริบิวต์ **PR_key** ต้องไม่ซ้ำกัน
สำหรับแต่ละระบบโฮสต์ แอตทริบิวต์ **PR_key** ใช้เพื่อ ป้องกันการเข้าถึงอุปกรณ์โดยตัวเริ่มต้นจากระบบ
โฮสต์อื่น

PR_shared

ใช้ระเบียบวิธีการสำรองถาวร SCSI3 โฮสต์แบบแบ่งใช้เมื่อเปิด อุปกรณ์ ค่า **PR_key** ต้องไม่ซ้ำกัน สำหรับ
แต่ละระบบโฮสต์ ตัวเริ่มต้นจากระบบโฮสต์อื่นจำเป็นต้องลงทะเบียน ก่อนสามารถเข้าถึงอุปกรณ์ได้

PR_key

ต้องการเฉพาะถ้าอุปกรณ์สนับสนุนนโยบายการสำรองถาวร (**PR_exclusive** หรือ **PR_shared**)

หลักการที่เกี่ยวข้อง:

“การกำหนดคอนฟิกอุปกรณ์ MPIO” ในหน้า 595

การกำหนดคอนฟิกอุปกรณ์ที่มีความสามารถ MPIO ใช้คำสั่งเดียวกันกับ อุปกรณ์ที่ไม่ใช่ MPIO

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง Isattr

คำสั่ง chdev

แอตทริบิวต์โมดูลควบคุมพาร

นอกเหนือจากโมดูลควบคุมพาร (PCMs) ดีพอลต์ของ AIX ดีพอลต์แล้ว PCM เฉพาะอุปกรณ์อาจมีการจัดให้โดย ผู้ขาย
อุปกรณ์ ชุดของแอตทริบิวต์ที่เปลี่ยนได้โดยผู้มีการกำหนดโดยผู้ขาย อุปกรณ์ PCM เฉพาะอุปกรณ์สามารถมีแอตทริบิวต์
อุปกรณ์และพาร

ข้อมูลต่อไปนี้เป็นแอตทริบิวต์อุปกรณ์ของ AIX ดีพอลต์ PCMs:

ขั้นตอนวิธี

กำหนดระเบียบวิธีในการแจกจ่าย I/O บนพารต่างๆ ของอุปกรณ์ แอตทริบิวต์ขั้นตอนวิธีมีค่าต่อไปนี้:

หมายเหตุ: อุปกรณ์บางอย่างสนับสนุนเฉพาะชุดย่อยของ ค่าเหล่านี้

failover

ส่งการดำเนินการ I/O ทั้งหมดเป็นพารเดียว ถ้า พารถูกทำเครื่องหมายเป็นล้มเหลวหรือปิดใช้งาน พารที่
พร้อมใช้งานถัดไปถูกเลือก สำหรับการดำเนินการ I/O ทั้งหมด อัลกอริทึมนี้รักษาพารเปิดใช้งาน ทั้ง
หมดเพื่อแสดงรายการตามค่าน้อยไปมากของแอตทริบิวต์ **path priority** พารที่ถูกต้องที่มีลำดับความ
สำคัญของพารต่ำสุดถูกเลือก สำหรับการดำเนินการ I/O

round_robin

กระจายการดำเนินการ I/O ให้ทั่วพารต่างๆ ที่เปิดใช้งาน สำหรับอุปกรณ์ที่มีพารแบบแอคทีฟและพาสซีฟ
หรือ พารแบบ preferred และแบบ non-preferred เฉพาะชุดย่อยของพาร จะถูกใช้สำหรับการดำเนินการ

I/O ถ้าพารามิเตอร์ถูกทำเครื่องหมายเป็นลัมหรือปิดใช้งาน พารามิเตอร์จะไม่สามารถใช้ส่งการดำเนินการ I/O ได้อีกต่อไป การดำเนินการ I/O ถูกกระจายตามแอตทริบิวต์ **path priority** พารามิเตอร์ที่มีลำดับความสำคัญสูงกว่า จะได้รับส่วนแบ่งของการดำเนินการ I/O มากกว่า

shortest_queue

กระจายการดำเนินการ I/O ให้ทั่วพารามิเตอร์ต่างๆ ที่เปิดใช้งาน สำหรับอุปกรณ์ที่มีพารามิเตอร์แบบแอคทีฟและพาสซีฟ หรือ พารามิเตอร์แบบ preferred และแบบ non-preferred เฉพาะชุดย่อยของพารามิเตอร์จะถูกใช้สำหรับการดำเนินการ I/O อัลกอริทึมนี้ เหมือนกับอัลกอริทึม **round_robin** อย่างไรก็ตาม อัลกอริทึม **shortest_queue** จะกระจายการดำเนินการ I/O ตามจำนวนการดำเนินการ I/O ที่ค้างอยู่บน แต่ละพารามิเตอร์ที่มีการดำเนินการ I/O ที่ค้างอยู่น้อยที่สุด จะถูกเลือกสำหรับการดำเนินการต่อไป แอตทริบิวต์ **path priority** จะถูกละเว้นเมื่ออัลกอริทึมถูกตั้งค่าเป็น **shortest_queue**

hcheck_mode

พิจารณาว่าพารามิเตอร์ใดต้องถูกตรวจสอบเมื่อใช้ความสามารถในการ ตรวจสอบสถานภาพ แอตทริบิวต์สนับสนุนโหมดต่อไปนี้:

enabled

ส่งคำสั่ง **healthcheck** ผ่านพารามิเตอร์ต่างๆ ที่มีสถานะเป็นเปิดใช้งาน โหมดไม่ส่งคำสั่ง **healthcheck** ผ่านพารามิเตอร์ที่มีสถานะปิดใช้งานหรือหายไป

failed

ส่งคำสั่ง **healthcheck** ผ่านพารามิเตอร์ที่มีสถานะลัมหรือโหมดไม่ส่งคำสั่ง **healthcheck** ผ่านพารามิเตอร์ที่มีสถานะเปิดใช้งาน, ปิดใช้งาน หรือหายไป

nonactive

(ดีฟอลต์) ส่งคำสั่ง **healthcheck** ผ่านพารามิเตอร์ที่ไม่มี I/O แอคทีฟกับอุปกรณ์ รวมถึงพารามิเตอร์ที่มีสถานะลัมหรือเปิดใช้งาน โหมดไม่ส่งคำสั่ง **healthcheck** ผ่านพารามิเตอร์ที่มีสถานะปิดใช้งานหรือหายไป

hcheck_interval

กำหนดความถี่ในการทำการตรวจสอบความสมบูรณ์บนพารามิเตอร์ของอุปกรณ์ แอตทริบิวต์สนับสนุนในช่วง 0 - 3600 วินาที เมื่อเลือกค่า 0 การตรวจสอบความสมบูรณ์จะปิดใช้งาน

หมายเหตุ: การตรวจสอบ สถานภาพถูกดำเนินการต่อเมื่อติดสก็ถูกเปิดขึ้นโดยกระบวนการบางอย่างและยังไม่ถูกปิด ถ้าไม่มีเอนทิตีใดติดสก็ถูกเปิด โมดูล Path Control จะไม่ตรวจสอบพารามิเตอร์แม้ว่าแอตทริบิวต์ **hcheck_interval** ของอุปกรณ์นั้นจะถูกตั้งค่าไม่เป็นศูนย์

dist_tw_width

กำหนดช่วงเวลาของ "หน้าต่างเวลา" นี้คือกรอบเวลา ระหว่างที่อัลกอริทึมการตรวจหาข้อผิดพลาดถูกแจกจ่ายรวบรวม I/Os ที่ส่งคืนโดยมีข้อผิดพลาด หน่วยของแอตทริบิวต์ **dist_tw_width** คือมิลลิวินาที การลดค่าแอตทริบิวต์นี้จะลดช่วงเวลาในการเก็บตัวอย่างแต่ละครั้ง และลดความไวของขั้นตอนวิธีต่อชุดขนาดเล็ก ของข้อผิดพลาด I/O การเพิ่มค่าแอตทริบิวต์ นี้จะเพิ่มระดับความถี่อัลกอริทึมเป็นกลุ่มข้อผิดพลาดขนาดเล็ก และความน่าจะเป็นของความลัมหรือของพารามิเตอร์

dist_err_percent

กำหนดเปอร์เซ็นต์ของ "หน้าต่างเวลา" ที่มีข้อผิดพลาดซึ่งได้รับอนุญาตบน พารามิเตอร์ ก่อนพารามิเตอร์จะลัมหรือเนื่องจากประสิทธิภาพที่ไม่ดี โดย **dist_err_percent** มี ช่วง 0 - 100 อัลกอริทึมการตรวจหาข้อผิดพลาดที่แจกจ่ายถูกปิดใช้งานเมื่อแอตทริบิวต์ถูกตั้งค่าเป็นศูนย์ (0) ค่าดีดตั้งดีฟอลต์คือศูนย์ อัลกอริทึมการตรวจหาข้อผิดพลาดที่แจกจ่ายสู่ส่วนตัว

อย่างโครงสร้างที่เชื่อมต่อกับอะแดปเตอร์เพื่อหาข้อผิดพลาด ขั้นตอนวิธีคำนวณเปอร์เซ็นต์ของตัวอย่างที่มีข้อผิดพลาด จากนั้นจะคำนวณความล้มเหลวของพารามิเตอร์ที่คำนวณได้มากกว่าค่าแอ็ดทริบิวต์ `dist_err_percent`

ข้อมูลต่อไปนี้เป็นแอ็ดทริบิวต์พารามิเตอร์ของ AIX PCM:

ลำดับความสำคัญของพารามิเตอร์

แก้ไขลักษณะการทำงานของระบบขั้นตอนวิธีบนรายการของพารามิเตอร์

เมื่อค่าแอ็ดทริบิวต์ ขั้นตอนวิธีเป็น failover พารามิเตอร์จะถูกเก็บไว้ในรายการ ลำดับ ในรายการนี้เป็นตัวกำหนดพารามิเตอร์ที่จะถูกเลือกก่อน และถ้าพารามิเตอร์ล้มเหลว จะกำหนดพารามิเตอร์ที่จะถูกเลือกถัดไป ลำดับมีการกำหนดโดยค่าของแอ็ดทริบิวต์ ลำดับความสำคัญของพารามิเตอร์ ลำดับความสำคัญ 1 คือลำดับความสำคัญสูงสุด หลายพารามิเตอร์สามารถมีค่าลำดับความสำคัญเดียวกันได้ แต่ถ้าพารามิเตอร์ทั้งหมดมีค่าเหมือนกัน การเลือกจะพิจารณาจากเวลาในการกำหนดคอนฟิกแต่ละพารามิเตอร์

เมื่อ ค่าแอ็ดทริบิวต์อัลกอริทึมเป็น `round_robin`, อัลกอริทึม `path priority` กำหนดค่าลำดับความสำคัญ ให้กับแต่ละพารามิเตอร์ พารามิเตอร์ถูกเลือกไว้สำหรับการดำเนินการ I/O ตามสัดส่วน ของลำดับความสำคัญของพารามิเตอร์ ดังนั้น พารามิเตอร์ที่มีค่าลำดับความสำคัญสูง จะถูกเลือกสำหรับการดำเนินการ I/O มากกว่า ถ้าลำดับความสำคัญของพารามิเตอร์ทั้งหมดเท่ากัน พารามิเตอร์จะถูกเลือกอย่างเท่าเทียมกัน

`cntl_hcheck_int`

ลำดับการตรวจสอบสถานะภาพตัวควบคุมถูกเริ่มทำงานหลังจากตรวจพบ ความล้มเหลวของการส่งโครงสร้างหน่วยเก็บข้อมูล แอ็ดทริบิวต์ `cntl_delay_time` พิจารณาระยะเวลาสูงสุดเป็นวินาที เมื่อลำดับการตรวจสอบ สถานภาพตัวควบคุมแอ็ดทริบิวต์ ถ้าคำสั่งตรวจสอบสถานะภาพตัวควบคุมดำเนินการ สำเร็จ จะตรวจหาพารามิเตอร์ที่มี จากนั้นลำดับการตรวจสอบ สถานภาพตัวควบคุมจะออกโดยอนุญาตให้ I/O ทำงานต่อ ที่ตอนท้ายของ ลำดับการตรวจสอบสถานะภาพตัวควบคุม ถ้าไม่พบพารามิเตอร์ที่ต้องการ ดังนั้น I/O ที่ค้างอยู่และที่เกิดขึ้นภายหลังสำหรับอุปกรณ์ที่ล้มเหลว จนกระทั่งตัวตรวจสอบสถานะภาพอุปกรณ์ตรวจพบพารามิเตอร์ล้มเหลวถูกส่งคืน

ขณะที่ ลำดับการตรวจสอบสถานะภาพตัวควบคุมแอ็ดทริบิวต์ แอ็ดทริบิวต์ `cntl_hcheck_interval` จะเป็นจำนวนเวลาเป็นวินาที เมื่อคำสั่งตรวจสอบสถานะภาพตั้งค่า ถัดไปถูกเรียกใช้ แอ็ดทริบิวต์ `cntl_hcheck_interval` ต้องน้อยกว่า `cntl_delay_time` ยกเว้นถูกตั้งค่าเป็น 0 หรือปิดใช้งาน ถ้า `cntl_delay_time` หรือ `cntl_hcheck_interval` ค่าใดค่าหนึ่งถูกตั้งค่าเป็น 0 คุณลักษณะจะถูกปิดใช้งาน

`cntl_delay_time`

ลำดับการตรวจสอบสถานะภาพตัวควบคุมถูกเริ่มทำงานหลังจากตรวจพบ ความล้มเหลวของการส่งโครงสร้างหน่วยเก็บข้อมูล แอ็ดทริบิวต์ `cntl_delay_time` เป็นตัวกำหนดช่วงเวลาสูงสุดในวินาทีซึ่งมีลำดับการตรวจสอบการทำงานของตัวควบคุมทำงานอยู่ ถ้าคำสั่งตรวจสอบสถานะภาพตัวควบคุมดำเนินการ สำเร็จ จะตรวจหาพารามิเตอร์ที่มี จากนั้นลำดับการตรวจสอบ สถานภาพตัวควบคุมจะออกโดยอนุญาตให้ I/O ทำงานต่อ ที่ตอนท้ายของ ลำดับการตรวจสอบสถานะภาพตัวควบคุม ถ้าไม่พบพารามิเตอร์ที่ต้องการ ดังนั้น I/O ที่ค้างอยู่และที่เกิดขึ้นภายหลังสำหรับอุปกรณ์ที่ล้มเหลว จนกระทั่งตัวตรวจสอบสถานะภาพอุปกรณ์ตรวจพบพารามิเตอร์ล้มเหลวถูกส่งคืน

ขณะที่ ลำดับการตรวจสอบสถานะภาพตัวควบคุมแอ็ดทริบิวต์ แอ็ดทริบิวต์ `cntl_hcheck_interval` จะเป็นจำนวนเวลาเป็นวินาที ซึ่งคำสั่งตรวจสอบสถานะภาพตั้งค่า ถัดไปถูกเรียกใช้ แอ็ดทริบิวต์ `cntl_hcheck_interval` ต้องน้อยกว่า `cntl_delay_time` ยกเว้นถูกตั้งค่าเป็น 0 หรือปิดใช้งาน ถ้า `cntl_delay_time` หรือ `cntl_hcheck_interval` ค่าใดค่าหนึ่งถูกตั้งค่าเป็น 0 คุณลักษณะจะถูกปิดใช้งาน

timeout_policy

ปรับเปลี่ยนลักษณะการทำงานของ PCM สำหรับการหมดเวลาใช้งานคำสั่ง และข้อผิดพลาด การถ่ายโอน เมื่อ **timeout_policy** ถูกตั้งค่าเป็น **fail_path** หรือเป็น **disable_path** การลดประสิทธิภาพการทำงานอาจดีขึ้นเมื่ออุปกรณ์ Multiple Path I/O (MPIO) พบปัญหา storage area network (SAN) fabric เป็นช่วงๆ บนพาร์ไปยังอุปกรณ์บางพาร์ แอ็ททริบิวต์ **timeout_policy** มีค่าต่อไปนี้:

retry_path

ค่าแรกของการหมดเวลาใช้งานคำสั่งบนพาร์ไม่ ทำให้พาร์ล้มเหลวในทันที ถ้าพาร์ที่ล้มเหลวเนื่องจาก ปัญหาการถ่ายโอนถูกกู้คืนโดยการตรวจสอบทำเครื่องหมาย พาร์ที่กู้คืนสามารถใช้ได้ในทันที

fail_path

พาร์ที่อยู่ในค่าแรกของการหมดเวลาใช้งานคำสั่ง จะสมมติว่า ไม่ใช่พาร์สุดท้ายในกลุ่มพาร์ ถ้าพาร์ที่ล้มเหลวเนื่องจาก ปัญหาการถ่ายโอนได้รับการกู้คืน พาร์ไม่ถูกใช้สำหรับการดำเนินการ อ่านหรือเขียน I/O จนกว่าหมดช่วงเวลาที่ไม่มีควมล้มเหลวบนพาร์นั้น เมื่อคุณลักษณะนี้เปิดใช้งาน อาจเกิดการหน่วงเวลา ก่อนการอ่าน หรือเขียน I/O ถูกจัดเส้นทางไปยังพาร์ที่ถูกกู้คืนจากข้อผิดพลาดการถ่ายโอน

disable_path

พาร์ที่อยู่ในค่าแรกของการหมดเวลาใช้งานคำสั่ง จะสมมติว่า ไม่ใช่พาร์สุดท้ายในกลุ่มพาร์ ถ้าพาร์ที่ล้มเหลวเนื่องจาก ปัญหาการถ่ายโอนได้รับการกู้คืน พาร์ไม่ถูกใช้สำหรับการ อ่านหรือเขียน I/O จนกว่าหมดช่วงเวลาที่ไม่มีควมล้มเหลวบนพาร์นั้น ถ้าพาร์ นี้ยังคงประสบปัญหาหมดเวลาใช้งานหลายๆ คำสั่งระหว่างระยะเวลา อาจถูกปิดใช้งาน พาร์ที่เปิดใช้งานยังคงถูกปิดใช้งาน (และไม่สามารถใช้ได้) จนกว่าคุณจะทำแอ็ททริบิวต์ **chpath** เพื่อเปิดใช้งานพาร์ที่เปิดใช้งานอยู่ กำหนดค่าดีสก์ที่ได้รับผลกระทบอีกครั้ง หรือรีบูต ระบบ

หลักการที่เกี่ยวข้อง:

“การกำหนดคอนฟิกอุปกรณ์ MPIO” ในหน้า 595

การกำหนดคอนฟิกอุปกรณ์ที่มีความสามารถ MPIO ใช้คำสั่งเดียวกันกับ อุปกรณ์ที่ไม่ใช่ MPIO

แอ็ททริบิวต์เรพลิเคชัน SAN

AIX Multiple Path I/O (MPIO) ต้องถูกติดตั้ง และอุปกรณ์ต้องใช้ AIX path control module (PCM) แอ็ททริบิวต์ เหล่านี้มี การขึ้นต่อกันในการตั้งค่าและคุณลักษณะที่ถูกกำหนด โดยระบบย่อยหน่วยเก็บข้อมูล

แอ็ททริบิวต์ AIX ต่อไปนี้ เกี่ยวข้องกับลักษณะการทำงานของ logical unit numbers (LUNs) ซึ่ง ถูกเรพลิเคชันผ่านระบบย่อย หน่วยเก็บข้อมูล ระบบย่อยหน่วยเก็บข้อมูลทั้งหมด หรือระดับไมโครโค้ดอาจไม่สนับสนุนแอ็ททริบิวต์เหล่านี้ ซอฟต์แวร์ การคลัสเตอร์ หรือที่มีความพร้อมใช้งานสูง เช่น PowerHA® SystemMirror® ถูกติดตั้ง เพื่อประสานการจัดการของเรพลิเคชัน storage area network (SAN) ระหว่างโหนดในคลัสเตอร์ แอ็ททริบิวต์ต่อไปนี้ไม่สามารถเปลี่ยนแปลง บน Virtual I/O Server (VIOS)

san_rep_cfg

พิจารณาว่าอุปกรณ์ซึ่งโครนัสที่กำลังใช้ Peer-to-Peer Remote Copy (PPRC) ถูกกำหนด และกำหนดคอนฟิกใน ระบบปฏิบัติการ AIX อย่างไร ค่า **unique_id** ของดีสก์อินสแตนซ์ได้รับผลโดยแอ็ททริบิวต์นี้ และอาจเปลี่ยนแปลง ถ้าค่าแอ็ททริบิวต์ถูกเปลี่ยนแปลง แอ็ททริบิวต์ **san_rep_cfg** ไม่เปลี่ยนแปลงสภาวะของอุปกรณ์ PPRC บนระบบย่อย หน่วยเก็บข้อมูล แอ็ททริบิวต์มีค่าต่อไปนี้:

none [Default]

กำหนดค่า LUNs ในอุปกรณ์ซึ่งโครนัสที่ใช้ PPRC เป็น โลจิคัลดีสก์อินสแตนซ์ต่างหาก

new กำหนดและปรับแต่งค่าอุปกรณ์ซึ่งโครนัสที่ใช้ PPRC เป็นโลจิคัลอินสแตนซ์เดี่ยว อุปกรณ์ถูกกำหนด และปรับแต่งค่า ต่อเมื่อไม่มีดิสก์อินสแตนซ์ที่มีอยู่ตรงกับ LUNs ใดๆ ในอุปกรณ์ PPRC

new_and_existing

กำหนดและปรับแต่งค่าอุปกรณ์ซึ่งโครนัสที่ใช้ PPRC เป็นโลจิคัลอินสแตนซ์เดี่ยว ถ้าไม่มีโลจิคัลดิสก์อินสแตนซ์แสดง อุปกรณ์ PPRC ดิสก์อินสแตนซ์ใหม่จะถูกกำหนด

migrate_disk

กำหนดและปรับแต่งค่าอุปกรณ์ซึ่งโครนัสที่ใช้ PPRC เป็นอินสแตนซ์ hdisk เดี่ยว และใช้โลจิคัลดิสก์อินสแตนซ์ที่เลือก สำหรับอุปกรณ์ การดำเนินการได้รับการสนับสนุนบนอุปกรณ์ที่มีแอดทริบิวต์

san_rep_device ตั้งค่าเป็นสนับสนุน หรือตรวจพบเท่านั้น ถ้าอุปกรณ์ที่กำหนดเป็นปลายทางมีแอดทริบิวต์ **san_rep_device** ตั้งค่า เป็นสนับสนุน การดำเนินการทำงานต่อเมื่อเรพลิเคชัน SAN ถูก ตั้งค่าบนหน่วยเก็บข้อมูลตั้งแต่ครั้งล่าสุดที่ดิสก์ถูกปรับแต่งค่า การดำเนินการนี้ได้รับการสนับสนุนบนดิสก์ที่เปิด และใช้งาน โดยระบบปฏิบัติการ AIX ถ้าอุปกรณ์ไม่ถูกใช้เป็นดิสก์ที่เก็บภายในคลัสเตอร์ ค่า **unique_id** สำหรับดิสก์ที่ได้รับผลจะถูกอัปเดตเพื่อแสดง ID ของอุปกรณ์ PPRC

revert_disk

กำหนดค่าอุปกรณ์ซึ่งโครนัสที่มีอยู่ที่ใช้โลจิคัลดิสก์อินสแตนซ์ PPRC เป็นอินสแตนซ์ hdisk อุปกรณ์ที่ไม่ใช่ PPRC การดำเนินการนี้ได้รับการสนับสนุนบนอุปกรณ์ที่มีแอดทริบิวต์ **san_rep_device** ตั้งค่า เป็น yes เท่านั้น ชื่ออุปกรณ์โลจิคัลและไฟล์พิเศษของดิสก์อินสแตนซ์ ปลายทางยังคงไม่เปลี่ยนแปลง LUN หลัก (ต้นทาง) สำหรับอุปกรณ์ เรพลิเคชัน SAN ถูกใช้สำหรับอินสแตนซ์ hdisk ที่ย้อนกลับ ถ้า ไม่พบ LUN หลัก (ต้นทาง) หรือไม่รู้จักบนโฮสต์ LUN รอง (ปลายทาง) จะถูกใช้สำหรับอินสแตนซ์ hdisk ที่ย้อนกลับ การดำเนินการนี้ได้รับการสนับสนุนบนดิสก์ที่เปิด และใช้งานโดย ระบบปฏิบัติการ AIX ถ้าอุปกรณ์ไม่ถูกใช้เป็นดิสก์ที่เก็บภายในคลัสเตอร์ ค่า **unique_id** สำหรับดิสก์ที่ได้รับผลจะถูกอัปเดตเพื่อแสดง LUN ID ที่ได้รับผล

ค่า **none**, **new** และ **new_and_existing** ใช้สำหรับอัปเดตลักษณะการทำงานสำหรับอุปกรณ์ทั้งหมดของชนิดเฉพาะ Object Data Manager (ODM) เดียวกัน คำสั่ง **chdef** ใช้เพื่อตั้งค่า **none**, **new** และ **new_and_existing** คำสั่ง **chdef** อัปเดตค่าดีฟอลต์สำหรับ แอดทริบิวต์สำหรับอุปกรณ์ทั้งหมดของชนิดเฉพาะ ODM ที่ระบุ คำสั่ง **chdef** ยังอัปเดตค่าแอดทริบิวต์สำหรับ instance อุปกรณ์ที่มีอยู่ ที่ถูกกำหนดไว้แล้ว สำหรับอุปกรณ์ที่ถูกกำหนดคอนฟิกไว้แล้ว เมื่อคำสั่ง **chdef** กำลังรัน การเปลี่ยนแปลงยังไม่แสดงผลจนกว่าระบบจะรีบูต หรืออุปกรณ์เหล่านี้ถูกยกเลิกการกำหนดคอนฟิก และกำหนดคอนฟิกใหม่ คำสั่ง **chdef** จำเป็นต้องใช้ คลาส คลาสย่อย และชนิดของแอดทริบิวต์ เมื่อต้องการพิจารณาชนิดเฉพาะ ODM ของแอดทริบิวต์ **san_rep_cfg** ใช้คำสั่ง **lsattr -l hdisk# -F class,subclass,type** ตัวอย่างเช่น:

```
lsdev -l hdisk0 -F class,subclass,type
disk,fcpl,aixmpi0ds8k
chdef -a san_rep_cfg=none -c disk -s fcp -t aixmpi0ds8k
chdef -a san_rep_cfg=new -c disk -s fcp -t aixmpi0ds8k
chdef -a san_rep_cfg=new_and_existing -c disk -s fcp -t aixmpi0ds8k
```

ค่า **migrate_disk** and **revert_disk** จะอัปเดตลักษณะการทำงานสำหรับอินสแตนซ์อุปกรณ์เดี่ยว และที่ระบุ คำสั่ง **chdev** ต้องใช้เพื่อตั้งค่า **migrate_disk** หรือ **revert_disk** อย่างไม่อย่างหนึ่ง สำหรับอุปกรณ์ที่ระบุ คำสั่ง **chdev** อัปเดตค่าสำหรับอุปกรณ์ที่ระบุเท่านั้น ตัวอย่างเช่น:

```
chdev -a san_rep_cfg=migrate_disk -l hdisk0
chdev -a san_rep_cfg=revert_disk -l hdisk0
```

san_rep_device

ระบุว่าไลจิสต์ดิสก์อินสแตนซ์ถูกกำหนดเป็นอุปกรณ์เรพลิเคชัน SAN แอ็ททริบิวต์นี้ถูกตั้งค่าระหว่างการกำหนดคอนฟิกดิสก์ และอาจเป็นค่าเก่าถ้าสถานะของอุปกรณ์ถูกเปลี่ยนแปลงตั้งแต่ดิสก์ถูกกำหนดคอนฟิก แอ็ททริบิวต์มีค่าต่อไปนี้:

no อุปกรณ์ไม่ถูกกำหนดคอนฟิกในระบบปฏิบัติการ AIX เป็น อุปกรณ์เรพลิเคชัน SAN อุปกรณ์นี้ไม่ตรงตามข้อกำหนด ที่จะถูกกำหนดคอนฟิกเป็นอุปกรณ์เรพลิเคชัน SAN

supported

อุปกรณ์ไม่ถูกกำหนดคอนฟิกในระบบปฏิบัติการ AIX เป็น อุปกรณ์เรพลิเคชัน SAN อุปกรณ์นี้ตรงตามข้อกำหนด ที่จะถูกกำหนดคอนฟิกเป็นอุปกรณ์เรพลิเคชัน SAN อย่างไรก็ตาม ขณะนี้ อุปกรณ์ ถูกตั้งค่าเป็น อุปกรณ์เรพลิเคชัน SAN บนระบบย่อยหน่วยเก็บข้อมูล

detected

อุปกรณ์ไม่ถูกกำหนดคอนฟิกในระบบปฏิบัติการ AIX เป็น อุปกรณ์เรพลิเคชัน SAN ระบบปฏิบัติการ AIX ตรวจสอบว่าอุปกรณ์นี้ตรงตามข้อกำหนดที่จะ กำหนดคอนฟิกเป็นอุปกรณ์หน่วยเก็บข้อมูล SAN และขณะนี้ ถูกตั้งค่า เป็นอุปกรณ์เรพลิเคชัน SAN บนระบบย่อยหน่วยเก็บข้อมูล

yes อุปกรณ์ถูกกำหนดคอนฟิกในระบบปฏิบัติการ AIX เป็น อุปกรณ์เรพลิเคชัน SAN

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง chdef

คำสั่ง chdev

คำสั่ง lsdev

การเอาอะแดปเตอร์การสื่อสารออก

ก่อนคุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก คุณต้องยกเลิก กำหนดคอนฟิกอะแดปเตอร์นั้น

การยกเลิกกำหนดคอนฟิกอะแดปเตอร์การสื่อสารเกี่ยวข้องกับการกิจต่อไปนี้:

- การปิดแ็พพลิเคชันทั้งหมดที่กำลังใช้อะแดปเตอร์ซึ่งคุณกำลังจะเอาออกหรือ เปลี่ยน
- การตรวจสอบให้แน่ใจว่าอุปกรณ์ทั้งหมดที่เชื่อมต่อกับอะแดปเตอร์มีการระบุ และหยุด
- การแสดงรายการสล็อตทั้งหมดที่ใช้อยู่ในปัจจุบัน หรือสล็อตที่ใช้โดยอะแดปเตอร์เฉพาะ
- การระบุที่ตั้งสล็อตของอะแดปเตอร์
- การแสดงและการลบข้อมูลอินเตอร์เฟซออกจากรายการอินเตอร์เฟซ เครือข่าย
- การทำให้อะแดปเตอร์ไม่มีอยู่

เพื่อยกเลิกกำหนดคอนฟิกอะแดปเตอร์การสื่อสารด้วยโพรซีเจอร์ต่อไปนี้ คุณต้องล็อกอินเป็น ราก:

หลักการที่เกี่ยวข้อง:

“การจัดการฮอตปลั๊ก PCI” ในหน้า 586

คุณสามารถเสียบอะแดปเตอร์ฮอตปลั๊ก PCI ใหม่เข้าในสล็อต PCI ที่มีอยู่ในขณะที่ระบบปฏิบัติการกำลังรัน

งานที่เกี่ยวข้อง:

“การยกเลิกกำหนดคอนฟิกอะแดปเตอร์การสื่อสาร PCI” ในหน้า 589

ข้อมูลต่อไปนี้ เป็นภาพรวมของโปรเซสการยกเลิกกำหนดคอนฟิก อะแดปเตอร์การสื่อสาร PCI ซึ่งรวมถึงอะแดปเตอร์ อีเทอร์เน็ต, Token-ring, FDDI, และ ATM

“การเอาออกหรือการเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก PCI” ในหน้า 590

คุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก PCI จากยูนิตระกูลระบบ ได้โดยไม่ต้องปิดระบบปฏิบัติการหรือปิดระบบการเอาอะแดปเตอร์ออกทำให้รื้อฮาร์ดแวร์ที่นำเสนอโดยอะแดปเตอร์นั้น ไม่มีอยู่สำหรับระบบปฏิบัติการและแอปพลิเคชันต่างๆ

การยกเลิกกำหนดคอนฟิกอะแดปเตอร์อีเทอร์เน็ต, Token-ring, FDDI, และ ATM:

เมื่อต้องการยกเลิกกำหนดคอนฟิกอะแดปเตอร์อีเทอร์เน็ต, Token-ring, FDDI, หรือ ATM ให้ปฏิบัติตามขั้นตอนต่อไปนี้:

1. พิมพ์ `lsdev -c pci` เพื่อแสดงรายการ สล็อตฮอตปลั๊กทั้งหมดในยูนิตระกูลระบบและแสดงลักษณะของสล็อตนั้น
2. พิมพ์คำสั่ง SMIT ที่เหมาะสม ดังแสดงในตัวอย่างต่อไปนี้ เพื่อแสดงรายการอะแดปเตอร์ที่ติดตั้งและแสดง สถานะปัจจุบัน ของอุปกรณ์ทั้งหมดในยูนิตระกูลระบบ:

ไอเท็ม	คำอธิบาย
<code>smit lsdenet</code>	เพื่อแสดงรายการอะแดปเตอร์อีเทอร์เน็ต
<code>smit lsdtok</code>	เพื่อแสดงรายการอะแดปเตอร์ token-ring
<code>smit ls_atm</code>	เพื่อแสดงรายการอะแดปเตอร์ ATM

ระเบียบการตั้งชื่อต่อไปนี้ใช้สำหรับอะแดปเตอร์ชนิดต่างๆ:

ไอเท็ม	คำอธิบาย
ชื่อ	ชนิดอะแดปเตอร์
<code>atm0, atm1, ...</code>	อะแดปเตอร์ ATM
<code>ent0, ent1, ...</code>	อะแดปเตอร์อีเทอร์เน็ต
<code>tok0, tok1, ...</code>	อะแดปเตอร์ Token Ring

3. ปิดแอปพลิเคชันทั้งหมดที่กำลังใช้อะแดปเตอร์ซึ่งคุณกำลังจะยกเลิกกำหนดคอนฟิก เมื่อต้องการทำโปรซีเจอร์นี้ต่อไป ต้องปิดใช้งานที่ตั้งดัมพ์เครือข่าย บนระบบ เมื่อต้องการค้นหาและปิดใช้งานที่ตั้งดัมพ์เครือข่าย ให้ทำดังต่อไปนี้:

- a. พิมพ์ดังต่อไปนี้จากบรรทัดคำสั่ง:

```
smit dump
```

- b. เลือก แสดงอุปกรณ์ดัมพ์ปัจจุบัน

- c. ตรวจสอบว่าอุปกรณ์ดัมพ์ที่กำหนดคอนฟิกใดๆ แสดงที่ตั้งเครือข่ายหรือไม่ ถ้าไม่ให้ออกจาก SMIT และคุณพร้อมแล้วสำหรับขั้นตอนที่ 4 เมื่อต้องการเปลี่ยนอุปกรณ์ดัมพ์เป็นที่ตั้งโลคัล ให้เลือก **Cancel** หรือ กด F3 และทำขั้นตอนต่อไปนี้อย่างต่อเนื่อง

- d. หากอุปกรณ์ดัมพ์หลักแสดงที่ตั้งเครือข่าย ให้เปลี่ยน เป็นที่ตั้งโลคัลโดยเลือก **เปลี่ยนอุปกรณ์ดัมพ์หลัก** จากนั้น ป้อนที่ตั้งโลคัลในฟิลด์ อุปกรณ์ดัมพ์หลัก

- e. หากอุปกรณ์ดัมพ์รองแสดงที่ตั้งเครือข่าย ให้เปลี่ยน เป็นที่ตั้งโลคัลโดยเลือก **เปลี่ยนอุปกรณ์ดัมพ์รอง** จากนั้น ป้อนที่ตั้งโลคัลในฟิลด์ อุปกรณ์ดัมพ์รอง

- f. เมื่อทำเสร็จแล้ว คลิก ตกลง หรือกด Enter

4. พิมพ์ `netstat -i` เพื่อแสดงรายการของ อินเทอร์เน็ตที่กำหนดคอนฟิกทั้งหมดและพิจารณาว่าอะแดปเตอร์ของคุณมีการกำหนดคอนฟิกสำหรับ TCP/IP หรือไม่ เอาต์พุตที่คล้ายกับตัวอย่างต่อไปนี้แสดงขึ้น:

```
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
lo0 16896 link#1 076 0 118 0 0
lo0 16896 127 127.0.0.1 076 0 118 0 0
lo0 16896 ::1 076 0 118 0 0
```

tr0	1492	link#2	8.0.5a.b8.b.ec	151	0	405	11	0
tr0	1492	19.13.97	19.13.97.106	151	0	405	11	0
at0	9180	link#3	0.4.ac.ad.e0.ad	0	0	0	0	0
at0	9180	6.6.6	6.6.6.5	0	0	0	0	0
en0	1500	link#5	0.11.0.66.11.1	212	0	1	0	0
en0	1500	8.8.8	8.8.8.106	212	0	1	0	0

อะแด็ปเตอร์ Token-ring สามารถมีได้เพียงอินเตอร์เฟซเดียวเท่านั้น อะแด็ปเตอร์อีเทอร์เน็ตสามารถ มีสองอินเตอร์เฟซ
อะแด็ปเตอร์ ATM สามารถมีหลายอินเตอร์เฟซ

5. พิมพ์คำสั่ง **ifconfig** ที่เหมาะสม ดังแสดงในตัวอย่าง ต่อไปนี้ เพื่อลบอินเตอร์เฟซออกจากรายการอินเตอร์เฟซเครือข่าย

ไอเท็ม	คำอธิบาย
<code>ifconfig en0 detach</code>	เพื่อลบอินเตอร์เฟซอีเทอร์เน็ตมาตรฐาน
<code>ifconfig et0 detach</code>	เพื่อลบอินเตอร์เฟซอีเทอร์เน็ต IEEE 802.3
<code>ifconfig tr0 detach</code>	เพื่อลบอินเตอร์เฟซ token-ring
<code>ifconfig at0 detach</code>	เพื่อลบอินเตอร์เฟซ ATM

6. พิมพ์คำสั่ง **rmdev** ที่เหมาะสม ดังแสดงในตัวอย่าง ต่อไปนี้ เพื่อยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์และ เก็บ นิชยาม
อุปกรณ์ไว้ในคลาสอ็อบเจ็กต์อุปกรณ์แบบกำหนดเอง:

ไอเท็ม	คำอธิบาย
<code>rmdev -l ent0</code>	เพื่อยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์อีเทอร์เน็ต
<code>rmdev -l tok1</code>	เพื่อยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์ token-ring
<code>rmdev -l atml</code>	เพื่อยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์ ATM
<code>rmdev -p pci1</code>	เพื่อยกเลิกกำหนดคอนฟิกขายนซ์ของบัส PCI และอุปกรณ์อื่นทั้งหมด ภายใต้นั้นในขณะที่คณนิยามอุปกรณ์ไว้ใน คลาสอ็อบเจ็กต์ อุปกรณ์แบบกำหนดเอง

หมายเหตุ: เมื่อต้องการยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์และ ลบ นิชยามอุปกรณ์ ในคลาสอ็อบเจ็กต์อุปกรณ์แบบ
กำหนดเอง คุณสามารถใช้คำสั่ง **rmdev** พร้อมด้วยแฟล็ก **-d**

ข้อควรสนใจ: อย่าใช้แฟล็ก **-d** พร้อมกับคำสั่ง **rmdev** สำหรับการดำเนินงานฮอตปลั๊กยกเว้นว่า คุณตั้งใจจะ
เอาอะแด็ปเตอร์ออกและไม่เปลี่ยน

งานที่เกี่ยวข้อง:

“การยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์ ATM” ในหน้า 608

คุณต้องยกเลิกกำหนดคอนฟิกอุปกรณ์ที่เลียนแบบ LAN แต่ละรายการก่อน คุณสามารถเอาอะแด็ปเตอร์ออกได้

การยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์ WAN:

คุณสามารถยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์ WAN ได้

เมื่อต้องการยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์ WAN:

1. พิมพ์ `lsslot -c pci` เพื่อแสดงรายการ สล็อตฮอตปลั๊กทั้งหมดในยูนิตรระบบและแสดงลักษณะของสล็อตนั้น

2. พิมพ์คำสั่ง SMIT ที่เหมาะสม ดังแสดงในตัวอย่างต่อไปนี้ เพื่อแสดงรายการอะแดปเตอร์ที่ติดตั้งและแสดง สถานะ ปัจจุบัน ของอุปกรณ์ทั้งหมดในยูนิตรระบบ:

ไอเท็ม	คำอธิบาย
smit 331121b9_ls	เพื่อแสดงรายการอะแดปเตอร์ 2-พอร์ต Multiprotocol WAN
smit riciophx_ls	เพื่อแสดงรายการอะแดปเตอร์ ARTIC WAN

ระเบียบการตั้งชื่อต่อไปนี้ใช้สำหรับอะแดปเตอร์ ชนิดต่างๆ:

ชื่อ	ชนิดอะแดปเตอร์
dpmpa	อะแดปเตอร์ 2-พอร์ต Multiprotocol
riciop	อะแดปเตอร์ ARTIC960

3. พิมพ์ `lsdev -C -c port` เพื่อแสดงรายการพอร์ต X.25 บนโฮสต์ของคุณ ข้อความที่คล้ายกับตัวอย่างต่อไปนี้แสดงขึ้น:

```

sx25a0 Available 00-05-01-00 X.25 Port
x25s0 Available 00-05-01-00-00 V.3 X.25 Emulator

```

4. ปิดแอฟพลิเคชันทั้งหมดที่กำลังใช้อะแดปเตอร์ซึ่งคุณกำลังจะยกเลิกกำหนดคอนฟิก เมื่อต้องการทำโปรซีเดอร์นี้ต่อไป ต้องปิดใช้งานที่ตั้งดัมพ์เครือข่าย บนระบบ เมื่อต้องการค้นหาและปิดใช้งานที่ตั้งดัมพ์เครือข่าย ให้ทำดังต่อไปนี้:

- a. พิมพ์ดังต่อไปนี้จากบรรทัดคำสั่ง:

```
smit dump
```

- b. เลือก แสดงอุปกรณ์ดัมพ์ปัจจุบัน

- c. ตรวจสอบว่าอุปกรณ์ดัมพ์ที่กำหนดคอนฟิกใดๆ แสดงที่ตั้งเครือข่ายหรือไม่ ถ้าไม่ให้ออกจาก SMIT และคุณพร้อม แล้วสำหรับขั้นตอนที่ 5 ข้างล่าง เมื่อต้องการเปลี่ยนอุปกรณ์ดัมพ์เป็นที่ตั้งโลคัล ให้เลือก Cancel หรือ กด F3 และทำ ขั้นตอนต่อไปนี้อย่างต่อไป

- d. หากอุปกรณ์ดัมพ์หลักแสดงที่ตั้งเครือข่าย ให้เปลี่ยน เป็นที่ตั้งโลคัลโดยเลือก เปลี่ยนอุปกรณ์ดัมพ์หลัก จากนั้น ป้อนที่ตั้งโลคัลในฟิลด์ อุปกรณ์ดัมพ์หลัก

- e. หากอุปกรณ์ดัมพ์รองแสดงที่ตั้งเครือข่าย ให้เปลี่ยน เป็นที่ตั้งโลคัลโดยเลือก เปลี่ยนอุปกรณ์ดัมพ์รอง จากนั้น ป้อนที่ตั้งโลคัลในฟิลด์ อุปกรณ์ดัมพ์รอง

- f. เมื่อทำเสร็จแล้ว คลิก ตกลง หรือกด Enter

5. ใช้คำสั่งในตารางต่อไปนี้เพื่อยกเลิกกำหนดคอนฟิกและลบ ไดรเวอร์อุปกรณ์และพอร์ตตัวเลียนแบบสำหรับอะแดปเตอร์ เหล่านี้:

ชื่อ	อะแดปเตอร์หลายโปรโตคอล 2 พอร์ต
smit rmhdlcdmpdd	เพื่อยกเลิกกำหนดคอนฟิกอุปกรณ์
smit rmsdlcscied	เพื่อยกเลิกกำหนดคอนฟิกตัวเลียนแบบ SDLC COMIO

ชื่อ	อะแดปเตอร์ ARTIC960Hx PCI
smit rmtssdd	เพื่อยกเลิกกำหนดคอนฟิกไดรเวอร์อุปกรณ์
smit rmtsdports	เพื่อลบอิมูเลชันพอร์ต MPQP COMIO

การยกเลิกกำหนดคอนฟิกอะแดปเตอร์ IBM 4-พอร์ต 10/100 Base-TX Ethernet PCI:

อะแดปเตอร์ 4-พอร์ต 10/100 Base-TX Ethernet PCI มีพอร์ต อีเทอร์เน็ตสี่ตัว และต้องยกเลิกกำหนดคอนฟิกแต่ละตัวก่อนคุณสามารถเอา อะแดปเตอร์ออกได้

- พิมพ์ `lsslot -c pci` เพื่อแสดงรายการ สล็อตฮอตปลั๊กทั้งหมดในยูนิระบบและแสดงลักษณะของสล็อตนั้น
- พิมพ์ `smit lsdenet` เพื่อแสดงรายการอุปกรณ์ ทั้งหมดในคลาสย่อย PCI ข้อความที่คล้ายกับ ตัวอย่างต่อไปนี้แสดงขึ้น:


```
ent1 Available 1N-00 IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020) (Port 1)
ent2 Available 1N-08 IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020) (Port 2)
ent3 Available 1N-10 IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020) (Port 3)
ent4 Available 1N-18 IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020) (Port 4)
```
- ปิดแอฟพลิเคชันทั้งหมดที่กำลังใช้อะแดปเตอร์ซึ่งคุณกำลังจะ ยกเลิกกำหนดคอนฟิก เมื่อต้องการทำโปรซีเดอร์นี้ต่อไป ต้องปิดใช้งานที่ตั้งดัมพ์เครือข่าย บนระบบ เมื่อต้องการค้นหาและปิดใช้งาน ที่ตั้งดัมพ์เครือข่าย ให้ทำดังต่อไปนี้:
 - พิมพ์ดังต่อไปนี้จากบรรทัดคำสั่ง:


```
smit dump
```
 - เลือก แสดงอุปกรณ์ดัมพ์ปัจจุบัน
 - ตรวจสอบว่าอุปกรณ์ดัมพ์ที่กำหนดคอนฟิกใดๆ แสดงที่ตั้งเครือข่ายหรือไม่ ถ้าไม่ให้ออกจาก SMIT และคุณพร้อมแล้วสำหรับขั้นตอนที่ 4 เมื่อต้องการเปลี่ยนอุปกรณ์ดัมพ์เป็นที่ตั้งโลคัล ให้เลือก **Cancel** หรือ กด F3 และทำขั้นตอนต่อไปนี้อย่างต่อเนื่อง
 - หากอุปกรณ์ดัมพ์หลักแสดงที่ตั้งเครือข่าย ให้เปลี่ยน เป็นที่ตั้งโลคัลโดยเลือก **เปลี่ยนอุปกรณ์ดัมพ์หลัก** จากนั้น ป้อนที่ตั้งโลคัลในฟิลด์ อุปกรณ์ดัมพ์หลัก
 - หากอุปกรณ์ดัมพ์รองแสดงที่ตั้งเครือข่าย ให้เปลี่ยน เป็นที่ตั้งโลคัลโดยเลือก **เปลี่ยนอุปกรณ์ดัมพ์รอง** จากนั้น ป้อนที่ตั้งโลคัลในฟิลด์ อุปกรณ์ดัมพ์รอง
 - เมื่อทำเสร็จแล้ว คลิก ตกลง หรือกด Enter
- พิมพ์ `netstat -i` เพื่อแสดงรายการของ อินเทอร์เน็ตที่กำหนดคอนฟิกทั้งหมดและพิจารณาว่าอะแดปเตอร์ของคุณมีการ กำหนดคอนฟิกสำหรับ TCP/IP หรือไม่เอาต์พุตที่คล้ายกับตัวอย่างต่อไปนี้แสดงขึ้น:

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
lo0	16896	link#1		076	0	118	0	0
lo0	16896	127	127.0.0.1	076	0	118	0	0
lo0	16896	:::1		076	0	118	0	0
tr0	1492	link#2	8.0.5a.b8.b.ec	151	0	405	11	0
tr0	1492	19.13.97	19.13.97.106	151	0	405	11	0
at0	9180	link#3	0.4.ac.ad.e0.ad	0	0	0	0	0
at0	9180	6.6.6	6.6.6.5	0	0	0	0	0
en0	1500	link#5	0.11.0.66.11.1	212	0	1	0	0
en0	1500	8.8.8	8.8.8.106	212	0	1	0	0

อะแดปเตอร์อีเทอร์เน็ตสามารถมีสองอินเทอร์เน็ตเฟซ ตัวอย่างเช่น et0 และ en0

- ใช้คำสั่ง `ifconfig` เพื่อลบแต่ละอินเทอร์เฟซออกจากรายการอินเทอร์เฟซเครือข่าย ตัวอย่างเช่น พิมพ์ `ifconfig en0 detach` เพื่อลบอินเทอร์เฟซอีเทอร์เน็ตมาตรฐาน และพิมพ์ `ifconfig et0` เพื่อลบอินเทอร์เฟซ IEEE 802.3
- ใช้คำสั่ง `rmdev` เพื่อยกเลิกกำหนดคอนฟิกอะแดปเตอร์ ในขณะที่คณินยามอุปกรณ์ไว้ในคลาสอ็อบเจ็กต์อุปกรณ์แบบกำหนดเอง ตัวอย่างเช่น `rmdev -l ent0`

หมายเหตุ: เมื่อต้องการยกเลิกกำหนดคอนฟิกอะแดปเตอร์และลบ นินยามอุปกรณ์ ในคลาสอ็อบเจ็กต์อุปกรณ์แบบกำหนดเอง คุณสามารถใช้คำสั่ง `rmdev` พร้อมด้วยแฟล็ก `-d`

ข้อควรสนใจ: อย่าใช้แฟล็ก `-d` พร้อมกับคำสั่ง `rmdev` สำหรับการดำเนินงานฮอตปลั๊กยกเว้นว่า คุณตั้งใจจะเอาอะแดปเตอร์ออกและไม่เปลี่ยน

การยกเลิกกำหนดคอนฟิกอะแดปเตอร์ ATM:

คุณต้องยกเลิกกำหนดคอนฟิกอุปกรณ์ที่เลียนแบบ LAN แต่ละรายการก่อน คุณสามารถเอาอะแดปเตอร์ออกได้

โปรโตคอลการเลียนแบบ Classic IP และ LAN สามารถรับบนอะแดปเตอร์ ATM โปรโตคอลการเลียนแบบ LAN ช่วยให้สามารถนำ LANs ที่เลียนแบบไปใช้บน เครือข่าย ATM ได้ LANs ที่เลียนแบบสามารถเป็นอีเทอร์เน็ต/IEEE 802.3, Token-ring/IEEE 802.5, และ MPOA (MultiProtocol Over ATM)

เมื่อต้องการลบอินเทอร์เฟซ LAN ให้ทำดังต่อไปนี้:

- พิมพ์ `lsslot -c pci` เพื่อแสดงรายการ สล็อตฮอตปลั๊กทั้งหมดในยูนิตรระบบและแสดงลักษณะของสล็อตนั้น
- พิมพ์ `smittl atm` เพื่อแสดงรายการอะแดปเตอร์ ATM ทั้งหมด ข้อความที่คล้ายกับตัวอย่างต่อไปนี้แสดงขึ้น:


```
.
.
atm0 Available 04-04 IBM PCI 155 Mbps ATM Adapter (14107c00)
atm1 Available 04-06 IBM PCI 155 Mbps ATM Adapter (14104e00)
```
- พิมพ์ `smittl install_atml` เพื่อแสดงรายการไคลเอ็นต์ ที่เลียนแบบ LAN ทั้งหมดบนอะแดปเตอร์ ข้อความที่คล้ายกับ ตัวอย่างต่อไปนี้แสดงขึ้น:


```
ent1 Available ATM LAN Emulation Client (Ethernet)
ent2 Available ATM LAN Emulation Client (Ethernet)
ent3 Available ATM LAN Emulation Client (Ethernet)
tok1 Available ATM LAN Emulation Client (Token Ring)
tok2 Available ATM LAN Emulation Client (Token Ring)
```

อะแดปเตอร์ ATM ทั้งหมด สามารถมีไคลเอ็นต์ที่เลียนแบบและกำลังรันอยู่บนอะแดปเตอร์ได้หลายรายการ

- พิมพ์ `smittl install_mpoa` เพื่อแสดงรายการไคลเอ็นต์ ที่เลียนแบบ LAN ทั้งหมดบนอะแดปเตอร์ ข้อความที่คล้ายกับ ตัวอย่างต่อไปนี้ แสดงขึ้น:


```
mpc0 Available ATM LAN Emulation MPOA Client
```

`atm0` และ `atm1` คือ อะแดปเตอร์ ATM ฟิสิคัล `mpc0` คือไคลเอ็นต์ที่เลียนแบบ MPOA `ent1`, `ent2`, `ent3`, `tok1`, และ `tok2` คือไคลเอ็นต์ที่เลียนแบบ LAN

- พิมพ์ `entstat` เพื่อพิจารณาอะแดปเตอร์ ซึ่งไคลเอ็นต์กำลังรัน ข้อความที่คล้ายกับ ตัวอย่างต่อไปนี้แสดงขึ้น:

```
-----
ETHERNET STATISTICS (ent1) :
Device Type: ATM LAN EmulationATM Hardware Address: 00:04:ac:ad:e0:ad
.
```

.
.
ATM LAN Emulation Specific Statistics:

Emulated LAN Name: ETHElan3
Local ATM Device Name: atm0
Local LAN MAC Address:
.
.

6. ปิดแอ็พพลิเคชันทั้งหมดที่กำลังใช้อะแดปเตอร์ซึ่งคุณกำลังจะยกเลิกกำหนดคอนฟิก เมื่อต้องการทำโปรซีเดอร์นี้ต่อไป ต้องปิดใช้งานที่ตั้งดัมพ์เครือข่าย บนระบบ เมื่อต้องการค้นหาและปิดใช้งานที่ตั้งดัมพ์เครือข่าย ให้ทำดังต่อไปนี้:

- a. พิมพ์ดังต่อไปนี้จากบรรทัดคำสั่ง:

```
smit dump
```

- b. เลือก แสดงอุปกรณ์ดัมพ์ปัจจุบัน

- c. ตรวจสอบว่าอุปกรณ์ดัมพ์ที่กำหนดคอนฟิกใดๆ แสดงที่ตั้งเครือข่ายหรือไม่ ถ้าไม่ให้ออกจาก SMIT และคุณพร้อมแล้วสำหรับขั้นตอนที่ 7 เมื่อต้องการเปลี่ยนอุปกรณ์ดัมพ์เป็นที่ตั้งโลคัล ให้เลือก Cancel หรือ กด F3 และทำขั้นตอนต่อไปนี้อย่างต่อเนื่อง

- d. หากอุปกรณ์ดัมพ์หลักแสดงที่ตั้งเครือข่ายให้เปลี่ยน เป็นที่ตั้งโลคัลโดยเลือก เปลี่ยนอุปกรณ์ดัมพ์หลัก จากนั้น ป้อนที่ตั้งโลคัลในฟิลด์ อุปกรณ์ดัมพ์หลัก

- e. หากอุปกรณ์ดัมพ์รองแสดงที่ตั้งเครือข่ายให้เปลี่ยน เป็นที่ตั้งโลคัลโดยเลือก เปลี่ยนอุปกรณ์ดัมพ์รอง จากนั้น ป้อนที่ตั้งโลคัลในฟิลด์ อุปกรณ์ดัมพ์รอง

- f. เมื่อทำเสร็จแล้ว คลิก ตกลง หรือกด Enter

7. ใช้คำสั่ง `rmdev -l device` เพื่อยกเลิกกำหนดคอนฟิกอินเตอร์เฟซในลำดับต่อไปนี้:

- อินเตอร์เฟซที่เลียนแบบ = en1, et1, en2, et2, tr1, tr2 ...
- อินเตอร์เฟซที่เลียนแบบ = ent1, ent2, tok1, tok2 ...
- Multiprotocol Over ATM (MPOA) = mpc0
- อะแดปเตอร์ ATM = atm0

8. เมื่อต้องการยกเลิกกำหนดคอนฟิกอะแดปเตอร์ SCSI `scsil` และชายนทั้งหมดในขณะที่ยังคงนิยามอุปกรณ์ไว้ในคลาสอ็อบเจกต์ อุปกรณ์แบบกำหนดเอง ให้พิมพ์:

```
rmdev -R scsil
```

ระบบแสดงข้อความคล้ายกับต่อไปนี้:

```
rmt0 Defined  
hdisk1 Defined  
scsil Defined
```

9. เมื่อต้องการยกเลิกกำหนดคอนฟิกเฉพาะชายนของอะแดปเตอร์ SCSI `scsil` ไม่ใช่ตัวอะแดปเตอร์เอง ในขณะที่ยังคงนิยามอุปกรณ์ไว้ในคลาสอ็อบเจกต์ อุปกรณ์แบบกำหนดเอง ให้พิมพ์:

```
rmdev -p scsil
```

ระบบแสดงข้อความคล้ายกับต่อไปนี้:

```
rmt0 Defined  
hdisk1 Defined
```

10. เมื่อต้องการยกเลิกกำหนดคอนฟิกของ PCI บัส pci1 และอุปกรณ์อื่นทั้งหมดภายใต้มันในขณะที่ยังมีอุปกรณ์ไว้ในคลาสอ็อบเจ็กต์ อุปกรณ์แบบกำหนดเองให้พิมพ์:

```
rmdev -p pci1
```

ระบบแสดงข้อความคล้ายกับต่อไปนี้:

```
rmt0 Defined
hdisk1 Defined
scsil Defined
ent0 Defined
```

งานที่เกี่ยวข้อง:

“การยกเลิกกำหนดคอนฟิกอะแดปเตอร์อีเทอร์เน็ต, Token-ring, FDDI, และ ATM” ในหน้า 604
เมื่อต้องการยกเลิกกำหนดคอนฟิกอะแดปเตอร์อีเทอร์เน็ต, Token-ring, FDDI, หรือ ATM ให้ปฏิบัติตามขั้นตอนต่อไปนี้:

การยกเลิกกำหนดคอนฟิกอะแดปเตอร์หน่วยเก็บ

ก่อนคุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์หน่วยเก็บ คุณต้องยกเลิกกำหนดคอนฟิกอะแดปเตอร์นั้น

เพื่อทำการกิจเหล่านี้ คุณต้องล็อกอินเป็นผู้ใช้ราก

ขั้นตอนต่อไปนี้จะยกเลิกการกำหนดคอนฟิกอะแดปเตอร์หน่วยเก็บข้อมูล SCSI และไฟเบอร์แซนแนล

การยกเลิกกำหนดคอนฟิกอะแดปเตอร์ หน่วยเก็บเกี่ยวข้องกับภารกิจต่อไปนี้:

- การปิดแอฟพลิเคชันทั้งหมดที่กำลังใช้อะแดปเตอร์ซึ่งคุณกำลังจะเอาออก เปลี่ยน หรือย้าย
- การถอนติดตั้งระบบไฟล์
- การตรวจสอบให้แน่ใจว่าอุปกรณ์ทั้งหมดที่เชื่อมต่อกับอะแดปเตอร์มีการระบุ และหยุด
- การแสดงรายการสล็อตทั้งหมดที่ใช้อยู่ในปัจจุบัน หรือสล็อตที่ใช้ โดยอะแดปเตอร์เฉพาะ
- การระบุที่ตั้งสล็อตของอะแดปเตอร์
- การทำให้อุปกรณ์พารেন্টและชายนี่ไม่มีอยู่
- การทำให้อะแดปเตอร์ไม่มีอยู่

ยกเลิกการกำหนดคอนฟิกอะแดปเตอร์ SCSI และไฟเบอร์แซนแนล

โดยทั่วไป อะแดปเตอร์หน่วยเก็บเป็นอุปกรณ์พารেন্টของอุปกรณ์สื่อบันทึกต่างๆ เช่น ดิสก์หรือเทปไดรฟ์ การลบพารেন্টกำหนดว่าอุปกรณ์ชายนี่ต่อพ่วงทั้งหมดต้อง ถูกเอาออกหรือวางไว้ในสถานะที่กำหนด

เมื่อต้องการยกเลิกการกำหนดคอนฟิกอะแดปเตอร์ SCSI และไฟเบอร์แซนแนล ให้ทำตามขั้นตอนต่อไปนี้:

1. ปิดแอฟพลิเคชันทั้งหมดที่กำลังใช้อะแดปเตอร์ซึ่งคุณกำลังจะยกเลิกกำหนดคอนฟิก
2. พิมพ์ `lsslot -c pci` เพื่อแสดงรายการสล็อต ฮอตปลั๊กทั้งหมดในยูนิตรระบบและแสดงลักษณะของสล็อตนั้น
3. พิมพ์ `lsdev -C` เพื่อแสดงรายการ สถานะ ปัจจุบัน ของอุปกรณ์ทั้งหมดในยูนิตรระบบ
4. พิมพ์ `umount` เพื่อถอนติดตั้งระบบไฟล์ที่ติดตั้งไว้ ก่อนหน้านั้น ไดรฟ์หรือโฟลว์ซึ่งใช้อะแดปเตอร์นี้ สำหรับข้อมูลเพิ่มเติม ให้ดูที่ การติดตั้ง JFS หรือ JFS2
5. พิมพ์ `rmdev -l adapter -R` เพื่อทำให้อะแดปเตอร์ไม่มีอยู่

ข้อควรสนใจ: อย่าใช้แฟล็ก `-d` พร้อมกับคำสั่ง `rmdev` สำหรับการดำเนินงานฮอตปลั๊กเนื่องจาก จะทำให้คอนฟิกูเรชันของคุณถูกลบออก

หลักการที่เกี่ยวข้อง:

“การจัดการฮอตปลั๊ก PCI” ในหน้า 586

คุณสามารถเสียบอะแดปเตอร์ฮอตปลั๊ก PCI ใหม่เข้าไปในสล็อต PCI ที่มีอยู่ในขณะที่ระบบปฏิบัติการกำลังทำงานที่เกี่ยวข้อง:

“การเอาออกหรือการเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก PCI” ในหน้า 590

คุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก PCI จากยูนิตรระบบ ได้โดยไม่ต้องปิดระบบปฏิบัติการหรือปิดระบบการเอาอะแดปเตอร์ออกทำให้รีฮอส์ที่นำเสนอโดยอะแดปเตอร์นั้น ไม่มีอยู่สำหรับระบบปฏิบัติการและแอปพลิเคชันต่างๆ

การยกเลิกกำหนดคอนฟิกอะแดปเตอร์อะซิงค์

คุณสามารถยกเลิกกำหนดคอนฟิกอะแดปเตอร์อะซิงค์ได้

เพื่อทำภารกิจเหล่านี้ คุณต้องล็อกอินเป็นผู้ใช้ราก

ข้อมูลต่อไปนี้เป็นขั้นตอนสำหรับการยกเลิกกำหนดคอนฟิกอะแดปเตอร์อะซิงค์

ก่อนคุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์อะซิงค์ คุณต้องยกเลิก กำหนดคอนฟิกอะแดปเตอร์นั้น การยกเลิกกำหนดคอนฟิกอะแดปเตอร์อะซิงค์เกี่ยวข้องกับภารกิจต่อไปนี้:

- การปิดแอปพลิเคชันทั้งหมดที่กำลังใช้อะแดปเตอร์ซึ่งคุณกำลังจะเอาออก เปลี่ยน หรือย้าย
- การตรวจสอบให้แน่ใจว่าอุปกรณ์ทั้งหมดที่เชื่อมต่อกับอะแดปเตอร์มีการระบุ และหยุด
- การแสดงรายการสล็อตทั้งหมดที่ใช้อยู่ในปัจจุบัน หรือสล็อตที่ใช้โดยอะแดปเตอร์เฉพาะ
- การระบุที่ตั้งสล็อตของอะแดปเตอร์
- การทำให้อุปกรณ์พาวเวอร์และชายนี้อยู่
- การทำให้อะแดปเตอร์ไม่มีอยู่

โพรซีเจอร์

ก่อนคุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์อะซิงค์ คุณต้องยกเลิก กำหนดคอนฟิกอะแดปเตอร์นั้นและอุปกรณ์ทั้งหมดที่ควบคุมโดยอะแดปเตอร์นั้น เพื่อยกเลิกกำหนดคอนฟิกอุปกรณ์ คุณต้องยุติโปรเซสทั้งหมด ที่ใช้อุปกรณ์นั้นอยู่ ใช้ขั้นตอนต่อไปนี้:

1. ปิดแอปพลิเคชันทั้งหมดที่กำลังใช้อะแดปเตอร์ซึ่งคุณกำลังจะยกเลิกกำหนดคอนฟิก
2. พิมพ์ `lsslot -c pci` เพื่อแสดงรายการสล็อต ฮอตปลั๊กทั้งหมดในยูนิตรระบบและแสดงลักษณะของสล็อตนั้น
3. พิมพ์ `lsdev -C -c tty` เพื่อแสดงรายการอุปกรณ์ `tty` ที่มีอยู่ทั้งหมดและสถานะปัจจุบันของอุปกรณ์ทั้งหมดในยูนิตรระบบ
4. พิมพ์ `lsdev -C -c printer` เพื่อแสดงรายการ อุปกรณ์เครื่องพิมพ์และพล็อตเตอร์ทั้งหมดซึ่งเชื่อมต่อกับอะแดปเตอร์
5. ใช้คำสั่ง `rmdev` เพื่อทำให้อะแดปเตอร์ไม่มีอยู่

ข้อควรสนใจ: อย่าใช้แฟล็ก `-d` พร้อมกับคำสั่ง `rmdev` สำหรับการดำเนินงานฮอตปลั๊กเนื่องจาก จะทำให้คอนฟิกูเรชันของคุณถูกลบออก

หลักการที่เกี่ยวข้อง:

“การจัดการฮอตปลั๊ก PCI” ในหน้า 586

คุณสามารถเสียบอะแดปเตอร์ฮอตปลั๊ก PCI ใหม่เข้าไปในสล็อต PCI ที่มีอยู่ในขณะที่ระบบปฏิบัติการกำลังรัน

“สถานะอุปกรณ์” ในหน้า 580

อุปกรณ์ที่เชื่อมต่อกับระบบสามารถอยู่ในสถานะอย่างใดอย่างหนึ่งจาก สี่สถานะ

งานที่เกี่ยวข้อง:

“การเอาออกหรือการเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก PCI” ในหน้า 590

คุณสามารถเอาออกหรือเปลี่ยนอะแดปเตอร์ฮอตปลั๊ก PCI จากยูนิตรระบบ ได้โดยไม่ต้องปิดระบบปฏิบัติการหรือปิดระบบการเอาอะแดปเตอร์ออกทำให้รีฮอตที่นำเสนอโดยอะแดปเตอร์นั้น ไม่มีอยู่สำหรับระบบปฏิบัติการและแอปพลิเคชันต่างๆ

ข้อมูลที่เกี่ยวข้อง:

การจัดการพิมพ์

การแก้ไขปัญหาอุปกรณ์ I/O

คุณสามารถกำหนดสาเหตุของปัญหาอุปกรณ์ได้

การตรวจสอบซอฟต์แวร์อุปกรณ์

แก้ไขปัญหาซอฟต์แวร์อุปกรณ์ โดย:

- การตรวจสอบบันทึกข้อผิดพลาด
- การแสดงรายการอุปกรณ์ทั้งหมด
- การตรวจสอบสถานะของอุปกรณ์
- การตรวจสอบแอตทริบิวต์ของอุปกรณ์
- การเปลี่ยนแอตทริบิวต์ของอุปกรณ์
- การใช้อุปกรณ์กับแอปพลิเคชันอื่น
- การกำหนดอุปกรณ์ใหม่

การตรวจสอบบันทึกข้อผิดพลาด

ตรวจสอบบันทึกข้อผิดพลาดเพื่อดูว่ามีกรบันทึกข้อผิดพลาดใดๆ สำหรับอุปกรณ์ อะแดปเตอร์ของอุปกรณ์ หรือแอปพลิเคชันที่ใช้อุปกรณ์หรือไม่ ไปยัง ความสะดวกในการบันทึกข้อผิดพลาด เพื่อดูข้อมูลเกี่ยวกับการทำ การตรวจสอบนี้ กลับไปยังขั้นตอนหลังจากทำโพรซีเจอร์เสร็จสมบูรณ์แล้ว

คุณ แก้ไขปัญหาเกี่ยวกับอุปกรณ์แล้วหรือไม่?

หากคุณไม่สามารถแก้ไข ปัญหาโดยใช้เมธอดก่อนหน้านี้ ให้ไปยังขั้นตอนถัดไป (การแสดงรายการ อุปกรณ์) เพื่อแสดงรายการอุปกรณ์ทั้งหมด

การแสดงรายการอุปกรณ์

ใช้คำสั่ง `lsdev -C` เพื่อแสดงรายการอุปกรณ์ที่กำหนดไว้หรือที่มีอยู่ทั้งหมด คำสั่งนี้ แสดงลักษณะของอุปกรณ์ทั้งหมดในระบบของคุณ

หาก อุปกรณ์อยู่ในรายการของอุปกรณ์ ให้ไปยังขั้นตอนถัดไป (การตรวจสอบสถานะของอุปกรณ์) เพื่อตรวจสอบสถานะของอุปกรณ์

หากอุปกรณ์ไม่อยู่ใน รายการของอุปกรณ์ ให้กำหนดอุปกรณ์ใหม่ (ให้ดูที่ นิยามอุปกรณ์ใหม่ ด้านล่าง)

การตรวจสอบสถานะของอุปกรณ์

ค้นหาอุปกรณ์ในรายการที่สร้างขึ้นจากคำสั่ง `Isdev -C` ตรวจสอบว่าอุปกรณ์ อยู่ในสถานะ Available หรือไม่

หากอุปกรณ์อยู่ใน สถานะ Available ให้ไปยังขั้นตอนถัดไป (การตรวจสอบแอตทริบิวต์ อุปกรณ์) เพื่อตรวจสอบแอตทริบิวต์อุปกรณ์

หากอุปกรณ์ไม่อยู่ในสถานะ Available ให้กำหนดอุปกรณ์ใหม่ (ให้ดูที่ นิยามอุปกรณ์ใหม่ ด้านล่าง)

การตรวจสอบแอตทริบิวต์อุปกรณ์

ใช้คำสั่ง `Isattr -E -I DeviceName` เพื่อแสดงรายการ แอตทริบิวต์ของอุปกรณ์

คำสั่ง `Isattr` แสดง ลักษณะแอตทริบิวต์และค่าที่เป็นไปได้ของแอตทริบิวต์สำหรับอุปกรณ์ในระบบ โปรดอ้างอิงเอกสารคู่มือของอุปกรณ์เฉพาะเพื่อดู ค่าติดตั้งที่ถูกต้อง

หากแอตทริบิวต์อุปกรณ์มีการตั้งค่าอย่างถูกต้อง ให้ดูที่ การใช้อุปกรณ์ กับแอพพลิเคชั่นอื่น ด้านล่าง

หากแอตทริบิวต์อุปกรณ์ไม่มี การตั้งค่าอย่างถูกต้อง ให้ไปยังขั้นตอนถัดไป การเปลี่ยนแปลงแอตทริบิวต์อุปกรณ์

การเปลี่ยนแปลงแอตทริบิวต์อุปกรณ์

ใช้คำสั่ง `chdev -I Name -a Attribute=Value` เพื่อเปลี่ยนแอตทริบิวต์อุปกรณ์ ก่อนคุณรันคำสั่งนี้ โปรดอ้างอิง *Commands Reference, Volume 1*

คำสั่ง `chdev` เปลี่ยนลักษณะของอุปกรณ์ซึ่งคุณระบุด้วยแฟล็ก `-I Name`

ถ้า การเปลี่ยนแอตทริบิวต์ไม่ได้แก้ไขปัญหเกี่ยวกับอุปกรณ์ ให้ไปยัง ขั้นตอนถัดไป การใช้อุปกรณ์กับแอพพลิเคชั่นอื่น

การใช้อุปกรณ์กับแอพพลิเคชั่นอื่น

ลอง ใช้อุปกรณ์กับแอพพลิเคชั่นอื่น หากอุปกรณ์ทำงานอย่างถูกต้องกับ แอพพลิเคชั่นอื่น แสดงว่าอาจมีปัญหเกี่ยวกับแอพพลิเคชั่นแรก

หาก อุปกรณ์ทำงานอย่างถูกต้องกับแอพพลิเคชั่นอื่น แสดงว่าคุณอาจมีปัญห เกี่ยวกับแอพพลิเคชั่นแรก รายงานปัญหาให้ผู้แทนเซอร์วิสซัพพอร์ตแวร์ของคุณรับทราบ

หาก อุปกรณ์ไม่ได้ทำงานอย่างถูกต้องกับแอพพลิเคชั่นอื่น ให้ไปยังขั้นตอน ถัดไป นิยามอุปกรณ์ใหม่

นียมอุปกรณ์ใหม่

หมายเหตุ: คุณต้องมีสิทธิ์ผู้ใช้รากหรือเป็นสมาชิกของกลุ่มความปลอดภัยเพื่อใช้ คำสั่ง `mkdev`

ใช้คำสั่ง `mkdev` เพื่อเพิ่มอุปกรณ์ลงในระบบ

คำสั่ง `mkdev` สามารถกำหนดและทำให้อุปกรณ์ใหม่พร้อมใช้งาน หรือทำให้อุปกรณ์ที่กำหนดไว้แล้วพร้อมใช้งาน คุณสามารถระบุอุปกรณ์ที่กำหนดไว้ล่วงหน้าโดยเฉพาะได้โดยใช้ชุดของ แฟล็ก `-c`, `-s`, และ `-t` ก่อนคุณรันคำสั่งนี้ โปรดอ้างอิง *Commands Reference, Volume 3*

หาก การกำหนดอุปกรณ์ไม่ได้แก้ไขปัญหา คุณสามารถหยุดและรายงาน ปัญหาให้ผู้แทนเซอวิสของคุณรับทราบ หรือใช้โปรแกรมการวิเคราะห์เพื่อ ทดสอบอุปกรณ์ของคุณ

การตรวจสอบการเชื่อมต่ออุปกรณ์:

เมื่อต้องการตรวจสอบการเชื่อมต่ออุปกรณ์ของคุณ ให้ปฏิบัติตามขั้นตอน ต่อไปนี้:

1. ตรวจสอบว่ามีไฟฟ้าที่ช่องจ่ายไฟฟ้า
2. ตรวจสอบว่าสายไฟของอุปกรณ์มีการต่อเข้ากับอุปกรณ์และช่องจ่ายไฟฟ้าอย่างถูกต้อง
3. ตรวจสอบว่าสายเคเบิลสัญญาณของอุปกรณ์มีการต่อเข้ากับอุปกรณ์และการเชื่อมต่อบนยูนิเทรบบอย่างถูกต้อง
4. สำหรับอุปกรณ์ SCSI ให้ตรวจสอบว่าตัวต่อปลายสาย SCSI มีการต่อพ่วง อย่างถูกต้องและค่าติดตั้ง SCSI แอดเดรสถูกต้อง
5. สำหรับอุปกรณ์การสื่อสาร ให้ตรวจสอบว่าอุปกรณ์มีการต่อพ่วงกับ สายการเชื่อมต่ออย่างถูกต้อง
6. ตรวจสอบว่าอุปกรณ์เปิดอยู่

โปรดอ้างอิงเอกสารคู่มือของอุปกรณ์เฉพาะเพื่อดูโปรซีเดเจอร์การเดินสาย เคเบิลและการกำหนดคอนฟิก และข้อมูลการแก้ไข ปัญหา เพิ่มเติม

ไปยังขั้นตอนถัดไปถ้าขั้นตอนในหัวข้อนี้ ไม่ได้แก้ไขปัญหา

วิธีแก้ไขปัญหาการเอาอะแด็ปเตอร์ออก:

คุณอาจได้รับข้อความแสดงข้อผิดพลาดถ้าอุปกรณ์เปิดอยู่เมื่อคุณใช้ คำสั่ง `rmdev` เพื่อยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์

หากข้อความชนิดต่อไปนี้แสดงขึ้นเมื่อคุณใช้คำสั่ง `rmdev` เพื่อยกเลิกกำหนดคอนฟิกอะแด็ปเตอร์ นี่บ่งชี้ว่าอุปกรณ์เปิดอยู่ อาจเนื่องมาจาก แอ็พพลิเคชันยังคงพยายามเข้าถึงอะแด็ปเตอร์ซึ่งคุณกำลังพยายาม เอาออกหรือเปลี่ยน

```
#rmdev -l ent0
Method error (/usr/lib/methods/ucfgent):
0514-062
Cannot perform the requested function because the
specified device is busy.
```

เพื่อแก้ไขปัญหา คุณต้องระบุแอ็พพลิเคชันใดๆ ซึ่งจะยังคงใช้ อะแด็ปเตอร์อยู่และปิดแอ็พพลิเคชันเหล่านั้น แอ็พพลิเคชันเหล่านี้อาจรวมถึงดังต่อไปนี้:

- TCP/IP
- SNA

- OSI
- IPX/SPX
- Novell NetWare
- Streams
- Generic data link control (GDLC)
 - IEEE Ethernet DLC
 - Token-ring DLC
 - FDDI DLC

แอ็พพลิเคชั่น Systems Network Architecture

บางแอ็พพลิเคชั่น SNA ที่อาจกำลังใช้อะแด็ปเตอร์ของคุณรวมถึง:

- DB2[®]
- TXSeries[®] (CICS[®] & Encina)
- DirectTalk
- MQSeries[®]
- HCON
- ADSM

แอ็พพลิเคชั่น Streams

บางแอ็พพลิเคชั่นบนสตรีมที่อาจกำลังใช้อะแด็ปเตอร์ของคุณรวมถึง:

- IPX/SPX
- Novell NetWare V4 และ Novell NetWare Services 4.1
- Connections และ NetBios สำหรับระบบปฏิบัติการนี้

แอ็พพลิเคชั่นที่กำลังรันบนอะแด็ปเตอร์ WAN

แอ็พพลิเคชั่นที่อาจกำลังใช้อะแด็ปเตอร์ WAN ของคุณรวมถึง:

- SDLC
- Bisync
- X.25
- ISDN
- QLLC for X.25

แอ็พพลิเคชั่น TCP/IP

แอ็พพลิเคชั่น TCP/IP ทั้งหมดที่ใช้อินเทอร์เน็ตเฟสเลเยอร์สามารถต่อพ่วงได้ ด้วยคำสั่ง `ifconfig` ซึ่งส่งผลให้แอ็พพลิเคชั่นที่ใช้ TCP/IP ใหม่เอาต์และเตือนผู้ใช้ว่าอินเทอร์เน็ตเฟส ดาวน์ หลังจากคุณเพิ่มหรือเปลี่ยนอะแด็ปเตอร์และรันคำสั่ง `ifconfig` เพื่อต่อพ่วงอินเทอร์เน็ตเฟสแล้ว แอ็พพลิเคชั่นจะกลับมาทำงานอีกครั้ง

การตรวจสอบสภาพความพร้อมของอุปกรณ์:

คุณสามารถตรวจสอบเพื่อดูว่าอุปกรณ์อยู่ในสภาพพร้อมหรือไม่

เมื่อต้องการกำหนดว่าอุปกรณ์อยู่ในสภาพพร้อมหรือไม่ให้ทำดังต่อไปนี้:

1. ตรวจสอบว่าตัวบ่งชี้ความพร้อมของอุปกรณ์เปิดอยู่
2. ตรวจสอบว่าสื่อบันทึกที่ถอดออกได้ เช่น เทป ดิสเก็ต และอุปกรณ์ทางเลือก มีการเสียบอย่างถูกต้อง
3. ตรวจสอบ ribbon, กระจก และหมึกผงสำหรับเครื่องพิมพ์ และพล็อตเตอร์
4. ตรวจสอบว่าสื่อการเขียนสามารถเขียนได้ถ้าคุณกำลังพยายาม เขียนลงในอุปกรณ์

การตรวจสอบของคุณแก้ไขปัญหเกี่ยวกับอุปกรณ์หรือไม่? หากการตรวจสอบ สภาพความพร้อมของอุปกรณ์ไม่ได้แก้ไข ปัญหาให้ไปยังขั้นตอนถัดไป

การวิเคราะห์อุปกรณ์:

เมื่อต้องการกำหนดว่าอุปกรณ์บกพร่องหรือไม่ให้รันการวิเคราะห์ฮาร์ดแวร์ของคุณ

หากการรันการวิเคราะห์ฮาร์ดแวร์ล้มเหลวในการค้นหาปัญหาเกี่ยวกับอุปกรณ์ของคุณ ให้ตรวจสอบซอฟต์แวร์อุปกรณ์ หาก อุปกรณ์ผ่านการทดสอบเชิงวิเคราะห์ แสดงว่าคุณอาจมีปัญหเกี่ยวกับวิธีที่อุปกรณ์ทำงานกับซอฟต์แวร์ระบบ ของคุณ หาก เป็นไปได้ว่ามีปัญหาที่ระบุก่อนหน้านี้ให้รายงานปัญหาไปยังหน่วยงานเซอร์วิสซอฟต์แวร์ของคุณ

การกำหนดคอนฟิกอุปกรณ์ปลายทาง

คำสั่ง `cfgmgr` สามารถใช้ได้กับ แฟล็ก -c เป็นอ็อปชันการเชื่อมต่อสำหรับ ขอบเขตจำกัดของการกำหนดคอนฟิกปลายทางของ อุปกรณ์ I/O

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `cfgmgr`

คอนฟิกูเรชันปลายทางของอุปกรณ์ FC และ FCoE

อ็อปชัน `cfgmgr -c` ใช้กับอะแดปเตอร์ Fibre Channel (FC) และ Fibre Channel over Ethernet (FCoE) สำหรับคอนฟิกูเรชัน ปลายทาง

คำสั่ง `cfgmgr` สามารถใช้พร้อมแฟล็ก -c เป็นอ็อปชันการเชื่อมต่อสำหรับขอบเขตที่จำกัดของคอนฟิกูเรชันอุปกรณ์ สำหรับอะแดปเตอร์ FC และ FCoE ไวยากรณ์เป็นดังนี้:

```
cfgmgr -l fscsi0 -c "parameter=val[,parameter=val,...]"
```

โดยการใส่สตริงตัวกรองการเชื่อมต่อ คุณสามารถจำกัดขอบเขต ของการค้นหาอุปกรณ์โดยใช้พารามิเตอร์ต่อไปนี้หนึ่งหรือ หลายรายการ:

ตารางที่ 70. พารามิเตอร์สำหรับแฟล็ก cfmgr -c

ชื่อพารามิเตอร์	คำอธิบาย
ww_name	ชื่อพอร์ตทั่วโลกของอุปกรณ์ปลายทาง
node_name	ชื่อโหนดทั่วโลกของอุปกรณ์ปลายทาง
scsi_id	IDN_Port อุปกรณ์ปลายทางที่จะแม็พกับ Small Computer System Interface (SCSI) ID สำหรับอุปกรณ์หน่วยเก็บข้อมูล Fibre Channel Protocol (FCP)
lun_id	หมายเลขโลจิคัลยูนิต (LUN)

ตัวอย่างเช่น คำสั่งต่อไปนี้จะกำหนดคอนฟิก LUN เดียวของ lun_id 0x1000000000000 ที่พอร์ตปลายทางของหน่วยเก็บข้อมูล ซึ่งมีชื่อพอร์ตทั่วโลกเป็น 0x5001738000330191:

```
# cfmgr -l fscsi0 -c "ww_name=0x5001738000330191,lun_id=0x1000000000000"
```

การสแกนนี้เกิดขึ้นเฉพาะสำหรับพอร์ตของโฮสต์อะแดปเตอร์ fscsi0

หมายเหตุ:

- อักขระนำ 0x ในค่าพารามิเตอร์ เป็นทางเลือก
- พารามิเตอร์ทั้งหมดต้องมีการแสดงแทนเป็นตัวเลขฐานสิบหก

ในตัวอย่างต่อไปนี้มีกระบวนการหนึ่งพารามิเตอร์เท่านั้น:

```
# cfmgr -l fscsi0 -c "lun_id=0x1000000000000"
```

คำสั่งนี้จะสแกนพอร์ตของอุปกรณ์หน่วยเก็บข้อมูลทั้งหมดบน storage area network (SAN) และกำหนดคอนฟิกโลจิคัลยูนิตเดี่ยวนี้สำหรับทุก พอร์ตปลายทาง SAN ที่มี LUN นี้

แนวทางและกฎสำหรับพารามิเตอร์ตัวกรองการเชื่อมต่อ

พิจารณาจุดต่อไปนีเมื่อคุณใช้พารามิเตอร์ตัวกรองการเชื่อมต่อ:

- คอนฟิกูเรชันปลายทางสำหรับอุปกรณ์ FC และ FCoE ใช้กับสภาวะแวดล้อมการติดตั้ง สวิตช์เท่านั้น ถ้าคุณระบุสตริงการเชื่อมต่อที่ติดตั้งเข้ากับ พอร์ตปลายทางโดยตรง การเชื่อมต่อจะล้มเหลวพร้อมกับข้อความซึ่งบ่งชี้ว่า ไม่พบอุปกรณ์ขายน
- แฟล็ก -c ได้รับการสนับสนุนเฉพาะถ้าใช้พร้อมกับ แฟล็ก -l ของคำสั่ง cfmgr ที่จำกัดขอบเขตของคำสั่งเป็นหนึ่งในอุปกรณ์ fscsiX ในแต่ละครั้ง
- ถ้าคุณระบุ -? เป็นสตริงการเชื่อมต่อ สำหรับแฟล็ก -c ของคำสั่ง cfmgr พร้อมกับแฟล็ก -v ข้อมูลการใช้งานจะ แสดงขึ้น
- ถ้าคุณระบุพารามิเตอร์ซ้ำ (ตัวอย่างเช่น lun_id แสดง สองครั้ง) จะส่งผลให้เกิดข้อผิดพลาด ตรวจสอบไม่พบอุปกรณ์
- ชุดของพารามิเตอร์ lun_id, scsi_id, ww_name และ node_name สามารถใช้ผสมกันได้ ยกเว้น ว่าห้ามซ้ำ เพื่อระบุ LUN, ปลายทาง หรือโหนดหน่วยเก็บข้อมูลที่จะ กำหนดคอนฟิกโดยไม่ซ้ำกัน คุณต้องระบุเพียงหนึ่งหรือสองพารามิเตอร์ถ้า ต้องการ แม้ว่าสามารถใช้พารามิเตอร์ได้มากกว่านั้น รายการต่อไปนี้จะระบุพารามิเตอร์ หรือชุดของพารามิเตอร์ที่จำเป็นในการระบุ LUN, ปลายทาง หรือโหนดหน่วยเก็บข้อมูลโดยไม่ซ้ำกัน:
 - พารามิเตอร์ ww_name และ lun_id ระบุ LUN โดยไม่ซ้ำกันบนโหนดปลายทางที่จะกำหนดคอนฟิก
 - พารามิเตอร์ scsi_id และ lun_id ระบุ LUN โดยไม่ซ้ำกันบนพอร์ตปลายทางที่จะกำหนดคอนฟิก

- พารามิเตอร์ `node_name` และ `lun_id` กำหนดคอนฟิก LUN สำหรับพอร์ตปลายทางทั้งหมดสำหรับโหนดหน่วยเก็บข้อมูลเฉพาะ พารามิเตอร์เหล่านี้สามารถกำหนดคอนฟิกพอร์ตปลายทางได้เฉพาะถ้าพอร์ตปลายทาง ทั้งหมดมีพารามิเตอร์ `node_name` ตัวเดียวกัน ซึ่ง อาจเป็น `true` สำหรับบางอุปกรณ์หน่วยเก็บข้อมูล
- พารามิเตอร์ `pw_name` กำหนดคอนฟิก LUNs ทั้งหมด สำหรับปลายทางเฉพาะ
- พารามิเตอร์ `node_name` กำหนดคอนฟิกพอร์ตปลายทางทั้งหมด สำหรับโหนดหน่วยเก็บข้อมูลเฉพาะ (เฉพาะถ้าพอร์ตปลายทางทั้งหมดมี พารามิเตอร์ `node_name` ตัวเดียวกัน ซึ่งอาจเป็น `true` สำหรับบางอุปกรณ์หน่วยเก็บข้อมูล)
- พารามิเตอร์ `lun_id` กำหนดคอนฟิก LUN บน พอร์ตปลายทางทั้งหมดที่มองเห็นได้จากอุปกรณ์ `fscsi` นั้น
- ถ้าระบุพารามิเตอร์มากกว่าสองรายการ โค้ดคอนฟิกูเรชันอุปกรณ์ จะใช้ข้อมูลพิเศษนี้เพื่อตรวจสอบความถูกต้องของตำแหน่งของอุปกรณ์ ถ้าค่าพารามิเตอร์ที่ระบุใดๆ ขัดแย้งกับ ค่าที่รายงานบน SAN คำสั่งจะล้มเหลว และไม่มีกำหนดคอนฟิกอุปกรณ์

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง `cfgmgr`

เทปไดรฟ์

ฟังก์ชันการจัดการระบบซึ่งอธิบายที่นี่เกี่ยวข้องกับเทปไดรฟ์

ฟังก์ชันเหล่านี้จำนวนมากเปลี่ยนหรือเรียกใช้ข้อมูลจากฐานข้อมูลคอนฟิกูเรชัน อุปกรณ์ ซึ่งมีข้อมูลเกี่ยวกับอุปกรณ์บนระบบของคุณ ฐานข้อมูล คอนฟิกูเรชันอุปกรณ์ประกอบด้วยฐานข้อมูลคอนฟิกูเรชันที่กำหนดไว้ล่วงหน้า ซึ่งมีข้อมูลเกี่ยวกับชนิดทั้งหมดที่เป็นไปได้ของอุปกรณ์ซึ่งได้รับการสนับสนุนบน ระบบ และฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเอง ซึ่งมีข้อมูล เกี่ยวกับอุปกรณ์เฉพาะบนระบบในปัจจุบัน เพื่อให้ระบบปฏิบัติการ สามารถใช้เทปไดรฟ์หรืออุปกรณ์อื่นใด ต้องกำหนดอุปกรณ์ไว้ในฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเองและต้องกำหนดชนิดอุปกรณ์ไว้ในฐานข้อมูลคอนฟิกูเรชันที่กำหนดไว้ล่วงหน้า

แอตทริบิวต์เทปไดรฟ์

คุณสามารถปรับแอตทริบิวต์เทปไดรฟ์เหล่านี้เพื่อให้ตรงกับความต้องการ ของระบบ

แอตทริบิวต์สามารถแสดงหรือเปลี่ยนโดยใช้ SMIT หรือคำสั่ง (โดยเฉพาะอย่างยิ่ง `lsattr` และ `chdev`)

เทปไดรฟ์แต่ละชนิดใช้ชุดย่อยของแอตทริบิวต์ทั้งหมดเท่านั้น

ข้อมูลทั่วไปเกี่ยวกับแต่ละแอตทริบิวต์

ขนาดบล็อก

แอตทริบิวต์ขนาดบล็อกบ่งชี้ขนาดบล็อกที่จะใช้เมื่ออ่าน หรือเขียนเทป ข้อมูลถูกเขียนลงในเทปในบล็อกของข้อมูล โดยมีช่องว่างระหว่างเร็กคอร์ด คั่นระหว่างบล็อก เร็กคอร์ดที่ใหญ่ขึ้นมีประโยชน์เมื่อเขียนลงในเทปที่ไม่ได้จัดรูปแบบ เนื่องจากจำนวนของช่องว่างระหว่างเร็กคอร์ดลดลงบนเทป ส่งผลให้สามารถเขียนข้อมูลได้มากขึ้น ค่า 0 บ่งชี้บล็อกความยาว แปรโต ค่าที่ใช้ได้และค่าดีฟอลต์แตกต่างกันไป ขึ้นอยู่กับ เทปไดรฟ์

บัฟเฟอร์อุปกรณ์

การตั้งค่าแอตทริบิวต์บัฟเฟอร์อุปกรณ์ (โดยใช้คำสั่ง `chdev`) เป็น `mode=yes` บ่งชี้ว่าแอ็พพลิเคชันได้รับแจ้งถึงความสมบูรณ์ของ การเขียนหลังจากข้อมูลถูกโอนย้ายไปยังบัฟเฟอร์ข้อมูลของเทปไดรฟ์แล้ว แต่ไม่จำเป็นต้องเป็นหลังจากที่ข้อมูลถูกเขียนลงในเทปอย่างแท้จริง หากคุณระบุ `mode=no` แอ็พพลิเคชันจะได้รับแจ้งถึงความสมบูรณ์ของ การเขียนเฉพาะหลังจากข้อมูลถูกเขียนลงในเทปอย่างแท้จริงเท่านั้น ไม่สามารถรักษา โหมดการสตรีมสำหรับการอ่านหรือการเขียนถ้าแอตทริบิวต์นี้มีการตั้งค่า เป็น `mode=no` ค่าดีฟอลต์คือ `mode=yes`

ด้วยค่า mode=no เทปไดร์ฟจะทำงานช้าลงแต่มีข้อมูลที่สมบูรณ์มากขึ้น ในกรณีที่ไฟฟ้ดับหรือระบบล้มเหลว และช่วยให้สามารถจัดการกับ เงื่อนไขการสิ้นสุดสื่อบันทึกได้ดียิ่งขึ้น

เครื่องหมายไฟล์แบบขยาย

การตั้งค่าแอตทริบิวต์เครื่องหมายไฟล์แบบขยาย (สำหรับคำสั่ง `chdev`, แอตทริบิวต์ `extfm`) เป็น no จะเขียน เครื่องหมายไฟล์ปกติลงในเทปในทุกเมื่อที่มีการเขียนเครื่องหมายไฟล์ การตั้งค่าแอตทริบิวต์นี้เป็น yes จะเขียนเครื่องหมายไฟล์แบบขยาย สำหรับเทปไดร์ฟ สามารถตั้งค่าแอตทริบิวต์นี้เป็น on ค่าดีฟอลต์คือ no ตัวอย่างเช่น เครื่องหมายไฟล์แบบขยายบนเทปไดร์ฟ 8 มม. ใช้เทปขนาด 2.2 MB และสามารถใช้เวลาถึง 8.5 วินาทีในการเขียน เครื่องหมายไฟล์ปกติใช้ 184 KB และใช้เวลา ประมาณ 1.5 วินาทีในการเขียน

เพื่อลดข้อผิดพลาดเมื่อคุณใช้เทป 8 มม. ในโหมดผนวก ให้ใช้เครื่องหมายไฟล์แบบขยายสำหรับการจัดวางตำแหน่งที่ตีขึ้น หลังจากย้อนกลับการดำเนินงานที่เครื่องหมายไฟล์

การปรับความตึง

การตั้งค่าแอตทริบิวต์ *Retensioning* (สำหรับคำสั่ง `chdev`, แอตทริบิวต์ `ret`) เป็น `ret=yes` สั่งให้เทปไดร์ฟปรับความตึงเทปโดยอัตโนมัติในทุกครั้งที่ใส่เทป หรือรีเซตไดร์ฟ การปรับความตึงเทปหมายถึงการหมุนไปจนสุดเทป แล้วกรอกกลับไปยังตอนต้นของเทปเพื่อให้ความตึงของเทปเท่ากัน ตลอดทั้งเทป การปรับความตึงเทปสามารถลดข้อผิดพลาดได้ แต่การดำเนินการนี้อาจใช้เวลาหลายนาที หากคุณระบุค่า `ret=no` เทปไดร์ฟจะไม่ปรับความตึงเทปโดยอัตโนมัติ ค่าดีฟอลต์คือ `yes`

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

ค่าติดตั้งความหนาแน่น #1 (สำหรับคำสั่ง `chdev`, แอตทริบิวต์ `density_set_1`) ตั้งค่าความหนาแน่นที่เทปไดร์ฟจะเขียนเมื่อใช้ไฟล์พิเศษ `/dev/rmt*`, `/dev/rmt*.1`, `/dev/rmt*.2`, และ `/dev/rmt*.3` ค่าติดตั้งความหนาแน่น #2 (สำหรับคำสั่ง `chdev`, แอตทริบิวต์ `density_set_2`) ตั้งค่าความหนาแน่นที่เทปไดร์ฟจะเขียนเมื่อใช้ไฟล์พิเศษ `/dev/rmt*.4`, `/dev/rmt*.5`, `/dev/rmt*.6`, และ `/dev/rmt*.7` โปรดดูที่ “ไฟล์พิเศษสำหรับเทปไดร์ฟ” ในหน้า 630 สำหรับข้อมูลเพิ่มเติม

ค่าติดตั้งความหนาแน่นมีการแสดงแทนเป็นตัวเลขฐานสิบ ในช่วง 0 ถึง 255 ค่าติดตั้งศูนย์ (0) เลือกความหนาแน่นดีฟอลต์สำหรับเทปไดร์ฟ ซึ่งโดยปกติ เป็นค่าติดตั้งความหนาแน่นสูง ของไดร์ฟ ค่าที่ใช้ได้เฉพาะและความหมายของค่านั้นแตกต่างกันไปตาม ชนิดที่แตกต่างกันของเทปไดร์ฟ แอตทริบิวต์เหล่านี้ไม่กระทบต่อความสามารถของเทปไดร์ฟ ในการอ่านเทปที่เขียนในความหนาแน่นทั้งหมดซึ่งสนับสนุนโดยเทปไดร์ฟ มักมีการตั้งค่าติดตั้งความหนาแน่น #1 เป็นความหนาแน่นสูงสุดที่เป็นไปได้ บนเทปไดร์ฟ และค่าติดตั้งความหนาแน่น #2 เป็นความหนาแน่นสูงสุดลำดับสอง ที่เป็นไปได้บนเทปไดร์ฟ

การสนับสนุนสำรอง

สำหรับเทปไดร์ฟที่ใช้แอตทริบิวต์สำรอง (คำสั่ง `chdev`, แอตทริบิวต์ `res_support`) การระบุค่า `res_support=yes` ทำให้เทปไดร์ฟมีการสำรองไว้บนบัส SCSI ในขณะที่เปิด หากมีมากกว่าหนึ่ง อะแดปเตอร์ SCSI แบ่งใช้อุปกรณ์เทป แอตทริบิวต์นี้ช่วยให้มั่นใจถึงการเข้าถึง โดยอะแดปเตอร์ตัวหนึ่งในขณะที่อุปกรณ์เปิด เทปไดร์ฟ SCSI บางรายการไม่สนับสนุนคำสั่ง `reserve` หรือ `release` เทปไดร์ฟ SCSI บางรายการมีค่าที่กำหนดไว้ล่วงหน้าสำหรับแอตทริบิวต์นี้ เพื่อให้สนับสนุนคำสั่ง `reserve` หรือ `release` เสมอ

ขนาดบล็อกความยาวแปรได้

แอตทริบิวต์ขนาดบล็อกความยาวแปรได้ (สำหรับคำสั่ง `chdev`, แอตทริบิวต์ `var_block_size`) ระบุขนาดบล็อกที่ต้องการ โดยเทปไดร์ฟเมื่อเขียนเร็คคอร์ดความยาวแปรได้ เทปไดร์ฟ SCSI บางรายการกำหนดว่าต้องระบุขนาด

บล็อกที่ไม่ใช่ศูนย์ในข้อมูลเลือกโหมด เมื่อเขียนเร็กคอร์ดความยาวแปรได้ แอ็ททริบิวต์ ขนาด บล็อก มีการตั้งค่า เป็น 0 เพื่อบ่งชี้เร็กคอร์ดความยาวแปรได้ โปรดดูที่ข้อกำหนดคุณลักษณะเทปไดรฟ์ SCSI เฉพาะเพื่อพิจารณาว่า จำเป็นต้องทำเช่นนี้หรือไม่

การบีบอัดข้อมูล

การตั้งค่าแอ็ททริบิวต์การบีบอัดข้อมูล (สำหรับคำสั่ง chdev, แอ็ททริบิวต์ compress) เป็น compress=yes ทำให้ เทปไดรฟ์อยู่ในโหมดบีบอัด ถ้าไดรฟ์สามารถบีบอัดข้อมูลได้ ถ้าเป็นเช่นนั้น ไดรฟ์จะเขียนข้อมูลลงในเทปในรูปแบบ ที่บีบอัดเพื่อให้เก็บข้อมูลได้มากขึ้นบนเทปเดียว การตั้งค่าแอ็ททริบิวต์เป็น no บังคับให้ เทปไดรฟ์เขียนในโหมด native (ไม่บีบอัด) การดำเนินงานอ่านไม่ได้รับ ผลกระทบจากค่าติดตั้งของแอ็ททริบิวต์นี้ ค่าติดตั้งดีฟอลต์คือ yes

Autoloader

การตั้งค่าแอ็ททริบิวต์ Autoloader (สำหรับคำสั่ง chdev, แอ็ททริบิวต์ autoloader) เป็น autoloader=yes ส่งผลให้ Autoloader ใช้งานอยู่ ถ้ามีการติดตั้งบนไดรฟ์ ถ้าเป็นเช่นนั้น และมีเทปอื่นอยู่ในตัวโหลด การดำเนินงานอ่านหรือ เขียนใดๆ ที่ดำเนินไปจนสุดเทป จะทำต่อไปบนเทปถัดไปโดยอัตโนมัติ คำสั่งเทปไดรฟ์ ที่ถูกจำกัดบนเทปคาร์ทริดจ์ เดียวจะไม่ได้รับผลกระทบ ค่าติดตั้ง ดีฟอลต์คือ yes

เวลาหน่วงการลงซ้ำ

แอ็ททริบิวต์เวลาหน่วงการลงซ้ำตั้งค่าจำนวนวินาทีที่ระบรอ หลังจากคำสั่งล้มเหลว ก่อนจะออกใช้คำสั่งอีกครั้ง ระบบอาจออกใช้คำสั่งที่ล้มเหลวอีกครั้งได้สูงสุดถึงสี่ครั้ง แอ็ททริบิวต์นี้ใช้กับเทปไดรฟ์ชนิด OST เท่านั้น ค่าติดตั้ง ดีฟอลต์คือ 45

ไทม์เอาต์การอ่าน/เขียน

แอ็ททริบิวต์ไทม์เอาต์การอ่าน/เขียนหรือเวลาหน่วงสูงสุดสำหรับ READ/WRITE ตั้งค่าจำนวนวินาทีสูงสุดที่ระบบ อนุญาตให้คำสั่ง read หรือ write ทำให้เสร็จสมบูรณ์ แอ็ททริบิวต์นี้ใช้กับเทปไดรฟ์ชนิด OST เท่านั้น ค่าติดตั้ง ดีฟอลต์คือ 144

การส่งคืนข้อผิดพลาดเมื่อเปลี่ยนเทป

แอ็ททริบิวต์การส่งคืนข้อผิดพลาดเมื่อเปลี่ยนหรือรีเซ็ตเทป เมื่อตั้งค่า ทำให้มีการ ส่งคืนข้อผิดพลาดในการเปิดเมื่อ เทปไดรฟ์ถูกรีเซ็ต หรือเทป ถูกเปลี่ยน การดำเนินงานก่อนหน้านั้นบนเทปไดรฟ์ต้องเกิดขึ้นแล้ว ซึ่งทำให้ เทปอยู่ใน ตำแหน่งที่เลเยรตำแหน่งตอนต้นของเทปเมื่อเปิด ข้อผิดพลาด ที่ส่งคืนคือ -1 และ errno มีการตั้งค่าเป็น EIO หลังจาก แสดงในแอ็พพลิเคชันแล้ว เงื่อนไขข้อผิดพลาดจะถูกล้างออกไป นอกจากนี้ การกำหนดคอนฟิก เทปไดรฟ์เองอีก ครั้งจะล้างเงื่อนไขข้อผิดพลาดเช่นกัน

แอ็ททริบิวต์สำหรับเทปไดรฟ์ 2.0 GB 4 มม. (ชนิด 4mm2gb):

ข้อมูลต่อไปนี้เป็นแอ็ททริบิวต์สำหรับเทปไดรฟ์ 2.0 GB 4 มม. (ชนิด 4mm2gb)

ขนาดบล็อก

ค่าดีฟอลต์คือ 1024

บัฟเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอ็ททริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

แอ็ททริบิวต์ที่มีค่าคงที่

หากเทปไดรฟ์มีการกำหนดคอนฟิกเป็นเทปไดรฟ์ 2.0 GB 4 มม. แอ็ททริบิวต์การปรับความตึง การสนับสนุนสำรอง ขนาดบล็อกความยาวแปรได้ ค่าติดตั้งความหนาแน่น #1, และค่าติดตั้งความหนาแน่น #2 มีค่าที่กำหนดไว้ล่วงหน้า ซึ่งไม่สามารถ เปลี่ยนได้ ค่าติดตั้งความหนาแน่นมีการกำหนดไว้ล่วงหน้าเนื่องจากเทปไดรฟ์เขียนในโหมด 2.0 GB เสมอ

แฉัตริบิวต์สำหรับเทปไตร์ฟ 4.0 GB 4 มม. (ชนิด 4mm4gb):

ข้อมูลต่อไปนี้เป็นแฉัตริบิวต์สำหรับเทปไตร์ฟ 4.0 GB 4 มม. (ชนิด 4mm4gb)

ขนาดบล็อก

ค่าตีฟอลต์คือ 1024

บัพเฟอ์อุปกรณั

ข้อมูลทั่วไปสำหรับแฉัตริบิวต์นี้ใช้กับเทปไตร์ฟ ชนิดนี้

ค่าตีตั้งความหนาแน่น #1 และค่าตีตั้งความหนาแน่น #2

ผู้ใชไม่สามารถเปลี่ยนค่าตีตั้งความหนาแน่นของไตร์ฟนี้ อุปกรณักำหนดคอนฟัก ตัวเองอีกครั้โดยอัตโนมัติและขึ้นอยู่กับชนิดสื่อบันทัก Digital Data Storage (DDS) ที่ตีตั้งตีตั้งนี้:

ชนิดสื่อบันทัก	คอนฟักเรชันอุปกรณั
DDS	อ่านอย่างเดี่ยว
DDS III	อ่าน/เขียนในโหมด 2.0 GB เท่านั้น
DDS2	อ่านในความหนาแน่นอย่างใดอย่างหนึ่ง เขียนในโหมด 4.0 GB เท่านั้น
non-DDS	ไม่สนับสนุน คาร์ทรีดจจะตีตออก

การบีบอัดข้อมูล

ข้อมูลทั่วไปสำหรับแฉัตริบิวต์นี้ใช้กับเทปไตร์ฟ ชนิดนี้

แฉัตริบิวต์ที่มีค่าคงที่

หากเทปไตร์ฟมีการกำหนดคอนฟักเป็นเทปไตร์ฟ 4.0 GB 4 มม. แฉัตริบิวต์การปรับความตีง การสนับสนุนสำรองขนาดบล็อกความยาวแปรได้ ค่าตีตั้งความหนาแน่น #1, และค่าตีตั้งความหนาแน่น #2 มีค่าที่กำหนดไว้ล่วงหน้าซึ่งไม่สามารถเปลี่ยนได้

แฉัตริบิวต์สำหรับเทปไตร์ฟ 2.3 GB 8 มม. (ชนิด 8 มม.):

ข้อมูลต่อไปนี้เป็นแฉัตริบิวต์สำหรับเทปไตร์ฟ 2.3 GB 8 มม. (ชนิด 8 มม.)

ขนาดบล็อก

ค่าตีฟอลต์คือ 1024 ค่าเล็กลงลดจำนวนของข้อมูล ที่ตีเก็บบนเทป

บัพเฟอ์อุปกรณั

ข้อมูลทั่วไปสำหรับแฉัตริบิวต์นี้ใช้กับเทปไตร์ฟ ชนิดนี้

เครื่องหมายไฟล์แบบขยาย

ข้อมูลทั่วไปสำหรับแฉัตริบิวต์นี้ใช้กับเทปไตร์ฟ ชนิดนี้

แฉัตริบิวต์ที่มีค่าคงที่

หากเทปไตร์ฟมีการกำหนดคอนฟักเป็นเทปไตร์ฟ 2.3 GB 8 มม. แฉัตริบิวต์การปรับความตีง การสนับสนุนสำรองขนาดบล็อกความยาวแปรได้ การบีบอัดข้อมูล ค่าตีตั้งความหนาแน่น #1, และค่าตีตั้งความหนาแน่น #2 มีค่าที่กำหนดไว้ล่วงหน้าซึ่งไม่สามารถเปลี่ยนได้ ค่าตีตั้งความหนาแน่นมีการกำหนดไว้ล่วงหน้าเนื่องจากเทปไตร์ฟเขียนในโหมด 2.3 GB เสมอ

แฉัตริบิวต์สำหรับเทปไตร์ฟ 5.0GB 8 มม. (ชนิด 8mm5gb):

ข้อมูลต่อไปนี้เป็นแฉัตริบิวต์สำหรับเทปไตร์ฟ 5.0GB 8 มม. (ชนิด 8mm5gb)

ขนาดบล็อก

ค่าดีฟอลต์คือ 1024 ถ้ากำลังเขียนเทปในโหมด 2.3 GB ค่าเล็กจะลดจำนวนของข้อมูลที่จัดเก็บบนเทป

บัฟเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ชนิดนี้

เครื่องหมายไฟล์แบบขยาย

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ชนิดนี้

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

ค่าติดตั้งต่อไปนี้ใช้:

ค่าติดตั้ง	ความหมาย
140	โหมด 5 GB (สามารถบีบอัดได้)
21	โหมด 5 GB (เทปที่ไม่บีบอัด)
20	โหมด 2.3 GB
0	ดีฟอลต์ (โหมด 5.0 GB)

ค่าดีฟอลต์คือ 140 สำหรับค่าติดตั้งความหนาแน่น #1, และ 20 สำหรับค่าติดตั้งความหนาแน่น #2 ค่า 21 สำหรับค่าติดตั้งความหนาแน่น #1 หรือ #2 อนุญาตให้ผู้ใช้อ่านหรือเขียนเทปที่ไม่บีบอัดในโหมด 5 GB ได้

การบีบอัดข้อมูล

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ชนิดนี้

แอตทริบิวต์ที่มีค่าคงที่

หากเทปไดรฟ์มีการกำหนดคอนฟิกเป็นเทปไดรฟ์ 5.0 GB 8 มม. แอตทริบิวต์ การปรับความตึง การสนับสนุนสำรองขนาดบล็อกความยาวแปรได้ มีค่าที่กำหนดไว้ล่วงหน้าซึ่งไม่สามารถเปลี่ยนได้

แอตทริบิวต์สำหรับเทปไดรฟ์ 20000 MB 8 มม. (การกำหนดคอนฟิกเอง):

ข้อมูลต่อไปนี้เป็นแอตทริบิวต์สำหรับเทปไดรฟ์ 20000 MB 8 มม. (การกำหนดคอนฟิกเอง)

ขนาดบล็อก

ค่าดีฟอลต์คือ 1024

บัฟเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ชนิดนี้

เครื่องหมายไฟล์แบบขยาย

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ชนิดนี้

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

ไดรฟ์สามารถอ่านและเขียนคาร์ตริจข้อมูลในรูปแบบ 20.0 GB ในระหว่างคำสั่ง Read ไดรฟ์จะกำหนดรูปแบบที่ถูกเขียนบนเทปโดยอัตโนมัติในระหว่าง Write ค่าติดตั้งความหนาแน่นกำหนดรูปแบบข้อมูลที่ถูกรวบรวมโดย

ค่าติดตั้งต่อไปนี้ใช้:

ค่าติดตั้ง	ความหมาย
39	โหมด 20 GB (สามารถบีบอัดได้)
0	ดีฟอลต์ (โหมด 20.0 GB)

ค่าดีฟอลต์คือ 39 สำหรับค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

การบีบอัดข้อมูล

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

แอตทริบิวต์ที่มีค่าคงที่

หากเทปไดรฟ์มีการกำหนดคอนฟิกเป็นเทปไดรฟ์ 20.0 GB 8 มม. แอตทริบิวต์ การปรับความตึง การสนับสนุนสำรอง ขนาดบล็อกความยาวแปรได้ มีค่าที่กำหนดไว้ล่วงหน้าซึ่งไม่สามารถเปลี่ยนได้

แอตทริบิวต์สำหรับเทปไดรฟ์ 35 GB (ชนิด 35gb):

ข้อมูลต่อไปนี้เป็นแอตทริบิวต์สำหรับเทปไดรฟ์ 35 GB (ชนิด 35gb)

ขนาดบล็อก

ปริมาณงานของ IBM 7205 รุ่น 311 ขึ้นอยู่กับขนาดบล็อก ขนาดบล็อกขั้นต่ำที่แนะนำ สำหรับไดรฟ์นี้คือ 32 KB ขนาดบล็อกใดๆ ที่น้อยกว่า 32 KB จะจำกัดอัตราข้อมูล (สำรองข้อมูลหรือเวลาเรียกคืน) ตารางต่อไปนี้แสดงรายการขนาดบล็อก ที่แนะนำโดยเรียงตามคำสั่ง:

คำสั่งที่สนับสนุน	ขนาดบล็อกดีฟอลต์ (ไบต์)	คำแนะนำ
BACKUP	32 K หรือ 51.2 K (ดีฟอลต์)	ใช้ 32 K หรือ 51.2 K ขึ้นอยู่กับ Backup ตามชื่อ หรือไม่ จำเป็นต้องเปลี่ยนแปลง
TAR	10 K	มีข้อผิดพลาดในคู่มือที่แสดงขนาดบล็อก 512 KB ตั้งค่าพารามิเตอร์Blocking เป็น -N64
MKSYSB	ใหญ่ที่ BACKUP	คำสั่ง MKSYSB ใช้คำสั่ง BACKUP ไม่จำเป็นต้องเปลี่ยนแปลง
DD	n/a	ตั้งค่าพารามิเตอร์Blocking เป็น bs=32K
CPIO	n/a	ตั้งค่าพารามิเตอร์Blocking เป็น -C64

หมายเหตุ: คุณต้องรับรู้ถึงความจุและ throughput เมื่อคุณเลือกขนาดบล็อก ขนาดบล็อกเล็กมีผลกระทบต่อประสิทธิภาพ และส่งผลกระทบต่อความจุณ้อยที่สุด ความสามารถของฟอร์แมต 2.6 GB (ความจุ) และฟอร์แมต 6.0 GB (ความจุ) มีผลกระทบ เมื่อคุณใช้ขนาดบล็อกเล็กกว่าที่แนะนำ ตัวอย่างเช่น ใช้ขนาดบล็อก 1024 bytes เพื่อสำรองข้อมูลขนาด 32 GB ใช้เวลาประมาณ 22 ชั่วโมง การสำรองข้อมูลขนาด 32 GB เท่ากัน แต่ใช้ขนาดบล็อก 32 KB จะใช้เวลาประมาณ 2 ชั่วโมง

บัฟเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

เครื่องหมายไฟล์แบบขยาย

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

แผนภูมิต่อไปนี้แสดงชนิดคาร์ตริจข้อมูลที่สนับสนุนและ ค่าติดตั้งความหนาแน่น (ในฐานะลิบและฐานะลิบหก) สำหรับเทปไดรฟ์ IBM 7205-311 เมื่อคุณทำการดำเนินงานเรียกคืน (อ่าน) เทปไดรฟ์จะ ตั้งค่าความหนาแน่นให้ตรงกับค่าความหนาแน่นที่เขียนไว้โดยอัตโนมัติ เมื่อคุณดำเนินการสำรองข้อมูล (เขียน) คุณต้องตั้งค่าความหนาแน่นให้ตรงกับ คาร์ตริจข้อมูลที่คุณใช้

คาร์ทริดจ์ข้อมูลที่สนับสนุน	ความจุปกติ	ความจุข้อมูลที่บีบอัด	ค่าติดตั้งความหนาแน่น SMIT	ค่าติดตั้งความหนาแน่น HEX
DLTape III	2.6 GB	2.6 GB (ไม่มีการบีบอัด)	23	17h
	6.0 GB	6.0 GB (ไม่มีการบีบอัด)	24	18h
	10.0 GB	20.0 GB (ดีฟอลต์สำหรับ ไดรฟ์)	25	19h
DLTape IIIxt	15.0 GB	30.6 GB (ดีฟอลต์สำหรับ ไดรฟ์)	25	19h
DLTape IV	20.0 GB	40.0 GB	26	1Ah
	35.0 GB	70.0 GB (ดีฟอลต์สำหรับ ไดรฟ์)	27	1Bh

หมายเหตุ: หากคุณร้องขอความจุปกติที่ไม่สนับสนุนของคาร์ทริดจ์ข้อมูล ไดรฟ์จะตั้งค่าดีฟอลต์เป็นความจุสูงสุดที่สนับสนุนของคาร์ทริดจ์ข้อมูล ซึ่งโหลดเข้าในไดรฟ์

การบีบอัดข้อมูล

การบีบอัดจริงขึ้นอยู่กับชนิดของข้อมูลที่ถูกเขียน (ดูตารางก่อนหน้า) มีการสมมติอัตราส่วนการบีบอัดเป็น 2:1 สำหรับ ความจุข้อมูลที่บีบอัดนี้

แฉัตริบิตที่มีค่าคงที่

ข้อมูลทั่วไปสำหรับแฉัตริบิตนี้ใช้กับเทปไดรฟ์ ชนิดนี้

แฉัตริบิตสำหรับเทปไดรฟ์ 150 MB 1/4-นิ้ว (ชนิด 150mb):

ข้อมูลต่อไปนี้เป็นแฉัตริบิตสำหรับเทปไดรฟ์ 150 MB 1/4-นิ้ว (ชนิด 150mb)

ขนาดบล็อก

ขนาดบล็อกดีฟอลต์คือ 512 ขนาดบล็อกอื่นที่ถูกต้องคือ 0 เท่านั้นสำหรับ บล็อกความยาวแปรได้

บัฟเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแฉัตริบิตนี้ใช้กับเทปไดรฟ์ ชนิดนี้

เครื่องหมายไฟล์แบบขยาย

การเขียนในเทป 1/4-นิ้วสามารถเกิดขึ้นที่ตอนต้นของเทป (BOT) หรือหลังจากตรวจพบเทปที่ว่างเปล่าเท่านั้น ถ้ามีข้อมูลอยู่บนเทป คุณไม่สามารถเขียนทับ ข้อมูลนั้นได้ ยกเว้นที่ BOT ถ้าคุณต้องการเพิ่มข้อมูลลงในเทปซึ่งถูกเขียนแล้ว rewind คุณต้องเลื่อนไปข้างหน้าจนกว่าจะพบเครื่องหมายไฟล์ถัดไป ซึ่งส่งผลให้ระบบส่งคืนข้อผิดพลาด จากนั้น คุณสามารถเริ่มต้นการเขียน อีกครั้งได้

การปรับความตึง

ข้อมูลทั่วไปสำหรับแฉัตริบิตนี้ใช้กับเทปไดรฟ์ ชนิดนี้

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

ค่าติดตั้งต่อไปนี้ใช้:

ค่าติดตั้ง	ความหมาย
16	QIC-150
15	QIC-120
0	ดีฟอลต์ (QIC-150) หรือค่าติดตั้งความหนาแน่นล่าสุดใดๆ โดยระบบ ที่ขึ้นอยู่กับ

ค่าดีฟอลต์คือ 16 สำหรับค่าติดตั้งความหนาแน่น #1, และ 15 สำหรับ ค่าติดตั้งความหนาแน่น #2

แอ็ททริบิวต์ที่มีค่าคงที่

หากเทปไดร์ฟมีการกำหนดคอนฟิกเป็นเทปไดร์ฟ 150 MB 1/4-นิ้ว แอ็ททริบิวต์เครื่องหมายไฟล์แบบขยาย การสนับสนุนสำรอง ขนาดบล็อกความยาวแปรได้ และการบีบอัด ข้อมูลมีค่าที่กำหนดไว้ล่วงหน้าซึ่งไม่สามารถเปลี่ยนได้

แอ็ททริบิวต์สำหรับเทปไดร์ฟ 525 MB 1/4-นิ้ว (ชนิด 525mb):

ข้อมูลต่อไปนี้เป็นแอ็ททริบิวต์สำหรับเทปไดร์ฟ 525 MB 1/4-นิ้ว (ชนิด 525mb)

ขนาดบล็อก

ขนาดบล็อกดีฟอลต์คือ 512 ขนาดบล็อกที่ถูกต้องอื่นคือ 0 สำหรับบล็อก ความยาวแปรได้ และ 1024

บัฟเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอ็ททริบิวต์นี้ใช้กับเทปไดร์ฟ ชนิดนี้

เครื่องหมายไฟล์แบบขยาย

การเขียนในเทป 1/4-นิ้วสามารถเกิดขึ้นที่ตอนต้นของเทป (BOT) หรือหลังจากตรวจพบเทปที่ว่างเปล่าเท่านั้น ถ้ามีข้อมูลอยู่บนเทป คุณไม่สามารถเขียนทับ ข้อมูลนั้นได้ ยกเว้นที่ BOT ถ้าคุณต้องการเพิ่มข้อมูลลงในเทปซึ่งถูกเขียน แล้วrewound คุณต้องเลื่อนไปข้างหน้าจนกว่าจะพบเครื่องหมายไฟล์ถัดไป ซึ่งส่งผลให้ระบบส่งคืนข้อผิดพลาด จากนั้น คุณสามารถเริ่มต้นการเขียน อีกครั้งได้

การปรับความตึง

การเขียนในเทป 1/4-นิ้วสามารถเกิดขึ้นที่ตอนต้นของเทป (BOT) หรือหลังจากตรวจพบเทปที่ว่างเปล่าเท่านั้น ถ้ามีข้อมูลอยู่บนเทป คุณไม่สามารถเขียนทับ ข้อมูลนั้นได้ ยกเว้นที่ BOT ถ้าคุณต้องการเพิ่มข้อมูลลงในเทปซึ่งถูกเขียน แล้วrewound คุณต้องเลื่อนไปข้างหน้าจนกว่าจะพบเครื่องหมายไฟล์ถัดไป ซึ่งส่งผลให้ระบบส่งคืนข้อผิดพลาด จากนั้น คุณสามารถเริ่มต้นการเขียน อีกครั้งได้

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

ค่าติดตั้งต่อไปนี้ใช้:

ค่าติดตั้ง	ความหมาย
17	QIC-525*
16	QIC-150
15	QIC-120
0	ดีฟอลต์ (QIC-525) หรือค่าติดตั้งความหนาแน่นล่าสุดใดๆ โดยระบบ ที่ขึ้นอยู่กับ

* QIC-525 เป็นเพียงโหมดเดียวที่สนับสนุนขนาดบล็อก 1024

ค่าดีฟอลต์คือ 17 สำหรับ ค่าติดตั้งความหนาแน่น #1, และ 16 สำหรับค่าติดตั้งความหนาแน่น #2

แอ็ททริบิวต์ที่มีค่าคงที่

หากเทปไดร์ฟมีการกำหนดคอนฟิกเป็นเทปไดร์ฟ 525 MB 1/4-นิ้ว แอ็ททริบิวต์เครื่องหมายไฟล์แบบขยาย การสนับสนุนสำรอง ขนาดบล็อกความยาวแปรได้ และการบีบอัด ข้อมูลมีค่าที่กำหนดไว้ล่วงหน้าซึ่งไม่สามารถเปลี่ยนได้

แอ็ททริบิวต์สำหรับเทปไดรฟ์ 1200 MB 1/4-นิ้ว (ชนิด 1200mb-c):

ข้อมูลต่อไปนี้เป็นแอ็ททริบิวต์สำหรับเทปไดรฟ์ 1200 MB 1/4-นิ้ว (ชนิด 1200mb-c)

ขนาดบล็อก

ขนาดบล็อกดีฟอลต์คือ 512 ขนาดบล็อกที่ถูกต้องอื่นคือ 0 สำหรับบล็อก ความยาวแปรได้ และ 1024

บัฟเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอ็ททริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

เครื่องหมายไฟล์แบบขยาย

การเขียนในเทป 1/4-นิ้วสามารถเกิดขึ้นที่ตอนต้นของเทป (BOT) หรือหลังจากตรวจพบเทปที่ว่างเปล่าเท่านั้น ถ้ามีข้อมูลอยู่บนเทป คุณไม่สามารถเขียนทับ ข้อมูลนั้นได้ ยกเว้นที่ BOT ถ้าคุณต้องการเพิ่มข้อมูลลงในเทปซึ่งถูกเขียน แล้ว rewind คุณต้องเลื่อนไปข้างหน้าจนกว่าจะพบเครื่องหมายไฟล์ถัดไป ซึ่งส่งผลให้ระบบส่งคืนข้อผิดพลาด จากนั้น คุณสามารถเริ่มต้นการเขียน อีกครั้งได้

การปรับความตึง

ข้อมูลทั่วไปสำหรับแอ็ททริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

ค่าติดตั้งต่อไปนี้ใช้:

ค่าติดตั้ง	ความหมาย
21	QIC-1000*
17	QIC-525*
16	QIC-150
15	QIC-120
0	ดีฟอลต์ (QIC-1000) หรือค่าติดตั้งความหนาแน่นล่าสุดใดๆ โดยระบบ ที่ใช้อยู่

* QIC-525 และ QIC-1000 เป็นสองโหมดเท่านั้นที่สนับสนุนขนาดบล็อก 1024

ค่าดีฟอลต์คือ 21 สำหรับค่าติดตั้งความหนาแน่น #1, และ 17 สำหรับ ค่าติดตั้งความหนาแน่น #2

แอ็ททริบิวต์ที่มีค่าคงที่

หากเทปไดรฟ์มีการกำหนดคอนฟิกเป็นเทปไดรฟ์ 1200 MB 1/4-นิ้ว แอ็ททริบิวต์เครื่องหมาย ไฟล์แบบขยาย การสนับสนุนสำรอง ขนาดบล็อกความยาวแปรได้ และการบีบอัดข้อมูลมีค่าที่กำหนดไว้ล่วงหน้าซึ่งไม่สามารถเปลี่ยนได้

แอ็ททริบิวต์สำหรับเทปไดรฟ์ 12000 MB 4 มม. (การกำหนดคอนฟิกเอง):

ข้อมูลต่อไปนี้เป็นแอ็ททริบิวต์สำหรับเทปไดรฟ์ 12000 MB 4 มม. (การกำหนดคอนฟิกเอง)

ขนาดบล็อก

ปริมาณงานของเทปไดรฟ์ IBM 12000 MB 4 มม. ขึ้นอยู่กับขนาดบล็อก ขนาดบล็อกขั้นต่ำที่แนะนำ สำหรับไดรฟ์นี้คือ 32 KB ขนาดบล็อกใดๆ ที่น้อยกว่า 32 KB จะจำกัดอัตราข้อมูล (สำรองข้อมูลหรือเวลาเรียกคืน) ตารางต่อไปนี้แสดงรายการขนาดบล็อก ที่แนะนำโดยเรียงตามคำสั่ง:

คำสั่งที่สนับสนุน BACKUP	ขนาดบล็อกดีฟอลต์ (ไบต์) 32 K หรือ 51.2 K (ดีฟอลต์)	ข้อเสนอแนะ ใช้ 32 K หรือ 51.2 K ขึ้นอยู่กับ Backup ตามชื่อ หรือไม่ ไม่จำเป็นต้องเปลี่ยนแปลง มีข้อผิดพลาดในคู่มือที่แสดงขนาดบล็อก 512 KB ตั้งค่าพารามิเตอร์ BLocking เป็น -N64 คำสั่ง MKSYSB ใช้คำสั่ง BACKUP ไม่จำเป็นต้อง เปลี่ยนแปลง
TAR	10 K	ตั้งค่าพารามิเตอร์ BLocking เป็น bs=32K ตั้งค่าพารามิเตอร์ BLocking เป็น -C64
MKSYSB	ให้ดูที่ BACKUP	
DD CPIO		

หมายเหตุ: คุณต้องรับรู้ถึงความจุและ throughput เมื่อคุณเลือกขนาดบล็อก ขนาดบล็อกเล็กมีผลกระทบต่ออย่างมากต่อประสิทธิภาพ และส่งผลกระทบต่อความจุที่น้อยที่สุด

บัพเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ชนิดนี้

เครื่องหมายไฟล์แบบขยาย

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ชนิดนี้

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

แผนภูมิต่อไปนี้แสดงชนิดการเข้ารหัสข้อมูลที่สนับสนุนและค่าติดตั้งความหนาแน่น (ในฐานะลิบและฐานลิบหก) สำหรับเทปไดรฟ์ IBM 12000 MB 4 มม. เมื่อคุณทำการดำเนินงานเรียกคืน (อ่าน) เทปไดรฟ์จะตั้งค่าความหนาแน่นให้ตรงกับความหนาแน่นที่เขียนไว้โดยอัตโนมัติ เมื่อคุณดำเนินการสำรองข้อมูล (เขียน) คุณต้องตั้งค่าความหนาแน่น ที่เข้ากันกับคาร์ตริจข้อมูลที่คุณใช้

คาร์ตริจข้อมูลที่สนับสนุน	ความจุปกติ	ความจุข้อมูลที่บีบอัด	ค่าติดตั้งความหนาแน่น SMIT	ค่าติดตั้งความหนาแน่น HEX
DDS III	2.0 GB	4.0 GB	19	13h
DDS2	4.0 GB	8.0 GB	36	24h
DDS3	12.0 GB	24.0 GB	37	25h

หมายเหตุ: หากคุณร้องขอความจุปกติที่ไม่สนับสนุนของคาร์ตริจข้อมูล ไดรฟ์จะตั้งค่าดีฟอลต์เป็นความจุสูงสุดที่สนับสนุนของคาร์ตริจข้อมูล ซึ่งไหลดเข้าในไดรฟ์

การบีบอัดข้อมูล

การบีบอัดจริงขึ้นอยู่กับชนิดของข้อมูลที่ถูกเขียน (ดูตารางก่อนหน้า) มีการสมมติอัตราส่วนการบีบอัดเป็น 2:1 สำหรับ ความจุข้อมูลที่บีบอัดนี้

แอตทริบิวต์ที่มีค่าคงที่

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ชนิดนี้

แอตทริบิวต์สำหรับเทปไดรฟ์ 13000 MB 1/4-นิ้ว (การกำหนดคอนฟิกเอง):

ข้อมูลต่อไปนี้ เป็นแอตทริบิวต์สำหรับเทปไดรฟ์ 13000 MB 1/4-นิ้ว (การกำหนดคอนฟิกเอง)

ขนาดบล็อก

ขนาดบล็อกดีฟอลต์คือ 512 ขนาดบล็อกที่ถูกต้องอื่นคือ 0 สำหรับบล็อก ความยาวแปรได้ และ 1024

บัพเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ชนิดนี้

เครื่องหมายไฟล์แบบขยาย

การเขียนในเทป 1/4-นิ้วสามารถเกิดขึ้นที่ตอนต้นของเทป (BOT) หรือหลังจากตรวจพบเทปที่ว่างเปล่าเท่านั้น ถ้ามีข้อมูลอยู่บนเทป คุณไม่สามารถเขียนทับ ข้อมูลนั้นได้ ยกเว้นที่ BOT ถ้าคุณต้องการเพิ่มข้อมูลลงในเทปซึ่งถูกเขียนแล้ว rewind คุณต้องเลื่อนไปข้างหน้าจนกว่าจะพบเครื่องหมายไฟล์ถัดไป ซึ่งส่งผลให้ระบบส่งคืนข้อผิดพลาด จากนั้น คุณสามารถเริ่มต้นการเขียน อีกครั้งได้

การปรับความตึง

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

ค่าติดตั้งต่อไปนี้ใช้:

ค่าติดตั้ง	ความหมาย
33	QIC-5010-DC*
34	QIC-2GB*
21	QIC-1000*
17	QIC-525*
16	QIC-150
15	QIC-120
0	ดีฟอลต์ (QIC-5010-DC)*

* QIC-525, QIC-1000, QIC-5010-DC, และ QIC-2GB เป็นสีโหมดเท่านั้น ที่สนับสนุนขนาดบล็อก 1024

ค่าดีฟอลต์คือ 33 สำหรับ ค่าติดตั้งความหนาแน่น #1, และ 34 สำหรับค่าติดตั้งความหนาแน่น #2

แอตทริบิวต์ที่มีค่าคงที่

หากเทปไดรฟ์มีการกำหนดคอนฟิกเป็นเทปไดรฟ์ 13000 MB 1/4-นิ้ว แอตทริบิวต์เครื่องหมาย ไฟล์แบบขยาย การสนับสนุนสำรอง และ ขนาด บล็อกความยาวแปรได้ มีค่าที่กำหนดไว้ล่วงหน้าซึ่ง ไม่สามารถเปลี่ยนได้

แอตทริบิวต์สำหรับเทปไดรฟ์ 1/2-นิ้ว 9-แทร็ก (ชนิด 9trk):

ข้อมูลต่อไปนี้ เป็นแอตทริบิวต์สำหรับเทปไดรฟ์ 1/2-นิ้ว 9-แทร็ก (ชนิด 9trk)

ขนาดบล็อก

ขนาดบล็อกดีฟอลต์คือ 1024

บัฟเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอตทริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

ค่าติดตั้งต่อไปนี้ใช้:

ค่าติดตั้ง	ความหมาย
3	6250 บิตต่อนิ้ว (bpi)
2	1600 bpi
0	ความหนาแน่นในการเขียนใดก็ตามที่ใช้ก่อนหน้านี้

ค่าดีฟอลต์คือ 3 สำหรับค่าติดตั้งความหนาแน่น #1, และ 2 สำหรับค่าติดตั้งความหนาแน่น #2

แอตทริบิวต์ที่มีค่าคงที่

หากเทปไดรฟ์มีการกำหนดคอนฟิกเป็นเทปไดรฟ์ 1/2-นิ้ว 9-แทร็ก แอตทริบิวต์เครื่องหมาย ไฟล์แบบขยาย การปรับความตึง การสนับสนุนสำรอง ขนาดบล็อกความยาวแปรได้ และการบีบอัดข้อมูลมีค่าที่กำหนดไว้ล่วงหน้าซึ่ง ไม่สามารถเปลี่ยนได้

แอ็ตทริบิวต์สำหรับคาร์ทริดจ์ 3490e 1/2-นิ้ว (ชนิด 3490e):

ข้อมูลต่อไปนี้เป็นแอ็ตทริบิวต์สำหรับคาร์ทริดจ์ 3490e 1/2-นิ้ว (ชนิด 3490e)

ขนาดบล็อก

ขนาดบล็อกดีฟอลต์คือ 1024 ไดรฟ์นี้มีคุณลักษณะอัตราการโอนย้ายข้อมูลสูง และขนาดบล็อกนับเป็นสิ่งที่สำคัญมากต่อการดำเนินงานที่มีประสิทธิภาพ ขนาดบล็อกใหญ่ขึ้น สามารถเพิ่มความเร็วในการดำเนินงานได้เป็นอย่างมาก และโดยทั่วไป ควรจะใช้ขนาดบล็อกใหญ่ที่สุดเท่าที่เป็นไปได้

หมายเหตุ: อย่างไรก็ตาม การเพิ่มค่าบล็อกอาจส่งผลให้เกิด ปัญหาการเข้ากันไม่ได้กับโปรแกรมอื่นบนระบบของคุณ หากกรณีนี้เกิดขึ้น คุณจะ ได้รับข้อความแสดงข้อผิดพลาดต่อไปนี้ในขณะรันโปรแกรมดังกล่าว:

A system call received a parameter that is not valid.

บัฟเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอ็ตทริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

การบีบอัด

ข้อมูลทั่วไปสำหรับแอ็ตทริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

Autoloader

ไดรฟ์นี้มีคุณลักษณะตัวจัดลำดับเทปซึ่งเป็น autoloader ที่โหลด และดีดชุดของคาร์ทริดจ์เทปออกจากตัวโหลดคาร์ทริดจ์ตามลำดับ เพื่อให้ ฟังก์ชันนี้ทำงานอย่างถูกต้อง สวิตช์บนแผงด้านหน้าควรอยู่ใน ตำแหน่ง AUTO และแอ็ตทริบิวต์ Autoloader ต้องมีการตั้งค่าเป็น yes

แอ็ตทริบิวต์สำหรับเทป SCSI อื่น (ชนิด ost):

ข้อมูลต่อไปนี้เป็นแอ็ตทริบิวต์สำหรับเทป SCSI อื่น (ชนิด ost)

ขนาดบล็อก

ค่าดีฟอลต์ระบบคือ 512 แต่ควรจะใช้เป็นขนาดบล็อกดีฟอลต์ ของเทปไดรฟ์ของคุณ ค่าปกติคือ 512 และ 1024 โดยปกติ เทปไดรฟ์ 8 มม. และ 4 มม. ใช้ 1024 และเป็นการสูญเสียพื้นที่ว่างบนเทปโดยเปล่าประโยชน์ ถ้าปล่อยแอ็ตทริบิวต์ขนาดบล็อกไว้ที่ 512 ค่า 0 บ่งชี้ขนาดบล็อกแปรได้ บนอุปกรณ์บางรายการ

บัฟเฟอร์อุปกรณ์

ข้อมูลทั่วไปสำหรับแอ็ตทริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

เครื่องหมายไฟล์แบบขยาย

ข้อมูลทั่วไปสำหรับแอ็ตทริบิวต์นี้ใช้กับเทปไดรฟ์ ชนิดนี้

ค่าติดตั้งความหนาแน่น #1 และค่าติดตั้งความหนาแน่น #2

ค่าดีฟอลต์เป็น 0 สำหรับค่าติดตั้งทั้งสองนี้ ค่าอื่นและความหมายของค่านั้น แตกต่างกันไปสำหรับเทปไดรฟ์ที่แตกต่างกัน

การสนับสนุนสำรอง

ค่าดีฟอลต์เป็น no แต่สามารถตั้งค่าเป็น yes ได้ถ้าไดรฟ์สนับสนุนคำสั่ง reserve/release หากคุณไม่แน่ใจ no เป็นค่าที่ปลอดภัยกว่า

ขนาดบล็อกความยาวแปรได้

ค่าดีฟอลต์ของขนาดบล็อกความยาวแปรได้คือ 0 โดยส่วนใหญ่แล้ว ค่าที่ไม่ใช่ ศูนย์ ใช้บนไดรฟ์ quarter inch cartridge (QIC) โปรดดูที่ข้อกำหนดคุณลักษณะ SCSI สำหรับเทปไดรฟ์เฉพาะสำหรับคำแนะนำ

เวลาหน่วงการลงซ้ำ

แอ็ททริบิวต์นี้ใช้กับเทปไดรฟ์ชนิด ost เท่านั้น

ไทม์เอาต์อ่าน/เขียน

แอ็ททริบิวต์นี้ใช้กับเทปไดรฟ์ชนิด ost เท่านั้น

แอ็ททริบิวต์ที่มีค่าคงที่

หากเทปไดรฟ์มีการกำหนดคอนฟิกเป็นเทปไดรฟ์ SCSI อื่น แอ็ททริบิวต์ เครื่องหมายไฟล์แบบขยาย การปรับความตึง และการบีบอัดข้อมูล มีค่าที่กำหนดไว้ล่วงหน้าซึ่งไม่สามารถเปลี่ยนได้

แอ็ททริบิวต์เทป MPIO

อุปกรณ์เทปที่สนับสนุน MPIO จะมีแอ็ททริบิวต์เพิ่มเติมซึ่งแสดงรายการอยู่ ภายใต้แอ็ททริบิวต์อุปกรณ์ MPIO

หลักการที่เกี่ยวข้อง:

“Multiple Path I/O” ในหน้า 591

ด้วย Multiple Path I/O (MPIO) อุปกรณ์สามารถตรวจพบได้ไม่ซ้ำกัน ผ่านทางการเชื่อมต่อฟิสิคัลตั้งแต่หนึ่งรายการขึ้นไป หรือ *พาท*

ไฟล์พิเศษสำหรับเทปไดรฟ์

มีไฟล์พิเศษหลายไฟล์ที่เชื่อมโยงกับเทปไดรฟ์แต่ละรายการซึ่ง ระบบปฏิบัติการรู้จัก

การเขียนและการอ่านจากไฟล์บนเทปทำโดยใช้ไฟล์พิเศษ rmt ไฟล์พิเศษเหล่านี้คือ /dev/rmt*, /dev/rmt*.1, /dev/rmt*.2, ถึง /dev/rmt*.7 rmt* เป็น ชื่อทางตรรกะของเทปไดรฟ์ เช่น rmt0, rmt1, และต่อไป

โดยการเลือกไฟล์พิเศษที่เชื่อมโยงกับเทปไดรฟ์ไฟล์ใดไฟล์หนึ่ง นั้นแสดงว่า คุณเลือกวิธีการทำการดำเนินงาน I/O ที่เกี่ยวข้องกับ เทปไดรฟ์

ไอเอ็ม

ความหนาแน่น

คำอธิบาย

คุณสามารถเลือกที่จะเขียนด้วยค่าติดตั้งความหนาแน่นเทปไดรฟ์ #1 หรือด้วยค่าติดตั้งความหนาแน่นเทปไดรฟ์ #2 สำหรับค่าติดตั้งความหนาแน่น เหล่านี้เป็นส่วนหนึ่งของแอ็ททริบิวต์ของเทปไดรฟ์ เนื่องจากโดยทั่วไป มักมีการตั้งค่าติดตั้ง ความหนาแน่น #1 เป็นความหนาแน่นสูงสุดที่เป็นไปได้สำหรับเทปไดรฟ์ และค่าติดตั้งความหนาแน่น #2 เป็นความหนาแน่นสูงสุดที่เป็นไปได้ถัดไปสำหรับเทปไดรฟ์ ไฟล์พิเศษที่ใช้ค่าติดตั้งความหนาแน่น #1 ในบางครั้ง จึงเรียกว่าเป็นความหนาแน่น สูง และไฟล์พิเศษที่ใช้ค่าติดตั้งความหนาแน่น #2 ในบางครั้ง จึงเรียกว่าเป็น ความหนาแน่นต่ำ แต่ลักษณะนี้ไม่ถูกต้องเสมอไป เมื่ออ่านจาก เทป ค่าติดตั้งความหนาแน่นจะถูกละเว้นไป

กรอกลับเมื่อปิด

คุณสามารถเลือกที่จะกรอกลับเทปหรือไม่ เมื่อไฟล์พิเศษที่อ้างอิง เทปไดรฟ์ถูกปิด หากเลือก กรอกลับเมื่อปิด เทปจะถูกจัดตำแหน่งไว้ที่ ตอนต้นของเทปเมื่อปิดไฟล์

การปรับความตึงเมื่อเปิด

คุณสามารถเลือกที่จะปรับความตึงเทปหรือไม่ เมื่อไฟล์เปิดขึ้น การปรับความตึงหมายถึงการหมุนไปจนสุด เทป แล้วกรอกลับไปยัง ตอนต้นของเทปเพื่อลดข้อผิดพลาด หากเลือก ปรับความตึงเมื่อเปิด เทปจะถูก จัดตำแหน่งไว้ที่ตอนต้นของเทป โดยเป็นส่วนหนึ่งของโปรเซสการเปิด

ตารางต่อไปนี้แสดงชื่อของไฟล์พิเศษ rmt และลักษณะของไฟล์นั้น

ไฟล์พิเศษ	กรอกลับเมื่อปิด	ปรับความตึงเมื่อเปิด	ค่าติดตั้งความหนาแน่น
/dev/rmt*	ใช่	ไม่ใช่	#1
/dev/rmt*.1	ไม่ใช่	ไม่ใช่	#1
/dev/rmt*.2	ใช่	ใช่	#1
/dev/rmt*.3	ไม่ใช่	ใช่	#1
/dev/rmt*.4	ใช่	ไม่ใช่	#2
/dev/rmt*.5	ไม่ใช่	ไม่ใช่	#2
/dev/rmt*.6	ใช่	ใช่	#2
/dev/rmt*.7	ไม่ใช่	ใช่	#2

สมมติว่าคุณต้องการเขียนสามไฟล์ลงบนเทปในเทปไดรฟ์ rmt2 ไฟล์แรก อยู่ที่ตอนต้นของเทป ไฟล์ที่สองอยู่ต่อจากไฟล์แรก และไฟล์ที่สามอยู่ต่อจากไฟล์ที่สอง ยิ่งไปกว่านั้น สมมติว่าคุณ ต้องการค่าติดตั้งความหนาแน่น #1 สำหรับเทปไดรฟ์ รายการของไฟล์พิเศษต่อไปนี้ ในลำดับที่กำหนด สามารถใช้สำหรับการเขียนเทปได้

1. /dev/rmt2.3
2. /dev/rmt2.1
3. /dev/rmt2

มีการเลือกไฟล์พิเศษเฉพาะเหล่านี้เนื่องจาก:

- /dev/rmt2.3 มีการเลือกเป็นไฟล์แรกเนื่องจากไฟล์นี้มี ปรับความตึงเมื่อเปิด ซึ่งทำให้มั่นใจว่าไฟล์แรกอยู่ที่ตอนต้น ของเทป ไม่ได้เลือก กรอกลับเมื่อปิด เนื่องจากการดำเนินงาน I/O ถัดไป จะเริ่มขึ้นในตำแหน่งที่ไฟล์นี้สิ้นสุดลง หากเทปอยู่ที่ตอนต้นอยู่แล้ว เมื่อเปิดไฟล์แรก การใช้ไฟล์ /dev/rmt2.1 เป็น ไฟล์แรกจะเร็วขึ้นเนื่องจากไม่ต้องใช้เวลาในการปรับความตึง เทป
- /dev/rmt2.1 มีการเลือกเป็นไฟล์ที่สองเนื่องจากไฟล์นี้ ไม่มีการเลือกทั้ง ปรับความตึงเมื่อเปิด และ กรอกลับเมื่อปิด ไม่มีเหตุผลที่จะ ไปที่ตอนต้นของเทปเมื่อเปิดไฟล์หรือ เมื่อปิดไฟล์
- /dev/rmt2 มีการเลือกเป็นไฟล์ที่สามและไฟล์สุดท้ายเนื่องจากไม่ต้อง การ ปรับความตึงเมื่อเปิด เนื่องจากไฟล์ที่สามต้อง ต่อจากไฟล์ ที่สอง มีการเลือก กรอกลับเมื่อปิด เนื่องจากไม่มีแผนที่จะทำการเขียนเพิ่มเติม บนเทปหลังจากไฟล์ที่สาม การใช้งานครั้งถัดไปของเทปจะเริ่มขึ้น ที่ตอนต้นของเทป

นอกเหนือจากการควบคุมการดำเนินงานเทปโดยการเลือกไฟล์พิเศษ rmt เฉพาะแล้ว คุณยังสามารถใช้คำสั่ง `tctl` เพื่อควบคุมการดำเนินงานเทปได้เช่นกัน

การสนับสนุนอุปกรณ์ USB

เลขยี่ห้อที่แตกต่างกันของไดรเวอร์ในระบบย่อย Universal Serial Bus (USB) ทำงานร่วมกันเพื่อสนับสนุนสิ่งที่แนบของอุปกรณ์ USB ต่างๆ ซึ่งรวมถึง flash ไดรฟ์, Blu-ray ไดรฟ์ เทป, CD-ROM, คีย์บอร์ด เม้าส์ ลำโพง และอื่นๆ

การสนับสนุน USB flash ไดรฟ์

เริ่มต้นด้วย AIX 5.3 ที่มีระดับเทคโนโลยี 5300-09 และ AIX 6.1 ที่มีระดับเทคโนโลยี 6100-02, Universal Serial Bus (USB) flash ไดรฟ์ได้รับการสนับสนุน

การสนับสนุนอุปกรณ์เหล่านี้มีการรวมไว้ในแพ็คเกจอุปกรณ์ต่อไปนี้:

devices.usbif.08025002

การสนับสนุน USB flash ไดรฟ์ของ AIX มีการตรวจสอบโดยเปรียบเทียบกับตัวอย่างของมาตรฐานอุตสาหกรรม OEM USB flash ไดรฟ์ไดรเวอร์อุปกรณ์สำหรับ AIX USB-โฮสต์ คอนโทรลเลอร์สนับสนุน USB 2.0 USB flash ไดรฟ์มีการกำหนดคอนฟิกด้วยชื่อ โลจิกคัล เช่น usbms0 และ usbms1 และนำเสนอทั้งไฟล์พิเศษ raw และบล็อก ตัวอย่างเช่น ไฟล์พิเศษ raw สำหรับ usbms0 คือ /dev/rusbms0 และไฟล์พิเศษบล็อกคือ /dev/usbms0 ก่อนหน้า AIX เวอร์ชัน 5.3 ที่มีระดับเทคโนโลยี 5300-11 และ AIX เวอร์ชัน 6.1 ที่มีระดับเทคโนโลยี 6100-04, USB flash ไดรฟ์มีการกำหนดคอนฟิก เป็น /dev/flashdrive0

ระบบไฟล์ International Organization for Standardization (ISO) (อ่านอย่างเดียว ISO 9660) ได้รับการสนับสนุนบนไดรฟ์เหล่านี้ คุณสามารถสร้างสำเนาสำรองระบบบนไดรฟ์โดยใช้คำสั่ง tar, คำสั่ง cpio หรือสำเนาสำรองหรือเรียกคืนที่เก็บถาวร คุณยังสามารถใช้คำสั่ง dd เพื่อเพิ่มอิมเมจ ISO ลงในไดรฟ์

ระบบปฏิบัติการ AIX ไม่สนับสนุน plug-and-play สำหรับ USB flash ไดรฟ์ เมื่อต้องการทำให้ flash ไดรฟ์พร้อมใช้งานสำหรับผู้ใช้อื่นๆ ผู้ใช้ root ต้องเชื่อมต่อไดรฟ์กับพอร์ต USB ของระบบ และรัน คำสั่งต่อไปนี้:

```
cfgmgr -l usb0
```

ข้อควรสนใจ: ใช้ความระมัดระวังเมื่อคุณถอด flash ไดรฟ์ออกจากพอร์ต หากปิดหรือถอนการติดตั้งไดรฟ์อย่างไม่ถูกต้องก่อนที่คุณจะถอดไดรฟ์ออก ข้อมูลบนไดรฟ์อาจเสียหาย

หลังจากที่ถอดไดรฟ์แล้ว ไดรฟ์ยังคงอยู่ในสถานะพร้อมใช้งาน ในฐานะข้อมูล Object Data Manager (ODM) จนกว่าผู้ใช้ root จะรันคำสั่งต่อไปนี้:

```
rmdev -l usbmsn
```

เมื่อไดรฟ์อยู่ในสถานะพร้อมใช้งานแล้ว คุณสามารถเชื่อมต่อไดรฟ์ กับระบบได้อีกครั้ง และไดรฟ์สามารถประกอบหรือเปิดขึ้นใหม่ได้ หากยกเลิกการเชื่อมต่อไดรฟ์จากพอร์ต USB ของระบบขณะที่ไดรฟ์ยังคงเปิดสำหรับผู้ใช้อื่นๆ คุณไม่สามารถใช้ไดรฟ์นั้นซ้ำจนกว่าผู้ใช้จะปิดและเปิดไดรฟ์ใหม่

การสนับสนุนบลูเรย์ไดรฟ์ USB แบบอ่านอย่างเดียว

AIX Version 6.1 with the 6100-06 Technology Level และใหม่กว่า รู้จักและกำหนดคอนฟิกไดรฟ์ Blu-ray แบบ USB

คุณลักษณะนี้รวมอยู่ในแพ็คเกจอุปกรณ์ต่อไปนี้:

devices.usbif.08025002

ความสามารถของระบบปฏิบัติการ AIX ในการอ่านสื่อบันทึกแบบ Blu-ray ถูกตรวจสอบกับตัวอย่างมาตรฐานอุตสาหกรรม original equipment manufacturer (OEM) ของไดรฟ์ Blu-ray แบบ USB

ไดรฟ์ Blu-ray แบบ USB ถูกกำหนดคอนฟิกโดยใช้ชื่อโลจิกคัล เช่น cd0 และ cd1 ไดรฟ์แสดงทั้งไฟล์ดิบและไฟล์บล็อกพิเศษ ตัวอย่างเช่น ไฟล์ดิบพิเศษสำหรับ cd0 คือ /dev/rcd0 และ ไฟล์บล็อกพิเศษคือ /dev/cd0

ความสามารถในการอ่านอย่างเดียวมีไว้สำหรับระบบไฟล์แบบ International Organization for Standardization (ISO) (อ่านอย่างเดียว ISO 9660), ระบบไฟล์แบบ Universal Disk Format (UDF) (เวอร์ชัน 2.01 หรือก่อนหน้า) และคำสั่งการเข้าถึงสื่อบันทึกออปติคัลต่างๆ เช่น dd และ tar

ระบบปฏิบัติการ AIX ไม่สนับสนุนการเขียนลงสื่อบันทึกแบบ CD, DVD หรือ Blu-ray ที่มีอยู่ในไดรฟ์ Blu-ray แบบ USB แม้ว่าจะไม่มีการป้องกันการเขียน (หากไดรฟ์สามารถเขียนได้) ไว้ก็ตาม IBM จะไม่รองรับปัญหาใดๆ ที่เกิดขึ้นในการเขียน

ระบบปฏิบัติการ AIX ไม่สนับสนุน plug-and-play สำหรับไดรฟ์ Blu-ray แบบ USB เมื่อต้องการทำให้ไดรฟ์ Blu-ray แบบ USB พร้อมใช้งานสำหรับผู้ใช้ AIX ผู้ใช้ต้องเชื่อมต่อไดรฟ์นั้นกับพอร์ต USB ของระบบ และรันคำสั่งต่อไปนี้:

```
cfgmgr -l usb0
```

หลังจากที่ถอดไดรฟ์แล้ว ไดรฟ์ยังคงอยู่ในสถานะพร้อมใช้งาน ในฐานะข้อมูล Object Data Manager (ODM) จนกว่าผู้ใช้จะรันคำสั่งต่อไปนี้:

```
rmdev -l cdn
```

เมื่อไดรฟ์อยู่ในสถานะพร้อมใช้งานแล้ว คุณสามารถเชื่อมต่อไดรฟ์ กับระบบได้อีกครั้ง หากยกเลิกการเชื่อมต่อไดรฟ์จากพอร์ต USB ของระบบขณะที่ยังคงเปิดให้กับผู้ใช้ คุณไม่สามารถใช้ไดรฟ์นั้นจนกว่าคุณจะปิดและเปิดไดรฟ์ใหม่

การแคชข้อมูลหน่วยเก็บข้อมูล

AIX Version 7.1 with Technology Level 4 Service Pack 5 (7100-04-05) และใหม่กว่า สนับสนุนการแคชฝั่งเซิร์ฟเวอร์ของข้อมูลหน่วยเก็บข้อมูล

อุปกรณ์แคชสามารถเป็นหนึ่งในอุปกรณ์ชนิดต่อไปนี้:

- อุปกรณ์แฟลชที่เชื่อมต่อกับเซิร์ฟเวอร์ เช่น โซลิดสเตตไดรฟ์ในตัว (SSD) ในเซิร์ฟเวอร์
- อุปกรณ์แฟลชที่เชื่อมต่อกับเซิร์ฟเวอร์โดยตรงโดยใช้คอนโทรลเลอร์ Serial Attached SCSI (SAS)
- ธีซอร์สแฟลชใน storage area network (SAN)

แนวคิดเกี่ยวกับการแคชข้อมูลหน่วยเก็บข้อมูล

คุณสามารถแคชข้อมูลแคชได้แบบไดนามิก (เริ่มต้นหรือหยุดการแคช) ขณะที่เวิร์กโหนด รันอยู่ เวิร์กโหนดไม่จำเป็นต้องทำให้อยู่ในสถานะไม่แอ็คทีฟเพื่อดำเนินการ การแคช

คำศัพท์ต่อไปนี้ใช้เพื่ออธิบายแนวคิดเกี่ยวกับการแคช:

อุปกรณ์แคช

อุปกรณ์แคชเป็นไดรฟ์โซลิดสเตต (SSD) หรือแฟลชไดรฟ์ที่ใช้สำหรับการแคช

แคชพูล

แคชพูลเป็นกลุ่มของอุปกรณ์แคชที่ใช้สำหรับการแคชหน่วยเก็บข้อมูลเท่านั้น

แคชพาร์ติชัน

แคชพาร์ติชันเป็นอุปกรณ์แคชแบบโลจิคัลที่สร้างจากแคชพูล

อุปกรณ์เป้าหมาย

อุปกรณ์เป้าหมายเป็นอุปกรณ์หน่วยเก็บข้อมูลที่จะถูกแคช

แคชพาร์ติชันเดียวสามารถใช้เพื่อแคชหนึ่งหรือหลายอุปกรณ์เป้าหมาย เมื่ออุปกรณ์เป้าหมายถูกแคช คำร้องขอการอ่านทั้งหมดสำหรับบล็อกของอุปกรณ์จะถูกกำหนดเส้นทางไปยังซอฟต์แวร์การแคช หาก พบบล็อกเฉพาะเจาะจงในแคช คำร้องขอ I/O จะถูกประมวลผลจากอุปกรณ์แคช หาก ไม่พบบล็อกที่ร้องขอในแคช หรือหากเป็นคำร้องขอการเขียน คำร้องขอ I/O จะกลับไปยัง อุปกรณ์เป้าหมาย

ข้อดีของการแคชข้อมูลหน่วยเก็บข้อมูล

การแคชฝั่งเซิร์ฟเวอร์ของข้อมูลหน่วยเก็บข้อมูลสามารถเพิ่มความหนาแน่น virtualization ได้ โดยเฉพาะเมื่อระบบย่อยหน่วยเก็บข้อมูลมีการใช้งานมาก

การแคชข้อมูลหน่วยเก็บข้อมูลมีข้อดีต่อไปนี้:

เวลาแฝง

เวิร์กโหนดในการวิเคราะห์และเชิงธุรกิจรวมลดเวลาการตอบสนองของเคียวริงเนื่องจากหน่วยเก็บข้อมูลโซลิดสเตทไดรฟ์ (SSD) มีเวลาแฝงน้อยกว่า หากผู้ใช้การแคชฝั่งเซิร์ฟเวอร์ เวลาแฝงเฉลี่ยสำหรับ เวิร์กโหนดเชิงธุรกิจรวมสามารถลดลงได้ครึ่งหนึ่ง

ปริมาณงาน

เวิร์กโหนดการประมวลผลธุรกรรมออนไลน์ (OLTP) มีอัตราธุรกรรมที่ขึ้นเนื่องจากหน่วยเก็บข้อมูล SSD มีทรูพุดที่ดีกว่า

ทรูพุดในการเขียน

ในสภาวะแวดล้อมที่ storage area network (SAN) มีการใช้งานมาก อุปกรณ์แฟลช ซึ่งใช้เป็นแคช สามารถช่วยลดเปอร์เซ็นต์ของการร้องขอการอ่านลงได้อย่างชัดเจน เมื่อคำร้องขอการอ่านลดลง SAN สามารถมีทรูพุดในการเขียนที่ดีขึ้น และสามารถให้บริการกับไคลเอนต์และโฮสต์จำนวนมากได้อย่างมีประสิทธิภาพ

memory footprint น้อย

หากมีการกำหนดค่าอุปกรณ์แคชแบบแฟลช เวิร์กโหนดบางอย่างสามารถทำได้แม้ว่าเมื่อมี memory footprint น้อย

ข้อจำกัดสำหรับการแคชข้อมูลหน่วยเก็บข้อมูล

ตรวจสอบให้แน่ใจว่าคุณเข้าใจข้อจำกัดและข้อกำหนดการกำหนดคอนฟิกเพิ่มเติม เพื่อใช้คุณลักษณะการแคช คุณยังต้องพิจารณาข้อจำกัดของแอปพลิเคชันสำหรับอุปกรณ์เป้าหมาย ที่ต้องถูกแคชด้วย

พิจารณาข้อจำกัดต่อไปนี้สำหรับการแคชข้อมูลหน่วยเก็บข้อมูล:

- ซอฟต์แวร์การแคชถูกกำหนดค่าเป็นแคชที่อ่านได้อย่างเดียว ซึ่งหมายความว่าคำร้องขอการอ่านเท่านั้นที่สามารถผ่านจากแฟลชไดรฟ์โซลิดสเตท (SSD) คำร้องขอการเขียนทั้งหมดจะถูกประมวลผลโดย อุปกรณ์หน่วยเก็บข้อมูลต้นทาง
- ข้อมูลที่ถูกเขียนไปยังอุปกรณ์หน่วยเก็บข้อมูลจะไม่ถูกเติมแคชโดยอัตโนมัติ หากการดำเนินการเขียน ถูกดำเนินการบนบล็อกที่อยู่ในแคช ข้อมูลที่มีอยู่ในแคชจะถูกทำเครื่องหมายว่า ไม่ถูกต้อง บล็อกเดียวกันจะปรากฏอีกครั้งในแคชเมื่อมีความถี่และความใหม่ล่าสุดตรงกับ ความต้องการที่จะเติมในแคชใหม่เท่านั้น
- หน่วยความจำเพิ่มเติมจำเป็นสำหรับแต่ละโลจิคัลพาร์ติชัน AIX (LPAR) เนื่องจากซอฟต์แวร์การแคชจะจัดการข้อมูลเมตามบิตและบล็อกการอ่าน คุณต้องการหน่วยความจำอย่างน้อย 4 GB สำหรับ LPAR ใดๆ ที่เปิดใช้งานการแคช
- ซอฟต์แวร์การแคชจะโหลดข้อมูลลงในแคชตามรูปแบบการอ่านแบบโลคัล และทำให้รายการแคช ใช้ไม่ได้บนโลคัล อุปกรณ์เป้าหมายต้องไม่ถูกแบ่งใช้โดยมากกว่าหนึ่ง LPAR พร้อมกัน อุปกรณ์เป้าหมายไม่สามารถเป็นส่วนหนึ่งของหน่วยเก็บข้อมูลแบบคลัสเตอร์ใดๆ เช่น Oracle Real Application Clusters (RAC), DB2 pureScale® และ General Parallel File System (GPFS™) อุปกรณ์เป้าหมายที่เป็นส่วนหนึ่งของคลัสเตอร์ความพร้อมใช้งานสูง สามารถแคชได้หากสิทธิ์ทำให้แน่ใจได้ว่าจะมีที่ละหนึ่งโฮสต์ที่กำลังอ่านหรือเขียนข้อมูลจาก เป้าหมายเท่านั้น และการแคชถูกเปิดใช้งานบนโหนดที่แอ็คทีฟเท่านั้น
- ดิสก์แคชสามารถจัดเตรียมให้กับ AIX LPAR หรือ Virtual I/O Server (VIOS) LPAR อุปกรณ์แคชไม่สามารถแบ่งใช้ได้

- ซอฟต์แวร์การแคชต้องเปิดอุปกรณ์เป้าหมายเพื่อตักคำร้องขอ I/O ใดๆ ไปยัง อุปกรณ์เป้าหมาย หากเวิร์กโหนดต้องการเปิดอุปกรณ์เป้าหมายเป็นพิเศษ หลังจากการแคชเริ่มทำงานแล้ว การดำเนินการเปิดเพิ่มเติมจะล้มเหลว ในกรณีเหล่านี้ การแคชต้องหยุดการทำงานและรีสตาร์ทหลังจาก เวิร์กโหนดเริ่มทำงาน
- หากดิสก์ถูกใช้เป็นอุปกรณ์เป้าหมาย ดังนั้น แอ็ททริบิวต์ `reserve_policy` ของ ดิสก์ต้องไม่ถูกตั้งค่าเป็น `single_path`
- เมื่อการดำเนินการแคชเริ่มทำงานสำหรับอุปกรณ์เป้าหมาย ทรกระแคชเอ็นจินจะหน่วงเวลาการโปรโมต ข้อมูลลงในแคชออกไป การหน่วงเวลานี้จำเป็นเพื่อให้แน่ใจว่าการดำเนินการ I/O ที่ค้างอยู่ทั้งหมด บนอุปกรณ์เป้าหมาย ซึ่งเรียกทำงานก่อนที่การดำเนินการแคชจะเริ่มทำงาน จะเสร็จสมบูรณ์ก่อนที่จะเริ่มต้นการดำเนินการแคช เวลาที่แน่นอนของการหน่วงเวลาถูกคำนวณภายใน ตามจำนวนของพาร์ทิชันที่มี และแอ็ททริบิวต์ `rw_timeout` (หากมี) ของ ดิสก์เป้าหมาย หากเวลาที่คำนวณได้ภายในต้องถูกแทนที่ด้วยเวลาที่ผู้ใช้กำหนด คุณสามารถตั้งค่า ตัวแปรสถานะแวดล้อม `DEFAULT_IO_DRAIN_TIMEOUT_PD` ในไฟล์ `/etc/environment` เป็นค่าการหน่วงเวลาที่กำหนดเอง หน่วยเป็นวินาที

คอมโพเนนต์ของการแคชข้อมูลหน่วยเก็บข้อมูล

ซอฟต์แวร์การแคชประกอบด้วยการจัดการแคชและคอมโพเนนต์แคชเอ็นจิน

การจัดการแคช

คุณสามารถจัดการการแคชข้อมูลหน่วยเก็บข้อมูลโดยใช้คำสั่ง `cache_mgt` ซึ่ง พร้อมใช้งานบนระบบปฏิบัติการ the AIX และบน Virtual I/O Server (VIOS) คุณสามารถใช้คำสั่ง `cache_mgt` เพื่อดำเนินการงานต่อไปนี้:

- เพื่อสร้างและแบ่งพาร์ติชันแคชพูล
- เพื่อกำหนดแคชพาร์ติชันให้กับอุปกรณ์เป้าหมาย หรือโลจิคัลพาร์ติชันของ AIX (LPAR) เป็นอุปกรณ์ Small Computer System Interface (vSCSI) เสมือน
- เพื่อเริ่มต้นและหยุดการดำเนินการแคช

แคชเอ็นจิน

แคชเอ็นจินเป็นส่วนที่จำเป็นที่สุดของซอฟต์แวร์การแคช แคชเอ็นจินจะตัดสินใจ ว่าบล็อกใดในหน่วยเก็บข้อมูลต้อง และข้อมูลต้องถูกดึงจากแคชหรือ หน่วยเก็บข้อมูลหลัก

อัลกอริทึมการแคชขึ้นอยู่กับกลไก `populate-on-read` ที่จะเติมแคชด้วยข้อมูล ที่มีช่องว่างแบบโลคัล (ใกล้เคียงบล็อกอื่นที่ถูกอ่านล่าสุด) อัลกอริทึมการแคชจะเติมข้อมูล ในแคชเร็วขึ้นเมื่อแคชว่าง

บล็อกทั้งหมดในแคชจะถูกมอนิเตอร์เพื่อตรวจสอบว่าถูกอ่านบ่อยเพียงใด และแผนผังความร้อน จะถูกสร้างขึ้น แผนผังความร้อนจะพิจารณาทั้งความถี่และความใหม่ล่าสุดของการเข้าถึง เมื่อแคชถูกเติมข้อมูลเต็มแล้ว รายการใหม่จะถูกเพิ่มเข้ากับแคชเมื่อบล็อกใหม่อุ่นกว่าบล็อกที่เย็นที่สุดในแคช บล็อกที่เย็นที่สุดจะถูกลบออกจากแคช และรายการใหม่จะถูกเพิ่ม

การเติมข้อมูลแบบแข่งขันนี้ทำให้แน่ใจว่าจะมีเวลาอุ่นเครื่องที่ทำให้แคชมีประสิทธิภาพสั้นทันทีที่ถูกเปิดใช้งาน นโยบายการลบข้อมูลออก ซึ่งขึ้นอยู่กับแผนผังความร้อน ทำให้แน่ใจว่าการแคชจะเป็นแบบไดนามิก และปรับให้เข้ากับการเปลี่ยนแปลงรูปแบบของเวิร์กโหนด

การกำหนดค่าการแคชข้อมูลหน่วยเก็บข้อมูล

ในระบบปฏิบัติการ AIX การแคชฝั่งเซิร์ฟเวอร์ของอุปกรณ์แฟลชได้รับการสนับสนุนในคอนฟิกูเรชันที่แตกต่างกันจำนวนมาก คอนฟิกูเรชันเหล่านี้แตกต่างกันในวิธีการจัดเตรียมอุปกรณ์แคชให้กับโลจิคัลพาร์ติชัน (LPAR) ของ AIX

การแคชฝั่งเซิร์ฟเวอร์สนับสนุนโหมดต่อไปนี้ในระบบปฏิบัติการ AIX:

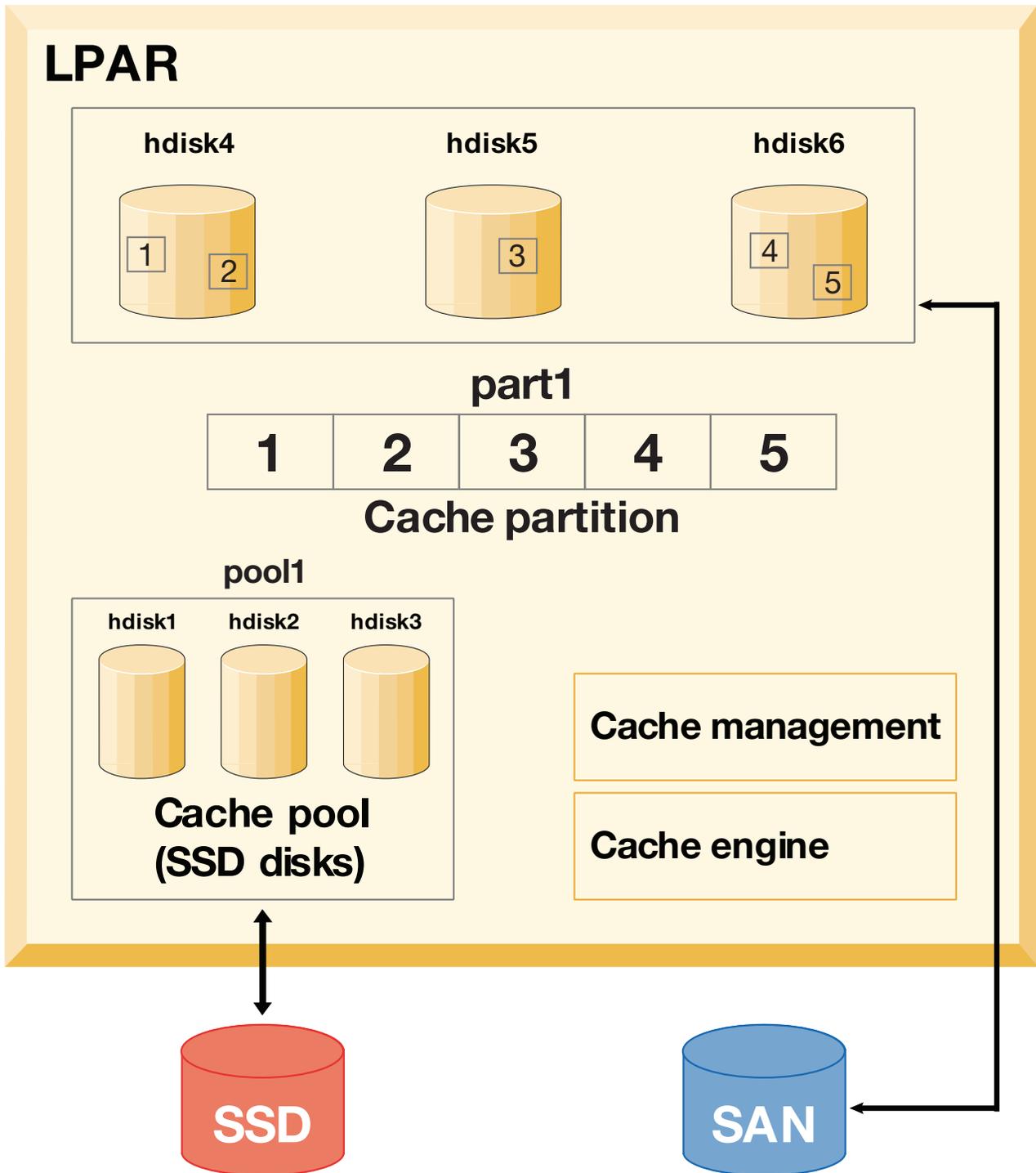
- โหมดเฉพาะ
- โหมดเสมือน
- โหมด N_Port ID Virtualization (NPIV)

การแคชข้อมูลหน่วยเก็บข้อมูลในโหมดที่แยกต่างหาก:

ในโหมดที่แยกต่างหาก อุปกรณ์แคชจะถูกจัดเตรียมสำหรับโลจิคัลพาร์ติชัน (LPAR) ของ AIX โดยตรง

คุณต้องสร้างแคชพูล และจากนั้นแคชพาร์ติชันบนอุปกรณ์แคช จะมีเพียงแคชพาร์ติชันเดียวที่สามารถสร้างในอุปกรณ์แคชที่แยกต่างหาก คุณสามารถใช้แคชพาร์ติชันเพื่อแคชอุปกรณ์เป้าหมายจำนวนเท่าใดก็ได้บน AIX LPAR LPAR ไม่ใช่แบบโมบายล์เนื่องจากอุปกรณ์แคชเป็นอุปกรณ์เฉพาะสำหรับ LPAR นี้ หาก LPAR ต้องถูกโอนย้ายไปยังเซิร์ฟเวอร์อื่น คุณต้องหยุดการแคชเองและยกเลิกการกำหนดค่าอุปกรณ์แคชก่อนการโอนย้าย

รูปต่อไปนี้แสดงตัวอย่างของการกำหนดคอนฟิกการแคชบน AIX LPAR สำหรับอุปกรณ์แคชเฉพาะ



รูปที่ 15. การแคชข้อมูลหน่วยเก็บข้อมูล: คอนฟิกูเรชันสำหรับอุปกรณ์แคชที่แยกต่างหาก

พิจารณาอุปกรณ์แคช คือ hdisk1, hdisk2 และ hdisk3 และอุปกรณ์เป้าหมาย คือ hdisk4, hdisk5 และ hdisk6 เมื่อต้องการเริ่มต้นและมอนิเตอร์การแคชของอุปกรณ์เป้าหมายให้ทำตามขั้นตอนต่อไปนี้:

1. สร้างแคชพูลบนหน่วยเก็บข้อมูล SSD

```
# cache_mgt pool create -d hdisk1,hdisk2,hdisk3 -p cmpool0
```
2. สร้างแคชพาร์ติชันขนาด 80 MB จากแคชพูล

```
# cache_mgt partition create -p cmpool0 -s 80M -P part1
```

3. กำหนดแคชพาร์ติชันให้กับดิสก์เป้าหมายที่คุณต้องการแคช

```
# cache_mgt partition assign -t hdisk4 -P part1  
# cache_mgt partition assign -t hdisk5 -P part1  
# cache_mgt partition assign -t hdisk6 -P part1
```

4. เริ่มต้นการแคชของอุปกรณ์เป้าหมาย

```
# cache_mgt cache start -t hdisk4  
# cache_mgt cache start -t hdisk5  
# cache_mgt cache start -t hdisk6
```

5. มอนิเตอร์สถิติเกี่ยวกับแคชฮิต

```
# cache_mgt monitor get -h -s
```

ข้อมูลที่เกี่ยวข้อง:

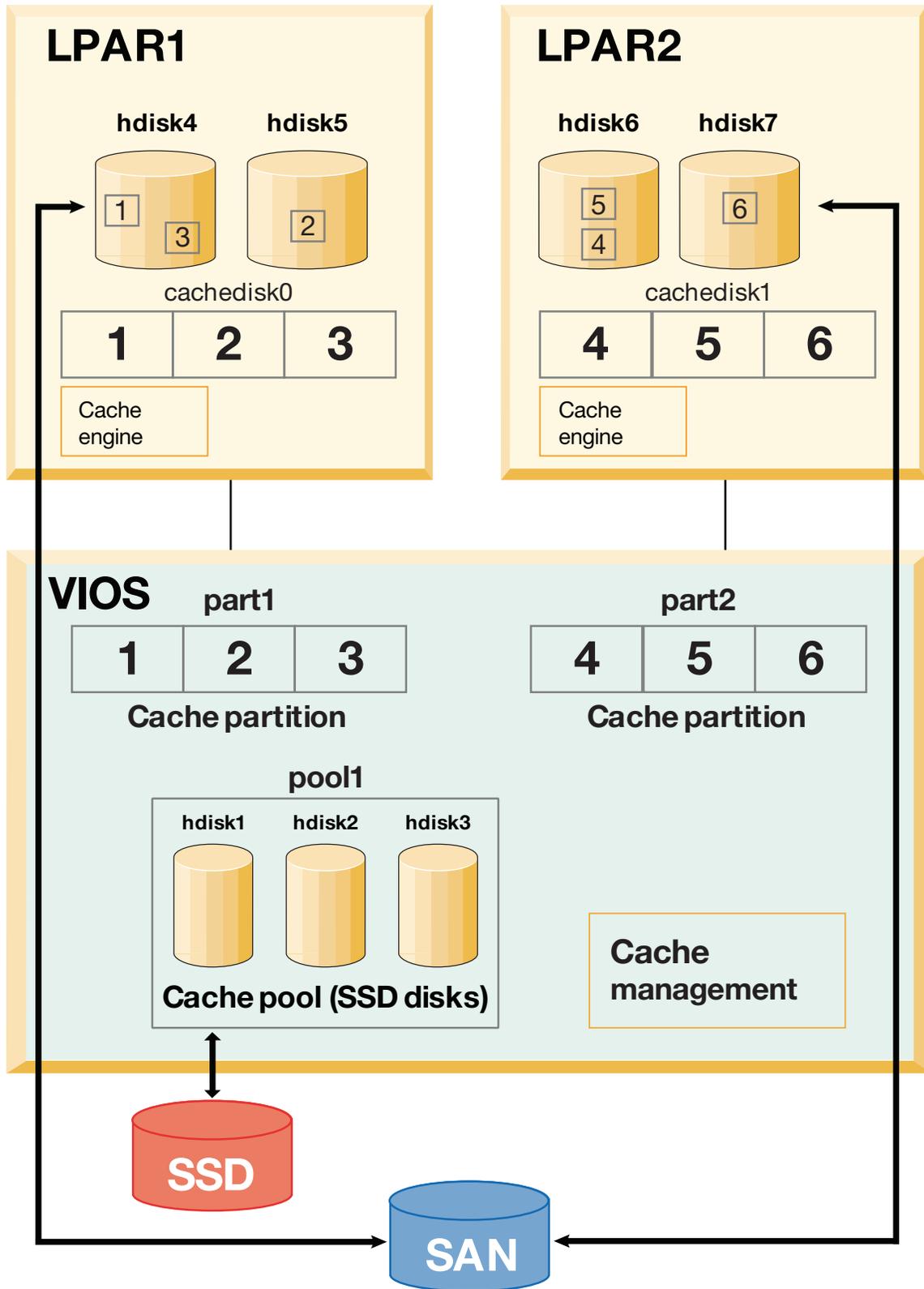
คำสั่ง cache_mgt

การแคชข้อมูลหน่วยเก็บข้อมูลในโหนดเสมือน:

ในโหนดเสมือน อุปกรณ์แคชจะถูกกำหนดให้กับ Virtual I/O Server (VIOS)

ในโหนดเสมือน แคชพูลถูกสร้างขึ้นบน VIOS จากนั้น แคชพูลจะถูกแยกเป็นพาร์ติชัน บน VIOS แต่ละแคชพาร์ติชันสามารถกำหนดให้กับอะแดปเตอร์โฮสต์เสมือน (vhost) เมื่อ แคชพาร์ติชันถูกพบบนโลจิคัลพาร์ติชัน AIX (LPAR) แคชพาร์ติชันสามารถใช้สำหรับการแคชอุปกรณ์เป้าหมายได้ แคชพาร์ติชันสามารถโอนย้ายไปยังเซิร์ฟเวอร์อื่นได้ เนื่องจากอุปกรณ์แคชเป็นอุปกรณ์เสมือน ก่อนการโอนย้าย การแคช จะหยุดทำงานโดยอัตโนมัติบน LPAR ต้นทาง เนื่องจากเป็นส่วนหนึ่งของการดำเนินการโอนย้าย แคชพาร์ติชัน ที่มีขนาดเดียวกันจะถูกสร้างขึ้นโดยอัตโนมัติบน VIOS เป้าหมาย หากมีการติดตั้งซอฟต์แวร์การแคชและ แคชพูลพร้อมใช้งานบน VIOS เป้าหมาย ระหว่างการโอนย้าย แคชพาร์ติชันจะถูกทำให้ พร้อมใช้งานสำหรับ LPAR เมื่อการโอนย้ายเสร็จสมบูรณ์ การแคชจะเริ่มทำงานโดยอัตโนมัติบน LPAR ปลายทาง ในกรณีนี้ การแคชจะเริ่มทำงานในสถานะว่างเปล่า (ไม่เต็มข้อมูล)

รูปต่อไปนี้จะแสดงตัวอย่างของการกำหนดคอนฟิกการแคชในโหนดเสมือน ที่อุปกรณ์แคช อยู่บน VIOS LPAR และอุปกรณ์เป้าหมายอยู่บน AIX LPAR



รูปที่ 16. การแคชข้อมูลหน่วยเก็บข้อมูล: คอนฟิกรัชันสำหรับอุปกรณ์แคชเสมือน

พิจารณาอุปกรณ์แคช คือ hdisk1, hdisk2 และ hdisk3 (บน VIOS LPAR) และอุปกรณ์เป้าหมาย คือ hdisk4 และ hdisk5 (บน AIX LPAR) เมื่อต้องการเริ่มต้นและมอนิเตอร์การแคชของอุปกรณ์เป้าหมายให้ทำตามขั้นตอนต่อไปนี้:

1. ใน VIOS LPAR สร้างแคชพูลโดยใช้หน่วยเก็บข้อมูล SSD
cache_mgt pool create -d hdisk1,hdisk2,hdisk3 -p cmpool0
2. ใน VIOS LPAR สร้างแคชพาร์ติชันขนาด 80 MB จากแคชพูล
cache_mgt partition create -p cmpool0 -s 80M -P part1
3. ใน VIOS LPAR กำหนดพาร์ติชันให้กับอะแดปเตอร์ไฮสปีดเสมือน
cache_mgt partition assign -P part1 -v vhost0
4. ใน AIX LPAR กำหนดแคชพาร์ติชันให้กับ อุปกรณ์เป้าหมาย
cache_mgt partition assign -t hdisk4 -P cachedisk0
cache_mgt partition assign -t hdisk5 -P cachedisk0
5. ใน AIX LPAR เริ่มต้นการแคชของ อุปกรณ์เป้าหมาย
cache_mgt cache start -t hdisk4
cache_mgt cache start -t hdisk5
6. ใน AIX LPAR ตรวจสอบสถานะเกี่ยวกับแคชฮิต
cache_mgt monitor get -h -s

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง cache_mgt

คำสั่ง cache_mgt บน VIOS

การแคชข้อมูลหน่วยเก็บข้อมูลในโหมด NPIV:

ในโหมดนี้ อุปกรณ์แคชจะพร้อมใช้งานเป็นอุปกรณ์ Fibre Channel (N_Port ID Virtualization) เสมือนบนโลจิคัลพาร์ติชันของ AIX (LPAR)

คุณต้องสร้างแคชพูล และจากนั้นแคชพาร์ติชันบน AIX LPAR จะมีเพียงแคชพาร์ติชันเดียวเท่านั้นที่สามารถสร้างบน AIX LPAR คุณสามารถใช้แคชพาร์ติชันเพื่อแคชอุปกรณ์เป้าหมายจำนวนเท่าใดก็ได้บน AIX LPAR LPAR สามารถโอนย้ายไปยังเซิร์ฟเวอร์อื่นได้เนื่องจาก อุปกรณ์แคชพร้อมใช้งานจาก storage area network (SAN) อุปกรณ์แคชต้องทำให้สามารถเห็นได้บนระบบปลายทาง การดำเนินการการแคชสามารถทำงานต่อระหว่างกระบวนการโอนย้าย และ แคชจะถูกเติมข้อมูลหลังจากการโอนย้ายเสร็จสมบูรณ์

การแคชอุปกรณ์เป้าหมายในโหมด NPIV เหมือนกับการแคชข้อมูลหน่วยเก็บข้อมูลในโหมดที่แยกต่างหาก ยกเว้นอุปกรณ์แคชจะพร้อมใช้งานจาก SAN อย่างไรก็ตาม ขั้นตอนในการแคชอุปกรณ์เป้าหมายจะเหมือนกับการแคชข้อมูลหน่วยเก็บข้อมูลในโหมดที่แยกต่างหาก

การจัดการแคชข้อมูลหน่วยเก็บข้อมูล

แม้ว่า การแคชจะถูกกำหนดค่าไว้แล้ว ข้อกำหนดสำหรับการแคชอาจเปลี่ยนแปลงได้เมื่อเวลาผ่านไป คุณอาจต้องการเพิ่มเวิร์กโหลดใหม่ที่จะถูกแคช เพื่อให้เป็นไปตามความต้องการการเปลี่ยนแปลง แคชพูล สามารถขยายได้โดยใช้อุปกรณ์แคชเพิ่มเติม แคชพาร์ติชันใหม่สามารถสร้างในพูลที่มีอยู่ หรือสามารถเพิ่มขนาดของพาร์ติชันที่มีอยู่ได้

คุณสามารถใช้ตัวอย่างต่อไปนี้เพื่อจัดการการกำหนดคอนฟิกการแคช:

- เมื่อต้องการเพิ่มอุปกรณ์แคชเข้ากับพูล ให้ป้อนคำสั่งต่อไปนี้:

```
# cache_mgt pool extend -p pool1 -d hdisk4 -f
```

แฟล็ก -f จะแทนที่การใช้งานดิสก์ที่มีอยู่ (hdisk4) หากดิสก์มีกลุ่มวอลุ่ม อยู่แล้ว

- เมื่อต้องการสร้างพาร์ติชันสำหรับเวิร์กโพลิตใหม่ที่มีขนาด 100 MB ให้ป้อนคำสั่งต่อไปนี้:

```
# cache_mgt partition create -p pool1 -s 100M -P part2
```

- เมื่อต้องการเพิ่มขนาดของพาร์ติชันที่มีอยู่อีก 20 MB ให้ป้อนคำสั่งต่อไปนี้:

```
# cache_mgt partition extend -p part1 -s 20M
```

ข้อควรพิจารณาเกี่ยวกับความพร้อมใช้งานสูง:

หากอุปกรณ์เป้าหมายซึ่งถูกแคช เป็นส่วนหนึ่งของกลุ่มรีซอร์สที่ได้รับการจัดการใน คลัสเตอร์ความพร้อมใช้งานสูง การดำเนินการ failover ต้องได้รับการวางแผนอย่างเหมาะสม

การแคชสามารถเปิดใช้งานได้ที่ละโหนดเท่านั้น ก่อนที่จะเหตุการณ์ failover จะเริ่มต้น คุณต้องแน่ใจว่าการดำเนินการการแคชถูกปิดใช้งานบนระบบต้นทาง หลังจาก failover ไปยังระบบใหม่เสร็จสมบูรณ์ คุณต้องเปิดใช้งานซอฟต์แวร์การแคช

เมื่อต้องการเปิดใช้งานซอฟต์แวร์การแคชให้ทำตามขั้นตอนต่อไปนี้:

1. หยุดการแคชบนระบบต้นทาง

```
# cache_mgt cache stop -t hdisk2
```

2. เริ่มต้นการแคชบนระบบใหม่หลังจากการกู้คืนจากความล้มเหลวเสร็จสมบูรณ์

```
# cache_mgt cache start -t hdisk2
```

ชื่อล็อกอิน, IDs ระบบ, และรหัสผ่าน

ระบบปฏิบัติการต้องทราบว่าคุณเป็นใคร เพื่อนำเสนอสถานะแวดล้อมที่ถูกต้องให้คุณ

เพื่อระบุตัวคุณเองให้ระบบปฏิบัติการรับทราบให้ล็อกอิน โดยพิมพ์ **ชื่อล็อกอิน** ของคุณ (รู้จักในอีกชื่อหนึ่งว่า ID ผู้ใช้หรือชื่อผู้ใช้) และ **รหัสผ่าน** รหัสผ่านอยู่ในรูปแบบของการรักษาความปลอดภัย บุคคลที่ทราบชื่อล็อกอินของคุณ ไม่สามารถล็อกอินเข้าสู่ระบบของคุณได้ ยกเว้นว่าพวกเขาทราบรหัสผ่านของคุณ

หากระบบมีการตั้งค่าเป็นระบบหลายผู้ใช้ ผู้ใช้ที่ได้รับอนุญาตแต่ละราย จะมีแอคเคาต์ รหัสผ่าน และชื่อล็อกอินบนระบบ ระบบปฏิบัติการเก็บประวัติของรีซอร์สที่ใช้โดยผู้ใช้แต่ละราย ซึ่ง เรียกว่าเป็น **บัญชีระบบ** ผู้ใช้แต่ละรายจะได้รับมอบพื้นที่ส่วนตัว ในพื้นที่ว่างหน่วยเก็บของระบบ ที่เรียกว่า **ระบบไฟล์** เมื่อคุณ ล็อกอิน ระบบไฟล์ปรากฏขึ้นพร้อมกับไฟล์ของคุณเท่านั้น แม้ว่ามีไฟล์อื่น นับพันไฟล์อยู่บนระบบ

สามารถมีชื่อล็อกอินที่ต้องการมากกว่าหนึ่งชื่อ บนระบบได้ ถ้าคุณต้องการเปลี่ยนจากชื่อล็อกอินหนึ่งเป็นอีกชื่อหนึ่ง คุณไม่ต้อง ล็อกเอาต์ออกจากระบบ แต่คุณสามารถใช้ชื่อล็อกอินอื่นพร้อมกัน ในเซลล์อื่น หรือใช้อย่างต่อเนื่องในเซลล์เดียวกันได้ โดยไม่ต้องล็อกเอาต์ นอกจากนี้ ถ้าระบบของคุณเป็นส่วนหนึ่งของเครือข่ายที่มี การเชื่อมต่อกับระบบอื่น คุณสามารถล็อกอินเข้าสู่ระบบอื่นใดซึ่งคุณ มีชื่อล็อกอินได้ด้วย ลักษณะนี้เรียกว่า **รีโมตล็อกอิน**

เมื่อคุณทำงานบนระบบปฏิบัติการเสร็จแล้ว ให้ล็อกเอาต์เพื่อให้มั่นใจว่าไฟล์และข้อมูลของคุณปลอดภัย

การล็อกอินเข้าสู่ระบบปฏิบัติการ

เพื่อใช้ระบบปฏิบัติการ ระบบของคุณต้องกำลังรัน และคุณต้องล็อกอินอยู่ เมื่อคุณล็อกอินเข้าสู่ระบบปฏิบัติการ คุณระบุตัวคุณเองให้ระบบรับทราบและอนุญาตให้ระบบตั้งค่าสถานะแวดล้อมของคุณ

ระบบของคุณอาจมีการตั้งค่าเพื่อให้คุณสามารถ ล็อกอินได้ในระหว่างบางชั่วโมงของวันและบางวันของสัปดาห์เท่านั้น หากคุณพยายามล็อกอินในเวลาอื่นนอกเหนือจากเวลาที่ได้รับอนุญาต การเข้าถึง จะถูกปฏิเสธ ผู้ดูแลระบบสามารถตรวจสอบเวลาล็อกอินของคุณได้

คุณ ล็อกอินที่ล็อกอินพร้อมท์ เมื่อคุณล็อกอินเข้าสู่ระบบปฏิบัติการ คุณจะถูกวางไว้ในไต่เร็กทอรีโฮม ของคุณ (เรียกอีกอย่างว่า *ไต่เร็กทอรีล็อกอิน*) โดยอัตโนมัติ

หลังจากเปิดระบบ ให้ล็อกอินเข้าสู่ระบบ เพื่อเริ่มต้นเซสชัน

1. พิมพ์ชื่อล็อกอินของคุณต่อจากพร้อมท์ **login::**

login: *LoginName*

ตัวอย่าง เช่น ถ้าชื่อล็อกอินของคุณคือ denise:

login: denise

2. ถ้าพร้อมท์ **password:** แสดงขึ้น ให้พิมพ์รหัสผ่านของคุณ (หน้าจอไม่แสดงรหัสผ่านของคุณ ในขณะที่คุณพิมพ์)

password: [your password]

หากพร้อมท์รหัสผ่านไม่แสดงขึ้น คุณไม่ได้ กำหนดรหัสผ่าน คุณสามารถเริ่มต้นการทำงานในระบบปฏิบัติการได้

หาก เครื่องของคุณไม่เปิด ให้ทำต่อไปนีก่อนคุณล็อกอิน:

1. ตั้งค่าสวิตช์เปิด/ปิดของแต่ละอุปกรณ์ที่ต่อพ่วงไปที่ On
2. เริ่มต้นยูนิตรระบบโดยการตั้งค่าสวิตช์เปิด/ปิดไปที่ On (I)
3. ดูที่การแสดงผลตัวเลขสามตัว เมื่อการทดสอบตัวเองเสร็จสมบูรณ์โดยไม่มี ข้อผิดพลาด การแสดงผลตัวเลขสามตัวจะว่างเปล่า

หากเกิดข้อผิดพลาดที่จำเป็นต้องแก้ไข โค้ด ตัวเลขสามตัวจะคงอยู่ และยูนิตรระบบหยุด โปรดติดต่อผู้ดูแลระบบของคุณเพื่อขอ ข้อมูลเกี่ยวกับโค้ดระบุความผิดพลาดและวิธีแก้ไข

เมื่อ การทดสอบตัวเองเสร็จเรียบร้อยแล้ว ล็อกอินพร้อมท์คล้ายกับตัวอย่างต่อไปนี้ จะแสดงขึ้นบนหน้าจอ:

login:

หลังจากคุณล็อกอินแล้ว ระบบจะเริ่มต้นในอินเทอร์เฟซบรรทัดคำสั่ง (เชลล์) หรืออินเทอร์เฟซเชิงกราฟิก (ตัวอย่างเช่น AIXwindows หรือ Common Desktop Environment (CDE)) อย่างใดอย่างหนึ่ง ขึ้นอยู่กับวิธีการตั้งค่าระบบปฏิบัติการของคุณ

หากคุณมีคำถามเกี่ยวกับ คอนฟิกูเรชันของรหัสผ่านหรือชื่อผู้ใช้ โปรดปรึกษาผู้ดูแลระบบของคุณ

การล็อกอินมากกว่าหนึ่งครั้ง (คำสั่ง login)

หากคุณกำลังทำงานในมากกว่าหนึ่งโครงการและคุณต้องการรักษา แอคเคาต์แยกต่างหากกัน คุณสามารถมีล็อกอินพร้อมท์มากกว่าหนึ่งรายการได้ คุณทำเช่นนี้ โดยใช้ชื่อล็อกอินเดียวกัน หรือโดยใช้ชื่อล็อกอินที่แตกต่างกันเพื่อล็อกอินเข้าสู่ระบบของคุณ

หมายเหตุ: แต่ละระบบมีจำนวนสูงสุดของชื่อล็อกอิน ที่สามารถใช้งานได้ในเวลาที่กำหนด จำนวนนี้ถูกกำหนดโดยข้อตกลงใบอนุญาต และแตกต่างกันไปในแต่ละการติดตั้ง

ตัวอย่าง เช่น ถ้าคุณล็อกอินเป็น denise1 แล้วและ ชื่อล็อกอินอื่นของคุณคือ denise2 ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:
login denise2

ถ้าพร้อมท์ password: แสดงขึ้น ให้พิมพ์รหัสผ่านของคุณ (หน้าจอไม่แสดงรหัสผ่านของคุณ ในขณะที่คุณพิมพ์) ขณะนี้ คุณมีสองล็อกอินที่กำลังรันบนระบบ

ให้ดูที่คำสั่ง login ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

การกลายเป็นผู้ใช้อื่นบนระบบ (คำสั่ง su)

คุณสามารถเปลี่ยน ID ผู้ใช้ที่เชื่อมโยงกับเซสชันได้ (ถ้าคุณทราบชื่อล็อกอิน ของผู้ใช้นั้น) โดยใช้คำสั่ง su (switch user)

ตัวอย่างเช่น ถ้าคุณต้องการสลับเป็น ผู้ใช้ joyce ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
su joyce
```

ถ้าพร้อมท์รหัสผ่าน: แสดงขึ้น ให้พิมพ์รหัสผ่านของผู้ใช้ joyce ขณะนี้ ID ผู้ใช้ของคุณคือ joyce หาก你不ทราบรหัสผ่าน คำร้องขอ จะถูกปฏิเสธ

เมื่อต้องการตรวจสอบว่า ID ผู้ใช้ของคุณคือ joyce ให้ใช้คำสั่ง id

หลักการที่เกี่ยวข้อง:

“การแสดง ID ผู้ใช้” ในหน้า 644

เมื่อต้องการแสดงหมายเลขประจำระบบ (IDs) สำหรับผู้ใช้ที่ระบุ ให้ใช้คำสั่ง id IDs ระบบคือหมายเลขที่ระบุ ผู้ใช้และกลุ่มผู้ใช้งานบนระบบ

ข้อมูลที่เกี่ยวข้อง:

ไวยากรณ์คำสั่ง su

การยับยั้งข้อความล็อกอิน

หลังจากล็อกอินเสร็จเรียบร้อยแล้ว คำสั่ง login แสดงข้อความของวัน วันที่และเวลาของความพยายามล็อกอินที่สำเร็จและไม่สำเร็จครั้งล่าสุดสำหรับผู้ใช้นี้ และจำนวนทั้งหมดของความพยายามล็อกอินที่ไม่สำเร็จสำหรับผู้ใช้นี้ ตั้งแต่การเปลี่ยนแปลงข้อมูลการพิสูจน์ตัวตนครั้งล่าสุด (โดยปกติคือรหัสผ่าน) คุณสามารถยับยั้งข้อความเหล่านี้โดยการรวมไฟล์ .hushlogin ไว้ในไดเรกทอรีโฮมของคุณ

ที่พร้อมท์ในไดเรกทอรีโฮมของคุณ พิมพ์คำสั่ง ต่อไปนี้:

```
touch .hushlogin
```

คำสั่ง touch สร้างไฟล์ว่างชื่อ .hushlogin หาก ไม่มีอยู่ในครั้งถัดไปที่คุณล็อกอิน ข้อความล็อกอินทั้งหมดจะถูกยับยั้งไว้ คุณสามารถสั่งให้ระบบเก็บรักษาเฉพาะข้อความประจำวัน และระงับข้อความล็อกอินอื่น

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง touch

การล็อกเอาต์ของระบบปฏิบัติการ (คำสั่ง exit และ logout)

เมื่อต้องการล็อกเอาต์ออกจากระบบปฏิบัติการให้ทำอย่างใดอย่างหนึ่งต่อไปนี้ ที่พร้อมท์ระบบ

กดลำดับคีย์ควบคุมการสิ้นสุดไฟล์ (คีย์ Ctrl-D)

หรือ

พิมพ์ exit

หรือ

พิมพ์ logout

หลังจาก คุณล็อกเอาต์ ระบบแสดงพร้อมต์ login:

การแสดงผล ID ผู้ใช้

เมื่อต้องการแสดงหมายเลขประจำระบบ (IDs) สำหรับผู้ใช้ที่ระบุ ให้ใช้คำสั่ง id IDs ระบบคือหมายเลขที่ระบุ ผู้ใช้และกลุ่มผู้ใช้งานระบบ

คำสั่ง id แสดงข้อมูลต่อไปนี้ เมื่อใช้ได้:

- ชื่อผู้ใช้และ ID ผู้ใช้จริง
- ชื่อของกลุ่มผู้ใช้และ ID กลุ่มจริง
- ชื่อของกลุ่มเสริมของผู้ใช้และ IDs กลุ่มเสริม ถ้ามี

ตัวอย่างเช่น ที่พร้อมต์ให้พิมพ์ดังต่อไปนี้:

```
id
```

ระบบแสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
uid=1544(sah) gid=300(build) euid=0(root) egid=9(printq) groups=0(system),10(audit)
```

ในตัวอย่างนี้ ผู้ใช้มีชื่อผู้ใช้ sah ที่มีหมายเลข ID เป็น 1544; ชื่อกลุ่มหลัก build ที่มี หมายเลข ID เป็น 300; ชื่อผู้ใช้ที่มีประสิทธิภาพ root ที่มี หมายเลข ID เป็น 0; ชื่อกลุ่มที่มีประสิทธิภาพ printq ที่มี หมายเลข ID เป็น 9; และชื่อกลุ่มเสริมสองชื่อคือ system และ audit ที่มีหมายเลข ID เป็น 0 และ 10 ตามลำดับ

ตัวอย่างเช่น ที่พร้อมต์ให้พิมพ์ดังต่อไปนี้:

```
id denise
```

ระบบแสดงข้อมูลคล้ายกับตัวอย่างต่อไปนี้:

```
uid=2988(denise) gid=1(staff)
```

ในตัวอย่างนี้ ผู้ใช้ denise มี หมายเลข ID เป็น 2988 และมีชื่อกลุ่มหลักเพียงชื่อเดียวคือ staff ที่มี หมายเลข ID เป็น 1

ให้ดูที่คำสั่ง id ใน *Commands Reference, Volume 3* สำหรับ ไวยากรณ์ทั้งหมด

งานที่เกี่ยวข้อง:

“การกลายเป็นผู้ใช้อื่นบนระบบ (คำสั่ง su)” ในหน้า 643

คุณสามารถเปลี่ยน ID ผู้ใช้ที่เชื่อมโยงกับเซสชันได้ (ถ้าคุณทราบชื่อล็อกอิน ของผู้ใช้นั้น) โดยใช้คำสั่ง su (switch user)

การแสดงชื่อล็อกอินของคุณ (คำสั่ง whoami และ logname):

เมื่อคุณมีมากกว่าหนึ่งล็อกอินพร้อมกัน อาจง่ายที่จะสูญเสียแทร็ก ของชื่อล็อกอิน โดยเฉพาะอย่างยิ่ง ชื่อล็อกอินซึ่งคุณใช้อยู่ ใน ปัจจุบัน คุณสามารถใช้คำสั่ง whoami และ logname เพื่อแสดงข้อมูลนี้

การใช้คำสั่ง whoami

เมื่อต้องการกำหนดชื่อล็อกอินที่กำลังใช้อยู่ที่พร้อมตัวให้พิมพ์ดังต่อไปนี้:

```
whoami
```

ระบบแสดงข้อมูลคล้ายกับตัวอย่าง ต่อไปนี้:

```
denise
```

ใน ตัวอย่างนี้ ชื่อล็อกอินที่กำลังใช้อยู่คือ denise

ให้ดูที่คำสั่ง whoami ใน *Commands Reference, Volume 6* สำหรับ ไวยากรณ์ทั้งหมด

การใช้คำสั่ง whoami

ส่วนแปรผันของคำสั่ง who อย่างหนึ่งคือ คำสั่ง whoami ช่วยให้คุณสามารถแสดงชื่อล็อกอิน ชื่อเทอร์มินัล และเวลา การล็อกอิน ที่พร้อมตัวให้พิมพ์ดังต่อไปนี้:

```
whoami
```

ระบบแสดงข้อมูลคล้ายกับตัวอย่าง ต่อไปนี้:

```
denise pts/0 Jun 21 07:53
```

ใน ตัวอย่างนี้ ชื่อล็อกอินคือ denise ชื่อของเทอร์มินัล คือ pts/0 และผู้ใช้นี้ล็อกอินเมื่อเวลา 7:53 a.m. ในวันที่ June 21

ให้ดูที่คำสั่ง who ใน *Commands Reference, Volume 6* สำหรับ ไวยากรณ์ทั้งหมด

การใช้คำสั่ง logname

ส่วนแปรผันอีกอย่างหนึ่งของคำสั่ง who คือ คำสั่ง logname แสดงข้อมูลเดียวกันกับคำสั่ง who

ที่พร้อมตัวให้พิมพ์ดังต่อไปนี้:

```
logname
```

ระบบแสดงข้อมูลคล้ายกับตัวอย่าง ต่อไปนี้:

```
denise
```

ใน ตัวอย่างนี้ ชื่อล็อกอินคือ denise

การแสดงชื่อระบบปฏิบัติการ (คำสั่ง `uname`):

เมื่อต้องการแสดงชื่อของระบบปฏิบัติการให้ใช้คำสั่ง `uname`

ตัวอย่างเช่น ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
uname
```

ระบบแสดงข้อมูลคล้ายกับตัวอย่าง ต่อไปนี้:

```
AIX
```

ในตัวอย่างนี้ชื่อระบบปฏิบัติการคือ AIX

ให้ดูที่คำสั่ง `uname` ใน *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงชื่อระบบของคุณ (คำสั่ง `uname`):

เมื่อต้องการแสดงชื่อระบบของคุณถ้าคุณอยู่บนเครือข่ายให้ใช้ คำสั่ง `uname` พร้อมด้วยแฟล็ก `-n` ชื่อระบบของคุณ ระบบระบบของคุณบนเครือข่าย ไม่เหมือนกับ ID ล็อกอิน ของคุณ

ตัวอย่างเช่น ที่พร้อมท์ให้พิมพ์ดังต่อไปนี้:

```
uname -n
```

ระบบแสดงข้อมูลคล้ายกับตัวอย่าง ต่อไปนี้:

```
barnard
```

ใน ตัวอย่างนี้ชื่อระบบคือ barnard

ให้ดูที่ คำสั่ง `uname` ในคู่มือ *Commands Reference, Volume 5* สำหรับ ไวยากรณ์ทั้งหมด

การแสดงผลผู้ใช้ทั้งหมดที่ล็อกอิน:

เมื่อต้องการแสดงข้อมูลเกี่ยวกับผู้ใช้ทั้งหมดที่ล็อกอินเข้าสู่ระบบโลคัล ในปัจจุบัน ให้ใช้คำสั่ง `who`

ข้อมูลต่อไปนี้แสดงขึ้น: ชื่อ ล็อกอิน ชื่อระบบ และวันที่และเวลาของล็อกอิน

หมายเหตุ: คำสั่งนี้ระบุเฉพาะ ผู้ใช้ที่ล็อกอินบนโลคัลโหนดเท่านั้น

เมื่อต้องการ แสดงข้อมูลเกี่ยวกับบุคคลที่กำลังใช้ โหนดระบบโลคัล ให้พิมพ์ดังต่อไปนี้:

```
who
```

ระบบแสดงข้อมูล คล้ายกับตัวอย่างต่อไปนี้:

```
joe 1ft/0 Jun 8 08:34
```

```
denise pts/1 Jun 8 07:07
```

ในตัวอย่างนี้ ผู้ใช้ joe บนเทอร์มินัล 1ft/0 ล็อกอิน เมื่อเวลา 8:34 a.m. ในวันที่ 8 มิถุนายน

โปรดดูที่คำสั่ง who

รหัสผ่าน

รหัสผ่านที่ไม่ซ้ำกันช่วยรักษาความปลอดภัยไฟล์ของคุณในบางระบบ

ระบบของคุณเชื่อมโยงรหัสผ่านกับแต่ละแอคเคาต์ การรักษาความปลอดภัยเป็นส่วนที่สำคัญของระบบคอมพิวเตอร์เนื่องจากช่วยป้องกันไม่ให้บุคคลที่ไม่ได้รับอนุญาตเข้าถึง ระบบและยังเกี่ยวกับไฟล์ของผู้ใช้อื่น การรักษาความปลอดภัยยังอาจมอบสิทธิเฉพาะให้แก่ผู้ใช้บางราย เช่น คำสั่งที่ผู้ใช้สามารถใช้ได้ และไฟล์ที่ผู้ใช้สามารถเข้าถึงได้ เพื่อป้องกัน ผู้ดูแลระบบบางราย อนุญาตให้ผู้ใช้เข้าถึงได้เฉพาะบางคำสั่งหรือไฟล์เท่านั้น

คำแนะนำรหัสผ่าน:

คุณควรมีรหัสผ่านที่ไม่ซ้ำกัน *ไม่ควรมีการแบ่งใช้รหัสผ่าน* ปกป้อง รหัสผ่านเหมือนกับที่คุณปกป้องสินทรัพย์อื่นของบริษัท เมื่อสร้างรหัสผ่าน ตรวจสอบให้แน่ใจว่า รหัสผ่านคาดเดาได้ยาก แต่ไม่ยากเกินจนคุณต้องเขียนออกมา เพื่อจดจำ

การใช้รหัสผ่านที่คาดเดาได้ยากช่วยให้ ID ผู้ใช้ของคุณปลอดภัย รหัสผ่านที่สร้างขึ้นจากข้อมูลส่วนบุคคล เช่น ชื่อหรือวันเกิด เป็นรหัสผ่านที่ไม่ดี แม้แต่คำทั่วไป สามารถคาดเดาได้ง่าย

รหัสผ่านที่ดีต้องมีอย่างน้อยหกอักขระและมีอักขระที่ไม่ใช่ตัวอักษร ส่วนผสมของคำที่แปลกและคำซึ่งตั้งใจสะกดผิดยังเป็นตัวเลือกที่ดีเช่นกัน

หมายเหตุ: หากรหัสผ่านของคุณยากต่อการจดจำจนคุณต้องจดออกมา นั่นไม่ใช่รหัสผ่านที่ดี

ใช้คำแนะนำต่อไปนี้เมื่อเลือกรหัสผ่าน:

- อย่าใช้ ID ผู้ใช้ของคุณเป็นรหัสผ่าน อย่าใช้ในลำดับย้อนกลับ ต่อกันเป็นสองเท่า หรือมีฉนั้น แก่ไข
- อย่านำรหัสผ่านไปใช้ซ้ำ ระบบอาจถูกตั้งค่าให้ปฏิเสธการใช้รหัสผ่าน ซ้ำ
- อย่าใช้ชื่อของบุคคลใดๆ เป็นรหัสผ่านของคุณ
- อย่าใช้คำที่พบได้ในพจนานุกรมตรวจสอบการสะกดทางออนไลน์ เป็นรหัสผ่านของคุณ
- อย่าใช้รหัสผ่านที่สั้นกว่าหกอักขระ
- อย่าใช้คำลามก เนื่องจากเป็นคำแรกๆ ที่ถูกตรวจสอบเมื่อ คาดเดารหัสผ่าน
- ใช้รหัสผ่านที่จดจำได้ง่าย เพื่อให้คุณไม่ต้องจด รหัสผ่านนั้นออกมา
- ใช้รหัสผ่านที่ใช้ทั้งตัวอักษรและตัวเลขและมีทั้งตัวอักษรพิมพ์เล็ก และพิมพ์ใหญ่
- ใช้สองคำที่คั่นด้วยตัวเลขเป็นรหัสผ่าน
- ใช้รหัสผ่านที่สะกดได้ ซึ่งจดจำได้ง่ายขึ้น
- อย่าจดรหัสผ่านออกมา อย่างไรก็ตาม ถ้าคุณต้องจดรหัสผ่านออกมา ควรเก็บ รหัสผ่านไว้ในสถานที่ที่ปลอดภัย เช่น ตู้เก็บเอกสารที่ล็อกกุญแจ

การเปลี่ยนรหัสผ่าน (คำสั่ง passwd):

เมื่อต้องการเปลี่ยนรหัสผ่านของคุณ ให้ใช้คำสั่ง passwd

1. ที่พร้อมท์ ให้พิมพ์ดังต่อไปนี้:

passwd

หาก คุณยังไม่มีรหัสผ่าน ให้ข้ามขั้นตอนที่ 2

2. พร้อมต์ต่อไปนี้แสดงขึ้น:

```
Changing password for UserID
UserID's Old password:
```

คำร้องขอนี้ช่วยป้องกันไม่ให้ผู้ใช้ที่ไม่ได้รับอนุญาตเปลี่ยนรหัสผ่านของคุณในขณะที่คุณ ห่างจากระบบ พิมพ์รหัสผ่านปัจจุบันของคุณ และกด Enter

3. พร้อมต์ต่อไปนี้แสดงขึ้น:

```
UserID's New password:
```

พิมพ์รหัสผ่านใหม่ซึ่งคุณต้องการ และกด Enter

4. พร้อมต์ต่อไปนี้แสดงขึ้น เพื่อขอให้คุณพิมพ์รหัสผ่านใหม่ อีกครั้ง

```
Enter the new password again:
```

คำร้องขอนี้ช่วยป้องกันให้คุณตั้งคาร์รหัสผ่านเป็นสตริงที่พิมพ์ผิด ซึ่งคุณไม่สามารถพิมพ์ใหม่ได้

ให้ดูที่คำสั่ง `passwd` ใน *Commands Reference, Volume 4* สำหรับ ไวยากรณ์ทั้งหมด

การโมฆะรหัสผ่าน (คำสั่ง `passwd`):

ถ้าคุณไม่ต้องการพิมพ์รหัสผ่านในแต่ละครั้งที่คุณล็อกอิน ให้ตั้งคาร์รหัสผ่านของคุณเป็น `null` (ว่างเปล่า)

เมื่อต้องการตั้งคาร์รหัสผ่านของคุณเป็น `null` ให้พิมพ์ดังต่อไปนี้:

```
passwd
```

เมื่อคุณได้รับการพร้อมต์ให้ป้อนรหัสผ่านใหม่ ให้กด Enter หรือ Ctrl-D

คำสั่ง `passwd` ไม่ได้พร้อมต์ให้ป้อนรายการรหัสผ่าน อีกครั้ง ข้อความที่ตรวจสอบรหัสผ่าน `null` แสดงขึ้น

ให้ดูที่คำสั่ง `passwd` สำหรับข้อมูลเพิ่มเติม และไวยากรณ์ทั้งหมด

ข้อมูลสรุปคำสั่งสำหรับชื่อล็อกอิน, IDs ระบบ, และรหัสผ่าน

มีคำสั่งสำหรับการทำงานกับชื่อล็อกอิน, IDs ระบบ, และรหัสผ่าน

คำสั่งล็อกอินและล็อกเอาต์

ไอเท็ม	คำอธิบาย
login	เริ่มต้นเซสชันของคุณ
logout	หยุดโปรเซสทั้งหมดของคุณ
shutdown	สิ้นสุดการดำเนินงานระบบ
su	เปลี่ยน ID ผู้ใช้ที่เชื่อมโยงกับเซสชัน
touch	อัปเดตเวลาเข้าถึงและเวลาการแก้ไขของไฟล์ หรือสร้างไฟล์ว่าง

คำสั่งการระบุผู้ใช้และระบบ

ไอเท็ม	คำอธิบาย
id	แสดงการระบุระบบของผู้ใช้ที่ระบุ
logname	แสดงชื่อล็อกอิน
uname	แสดงชื่อของระบบปฏิบัติการปัจจุบัน
who	ระบุผู้ใช้ที่ล็อกอินอยู่ในปัจจุบัน
whoami	แสดงชื่อล็อกอินของคุณ

คำสั่งรหัสผ่าน

ไอเท็ม	คำอธิบาย
passwd	เปลี่ยนรหัสผ่านของผู้ใช้

Common Desktop Environment

ด้วย Common Desktop Environment (CDE) คุณสามารถเข้าถึงอุปกรณ์บนเครือข่าย และเครื่องมือโดยไม่ต้องทราบถึงตำแหน่งของอุปกรณ์เหล่านั้น คุณสามารถแลกเปลี่ยนข้อมูลข้ามแอ็พพลิเคชัน โดยการลากและวาง อ็อบเจกต์เหล่านั้น

หลายๆงานที่ต้องการไวยากรณ์บรรทัดรับคำสั่งซับซ้อนก่อนหน้านี้ สามารถทำได้ง่ายยิ่งขึ้น และคล้ายกับจากแพลตฟอร์มไปอีกแพลตฟอร์ม ตัวอย่างเช่น คุณสามารถกำหนดคอนฟิกแบบศูนย์กลาง และกระจายแอ็พพลิเคชันไปยังผู้ใช้ คุณยังสามารถจัดการการรักษาความปลอดภัย ความพร้อมใช้ และการทำงานร่วมกันของแอ็พพลิเคชัน สำหรับผู้ใช้ที่คุณสนับสนุน

หมายเหตุ: วอลุ่มวิธีใช้ Common Desktop Environment (CDE) 1.0, เอกสารคู่มือบนเว็บ และคู่มือฉบับพิมพ์อาจอ้างอิงไปยัง เดสก์ท็อปเป็น Common Desktop Environment, เดสก์ท็อป AIXwindows, CDE 1.0 หรือ เดสก์ท็อป

การเปิดใช้งานและการปิดใช้งานการเริ่มต้นอัตโนมัติของเดสก์ท็อป

คุณอาจพบว่าเป็นการสะดวกกว่าที่จะตั้งค่าระบบของคุณเพื่อเริ่มต้น Common Desktop Environment โดยอัตโนมัติเมื่อเปิดระบบ

คุณสามารถทำเช่นนี้ผ่านทาง System Management Interface Tool (SMIT) หรือจากบรรทัดรับคำสั่ง สิ่งจำเป็นต้องมี

คุณ ต้องมีสิทธิ์ผู้ดูแลระบบเพื่อเปิดใช้งานหรือปิดใช้งานการเริ่มต้นอัตโนมัติของเดสก์ท็อป

ศึกษาตารางต่อไปนี่เพื่อกำหนดวิธีการเปิดใช้งานหรือปิดใช้งาน การเริ่มต้นอัตโนมัติของเดสก์ท็อป

การเริ่มต้น และการหยุด Common Desktop Environment โดยอัตโนมัติ

ภารกิจ	พารามิเตอร์ SMIT	คำสั่งหรือไฟล์
การเปิดใช้งานการเริ่มต้นอัตโนมัติของเดสก์ท็อป ¹	smit dtconfig	/usr/dt/bin/dtconfig -e
การปิดใช้งานการเริ่มต้นอัตโนมัติของเดสก์ท็อป ¹	smit dtconfig	/usr/dt/bin/dtconfig -d

หมายเหตุ: รีสตาร์ทระบบหลังจากทำภารกิจนี้เสร็จสมบูรณ์แล้ว

การเริ่มต้นสภาวะแวดล้อมเดสก์ท็อปด้วยตนเอง

ใช้ไพรซีเดอร์นี้เพื่อเริ่มต้นสภาวะแวดล้อมเดสก์ท็อปด้วยตนเอง

1. ล็อกอินเข้าสู่ระบบของคุณเป็นราก
2. ที่บรรทัดคำสั่ง ให้พิมพ์ดังต่อไปนี้:

```
/usr/dt/bin/dtlogin -daemon
```

หน้าจอ เดสก์ท็อปล็อกอิน แสดงขึ้น เมื่อคุณล็อกอิน เดสก์ท็อปเซสชันจะเริ่มต้น

การหยุดสภาวะแวดล้อมเดสก์ท็อปด้วยตนเอง

เมื่อคุณหยุด Login Manager ด้วยตนเอง เซิร์ฟเวอร์ X และเดสก์ท็อป เซสชันทั้งหมดที่ Login Manager เริ่มต้นจะถูกหยุด

1. เปิดหน้าต่างโปรแกรมเลียนแบบเทอร์มินัล และล็อกอินเป็นราก
2. รับ ID โพรเซสของ Login Manager โดยพิมพ์ดังต่อไปนี้:

```
cat /var/dt/Xpid
```

3. หยุด Login Manager โดยพิมพ์:

```
kill -term process_id
```

การแก้ไขเดสก์ท็อปโปรไฟล์

เมื่อคุณล็อกอินเข้าสู่เดสก์ท็อป ไฟล์สภาวะแวดล้อมเซลล์ (.profile หรือ .login) ไม่มีการอ่านโดยอัตโนมัติ เนื่องจากเดสก์ท็อปรันเซิร์ฟเวอร์ X ก่อนคุณล็อกอิน ดังนั้นฟังก์ชันที่นำเสนอโดยไฟล์ .profile หรือไฟล์ .login ต้องมีการระบุโดย Login Manager ของเดสก์ท็อป

ตัวแปรสภาวะแวดล้อมเฉพาะผู้ใช้มีการตั้งค่าใน `/Home Directory/.dtprofile` เพิ่มเพลตสำหรับไฟล์นี้มีอยู่ใน `/usr/dt/config/sys.dtprofile` วางตัวแปรและคำสั่งเซลล์ไว้ใน `.dtprofile` ที่ใช้กับเดสก์ท็อปเท่านั้น เพิ่มบรรทัดที่ตอนท้ายของ `.dtprofile` เพื่อ ประสานไฟล์สภาวะแวดล้อมเซลล์

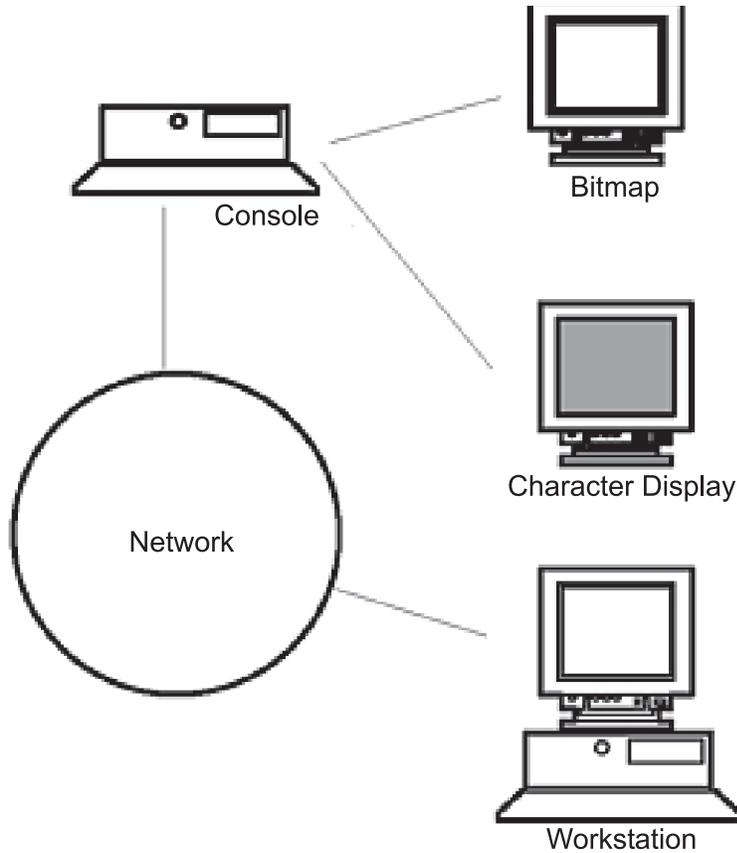
ตัวแปรสภาวะแวดล้อมทั่วทั้งระบบสามารถตั้งค่าได้ในไฟล์คอนฟิกูเรชัน Login Manager หากต้องการรายละเอียดเกี่ยวกับการกำหนดคอนฟิกตัวแปรสภาวะแวดล้อม ให้อ่าน *Common Desktop Environment 1.0: Advanced User's and System Administrator's Guide*

การเพิ่มและการลบจอแสดงผลและเทอร์มินัลสำหรับ Common Desktop Environment

คุณสามารถเพิ่มและลบจอแสดงผลและเทอร์มินัลสำหรับ Common Desktop Environment

Login Manager สามารถเริ่มต้นขึ้นจากระบบที่มีโลคัลบิตแม็พ หรือกราฟิกคอนโซลเดียว อย่างไรก็ตาม สามารถใช้สถานการณ์อื่นได้เช่นกัน (โปรดดูรูปภาพต่อไปนี้) คุณสามารถเริ่มต้น Common Desktop Environment จาก:

- โลคัลคอนโซล
- รีโมตคอนโซล
- ระบบบิตแม็พหรือเทอร์มินัล X แสดงอักขระที่กำลังรันบนระบบโฮสต์ บนเครือข่าย



รูปที่ 17. จุดอินเทอร์เฟซ CDE. ภาพสาริตนี้แสดงจุดเชื่อมต่อระหว่าง คอนโซล เครื่องข่าย จอแสดงผลบิตแม็พ จอแสดงผลอักขระ และเว็กรีสเตชัน

ระบบเทอร์มินัล X ประกอบด้วยอุปกรณ์จอแสดงผล คีย์บอร์ด และเมาส์ที่รันเซิร์ฟเวอร์ X เท่านั้น โคลเอ็นต์ รวมถึง Common Desktop Environment มีการรันบนระบบโฮสต์ตั้งแต่หนึ่งระบบขึ้นไปบนเครือข่าย เอาต์พุตจากโคลเอ็นต์ ถูกกำหนดทิศทางไปยังจอแสดงผลเทอร์มินัล X

ภารกิจคอนฟิกูเรชัน Login Manager ต่อไปนี้สนับสนุนคอนฟิกูเรชันที่เป็นไปได้จำนวนมาก:

- การลบจอแสดงผลโลคัล
- การเพิ่ม ASCII หรือเทอร์มินัลแสดงอักขระ

เมื่อต้องการใช้เว็กรีสเตชันเป็นเทอร์มินัล X ให้พิมพ์ดังต่อไปนี้ที่บรรทัด คำสั่ง:

```
/usr/bin/X11/X -query hostname
```

เซิร์ฟเวอร์ X ของเว็กรีสเตชันซึ่งทำหน้าที่เป็นเทอร์มินัล X ต้อง:

- สนับสนุน XDMCP และอ็อปชันบรรทัดคำสั่ง `-query`

- นำเสนอสิทธิอนุญาต xhost (ใน /etc/X*.hosts) ที่เทอร์มินัลโฮสต์

เมื่อต้องการลบจอแสดงผลโลคัล ให้ลบรายการของจอแสดงผลนั้นในไฟล์ Xservers ในไดเรกทอรี /usr/dt/config

เทอร์มินัลแสดง อักขระ หรือ เทอร์มินัล ASCII คือคอนฟิกูเรชันซึ่งเทอร์มินัล ไม่ใช่อุปกรณ์บิตแมพ

เมื่อต้องการเพิ่ม ASCII หรือคอนโซลแสดงอักขระ ถ้าไม่มีจอแสดงผลบิตแมพ ให้ปฏิบัติตามขั้นตอนต่อไปนี้:

1. ถ้าไม่มีไฟล์ /etc/dt/config/Xservers ให้คัดลอกไฟล์ /usr/dt/config/Xservers ไปยังไดเรกทอรี /etc/dt/config
2. หากคุณต้องคัดลอกไฟล์ Xservers ไปยัง /etc/dt/config ให้เปลี่ยนหรือเพิ่มบรรทัด Dtlogin.servers: ใน /etc/dt/config/Xconfig เป็น:

```
Dtlogin*servers: /etc/dt/config/Xservers
```

3. แสดงข้อคิดเห็นบรรทัดใน /etc/dt/config/Xservers ที่เริ่มต้นเซิร์ฟเวอร์ X

```
# * Local local@console /path/X :0
```

คำสั่งนี้ปิดใช้งาน เมนูล็อกอินอ้อพชัน

4. อ่านไฟล์คอนฟิกูเรชัน Login Manager

เมื่อต้องการเพิ่มคอนโซลแสดงอักขระถ้ามีจอแสดงผลบิตแมพอยู่ ให้ปฏิบัติตามขั้นตอนต่อไปนี้:

1. ถ้าไม่มีไฟล์ /etc/dt/config/Xservers ให้คัดลอกไฟล์ /usr/dt/config/Xservers ไปยังไดเรกทอรี /etc/dt/config
2. หากคุณต้องคัดลอกไฟล์ Xservers ไปยัง /etc/dt/config ให้เปลี่ยนหรือเพิ่มบรรทัด Dtlogin.servers: ใน /etc/dt/config/Xconfig เป็น:

```
Dtlogin*servers: /etc/dt/config/Xservers
```

3. แก้ไขบรรทัดใน /etc/dt/config/Xservers ที่เริ่มต้น เซิร์ฟเวอร์ X เป็น:

```
* Local local@none /path/X :0
```

4. อ่านไฟล์คอนฟิกูเรชัน Login Manager

การแสดงผลการกำหนดอุปกรณ์เองสำหรับ Common Desktop Environment

คุณสามารถกำหนดคอนฟิก Common Desktop Environment Login Manager เพื่อรันบนระบบที่มีอุปกรณ์แสดงผลตั้งแต่สองรายการขึ้นไป

เมื่อระบบมีหลายจอแสดงผล ต้องปฏิบัติตามข้อกำหนดคอนฟิกูเรชัน ต่อไปนี้:

- ต้องเริ่มต้นเซิร์ฟเวอร์บนแต่ละจอแสดงผล
- No Windows Mode must be configured for each display.

อาจจำเป็นหรือต้องการที่จะใช้รีซอร์ส dtlogin ที่แตกต่างกัน สำหรับแต่ละจอแสดงผล

และยังอาจจำเป็นหรือต้องการที่จะใช้ตัวแปรสภาวะแวดล้อมทั่วทั้งระบบที่แตกต่างกัน สำหรับแต่ละอุปกรณ์แสดงผล

การเริ่มต้นเซิร์ฟเวอร์บนอุปกรณ์จอแสดงผลแต่ละรายการ:

เริ่มต้นเซิร์ฟเวอร์บนอุปกรณ์จอแสดงผลแต่ละรายการโดยใช้โพธิ์ซีเตอร์นี้

1. ถ้าไม่มีไฟล์ `/etc/dt/config/Xservers` ให้คัดลอกไฟล์ `/usr/dt/config/Xservers` ไปยังไดเรกทอรี `/etc/dt/config`
2. หากคุณต้องคัดลอกไฟล์ `Xservers` ไปยัง `/etc/dt/config` ให้เปลี่ยนบรรทัด `Dtlogin.servers:` ใน `/etc/dt/config/Xconfig` เป็น:
`Dtlogin*servers: /etc/dt/config/Xservers`
3. แก้ไข `/etc/dt/config/Xservers` เพื่อเริ่มต้นเซิร์ฟเวอร์ X บนอุปกรณ์จอแสดงผลแต่ละรายการ

ไวยากรณ์ทั่วไปสำหรับการเริ่มต้นเซิร์ฟเวอร์คือ:

```
DisplayName DisplayClass DisplayType [ @ite ] Command
```

Only displays with an associated Internal Terminal Emulator (ITE) can operate in *No Windows Mode*. *No Windows Mode* temporarily disables the desktop for the display and runs a *getty* process if one is not already started. กระบวนการ *getty* เป็นโปรแกรม UNIX ที่ตั้งค่าชนิดเทอร์มินัลและใช้ในโปรเซสล็อกอิน

ซึ่งช่วยให้คุณสามารถล็อกอินและทำภารกิจที่ไม่สามารถทำได้ภายใต้ Common Desktop Environment เมื่อคุณล็อกเอาต์ จะมีการรีสตาร์ทเดสก์ทอปสำหรับอุปกรณ์จอแสดงผล If a *getty* process is not already running on a display device, Login Manager starts one when *No Windows Mode* is initiated.

ในดีฟอลต์คอนฟิกูเรชัน เมื่อ `ite` ถูกข้ามไป จะมีการเชื่อมโยง `display:0` เข้ากับ ITE (`/dev/console`)

การตั้งค่าจอแสดงผลอื่นเป็น ITE:

ใช้โพธิ์ซีเตอร์นี้เพื่อตั้งค่าจอแสดงผลอื่นเป็น ITE

เมื่อต้องการระบุจอแสดงผลอื่นเป็น ITE:

- บนจอแสดงผล ITE ให้ตั้งค่า ITE เป็นอุปกรณ์อักขระ
- บนจอแสดงผลอื่นทั้งหมด ให้ตั้งค่า ITE เป็น ไม่มี

ตัวอย่างต่อไปนี้แสดงรายการในไฟล์ `Xserver` ซึ่งเริ่มต้นเซิร์ฟเวอร์บนจอแสดงผลโลคัลสามจอบน `sysaaa:0` จอแสดงผล :0 จะเป็นคอนโซล (ITE)

```
sysaaa:0 Local local /usr/bin/X11/X :0
sysaaa:1 Local local /usr/bin/X11/X :1
sysaaa:2 Local local /usr/bin/X11/X :2
```

บนโฮสต์ `sysbbb` จอแสดงผล บิตแม็พ :0 ไม่ใช่ ITE; ITE เชื่อมโยงกับอุปกรณ์ `/dev/tty1` รายการต่อไปนี้ในไฟล์ `Xserver` เริ่มต้นเซิร์ฟเวอร์บน จอแสดงผลบิตแม็พสองจอที่ไม่มีโหมด Windows ที่เปิดใช้งานบน :1

```
sysaaa:0 Local local@none /usr/bin/X11/X :0
sysaaa:1 Local local@tty1 /usr/bin/X11/X :1
```

การตั้งค่าชื่อจอแสดงผลใน Xconfig:

คุณไม่สามารถใช้ไวยากรณ์ hostname:0 ปกติสำหรับ ชื่อจอแสดงผลใน /etc/dt/config/Xconfig

เมื่อต้องการระบุชื่อจอแสดงผลใน Xconfig:

- ใช้ขีดเส้นใต้แทนเครื่องหมายจุด
- ในชื่อโฮสต์ที่ถูกต้องแบบเต็ม ให้ใช้ขีดเส้นใต้แทนเครื่องหมายหัวภาค

ตัวอย่างต่อไปนี้แสดงเซตอ็อปของชื่อจอแสดงผลใน Xconfig:

```
Dtlogin.claaa_0.resource: value
Dtlogin.sysaaa_prsm_ld_edu_0.resource: value
```

การใช้รีซอร์ส Login Manager ที่แตกต่างกันสำหรับแต่ละจอแสดงผล:

เมื่อต้องการใช้รีซอร์ส Login Manager ที่แตกต่างกันสำหรับแต่ละจอแสดงผล ให้ปฏิบัติตามขั้นตอนต่อไปนี้:

1. ถ้าไม่มีไฟล์ /etc/dt/config/Xconfig ให้คัดลอกไฟล์ /usr/dt/config/Xconfig ไปยัง /etc/dt/config directory
2. แก้ไขไฟล์ /etc/dt/config/Xconfig เพื่อระบุ ไฟล์รีซอร์สที่แตกต่างกันสำหรับแต่ละจอแสดงผล ตัวอย่างเช่น:

```
Dtlogin.DisplayName.resources: path/file
```

โดยที่ *path* คือชื่อพาร ของไฟล์ Xresource ที่จะใช้ และ *file* คือชื่อไฟล์ ของไฟล์ Xresource ที่จะใช้

3. สร้างไฟล์รีซอร์สแต่ละไฟล์ซึ่งระบุไว้ในไฟล์ Xconfig ไฟล์ Xresources เฉพาะภาษาที่มีการติดตั้งใน /usr/dt/config/<LANG>
4. ในแต่ละไฟล์ ให้วางรีซอร์ส dtlogin สำหรับ จอแสดงผลนั้น

ตัวอย่างต่อไปนี้แสดงบรรทัดในไฟล์ Xconfig ซึ่งระบุไฟล์รีซอร์สที่แตกต่างกันสำหรับสามจอแสดงผล:

```
Dtlogin.sysaaa_0.resources: /etc/dt/config/Xresources0
Dtlogin.sysaaa_1.resources: /etc/dt/config/Xresources1
Dtlogin.sysaaa_2.resources: /etc/dt/config/Xresources2
```

การรันสคริปต์ที่แตกต่างกันสำหรับแต่ละจอแสดงผล:

ใช้ไพรซีเดอร์นี้เพื่อรันสคริปต์เฉพาะสำหรับจอแสดงผลเฉพาะ

1. ถ้าไม่มีไฟล์ /etc/dt/config/Xconfig ให้คัดลอกไฟล์ /usr/dt/config/Xconfig ไปยังไดเรกทอรี /etc/dt/config
2. ใช้รีซอร์สสตาร์ทอัพ รีเซ็ต และเซตอ็อปใน /etc/dt/config/Xconfig เพื่อระบุสคริปต์ที่แตกต่างกันสำหรับแต่ละจอแสดงผล (ไฟล์เหล่านี้จะรันแทน ไฟล์ Xstartup, Xreset, และ Xsetup):

```
Dtlogin*DisplayName*startup: /path/file
Dtlogin*DisplayName*reset: /path/file
Dtlogin*DisplayName*setup: /path/file
```

โดยที่ *path* คือชื่อพาร ของไฟล์ที่จะใช้ และ *file* คือชื่อไฟล์ ของไฟล์ที่จะใช้ สตาร์ทอัพสคริปต์มีการรันเป็นรากหลังจากผู้ใช้ล็อกอินแล้ว ก่อนเซสชัน Common Desktop Environment เริ่มต้น ขึ้น

สามารถใช้สคริปต์ /usr/dt/config/Xreset เพื่อ ย้อนกลับค่าติดตั้งที่ทำในไฟล์ Xstartup ไฟล์ Xreset รันเมื่อผู้ใช้ล็อกเอาต์

ตัวอย่าง ต่อไปนี้แสดงบรรทัดในไฟล์ Xconfig ซึ่งระบุ สคริปต์ที่แตกต่างกันสำหรับสองจอแสดงผล:

```
Dtlogin.sysaaa_0*startup: /etc/dt/config/Xstartup0
Dtlogin.sysaaa_1*startup: /etc/dt/config/Xstartup1
Dtlogin.sysaaa_0*setup: /etc/dt/config/Xsetup0
Dtlogin.sysaaa_1*setup: /etc/dt/config/Xsetup1
Dtlogin.sysaaa_0*reset: /etc/dt/config/Xreset0
Dtlogin.sysaaa_1*reset: /etc/dt/config/Xreset1
```

การตั้งค่าตัวแปรสถานะแวดล้อมทั่วทั้งระบบที่แตกต่างกันสำหรับแต่ละจอแสดงผล:

ใช้ไพรซีเตอร์นี้เพื่อกำหนดตัวแปรสถานะแวดล้อมทั่วทั้งระบบเอง สำหรับแต่ละจอแสดงผล

เมื่อต้องการตั้งค่าตัวแปรสถานะแวดล้อมทั่วทั้งระบบที่แตกต่างกันสำหรับแต่ละจอแสดงผล:

1. ถ้าไม่มีไฟล์ /etc/dt/config/Xconfig ให้คัดลอกไฟล์ /usr/dt/config/Xconfig ไปยังไดเรกทอรี /etc/dt/config
2. ตั้งค่ารีซอร์สสถานะแวดล้อมใน /etc/dt/config/Xconfig แยกต่างหากกันสำหรับแต่ละจอแสดงผล:

```
Dtlogin*DisplayName*environment: value
```

กฎต่อไปนี้ใช้กับตัวแปรสถานะแวดล้อมสำหรับแต่ละจอแสดงผล:

- แบ่งระหว่างการกำหนดตัวแปรแต่ละค่าด้วยช่องว่างหรือแท็บ
- อย่าใช้รีซอร์สสถานะแวดล้อมเพื่อตั้งค่า TZ และ LANG
- ไม่มีการประมวลผลเซลล์ภายในไฟล์ Xconfig

ตัวอย่างต่อไปนี้แสดงบรรทัดในไฟล์ Xconfig ซึ่งตั้งค่าตัวแปรสำหรับสองจอแสดงผล:

```
Dtlogin*syshere_0*environment:EDITOR=vi SB_DISPLAY_ADDR=0xB00000
Dtlogin*syshere_1*environment:EDITOR=emacs \
SB_DISPLAY_ADDR=0xB00000
```

การพิมพ์และงานพิมพ์

การพิมพ์ใน AIX จะมี การกำหนดคอนฟิกและอ็อปชันการตั้งค่าจำนวนมาก

ขึ้นอยู่กับเครื่องพิมพ์ของคุณ ระบบปฏิบัติการ AIX ควบคุมรูปลักษณะและคุณสมบัติของเอาต์พุตสุดท้าย เครื่องพิมพ์ไม่จำเป็นต้องอยู่ในบริเวณเดียวกันกับยูนิตรระบบ และคอนโซลระบบ คุณอาจเลือกเชื่อมต่อเครื่องพิมพ์ของคุณกับระบบโลคัลโดยตรง หรือสถานการณ์ของคุณอาจต้องการให้คุณส่งงานพิมพ์ผ่านเครือข่ายไปยัง ระบบแบบรีโมต

เมื่อต้องการจัดการงานพิมพ์ด้วยประสิทธิภาพสูงสุด ระบบปฏิบัติการ AIX จะวางงานแต่ละงานลงในคิว เพื่อรอสภาพพร้อมใช้ของเครื่องพิมพ์ ระบบสามารถประหยัดเอาต์พุตจาก หนึ่งไฟล์ขึ้นไปในคิว ในขณะที่เครื่องพิมพ์จัดทำเอาต์พุตจากไฟล์หนึ่ง ระบบจะประมวลผลงานถัดไปในคิว กระบวนการนี้ ทำต่อเนื่องไปจนกว่าแต่ละงานในคิวถูกพิมพ์ครบแล้ว

การเริ่มต้นงานพิมพ์

ใช้คำสั่ง qprt หรือ smit เพื่อร้องของานพิมพ์

- สำหรับงานพิมพ์โลโก้ เครื่องพิมพ์ต้องมีการต่อพ่วงเข้ากับระบบของคุณทาง ภายนอก หรือในกรณีของเครื่องพิมพ์เครือข่าย ต้องต่อพ่วงและกำหนดคอนฟิก บนเครือข่าย
- สำหรับงานพิมพ์รีโมต ระบบของคุณต้องมีการกำหนดคอนฟิกเพื่อสื่อสารกับ เซิร์ฟเวอร์การพิมพ์รีโมต
- ก่อนคุณสามารถพิมพ์ไฟล์ได้ คุณต้องมีสิทธิ์การอ่าน ในไฟล์นั้น เมื่อต้องการ ลบไฟล์หลังจากที่พิมพ์ไฟล์นั้นแล้ว คุณต้องมีสิทธิ์การเขียน ในไดเรกทอรีที่มีไฟล์

ระบุข้อมูลต่อไปนี้เพื่อร้องของานพิมพ์:

- ชื่อของไฟล์ที่จะพิมพ์
- ชื่อคิวงานพิมพ์
- จำนวนสำเนาที่จะพิมพ์
- จัดทำสำเนาของไฟล์บนรีโมตโฮสต์หรือไม่
- ลบไฟล์หลังจากการพิมพ์หรือไม่
- ส่งการแจ้งเตือนเกี่ยวกับสถานะงานหรือไม่
- ส่งการแจ้งเตือนเกี่ยวกับสถานะงานทางเมลระบบหรือไม่
- สถานะการส่งเป็นชุด
- ชื่อผู้ใช้สำหรับเลเบล "จัดส่งไปยัง"
- ข้อความการตอบรับคอนโซลสำหรับการพิมพ์รีโมต
- ข้อความการตอบรับไฟล์สำหรับการพิมพ์รีโมต
- ระดับของระดับความสำคัญ

ใช้คำสั่ง `qprt` เพื่อสร้างและจัดคิวงานพิมพ์ เพื่อพิมพ์ไฟล์ซึ่งคุณระบุ หากคุณระบุมากกว่าหนึ่งไฟล์ ไฟล์ทั้งหมด จะรวมเข้าด้วยกันเป็นหนึ่งงานพิมพ์ ไฟล์เหล่านี้มีการพิมพ์ในลำดับ ที่ระบุบนบรรทัดคำสั่ง

รูปแบบพื้นฐาน ของคำสั่ง `qprt` คือ:

`qprt -PQueueName FileName`

แฟล็กของคำสั่ง `qprt` ต่อไปนี้มีประโยชน์คือ:

ไอเท็ม	Descriptor
<code>-b Number</code>	ระบุขอบด้านล่าง ขอบด้านล่างคือจำนวนของ บรรทัดเปล่าที่ปล่อยไว้ที่ด้านล่างของแต่ละหน้า
<code>-B Value</code>	ระบุว่าควรจะพิมพ์เพจการส่งเป็นชุด (หน้าต่อเนื่องซึ่งมีการแบ่งที่ perforations) หรือไม่ ตัวแปร Value ประกอบด้วยสตริงสองอักขระ อักขระแรกใช้กับเพจ ส่วนหัว อักขระที่สอง ใช้กับเพจส่วนท้าย แต่ละอักขระ ทั้งสองตัวนี้ สามารถเป็นอย่างใดอย่างหนึ่งต่อไปนี้: a พิมพ์หน้า (ส่วนหัวหรือส่วนท้าย) ของแต่ละไฟล์ในแต่ละงานพิมพ์ เสมอ n ไม่พิมพ์หน้า (ส่วนหัวหรือส่วนท้าย) g พิมพ์หน้า (ส่วนหัวหรือส่วนท้าย) หนึ่งครั้งสำหรับแต่ละงานพิมพ์ (กลุ่มของ ไฟล์) ตัวอย่างเช่น แฟล็ก <code>-B ga</code> ระบุว่าเพจส่วนหัวจะถูกพิมพ์ ที่ตอนต้นของแต่ละงานพิมพ์ และเพจส่วนท้ายจะถูกพิมพ์ หลังจากแต่ละไฟล์ในแต่ละงานพิมพ์ ในสภาวะแวดล้อมการพิมพ์รีโมต คำศัพท์มีการกำหนดโดยคิวรีโมตบนเซิร์ฟเวอร์

ไอเท็ม	Descriptor
-e Option	ระบุว่าต้องการการพิมพ์แบบเน้นหรือไม่ + บ่งชี้ว่าต้องการการพิมพ์แบบเน้น ! บ่งชี้ว่าไม่ต้องการการพิมพ์แบบเน้น
-E Option	ระบุว่าต้องการการพิมพ์แบบ double-high หรือไม่ + บ่งชี้ว่าต้องการการพิมพ์แบบ double-high ! บ่งชี้ว่าไม่ต้องการการพิมพ์แบบ double-high
-f FilterType	ตัวระบุหนึ่งอักขระที่ระบุตัวกรองซึ่งไฟลงานพิมพ์หรือไฟล์ของคุณ จะถูกส่งผ่าน ก่อนส่งไปยังเครื่องพิมพ์ ตัวระบุตัวกรองที่มีอยู่คือ p ซึ่งจะเรียกใช้ตัวกรอง pr และ n ซึ่งจะประมวลผลเอาต์พุตจากคำสั่ง troff
-i Number	ส่งผลให้ย่อหน้าแต่ละบรรทัดตามจำนวนช่องว่าง ที่ระบุ ตัวแปร Number ต้องถูกรวมไว้ในความกว้างหน้าที่ ระบุ โดยแฟล็ก -w
-K Option	ระบุว่าต้องการการพิมพ์แบบบีบอัดหรือไม่ + บ่งชี้ว่าต้องการการพิมพ์แบบบีบอัด ! บ่งชี้ว่าไม่ต้องการการพิมพ์แบบบีบอัด
-l Number	ตั้งค่าความยาวของหน้าเป็นจำนวนบรรทัดที่ระบุ หากตัวแปร Number เป็น 0 ความยาวของหน้าจะถูก ละเว้น และมีการพิจารณาว่า เอาต์พุตเป็นหน้าต่อเนื่องหนึ่งหน้า ความยาวหน้ากระดาษจะรวมขอบด้านบน และขอบด้านล่าง และบ่งชี้ความยาวที่สามารถพิมพ์ได้ของกระดาษ
-L Option	ระบุว่าบรรทัดที่ยาวกว่าความกว้างของหน้าจะถูก ตัดไปขึ้นบรรทัดใหม่ หรือตัดให้สั้นลงที่ขอบด้านขวา + บ่งชี้ว่าควรตัดบรรทัดที่ยาวไปขึ้นบรรทัดถัดไป ! บ่งชี้ว่าไม่ควรตัดบรรทัดที่ยาว แต่ควรตัดให้สั้นลงที่ ขอบด้านขวาแทน
-N Number	ระบุจำนวนสำเนาที่จะพิมพ์ หากไม่ได้ระบุแฟล็กนี้ ระบบจะพิมพ์หนึ่งสำเนา
-p Number	ตั้งค่าระดับเป็น Number อักขระ ต่อนี้ว ค่าปกติของ Number คือ 10 และ 12 pitch ของอักขระที่พิมพ์ยังได้ รับผลกระทบโดยค่าสำหรับแฟล็ก -K (แบบย่อ) และแฟล็ก -W (กว้างเป็นสองเท่า)
-P Queue[:QueueDevice]	ระบุชื่อคิวการพิมพ์และชื่ออุปกรณ์คิวที่เป็น ทางเลือก หากไม่ได้ระบุแฟล็กนี้ จะใช้เครื่องพิมพ์ดีฟอลต์
-Q Value	ระบุขนาดกระดาษสำหรับงานพิมพ์ Value ของ ขนาดกระดาษขึ้นอยู่กับเครื่องพิมพ์ โดยทั่วไปค่าคือ 1 สำหรับกระดาษ letter 2 สำหรับ legal เป็นต้น ศึกษา คู่มือการใช้งานพริ้นเตอร์สำหรับค่าที่กำหนดไว้ให้กับ ขนาดของกระดาษที่ระบุเฉพาะ
-t Number	ระบุขอบด้านบน ขอบด้านบนคือจำนวนของ บรรทัดเปล่าที่ปล่อยไว้ที่ด้านบนของแต่ละหน้า
-w Number	ตั้งค่าความกว้างของหน้าเป็นจำนวนอักขระที่ระบุ โดยพารามิเตอร์ Number ความกว้างของหน้าต้องรวม จำนวนช่องว่างการย่อหน้าที่ระบุด้วยแฟล็ก -i
-W Option	ระบุว่าต้องการการพิมพ์แบบความกว้างสองเท่าหรือไม่ + บ่งชี้ว่าต้องการการพิมพ์แบบความกว้างสองเท่า ! บ่งชี้ว่าไม่ต้องการการพิมพ์แบบความกว้างสองเท่า

ไอเท็ม	Descriptor
-z Value	<p>หมุนหน้าเอาต์พุตของเครื่องพิมพ์ตามจำนวนรอบเศษหนึ่งส่วนสี่ ตามเข็มนาฬิกาที่ระบุโดยตัวแปร Value ค่าความยาว (-l) และความกว้าง (-w) มีการปรับโดยอัตโนมัติตามการหมุนนั้น</p> <p>0 แนวตั้ง</p> <p>1 แนวนอนชิดขวา</p> <p>2 แนวตั้งจากบนลงล่าง</p> <p>3 แนวนอนชิดซ้าย</p>
-# Value	<p>ระบุฟังก์ชันพิเศษ</p> <p>j แสดงหมายเลขงานสำหรับงานพิมพ์ที่ระบุ</p> <p>h จัดคิวงานพิมพ์ แต่วางไว้ในสถานะ HELD จนกว่าถูก รีลีสอีกครั้ง</p> <p>v ตรวจสอบค่าแฟล็กด้านหลักของพริ้นเตอร์ที่ระบุเฉพาะ การตรวจสอบนี้มีประโยชน์ในการตรวจสอบค่าแฟล็กที่ไม่ถูกต้องในเวลาที่ส่งงาน พิมพ์ หากไม่ได้รับผลการตรวจสอบความถูกต้อง ค่าแฟล็กที่ไม่ถูกต้องจะหยุดงานพิมพ์ในภายหลังในขณะกำลังประมวลผลงานจริง</p>

รายการต่อไปนี้มีตัวอย่างวิธีการใช้แฟล็กคำสั่ง qprt :

- หากต้องการร้องขอให้ไฟล์ myfile พิมพ์บน เครื่องพิมพ์ที่พร้อมใช้งานตัวแรกที่กำหนดคอนฟิกสำหรับคิวการพิมพ์ค่าดีฟอลต์โดยใช้ ค่าดีฟอลต์ให้พิมพ์:

```
qprt myfile
```

- หากต้องการร้องขอให้ไฟล์ myfile พิมพ์บนคิวที่ระบุ โดยใช้ค่าแฟล็กที่ระบุ และเพื่อตรวจสอบค่าแฟล็กในขณะที่ส่งงานพิมพ์ให้พิมพ์:

```
qprt -f p -e + -Pfastest -# v myfile
```

คำสั่งนี้ ส่งผ่านไฟล์ myfile ผ่านทางคำสั่งตัวกรอง pr (แฟล็ก -fp) และพิมพ์ไฟล์โดยใช้โหมดที่เน้น (แฟล็ก -e +) บนเครื่องพิมพ์แรกที่พร้อมใช้งานซึ่งกำหนดคอนฟิกสำหรับคิวที่ชื่อ fastest (แฟล็ก -Pfastest)

- หากต้องการพิมพ์ไฟล์ myfile บนกระดาษขนาด legal ให้พิมพ์:

```
qprt -Q2 myfile
```

- หากต้องการสามสำเนาของไฟล์ new.index.c, print.index.c, และ more.c แต่ละไฟล์ที่คิวการพิมพ์ Msp1 ให้พิมพ์:

```
qprt -PMsp1 -N 3 new.index.c print.index.c more.c
```

- หากต้องการพิมพ์สามสำเนาของสามไฟล์แบบเรียงติดกัน new.index.c, print.index.c, และ more.c ให้พิมพ์:

```
cat new.index.c print.index.c more.c | qprt -PMsp1 -N 3
```

หมายเหตุ: ระบบปฏิบัติการพื้นฐานยังสนับสนุนคำสั่งพิมพ์ BSD UNIX (lpr) และคำสั่งพิมพ์ System V UNIX (lp) สำหรับไวยากรณ์ที่สมบูรณ์ โปรดดูที่คำสั่ง lpr และ lp ใน *Commands Reference, Volume 3*

สำหรับ ไวยากรณ์ที่สมบูรณ์ โปรดดูที่คำสั่ง qprt ใน *Commands Reference, Volume 4*

คุณยังสามารถใช้ SMIT เพื่อร้องของานพิมพ์ หากต้องการเริ่มงานพิมพ์โดยใช้ SMIT ให้พิมพ์:

```
smit qprt
```

การยกเลิกงานพิมพ์ (คำสั่ง qcan)

คุณสามารถใช้คำสั่ง qcan เพื่อยกเลิกงานพิมพ์

- สำหรับงานพิมพ์ไลคัล เครื่องพิมพ์ต้องมีการต่อพ่วงเข้ากับระบบของคุณทาง ภายนอก หรือในกรณีของเครื่องพิมพ์เครือข่าย ต้องต่อพ่วงและกำหนดคอนฟิก บนเครือข่าย
- สำหรับงานพิมพ์รีโมต ระบบของคุณต้องมีการกำหนดคอนฟิกเพื่อสื่อสารกับ เซิร์ฟเวอร์การพิมพ์รีโมต

เมื่อคุณยกเลิกงานพิมพ์ คุณจะได้รับการพร้อมตัวให้ระบุชื่อของคิวการพิมพ์ที่งานตั้งอยู่ และหมายเลขงานซึ่งจะถูกยกเลิก

โปรแกรมนี้ใช้กับทั้งงานพิมพ์ไลคัลและรีโมต

ใช้คำสั่ง qcan เพื่อยกเลิกหมายเลขงานเฉพาะในคิวการพิมพ์แบบไลคัลหรือรีโมต หรืองานทั้งหมดในคิวการพิมพ์ ไลคัล เมื่อต้องการกำหนดหมายเลขงานพิมพ์ ให้ใช้คำสั่ง qchk

รูปแบบพื้นฐาน ของคำสั่ง qcan คือ:

```
qcan -P QueueName -x JobNumber
```

สำหรับไวยากรณ์โดยสมบูรณ์ทั้งหมด ดูที่คำสั่ง qcan ใน *Commands Reference, Volume 4*

รายการต่อไปนี้มีตัวอย่างวิธีการใช้ คำสั่ง qcan:

- เมื่อต้องการยกเลิกหมายเลขงาน 123 บนเครื่องพิมพ์ใดก็ตามที่มีงานอยู่ ให้พิมพ์: qcan -x 123
- เมื่อต้องการยกเลิกคิวงานทั้งหมดบนเครื่องพิมพ์ lp0 ให้พิมพ์: qcan -X -Plp0

หมายเหตุ: ระบบปฏิบัติการพื้นฐานยังสนับสนุนคำสั่งยกเลิกการพิมพ์ของ BSD UNIX (lprm) และคำสั่งยกเลิกการพิมพ์ของ System V UNIX (cancel) สำหรับไวยากรณ์ทั้งหมด โปรดดูที่คำสั่ง lprm และ cancel ใน *Commands Reference*

การยกเลิกงานพิมพ์ (SMIT)

คุณสามารถใช้ SMIT เพื่อยกเลิกงานพิมพ์

- สำหรับงานพิมพ์ไลคัล เครื่องพิมพ์ต้องมีการต่อพ่วงเข้ากับระบบของคุณทาง ภายนอก หรือในกรณีของเครื่องพิมพ์เครือข่าย ต้องต่อพ่วงและกำหนดคอนฟิก บนเครือข่าย
- สำหรับงานพิมพ์รีโมต ระบบของคุณต้องมีการกำหนดคอนฟิกเพื่อสื่อสารกับ เซิร์ฟเวอร์การพิมพ์รีโมต

เมื่อต้องการยกเลิกงานพิมพ์โดยใช้ SMIT ให้พิมพ์:

```
smit qcan
```

จากนั้น คุณสามารถเลือกหมายเลขงานพิมพ์หรือเครื่องพิมพ์ได้

การจัดระดับความสำคัญของงานพิมพ์ (คำสั่ง qpri)

คุณสามารถกำหนดลำดับความสำคัญของงานบนคิวแบบไลคัลโดยใช้คำสั่ง qpri

ต้องต่อพ่วงเครื่องพิมพ์เข้ากับระบบของคุณทางกายภาพ

ค่าสูงสุด บ่งชี้ระดับความสำคัญสูงสุดของงานพิมพ์ ลำดับ ความสำคัญที่เป็นค่าดีฟอลต์คือ 15 ระดับความสำคัญสูงสุดคือ 20 สำหรับผู้ใช้ส่วนใหญ่ และ 30 สำหรับผู้ใช้ที่มีสิทธิ์ผู้ใช้รากและสมาชิกของกลุ่ม printq (กลุ่ม 9)

หมายเหตุ: คุณไม่สามารถกำหนดลำดับความสำคัญไปยังงานพิมพ์แบบรีโมต

ใช้คำสั่ง `qpri` เพื่อกำหนดระดับความสำคัญของงานพิมพ์ ซึ่งคุณส่งอีกครั้ง หากคุณมีสิทธิผู้ใช้รากหรือเป็นสมาชิกของกลุ่ม `printq` คุณสามารถกำหนดระดับความสำคัญให้กับงานใดๆ ในขณะที่งานอยู่ในคิวการพิมพ์ได้

รูปแบบพื้นฐาน ของคำสั่ง `qpri` คือ:

```
qpri -# JobNumber -a PriorityLevel
```

สำหรับ ไวยากรณ์ที่สมบูรณ์ โปรดดูที่คำสั่ง `qpri` ใน *Commands Reference, Volume 4*

รายการต่อไปนี้มีตัวอย่างวิธีการใช้คำสั่ง `qpri`:

- หากต้องเปลี่ยนหมายเลขของงาน 123 เป็นหมายเลขลำดับความสำคัญ 18 ให้พิมพ์:

```
qpri -# 123 -a 18
```

- หากต้องการจัดลำดับความสำคัญงานพิมพ์แบบโลคัลตามที่ส่งไปให้พิมพ์:

```
qpri -PQueueName -R PriorityLevel FileName
```

การกำหนดลำดับความสำคัญของงานพิมพ์ (SMIT)

คุณสามารถกำหนดลำดับความสำคัญงานเฉพาะบนคิวแบบโลคัลเท่านั้น

ต้องต่อพ่วงเครื่องพิมพ์เข้ากับระบบของคุณทางกายภาพ

ค่าสูงสุด บ่งชี้ระดับความสำคัญสูงสุดของงานพิมพ์ ลำดับความสำคัญ ที่เป็นค่าดีฟอลต์คือ 15 ระดับความสำคัญสูงสุดคือ 20 สำหรับผู้ใช้ส่วนใหญ่ และ 30 สำหรับผู้ใช้ที่มีสิทธิผู้ใช้รากและสมาชิกของกลุ่ม `printq` (กลุ่ม 9)

หมายเหตุ: คุณไม่สามารถกำหนดลำดับความสำคัญให้กับงานพิมพ์แบบรีโมต

หากต้องเปลี่ยนแปลงลำดับความสำคัญของงานพิมพ์โดยใช้ SMIT ให้พิมพ์:

```
smit qpri
```

Moving a print job to another print queue (คำสั่ง `qmov`)

คุณสามารถย้ายงานพิมพ์ระหว่างคิวต่างๆ ได้โดยใช้คำสั่ง `qmov`

เมื่อต้องการ ดำเนินการภารกิจนี้ คุณต้องมีสิ่งที่จำเป็นต้องมีต่อไปนี้:

- เครื่องพิมพ์ต้องต่อพ่วงเข้ากับระบบของคุณทางกายภาพ
- คุณต้องเป็นเจ้าของงานพิมพ์
- คุณมีสิทธิระดับรูท
- คุณต้องเป็นสมาชิกกลุ่ม `printq`

หมายเหตุ: คุณไม่สามารถย้ายงานพิมพ์แบบรีโมตไปยังคิวการพิมพ์อื่น

ใช้คำสั่ง `qmov` เพื่อย้ายงานพิมพ์ไปยัง คิวการพิมพ์อื่น คุณสามารถย้ายงานพิมพ์เฉพาะ หรือสามารถย้ายงานพิมพ์ทั้งหมดใน คิวการพิมพ์ที่ระบุ คุณยังสามารถย้ายงานพิมพ์ทั้งหมดที่ถูกส่งโดย ผู้ใช้ที่ระบุ เมื่อต้องการกำหนด หมายเลขงานพิมพ์ ให้ใช้คำสั่ง `qchk` สำหรับข้อมูลเพิ่มเติม ให้อูที่ `qchk`

รูปแบบ พื้นฐานของคำสั่ง `qmov` เป็นดังนี้:

```
qmov -mNewQueue { [ -#JobNumber ] [ -PQueue ] [ -uUser ] }
```

คุณสามารถย้ายงานพิมพ์โดยใช้คำสั่งอย่างใดอย่างหนึ่งต่อไปนี้:

- `qmov -m DestinationQueue -# JobNumber`
- `qmov -m DestinationQueue -P Queue`
- `qmov -m DestinationQueue -u User`

สำหรับไวยากรณ์ทั้งหมด โปรดดูที่คำสั่ง `qmov` ใน *Commands Reference*

รายการต่อไปนี้มีตัวอย่างของวิธีการใช้คำสั่ง `qmov`:

- เมื่อต้องการย้ายงานหมายเลข 280 ไปยังคิวการพิมพ์ hp2 ให้พิมพ์:

```
qmov -mhp2 -#280
```
- เมื่อต้องการย้ายงานพิมพ์ทั้งหมดบนคิวการพิมพ์ hp4D ไปยังคิวการพิมพ์ hp2 ให้พิมพ์:

```
qmov -mhp2 -Php4D
```

Moving a print job to another print queue (SMIT)

หากเครื่องพิมพ์ของคุณมีการเชื่อมต่อกับระบบของคุณ คุณสามารถย้ายงานพิมพ์ไปยังคิวการพิมพ์อื่นด้วย SMIT

ถ้าเครื่องพิมพ์ของคุณต่อกับระบบของคุณโดยตรง คุณสามารถย้ายงานพิมพ์ไปยังคิวงานอื่นได้ด้วย System Management Interface Tool (SMIT)

เมื่อต้องการดำเนินการภารกิจนี้ คุณต้องมีสิ่งที่จำเป็นต้องมีต่อไปนี้:

- ต้องต่อพ่วงเครื่องพิมพ์เข้ากับระบบของคุณทางกายภาพ
- คุณต้องเป็นเจ้าของงานพิมพ์
- คุณมีสิทธิ์ระดับรูท
- คุณต้องเป็นสมาชิกของกลุ่ม `printq`

หมายเหตุ: คุณไม่สามารถย้ายงานพิมพ์แบบรีโมตไปยังคิวการพิมพ์อื่นได้

พิมพ์คำสั่งต่อไปนี้:

```
smit qmov
```

Holding and releasing print jobs (คำสั่ง `qhld`)

คุณสามารถพักและรีลีสงานพิมพ์ได้ด้วยคำสั่ง `qhld`

หมายเหตุ: คุณไม่สามารถพักและรีลีสงานพิมพ์รีโมต

เมื่อต้องการดำเนินการภารกิจนี้ คุณต้องมีสิ่งที่จำเป็นต้องมีต่อไปนี้:

- ต้องต่อพ่วงเครื่องพิมพ์เข้ากับระบบของคุณทางกายภาพ
- คุณต้องเป็นเจ้าของงานพิมพ์
- คุณมีสิทธิ์ระดับรูท

- คุณต้องเป็นสมาชิกของกลุ่ม **printq**

ใช้คำสั่ง **qhld** เพื่อนำงานพิมพ์ที่ค้างไว้หลังจากที่ส่ง คุณสามารถพักงานพิมพ์เฉพาะ หรือคุณสามารถพักงานพิมพ์ทั้งหมดบนคิวการพิมพ์ที่ระบุ เมื่อต้องการกำหนดหมายเลขงานพิมพ์ให้ป้อนคำสั่ง **qchk** รูปแบบพื้นฐานของคำสั่ง **qhld** เป็นดังนี้:

```
qhld [ -r ] { [ -#JobNumber ] [ -PQueue ] [ -uUser ] }
```

คุณสามารถพักงานพิมพ์ได้โดยใช้หนึ่งในคำสั่งต่อไปนี้:

- `qhld -# JobNumber`
- `qhld -P Queue`
- `qhld -u User`

คุณสามารถปล่อยงานพิมพ์ได้โดยใช้หนึ่งใน คำสั่งต่อไปนี้:

- `qhld -r -# Jobnumber`
- `qhld -r -P Queue`
- `qhld -r -u User`

รายการต่อไปนี้มีตัวอย่างวิธีการใช้ คำสั่ง **qhld**:

1. เมื่อต้องการพักงานหมายเลข 452 บนคิวงานพิมพ์ ที่เปิดอยู่ พิมพ์คำสั่งต่อไปนี้:
`qhld -#452`
2. เมื่อต้องการพักงานที่จัดคิวทั้งหมดบนคิวการพิมพ์ hp2 พิมพ์ คำสั่งต่อไปนี้:
`qhld -P hp2`
3. เมื่อต้องการปล่อยงานหมายเลข 452 บนคิวงานพิมพ์ ที่เปิดอยู่ พิมพ์คำสั่งต่อไปนี้:
`qhld -#452 -r`
4. เมื่อต้องการปล่อยงานที่จัดคิวทั้งหมดบนคิวการพิมพ์ hp2 พิมพ์คำสั่งต่อไปนี้:
`qhld -P hp2 -r`

การพักและรีลีสงานพิมพ์ (SMIT)

คุณสามารถพักและรีลีสงานพิมพ์ได้โดยใช้ SMIT

หากต้องการพักหรือรีลีสงานพิมพ์ คุณต้องเป็นบุคคลใดบุคคลหนึ่งต่อไปนี้:

- เจ้าของงานพิมพ์
- ผู้ใช้ที่มีสิทธิ์ระดับราก
- สมาชิกของกลุ่ม **printq**

เมื่อต้องการพักหรือรีลีสงานพิมพ์:

- `smit qhld`

การตรวจสอบสถานะของงานพิมพ์ (คำสั่ง qchk)

คุณสามารถใช้คำสั่ง **qchk** เพื่อตรวจสอบสถานะ ของงานพิมพ์

- สำหรับงานพิมพ์ไคลด์ เครื่องพิมพ์ต้องมีการต่อพ่วงเข้ากับระบบของคุณทาง ภายนอก หรือในกรณีของเครื่องพิมพ์เครือข่าย ต้องต่อพ่วงและกำหนดคอนฟิก บนเครือข่าย
- สำหรับงานพิมพ์รีโมต ระบบของคุณต้องมีการกำหนดคอนฟิกเพื่อสื่อสารกับ เซิร์ฟเวอร์การพิมพ์รีโมต

ใช้คำสั่ง `qchk` เพื่อแสดงข้อมูลสถานะปัจจุบัน ของงานพิมพ์ คิวการพิมพ์ หรือผู้ใช้ ที่ระบุ

รูปแบบพื้นฐานของคำสั่ง `qchk` คือ:

```
qchk -P QueueName -# JobNumber -u OwnerName
```

หมายเหตุ: ระบบปฏิบัติการ พื้นฐานยังสนับสนุนคำสั่งตรวจสอบคิวการพิมพ์ของ BSD UNIX (`lpq`) และคำสั่งตรวจสอบคิวการพิมพ์ของ System V UNIX (`lpstat`) สำหรับไวยากรณ์ทั้งหมด โปรดดูที่คำสั่ง `lpq` และ `lpstat` in *Commands Reference*

รายการต่อไปนี้มีตัวอย่างวิธีการใช้ คำสั่ง `qchk`:

- หากต้องการแสดงคิวการพิมพ์ดีฟอลต์ให้พิมพ์:
`qchk -q`
- หากต้องการแสดงสถานะของคิวทั้งหมดแบบยาวจนกระทั่งว่างเปล่าขณะที่อัปเดต หน้าจอทุกๆ 5 วินาที ให้ป้อน:
`qchk -A -L -w 5`
- หากต้องการแสดงสถานะของคิวการพิมพ์ `lp0` ให้พิมพ์:
`qchk -P lp0`
- หากต้องการแสดงสถานะของหมายเลขงาน `123` ให้พิมพ์:
`qchk -# 123`
- หากต้องการตรวจสอบสถานะของทุกงานในคิวทั้งหมด ให้พิมพ์:
`qchk -A`

เงื่อนไขสถานะของคิวการพิมพ์

เงื่อนไขสถานะ บางอย่างที่คิวการพิมพ์สามารถมีได้:

ไอเท็ม	Descriptor
<code>DEV_BUSY</code>	<p>บ่งชี้ว่า:</p> <ul style="list-style-type: none"> • มีการกำหนดมากกว่าหนึ่งคิวให้กับอุปกรณ์เครื่องพิมพ์ (<code>lp0</code>) และคิวอื่นกำลังใช้อุปกรณ์เครื่องพิมพ์อยู่ในปัจจุบัน • <code>qdaemon</code> พยายามใช้อุปกรณ์พอร์ตเครื่องพิมพ์ (<code>lp0</code>) แต่แอ็พพลิเคชันอื่นกำลังใช้อุปกรณ์เครื่องพิมพ์นั้นอยู่ในปัจจุบัน <p>เมื่อต้องการกู้คืนจาก <code>DEV_BUSY</code> ให้รอจนกว่าคิวหรือแอ็พพลิเคชัน รีลีสอุปกรณ์เครื่องพิมพ์ หรือยกเลิกงานหรือกระบวนการที่กำลังใช้ พอร์ตเครื่องพิมพ์นั้น</p>
<code>DEV_WAIT</code>	<p>บ่งชี้ว่าคิวกำลังรอเครื่องพิมพ์อยู่เนื่องจาก เครื่องพิมพ์ออฟไลน์ ขาดกระดาษ กระดาษติด หรือสายเคเบิลหลวม อยู่ในสภาพ ไม่ดี หรือเชื่อมต่อไม่ถูกต้อง</p> <p>เมื่อต้องการกู้คืนจาก <code>DEV_WAIT</code> ให้แก้ไขปัญหาที่ทำให้คิวต้องรอ การทดสอบเชิงวินิจฉัยสามารถทำได้ง่ายขึ้นโดยใช้คำสั่ง <code>enq</code> เพื่อย้ายงานที่จัดคิวทั้งหมดออกจากคิว <code>DEV_WAIT</code> ไปยังคิวอื่นที่ กำลังพิมพ์หรือเป็น <code>DOWN</code> หลังจากแก้ไขปัญหาแล้ว คุณ สามารถย้ายงานที่ยังไม่ได้พิมพ์ใดๆ กลับไปยังคิวดั้งเดิม</p>

ไอเอ็ม	Descriptor
DOWN	โดยปกติ คิวเข้าสู่สถานะ DOWN หลังจากเคยอยู่ในสถานะ DEV_WAIT มาแล้ว สถานการณ์นี้เกิดขึ้นเมื่อไดรเวอร์อุปกรณ์ เครื่องพิมพ์ไม่สามารถบอกได้ว่าเครื่องพิมพ์อยู่ที่นั่นหรือไม่ เนื่องจากขาดการส่งสัญญาณ ที่ถูกต้อง อย่างไรก็ตาม เครื่องพิมพ์บางเครื่องอาจไม่มีความสามารถในการส่งสัญญาณแจ้งระบบที่จัดคิวว่าเครื่องพิมพ์ออฟไลน์ และส่งสัญญาณว่า เครื่องพิมพ์ปิดแทน หาก อุปกรณ์เครื่องพิมพ์ส่งสัญญาณหรือแสดงว่าปิด คิวจะเข้าสู่สถานะ DOWN
	เมื่อต้องการกู้คืนจากสถานะ DOWN ให้แก้ไขปัญหาที่ทำให้คิวควาน และขอให้ผู้ดูแลระบบ นำคิวกลับมา ต้องนำคิวกลับมาด้วยตนเองก่อนจะสามารถใช้คิว อีกครั้งได้
HELD	ระบุว่าการพิมพ์ถูกพักไว้ งานพิมพ์จะไม่มี การประมวลผลโดยสพูลเลอร์จนกว่าถูกรีลีส
QUEUED	ระบุไฟล์การพิมพ์ที่จัดคิวและกำลังรออยู่ในแถว ที่จะพิมพ์
READY	ระบุว่าทุกสิ่งที่เกี่ยวข้องกับคิวพร้อมที่จะ จัดคิวและพิมพ์งาน
RUNNING	ระบุว่าไฟล์การพิมพ์กำลังพิมพ์

Checking the status of a print job (SMIT)

คุณสามารถใช้คำสั่ง `smit` เพื่อ ตรวจสอบสถานะของงานพิมพ์

- สำหรับงานพิมพ์โลคัล เครื่องพิมพ์ต้องมีการต่อพ่วงเข้ากับระบบของคุณทาง ภายนอก หรือในกรณีของเครื่องพิมพ์เครือข่าย ต้องต่อพ่วงและกำหนดคอนฟิก บนเครือข่าย
- สำหรับงานพิมพ์รีโมต ระบบของคุณต้องมีการกำหนดคอนฟิกเพื่อสื่อสารกับ เซิร์ฟเวอร์การพิมพ์รีโมต

คุณสามารถแสดงข้อมูลสถานะปัจจุบันของ หมายเลขงาน คิว เครื่องพิมพ์ หรือผู้ใช้ที่ระบุได้ หากต้องการตรวจสอบสถานะของงานพิมพ์โดยใช้ SMIT ให้พิมพ์:

```
smit qchk
```

เงื่อนไขสถานะของคิวการพิมพ์

เงื่อนไขสถานะ บางอย่างที่คิวการพิมพ์สามารถมีได้:

ไอเอ็ม	รายละเอียด
DEV_BUSY	บงชี้ว่า: <ul style="list-style-type: none"> • มีการกำหนดมากกว่าหนึ่งคิวให้กับอุปกรณ์เครื่องพิมพ์ (IpO) และคิวอื่นกำลังใช้อุปกรณ์เครื่องพิมพ์อยู่ในปัจจุบัน • <code>qdaemon</code> พยายามใช้อุปกรณ์พอร์ตเครื่องพิมพ์ (IpO) แต่แอพลิเคชันอื่นกำลังใช้อุปกรณ์เครื่องพิมพ์นั้นอยู่ในปัจจุบัน <p>เมื่อต้องการกู้คืนจาก DEV_BUSY ให้รอจนกว่าคิวหรือแอพลิเคชัน รีลีสอุปกรณ์เครื่องพิมพ์ หรือยกเลิกงานหรือกระบวนการที่กำลังใช้ พอร์ตเครื่องพิมพ์นั้น</p>
DEV_WAIT	บงชี้ว่าคิวกำลังรอเครื่องพิมพ์อยู่เนื่องจาก เครื่องพิมพ์ออฟไลน์ ขาดกระดาษ กระดาษติด หรือสายเคเบิลหลวม อยู่ในสภาพ ไม่ดี หรือเชื่อมต่อไม่ถูกต้อง
	เมื่อต้องการกู้คืนจาก DEV_WAIT ให้แก้ไขปัญหา ที่ทำให้คิวต้องรอ การทดสอบเชิงวินิจฉัยสามารถทำได้ง่ายขึ้นโดยใช้คำสั่ง <code>enq</code> เพื่อย้ายงานที่จัดคิวทั้งหมดออกจากคิว DEV_WAIT ไปยังคิวอื่นที่ กำลังพิมพ์หรือเป็น DOWN หลังจากแก้ไขปัญหานั้นแล้ว คุณ สามารถย้ายงานที่ยังไม่ได้พิมพ์ใดๆ กลับไปยังคิวดั้งเดิม
DOWN	โดยปกติ คิวเข้าสู่สถานะ DOWN หลังจากเคยอยู่ในสถานะ DEV_WAIT มาแล้ว สถานการณ์นี้เกิดขึ้นเมื่อไดรเวอร์อุปกรณ์ เครื่องพิมพ์ไม่สามารถบอกได้ว่าเครื่องพิมพ์อยู่ที่นั่นหรือไม่ เนื่องจากขาดการส่งสัญญาณ ที่ถูกต้อง อย่างไรก็ตาม เครื่องพิมพ์บางเครื่องอาจไม่มีความสามารถในการส่ง สัญญาณแจ้งระบบที่จัดคิวว่าเครื่องพิมพ์ออฟไลน์ และส่งสัญญาณว่า เครื่องพิมพ์ปิดแทน หาก อุปกรณ์เครื่องพิมพ์ส่งสัญญาณหรือแสดงว่าปิด คิวจะเข้าสู่สถานะ DOWN
	เมื่อต้องการกู้คืนจากสถานะ DOWN ให้แก้ไขปัญหาที่ทำให้คิวควาน และขอให้ผู้ดูแลระบบ นำคิวกลับมา ต้องนำคิวกลับมาด้วยตนเองก่อนจะสามารถใช้คิว อีกครั้งได้
HELD	ระบุว่าการพิมพ์ถูกพักไว้ งานพิมพ์จะไม่มี การประมวลผลโดยสพูลเลอร์จนกว่าถูกรีลีส
QUEUED	ระบุไฟล์การพิมพ์ที่จัดคิวและกำลังรออยู่ในแถว ที่จะพิมพ์
READY	ระบุว่าทุกสิ่งที่เกี่ยวข้องกับคิวพร้อมที่จะ จัดคิวและพิมพ์งาน

การจัดรูปแบบไฟล์สำหรับพิมพ์ (คำสั่ง pr)

คุณสามารถใช้คำสั่ง pr เพื่อจัดรูปแบบ ไฟล์ที่ส่งไปยังเครื่องพิมพ์ได้อย่างง่ายดาย

คุณสามารถทำโพท์เอาต์พุตของคำสั่ง pr ไปยังคำสั่ง qprt เพื่อจัดรูปแบบข้อความ

แฟล็กคำสั่ง pr ที่มีประโยชน์บางแฟล็กคือ:

ไอเท็ม	Descriptor
-d	เอาต์พุตมีช่องว่างคู่
-h "String"	แสดงสตริงที่ระบุซึ่งใส่ไว้ในเครื่องหมายอัญประกาศ (" ") แทนชื่อไฟล์เป็นส่วนหัวของหน้า คั่นแฟล็ก และสตริงด้วยช่องว่าง
-l Lines	ยกเลิกตีฟอลต์ 66 บรรทัดและรีเซ็ตความยาวหน้า เป็นจำนวนบรรทัดซึ่งระบุโดยตัวแปร Lines ถ้าค่า Lines น้อยกว่าผลรวมของหัว และส่วนท้าย (ในหน่วยบรรทัด) ส่วนหัวและส่วนท้ายจะถูกตัด (เหมือนกับว่าใช้แฟล็ก -t)
-m	ผลานไฟล์ มีการจัดรูปแบบเอาต์พุตมาตรฐานเพื่อให้คำสั่ง pr เขียนหนึ่งบรรทัดจากแต่ละไฟล์ที่ระบุโดยตัวแปร File แบบด้นต่อด้านเข้าในคอลัมน์ข้อความซึ่งมีความกว้างที่ระบุเท่ากัน ตามข้อมูล จำนวนของตำแหน่งคอลัมน์ แฟล็กนี้ไม่ควรใช้ร่วมกับแฟล็ก -Column
-n [Width][Character]	แสดงการกำหนดหมายเลขบรรทัดตามข้อมูลตัวเลขที่ระบุ โดยตัวแปร Width ตัวเลขตีฟอลต์คือ 5 ถ้ามีการระบุตัวแปร Character (อักขระ ที่ไม่ใช่ตัวเลขใดๆ) คำนับจะถูกผนวกเข้ากับหมายเลขบรรทัด เพื่อแบ่งแยกออกจากสิ่งทีตามมานับบรรทัด
-o Offset	ย่อหน้าแต่ละบรรทัดตามตัวเลขของตำแหน่งอักขระที่ระบุ โดยตัวแปร Offset จำนวนทั้งหมดของตำแหน่งอักขระ ต่อบรรทัดคือผลรวมของความกว้างและออฟเซต ค่าตีฟอลต์ของ Offset คือ 0
-s Character	แบ่งคอลัมน์โดยใช้อักขระเดี่ยวที่ระบุโดยตัวแปร Character แทนการใช้จำนวนช่องว่างที่เหมาะสม ค่าตีฟอลต์ของ Character คือ อักขระ ASCII Tab
-t	ไม่แสดงส่วนหัวการระบุหัวบรรทัดและ ส่วนท้ายหัวบรรทัด หยุดหลังจากบรรทัดสุดท้ายของแต่ละไฟล์โดยไม่มีการเว้นช่องว่างที่ ตอนท้ายของหน้า
-w Width	ตั้งค่าจำนวนของตำแหน่งคอลัมน์ต่อบรรทัดเป็นค่าที่ ระบุโดยตัวแปร Width ค่าตีฟอลต์คือ 72 สำหรับเอาต์พุตหลายคอลัมน์ที่มีความกว้างเท่ากัน ไม่มีขีดจำกัดสำหรับเอาต์พุตรูปแบบอื่น ถ้าไม่ได้ระบุแฟล็ก -w และมีการระบุแฟล็ก -s ความกว้างตีฟอลต์คือ 512 ตำแหน่งคอลัมน์
-Column	ตั้งค่าจำนวนคอลัมน์เป็นค่าที่ระบุโดยตัวแปร Column ค่าตีฟอลต์คือ 1 อย่าใช้อ็พชันนี้กับแฟล็ก -m มีการสมมติแฟล็ก -e และ -i สำหรับเอาต์พุตหลายคอลัมน์ คอลัมน์ ข้อความไม่ควรเกินกว่าความยาวของหน้า (โปรดดูที่แฟล็ก -l) เมื่อใช้แฟล็กนี้ร่วมกับแฟล็ก -t ให้ใช้จำนวนบรรทัดต่ำสุด เพื่อเขียนเอาต์พุต
+Page	เริ่มต้นการแสดงผลด้วยหมายเลขหน้าทีระบุโดยตัวแปร Page ค่าตีฟอลต์คือ 1

สำหรับไวยากรณ์โดยสมบูรณ์ ดูที่คำสั่ง pr ใน in Commands Reference

ข้อมูลต่อไปนี้เป็นรายการของตัวอย่างวิธีการใช้แฟล็กคำสั่ง pr :

- เมื่อต้องการพิมพ์ไฟล์ที่ชื่อ prog.c พร้อมกับหัวเรื่องและ หมายเลขหน้าบนเครื่องพิมพ์ ให้พิมพ์:

```
pr prog.c | qprt
```

คำสั่งนี้ เพิ่มหัวเรื่องหน้าลงใน prog.c และส่งไปยัง คำสั่ง qprt หัวเรื่องประกอบด้วยวันที่ซึ่งแก้ไขไฟล์ ครั้งล่าสุด ชื่อไฟล์ และหมายเลขหน้า

- เมื่อต้องการระบุหัวเรื่องสำหรับไฟล์ที่ชื่อ prog.c ให้พิมพ์:

```
pr -h "MAIN PROGRAM" prog.c | qprt
```

คำสั่งนี้ พิมพ์ prog.c พร้อมด้วยหัวเรื่อง MAIN PROGRAM ใน ตำแหน่งของชื่อไฟล์ วันที่แก้ไขและหมายเลขหน้ายังคงมีการพิมพ์ด้วย

- เมื่อต้องการพิมพ์ไฟล์ที่ชื่อ word.lst ในหลายคอลัมน์ให้พิมพ์:

```
pr -3 word.lst | qprt
```

คำสั่งนี้พิมพ์ไฟล์ word.lst ในคอลัมน์แนวตั้งจำนวนสามคอลัมน์

- เมื่อต้องการพิมพ์หลายไฟล์แบบด้านต่อด้านบนกระดาษให้พิมพ์:

```
pr -m -h "Members and Visitors" member.lst visitor.lst | qprt
```

คำสั่งนี้พิมพ์ member.lst และ visitor.lst แบบด้านต่อด้านพร้อมด้วยหัวเรื่อง Members and Visitors

- เมื่อต้องการแก้ไขไฟล์ชื่อ prog.c สำหรับการใช้ในภายหลังให้พิมพ์:

```
pr -t -e prog.c > prog.notab.c
```

คำสั่งนี้แทนที่อักขระแท็บใน prog.c ด้วยช่องว่าง และวางผลลัพธ์ไว้ใน prog.notab.c ตำแหน่งแท็บอยู่ที่ คอลัมน์ 9, 17, 25, 33, และต่อไป แฟล็ก -e บอกคำสั่ง pr ให้แทนที่อักขระแท็บโดยแฟล็ก -t ตัดส่วนหัวของหน้า

- เมื่อต้องการพิมพ์ไฟล์ที่ชื่อ myfile ในสองคอลัมน์ในแนวนอน ด้วยข้อความ 7 พอยต์ให้พิมพ์:

```
pr -l66 -w172 -2 myfile | qprt -z1 -p7
```

การพิมพ์ไฟล์ ASCII บนเครื่องพิมพ์ PostScript

Text Formatting System ประกอบด้วยตัวกรอง **enscript** สำหรับการแปลงไฟล์งานพิมพ์ ASCII ไปเป็นไฟล์ PostScript สำหรับการพิมพ์บนเครื่องพิมพ์ PostScript

- ต้องต่อพ่วงเครื่องพิมพ์เข้ากับระบบของคุณทางกายภาพ
- ต้องกำหนดคอนฟิกและกำหนดเครื่องพิมพ์
- ต้องติดตั้งส่วนการถอดรหัสของเซอร์วิสการจัดรูปแบบข้อความ

ตัวกรอง **enscript** ถูกเรียกโดยคำสั่ง **qprt -da** เมื่อส่งงานพิมพ์ไปยังคิวการพิมพ์ PostScript แฟล็กต่างๆ อาจ ถูกระบุด้วยคำสั่ง **qprt** เพื่อปรับแต่งเอาต์พุต เมื่อส่งไฟล์ ASCII ไปยังคิวการพิมพ์ PostScript :

ไอเท็ม	Descriptor
-1+	เพิ่มส่วนหัวของหน้า
-2+	จัดรูปแบบเอาต์พุตในสองคอลัมน์
-3+	พิมพ์ส่วนหัวของหน้า วันที่ และหมายเลขหน้าในลักษณะ แพนซี ซึ่งบางครั้งเรียกว่าเป็นโหมด "จุดลาด"
-4+	พิมพ์ไฟล์ แมวที่มีอักขระที่ไม่สามารถพิมพ์ได้
-5+	แสดงรายการอักขระที่ไม่ได้รวมอยู่ในฟอนต์
-h string	ระบุสตริงที่จะใช้สำหรับส่วนหัวของหน้า ถ้าไม่ได้ระบุแฟล็กนี้ ส่วนหัวจะประกอบด้วยชื่อไฟล์ วันที่แก้ไข และหมายเลขหน้า
-l value	ระบุจำนวนบรรทัดสูงสุดที่พิมพ์ต่อหน้า บรรทัดที่ปรากฏขึ้นจริงสำหรับแต่ละหน้าอาจมีจำนวนน้อยลง ขึ้นอยู่กับขนาดพอยต์
-L!	ตัดบรรทัดที่ยาวเกินกว่าความกว้างของหน้าให้สั้นลง
-p	ระบุขนาดพอยต์ ถ้าไม่ได้ระบุแฟล็กนี้ ระบบจะสมมติขนาดพอยต์เป็น 10 ยกเว้นว่าระบุโหมดที่หมุนสองคอลัมน์ (-2+ -z1) ในกรณีใดก็ตาม ค่าที่ใช้คือ 7

ไอเท็ม

-s

Descriptor

ระบุลักษณะฟอนต์ หากไม่ระบุแฟล็กนี้ จะใช้ฟอนต์ Courier เครื่องพิมพ์ PostScript ต้องมีการเข้าถึงฟอนต์ ที่ระบุ ค่าที่สามารถยอมรับได้คือ:

Courier-Oblique
Helvetica
Helvetica-Oblique
Helvetica-Narrow
Helvetica-Narrow-Oblique
NewCenturySchlbk-Italic
Optima
Optima-Oblique
Palatino-Roman
Palatino-Italic
Times-Roman
Times-Italic

-z1

หมุนเอาต์พุต 90 องศา (โหมตแนวนอน)

รายการต่อไปนี้มีตัวอย่างของวิธีการใช้แฟล็กคำสั่ง **qrpt** เหล่านี้:

- หากต้องการส่งไฟล์ ACSII myfile.ascii ไปยังเครื่องพิมพ์ PostScript ที่ชื่อ **Msp1** ให้พิมพ์:

```
qrpt -da -PMsp1 myfile.ascii
```

- หากต้องการส่งไฟล์ ACSII myfile.ascii ไปยังเครื่องพิมพ์ PostScript ที่ชื่อ **Msp1** และพิมพ์ออกมาเป็นฟอนต์ Helvetica ให้พิมพ์:

```
qrpt -da -PMsp1 -sHelvetica myfile.ascii
```

- หากต้องการส่งไฟล์ ASCII myfile.ascii ไปยังเครื่องพิมพ์ PostScript ที่ชื่อ **Msp1** และพิมพ์ออกมาในขนาด 9 พอยต์ ให้พิมพ์:

```
qrpt -da -PMsp1 -p9 myfile.ascii
```

การทำการแปลง ASCII เป็น PostScript ให้เป็นระบบอัตโนมัติ:

คุณสามารถกำหนดคอนฟิกระบบให้ตรวจหาไฟล์การพิมพ์ ASCII ที่ส่งไปยังคิวการพิมพ์ PostScript และแปลงไฟล์ ASCII เหล่านี้เป็น PostScript โดยอัตโนมัติสำหรับเครื่องพิมพ์ PostScript

แอฟพลิเคชันจำนวนมากที่สร้างไฟล์การพิมพ์ PostScript จะปฏิบัติตามแบบแผนของการจัดทำอักษรสองตัวแรกของไฟล์ PostScript %! ซึ่งระบุไฟล์การพิมพ์เป็นไฟล์การพิมพ์ PostScript เมื่อต้องการกำหนดคอนฟิกระบบเพื่อตรวจหาไฟล์การพิมพ์ ASCII ที่ส่งไปยังคิวการพิมพ์ PostScript และแปลงไฟล์เหล่านั้นเป็นไฟล์ PostScript โดยอัตโนมัติ ก่อนจะส่งไปยังเครื่องพิมพ์ PostScript ให้ปฏิบัติตามขั้นตอนเหล่านี้:

1. เมื่อมีการพร้อมระบบ ให้พิมพ์: **smit chpq**
2. พิมพ์ชื่อคิว PostScript หรือใช้คุณลักษณะ **List** เพื่อเลือกจากรายการคิว
3. เลือกอ็อปชันเมนู **Printer Setup**
4. เปลี่ยนค่าของฟิลด์ จะทำการตรวจสอบอัตโนมัติของ ชนิดไฟล์การพิมพ์หรือไม่? ให้เป็น **ใช่**

ขณะนี้คำสั่งต่อไปนี้จะแปลงไฟล์ ASCII เป็นไฟล์ PostScript และพิมพ์ไฟล์นั้นบนเครื่องพิมพ์ PostScript หากต้องการแปลง myfile.ascii ให้พิมพ์บรรทัดคำสั่งตัวใดตัวหนึ่ง ต่อไปนี้:

```
qprt -Pps myfile.ps myfile.ascii  
lpr -Pps myfile.ps myfile.ascii  
lp -dps myfile.ps myfile.acsii
```

โดย ps เป็นคิวการพิมพ์ PostScript

การยกเลิกการกำหนดชนิดไฟล์การพิมพ์โดยอัตโนมัติ:

ในบางกรณี คุณอาจจำเป็นต้องยกเลิกการกำหนด ชนิดไฟล์งานพิมพ์โดยอัตโนมัติ

คุณสามารถยกเลิกการกำหนดชนิดไฟล์งานพิมพ์โดยอัตโนมัติสำหรับการพิมพ์ PostScript ด้วยแฟล็ก -d และ -s แฟล็ก -d จะยกเลิกชนิดไฟล์งานพิมพ์ที่เป็นค่าดีฟอลต์ และแฟล็ก -s ระบุการพิมพ์ PostScript แทน

คุณอาจจำเป็นต้องยกเลิกการกำหนด ชนิดไฟล์งานพิมพ์โดยอัตโนมัติสำหรับการพิมพ์ PostScript ในสถานการณ์ ต่อไปนี้:

- หากต้องการพิมพ์ไฟล์ PostScript file ที่ชื่อ myfile.ps ที่ไม่ได้เริ่มต้นด้วย %! ให้พิมพ์คำสั่งต่อไปนี้:
qprt -ds -Pps myfile.ps
- หากต้องการพิมพ์รายการซอร์สของไฟล์ PostScript ที่ชื่อ myfile.ps ที่เริ่มต้นด้วย %! ให้พิมพ์คำสั่งต่อไปนี้:
qprt -da -Pps myfile.ps

ข้อมูลสรุปคำสั่งสำหรับการพิมพ์

มีคำสั่งจำนวนมากที่ใช้สำหรับการพิมพ์และการจัดการคิว การพิมพ์

ไอเท็ม	Descriptor
cancel	ยกเลิกการร้องขอไปยังเครื่องพิมพ์บรรทัด
lp	ส่งคำร้องขอไปยังเครื่องพิมพ์บรรทัด
lpq	ตรวจสอบคิวเก็บพัก
lpr	จัดคิวงานพิมพ์
lprm	ลบงานออกจากคิวการเก็บพักของ เครื่องพิมพ์บรรทัด
lpstat	แสดงข้อมูลสถานะของเครื่องพิมพ์บรรทัด
pr	เขียนไฟล์ไปยังเอาต์พุตมาตรฐาน
qcan	ยกเลิกงานพิมพ์
qchk	แสดงสถานะของคิว การพิมพ์
qhld	พักหรือรีลีसानพิมพ์
qmov	ย้ายงานพิมพ์ไปยังคิวการพิมพ์อื่น
qpri	จัดลำดับความสำคัญงานในคิว การพิมพ์
qprt	เริ่มต้นงานพิมพ์

Live Partition Mobility with Host Ethernet Adapters

การใช้ Live Partition Mobility (LPM) พร้อมกับคุณลักษณะ Host Ethernet Adapters (HEA) ของซอฟต์แวร์ IBM PowerVM® คุณสามารถโอนย้าย AIX LPAR และแอปพลิเคชันที่โฮสต์จากฟิสิคัลพาร์ติชันไปยังฟิสิคัลพาร์ติชันอื่น ขณะที่ HEA ถูกกำหนดให้กับพาร์ติชันการโอนย้าย

ในระหว่างการย้าย HEA จะถูกลบออกจากพาร์ติชันที่ย้าย และ HEA จะถูกเรียกคืนบนพาร์ติชันเมื่อการย้ายเสร็จสมบูรณ์ อย่างไรก็ตาม การเชื่อมต่อเครือข่ายของคุณจะไม่ได้รับผลกระทบ

ข้อกำหนดสำหรับ Live Partition Mobility with HEA

ก่อนคุณสามารถเริ่มต้นโดยใช้ LPM with HEA คุณต้องตรวจสอบให้แน่ใจว่าสถานะแวดล้อมระบบตรงกับข้อกำหนดคอนฟิกูเรชันและการเข้าถึง

ข้อกำหนดพาร์ติชัน

- CEC ต้นทางและ CEC เป้าหมายต้องมีความสามารถในการย้ายพาร์ติชัน
- AIX LPAR ต้นทาง ต้องไม่มีฟิสิคัลรีซอร์สที่มีค่าติดตั้งเป็น บังคับ ใน โปรไฟล์
- AIX LPAR ต้นทาง ต้องไม่มีฟิสิคัลรีซอร์สใดๆ นอกเหนือจาก HEA

ข้อกำหนดการเข้าถึง

- คุณต้องมีสิทธิ์การใช้งานรากบนพาร์ติชันซึ่งคุณต้องการ ย้าย
- คุณต้องมีสิทธิ์ hscroot หรือสิทธิ์ที่เทียบเท่าซึ่งจำเป็น สำหรับการย้ายพาร์ติชันบน HMC ต้นทางและ HMC ปลายทาง

ข้อกำหนดคอนฟิกูเรชัน

- HEA ต้องไม่มีค่าติดตั้งเป็น บังคับ ในพาร์ติชันโปรไฟล์ แต่สามารถมีค่าติดตั้งเป็น ต้องการ ในโปรไฟล์ได้
- ต้องกำหนดคอนฟิก HEA ทั้งหมดภายใต้ EtherChannel เป็นอะแดปเตอร์หลัก
- อะแดปเตอร์หลักทั้งหมดใน EtherChannel ต้องเป็น HEA
- อะแดปเตอร์สำรองของ EtherChannel ต้องเป็นอะแดปเตอร์ Virtual Ethernet
- ต้องกำหนดคอนฟิกอย่างน้อยหนึ่ง EtherChannel โดยมี HEA เป็น อะแดปเตอร์หลักและอะแดปเตอร์ Virtual Ethernet เป็นอะแดปเตอร์สำรอง
- สนับสนุน EtherChannel สูงสุด 4 ตัว
- EtherChannel failover ต้องทำงานได้
- คุณต้องตรวจสอบว่าทั้งระบบต้นทางและระบบเป้าหมาย มีการตั้งค่าสำหรับการย้ายพาร์ติชัน
- หากคุณกำลังย้ายระหว่างสอง HMCs คุณต้องตั้งค่าการพิสูจน์ตัวตน SSH ระหว่าง HMC ต้นทางและ HMC รีโมต คุณต้องรันคำสั่ง `mkauthkeys` บน HMC ต้นทางก่อนเริ่มต้นการย้าย

การรัน Live Partition Mobility with HEA

คุณสามารถรัน LPM with HEA โดยใช้อินเตอร์เฟซ SMIT

อ่านบททวนหัวข้อ “ข้อกำหนดสำหรับ Live Partition Mobility with HEA” ในหน้า 669 ก่อนคุณพยายามใช้ LPM with HEA

เมื่อต้องการทำการย้ายพาร์ติชัน LPM with HEA ให้ทำขั้นตอนต่อไปนี้อย่างสมบูรณ์:

1. จากพร้อมต์คำสั่ง ป้อนพารามิเตอร์ SMIT ต่อไปนี้: **smitty migration** เพื่อแสดงเมนู Live Partition Mobility with Host Ethernet Adapter (HEA)
2. ระบุชื่อโฮสต์หรือ IP แอดเดรสของ HMC ต้นทาง
3. ระบุชื่อผู้ใช้ของ HMC ต้นทาง
4. ป้อน **ไม่** ถ้าระบบต้นทางและปลายทางมีการจัดการโดย HMC เดียวกันและไปยังขั้นตอนที่ 5 กด **ใช่** ถ้าระบบต้นทางและปลายทาง มีการจัดการโดย HMCs ที่แตกต่างกันและทำขั้นตอนต่อไปนี้อย่างสมบูรณ์:

หมายเหตุ: คุณต้องรันคำสั่ง **mkauthkeys** บน HMC ต้นทางก่อนคุณป้อน **ใช่**

- a. ระบุชื่อโฮสต์หรือ IP แอดเดรสของ HMC รีโมต
 - b. ระบุชื่อผู้ใช้ของ HMC ต้นทาง
5. ระบุชื่อของระบบต้นทาง
 6. ระบุชื่อของระบบปลายทาง
 7. ระบุชื่อของพาร์ติชันซึ่งคุณต้องการย้าย
 8. ป้อน **ไม่** ถ้าคุณต้องการทำการย้ายโดยไม่ตรวจสอบความถูกต้อง ป้อน **ใช่** ถ้าคุณต้องการทำการตรวจสอบความถูกต้องของการย้าย เท่านั้น หากคุณระบุ **ใช่** ระบบจะไม่ทำการย้าย และทำการตรวจสอบความถูกต้องเพียงอย่างเดียว

หมายเหตุ: คุณควรทำการตรวจสอบความถูกต้องของการย้ายพาร์ติชันก่อนทำการย้ายพาร์ติชัน

9. ตรวจสอบว่าฟิลด์ทั้งหมดมีข้อมูลที่ถูกต้อง และ กด **Enter** เพื่อทำการย้าย

หมายเหตุ: คุณจะได้รับการพร้อมต์ให้ป้อนรหัสผ่านสองครั้ง ป้อนรหัสผ่าน สำหรับชื่อผู้ใช้ของ HMC ต้นทางซึ่งคุณระบุก่อนหน้านี้ในขั้นตอน 4b

ในตัวอย่างนี้ พาร์ติชัน X กำลังย้ายจาก CEC C ที่จัดการโดย HMC A ไปยัง CEC D ที่จัดการโดย HMC B รันคำสั่ง **mkauthkeys** บน HMC A สำหรับการพิสูจน์ตัวตน ระหว่าง HMC A และ HMC B

สำหรับโปรเซสการย้ายนี้ มีการระบุค่าต่อไปนี้ใน SMIT:

```
Source HMC Hostname or IP address: A
Source HMC username: hscroot
Migration between two HMCs: yes
  Remote HMC hostname or IP address: B
  Remote HMC username: hscroot
Source system: C
Destination system: D
Migrating Partition name: X
Migration validation only: no
```

อีกตัวอย่างหนึ่งจะเป็น ถ้าพาร์ติชัน X กำลังย้ายจาก CEC C ที่จัดการโดย HMC A ไปยัง CEC D ที่จัดการโดย HMC A เช่นกัน

สำหรับโปรเซสการย้ายนี้ มีการระบุค่าต่อไปนี้ใน SMIT:

```
Source HMC Hostname or IP address: A
Source HMC username: hscroot
Migration between two HMCs: no
  Remote HMC hostname or IP address:
  Remote HMC username:
Source system: C
Destination system: D
Migrating Partition name: X
Migration validation only: no
```

ในกรณีที่การย้ายล้มเหลว ให้ปฏิบัติตามโพรซีเจอร์เพื่อทำ Live Partition Mobility จาก HMC ต้นทาง

การโอนย้ายในไคลเอ็นต์ NIM โดยใช้ LPM:

เมื่อใช้งาน Live Partition Mobility (LPM) เพื่อย้ายเครื่อง จากเซิร์ฟเวอร์แบบฟิสิคัลหนึ่งไปยังอีกเครื่องหนึ่งถูกกำหนดให้เป็นไคลเอ็นต์ Network Installation Management (NIM) ซึ่งผู้ดูแลระบบ NIM ต้องอัปเดตแอตทริบิวต์ *cpuid* สำหรับไคลเอ็นต์ NIM เพื่อแสดงค่าฮาร์ดแวร์ใหม่หลังจากที่การโอนย้าย LPM เสร็จสมบูรณ์

เมื่อต้องการอัปเดตแอตทริบิวต์ *cpuid* ให้ทำตาม ขั้นตอนต่อไปนี้:

1. บนไคลเอ็นต์ NIM ขอรหัส *cpuid* ID ใหม่โดยรัน คำสั่งต่อไปนี้:

```
uname -a
```

2. บน NIM หลัก ให้รันคำสั่งต่อไปนี้:

```
nim -o change -a cpuid=cpuid client
```

หมายเหตุ: โปรแกรมติดตั้งแบบเครือข่าย OS_install ไม่สนับสนุนการติดตั้งของระบบปฏิบัติการ Linux อีกต่อไป เนื่องจากการถอน การสนับสนุน Cluster Systems Management (CSM) ในระบบปฏิบัติการ AIX

การย้ายที่อะแด็ปเตอร์สำหรับ DLPAR

คุณต้องกำหนดค่ากราฟิกส์อะแด็ปเตอร์ก่อนย้ายที่ อะแด็ปเตอร์สำหรับการดำเนินการ dynamic logical partitioning (DLPAR)

ใช้คำสั่งต่อไปนี้เพื่อย้ายที่กราฟิกส์อะแด็ปเตอร์ แบบไดนามิก เช่น FC 5748:

1. ให้แน่ใจว่าไม่มีกระบวนการปัจจุบัน (Desktop และ Xserver) กำลังใช้กราฟิกส์อะแด็ปเตอร์ (ตัวอย่างเช่น /dev/lft0)
2. ตรวจสอบว่าคอนโซลไม่ถูกตั้งค่าเป็น lft0 เมื่อต้องการระบุ อินเทอร์เฟซที่ขึ้นต่อกันที่กำหนดหรือมีอยู่ (lft หรือ rcm) ป้อน คำสั่งต่อไปนี้:

```
lsdev -C | grep lft
```

```
lsdev -C | grep rcm
```

เมื่อต้องการจัดหาอะแด็ปเตอร์ Peripheral Component Interconnect (PCI) พาเรนต์หลังจากกราฟิกส์อะแด็ปเตอร์ถูกกำหนดให้คำสั่งต่อไปนี้:

```
odmget -q name=<cortina adapter name ตัวอย่างเช่น cor0> CuDv
```

คำสั่ง นี้ให้ข้อมูลเกี่ยวกับพาเรนต์และ cortina

3. เมื่อต้องการนำอินเทอร์เฟซที่ขึ้นต่อกันทั้งหมดออกเพื่อให้อะแดปเตอร์ พร้อมใช้สำหรับการดำเนินการ DLPAR ป้อนคำสั่งต่อไปนี้:

```
# pdisable lft0

# rmdev -l rcm0
rcm0 Defined

# rmdev -l lft0
lft0 Defined

# rmdev -Rdl pci23
cor0 deleted
pci23 deleted
```

อินเทอร์เฟซคอนโซลการจัดการสามารถใช้สำหรับการดำเนินการ DLPAR บนกราฟิกส์อะแดปเตอร์

อุปกรณ์ Loopback

อุปกรณ์ loopback คืออุปกรณ์ที่สามารถใช้เป็นอุปกรณ์บล็อก เพื่อเข้าถึงไฟล์

ไฟล์ loopback สามารถมีรูปภาพ ISO, รูปภาพดิสก์ ระบบไฟล์ หรือรูปภาพโลจิคัลวอลุ่ม ตัวอย่างเช่น โดยการแนบรูปภาพซีดีรอม ISO เข้ากับอุปกรณ์ loopback และติดตั้ง คุณสามารถเข้าถึงรูปภาพ ได้ในวิธีเดียวกับที่คุณสามารถเข้าถึงอุปกรณ์ซีดีรอม

ใช้คำสั่ง **loopmount** เพื่อสร้างอุปกรณ์ loopback เพื่อผูกไฟล์ที่ระบุเข้ากับอุปกรณ์ loopback และเพื่อติดตั้ง อุปกรณ์ loopback ใช้คำสั่ง **loopumount** เพื่อถอนติดตั้งไฟล์รูปภาพที่ติดตั้งก่อนหน้านี้บนอุปกรณ์ loopback และเพื่อเอาอุปกรณ์ออก ไม่มีการจำกัดจำนวนของอุปกรณ์ loopback ใน AIX อุปกรณ์ loopback ไม่เคยมีการสร้างโดยค่าดีฟอลต์ คุณ ต้องสร้างอุปกรณ์เอง ขนาดบล็อกของอุปกรณ์ loopback เป็น 512 ไบต์เสมอ

นอกจากนี้ ยังสามารถสร้างอุปกรณ์ใหม่โดยใช้คำสั่ง **mkdev** เปลี่ยนโดยใช้คำสั่ง **chdev** และเอาออกโดยใช้ คำสั่ง **rmdev** หลังจากสร้างอุปกรณ์แล้ว สามารถติดตั้ง อุปกรณ์นั้นเพื่อเข้าถึงรูปภาพที่ต้องการ หรือใช้เป็นอุปกรณ์บล็อก สำหรับ I/O ดิบ และสามารถดึงข้อมูลเกี่ยวกับรูปภาพที่ต้องการ โดยใช้คำสั่ง **ioctl (IOCINFO)**

ข้อจำกัดต่อไปนี้จะใช้กับอุปกรณ์ loopback ใน AIX:

- ไม่สนับสนุนคำสั่ง **varyonvg** บนรูปภาพดิสก์
- รูปภาพ CD ISO, และ DVD UDF+ISO, และ CD/DVD อื่นได้รับการสนับสนุนในรูปแบบอ่านอย่างเดียวเท่านั้น
- ไฟล์รูปภาพสามารถเชื่อมโยงกับอุปกรณ์ loopback เพียงรายการเดียวเท่านั้น
- ไม่สนับสนุนอุปกรณ์ loopback ในพาร์ติชันเวิร์กโหลด

ข้อมูลที่เกี่ยวข้อง:

คำสั่ง **loopmount**

คำสั่ง **loopumount**

คำสั่ง **ioctl**

AIX Event Infrastructure สำหรับ AIX และ AIX clusters-AHAFS

AIX Event Infrastructure สำหรับ AIX และคลัสเตอร์ AIX ประกอบด้วยเฟรมเวิร์กการมอนิเตอร์เหตุการณ์สำหรับการมอนิเตอร์เหตุการณ์ที่กำหนดไว้ล่วงหน้า และที่ผู้ใช้กำหนดเอง

บทนำเกี่ยวกับ AIX Event Infrastructure

AIX Event Infrastructure คือกรอบงานการมอนิเตอร์เหตุการณ์สำหรับการมอนิเตอร์ที่กำหนดไว้ก่อนและเหตุการณ์ที่ผู้ใช้กำหนดเอง

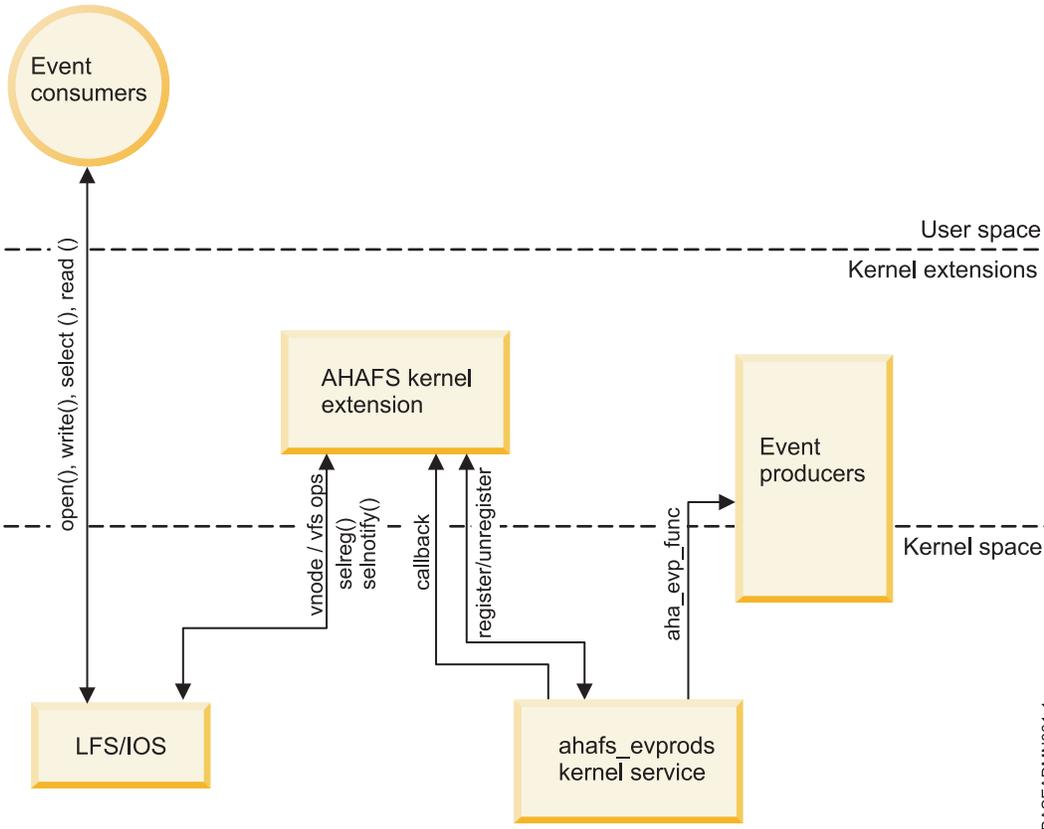
ใน AIX Event Infrastructure เหตุการณ์ถูกนิยามเป็นการเปลี่ยนแปลงสถานะใดๆ หรือค่าที่สามารถตรวจพบโดยเคอร์เนล หรือส่วนขยายเคอร์เนล ณ เวลาที่การเปลี่ยนแปลง เกิดขึ้น เหตุการณ์ที่สามารถมอนิเตอร์ถูกแสดงเป็นไฟล์ในระบบไฟล์ pseudo ข้อดีบางส่วนของ AIX Event infrastructure คือ:

- ไม่มีความจำเป็นสำหรับการโพลแบบคงที่ ผู้ใช้มอนิเตอร์เหตุการณ์ ถูกแจ้งเตือนเมื่อเหตุการณ์เหล่านี้เกิดขึ้น
- ข้อมูลโดยละเอียดเกี่ยวกับเหตุการณ์ (เช่น การติดตั้งแบบสแต็ก และข้อมูลผู้ใช้และกระบวนการ) ถูกจัดเตรียมไว้ให้กับผู้ใช้ที่มอนิเตอร์เหตุการณ์
- อินเทอร์เน็ตระบบไฟล์ที่มีอยู่จะถูกใช้เพื่อที่จะได้ไม่จำเป็นต้องใช้ application programming interface (API) ใหม่
- การควบคุมถูกทำกับ AIX Event Infrastructure ณ เวลาที่เหตุการณ์เกิดขึ้นอย่างแน่นอน

คอมโพเนนต์ AIX Event Infrastructure

AIX Event Infrastructure ประกอบด้วยคอมโพเนนต์ทั้งสี่ตัวต่อไปนี้:

- ส่วนขยายเคอร์เนลที่นำระบบไฟล์ pseudo ไปใช้
- ผู้บริโภครหัสเหตุการณ์ที่ใช้เหตุการณ์
- ผู้สร้างเหตุการณ์ที่สร้างเหตุการณ์
- คอมโพเนนต์เคอร์เนลที่ใช้เป็นอินเทอร์เน็ตระหว่าง ส่วนขยายเคอร์เนลและผู้สร้างเหตุการณ์



BASEADMIN001-1

ส่วนขยายเคอร์เนล AIX Event Infrastructure

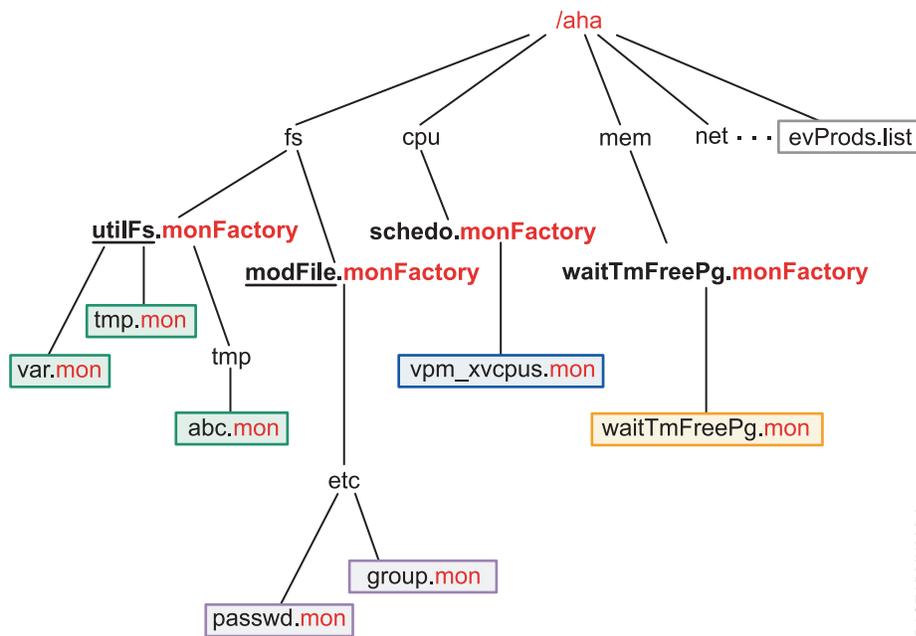
ส่วนขยายเคอร์เนล AIX Event Infrastructure นำระบบไฟล์ pseudo ไปใช้งาน

เหตุการณ์ทั้งหมดถูกแทนค่าเป็นไฟล์ในระบบไฟล์นี้ ซึ่งมีสี่ชนิดของอ็อบเจกต์ไฟล์:

- ไฟล์ .list: มีเพียงหนึ่งไฟล์ .list ในระบบไฟล์ pseudo evProds.list เท่านั้น นี่คือไฟล์พิเศษซึ่งเมื่ออ่านแล้วจะส่งคืนชื่อของผู้สร้างเหตุการณ์ที่กำหนดไว้ในปัจจุบันทั้งหมด
- ไดร็อกทอรี .monFactory: แฟกทอรีมอนิเตอร์ คือชนิดของไดเรกทอรีพิเศษ เหล่านี้คือการแทนค่าไดเรกทอรีของผู้สร้างเหตุการณ์ แฟกทอรีมอนิเตอร์และไดเรกทอรีย่อยหลัก ถูกสร้างขึ้นแบบอัตโนมัติสำหรับผู้ใช้
- ไดร็อกทอรีย่อย: ไดร็อกทอรีย่อยถูกใช้เพื่อให้ง่ายต่อการจัดการ และเพื่อแสดงชื่อพาธเต็มสำหรับไฟล์มอนิเตอร์ (โปรดดูไฟล์ .mon)
- ไฟล์ .mon: ไฟล์มอนิเตอร์แทนค่าเหตุการณ์ที่สามารถมอนิเตอร์ได้ ชื่อพาธเต็มของไฟล์มอนิเตอร์จากแฟกทอรีมอนิเตอร์หลัก ครอบคลุมขยาย .mon ที่เป็นการแทนค่าแบบเต็ม ของเหตุการณ์ที่ถูกมอนิเตอร์ ตัวอย่างเช่น ไฟล์ /aha/fs/modFile.monFactory/etc/password.mon ถูกใช้เพื่อมอนิเตอร์การแก้ไขไฟล์ /etc/passwd ไฟล์มอนิเตอร์สามารถอยู่ใกล้แฟกทอรีมอนิเตอร์ที่มีอยู่

ไม่มีไฟล์ปกติไฟล์อื่นที่สามารถสร้างขึ้นในระบบไฟล์ pseudo นี้ เนื่องจากระบบไฟล์ AIX Event Infrastructure คือระบบไฟล์ในหน่วยความจำ ซึ่งมีขนาดมากที่สุด 32 KB ของไอโหนดที่อาจมีอยู่แล้ว จำนวนของ inodes ที่ใช้จะถูกแสดงอยู่ในเอาต์พุตคำสั่ง df

ตัวอย่างของโครงสร้างของระบบไฟล์ AIX Event Infrastructure ที่แสดงอยู่ด้านล่าง:



หมายเหตุ:

ไฟล์ `evProds.list` มีอยู่ภายใต้รากของ ระบบไฟล์ และมีรายการของผู้สร้างเหตุการณ์ที่ ถูกนิยามไว้และสามารถใช้อยู่ภายใต้ อินสแตนซ์ของระบบปฏิบัติการ

การใช้อินเตอร์เฟส LFS AIX Event Infrastructure จะแปลข้อความอินพุตที่เขียนไปเป็นไฟล์มอนิเตอร์ ภายในข้อกำหนด คุณสมบัติเกี่ยวกับวิธีการที่ผู้ใช้ต้องการแจ้งให้ทราบถึงเหตุการณ์ ที่เกิดขึ้น เมื่อผู้ใช้ได้ใช้การเรียก `select()` หรือการบล็อก `read()` เพื่อกำหนดจุดเริ่มต้นของการมอนิเตอร์ AIX Event Infrastructure จะแจ้งเตือนผู้สร้าง เหตุการณ์ที่สอดคล้องกันเพื่อ เริ่มต้นการมอนิเตอร์เหตุการณ์ที่ระบุไว้

เมื่อตรวจพบเหตุการณ์ที่เกิดขึ้น AIX Event Infrastructure จะแจ้งเตือนผู้บริโศคที่รออยู่ทั้งหมดซึ่งมีเงื่อนไขการมอนิเตอร์ที่ ตรงกัน

ผู้บริโศคเหตุการณ์

ผู้บริโศคเหตุการณ์คือกระบวนการพื้นที่ของผู้ใช้ที่กำลังรอ ให้เหตุการณ์เกิดขึ้น

ผู้บริโศคตั้งค่าการมอนิเตอร์เหตุการณ์โดยเขียนข้อมูลไปยังไฟล์การมอนิเตอร์ โดยระบุวิธีและเวลาที่ควรแจ้งให้ทราบ ผู้ บริโศค อาจรอการแจ้งเตือนเหตุการณ์ในการเรียก `select()` หรือบล็อกการเรียก `read()`

AIX Event Infrastructure ไม่ใช่เซตที่ปลอดภัย กระบวนการไม่ควรใช้เซตจำนวนมาก เพื่อมอนิเตอร์เหตุการณ์ที่เหมือนกัน

ผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์คือส่วนของโค้ดภายในเคอร์เนล หรือส่วนขยายเคอร์เนลที่สามารถตรวจพบเหตุการณ์

เมื่อเหตุการณ์ที่ถูกมอนิเตอร์เกิดขึ้น ผู้สร้างเหตุการณ์แจ้งเตือนส่วนขยายเคอร์เนล AIX Event Infrastructure และส่งข้อมูลที่ เชื่อมโยงใดๆ เกี่ยวกับเหตุการณ์ เพื่อส่งผ่านไปยังผู้บริโศค

ในปัจจุบัน มีสองคลาสหลักของผู้สร้างเหตุการณ์:

- คลาสที่มอนิเตอร์สำหรับการเปลี่ยนสถานะ
- คลาสที่มอนิเตอร์สำหรับค่าที่เกินค่า thresholds ที่ผู้ใช้ระบุ

เซอร์วิสเคอร์เนล ahafs_evprods

เซอร์วิสเคอร์เนล ahafs_evprods อำนาจความเสถียรในเรื่องของการสื่อสารระหว่างส่วนขยายเคอร์เนล AIX Event Infrastructure และผู้สร้างเหตุการณ์

ในการอำนวยความสะดวกในเรื่องของการสื่อสารระหว่างส่วนขยายเคอร์เนล AIX Event Infrastructure และผู้สร้างเหตุการณ์ เซอร์วิสเคอร์เนล ahafs_evprods จะถูกเอ็กซ์พอร์ต ภายในเคอร์เนล รายการของผู้สร้างเหตุการณ์ที่ลงทะเบียนแล้วถูกใช้เพื่อค้นหาผู้สร้างเหตุการณ์และเพื่อส่งผ่านข้อมูลระหว่างผู้สร้างเหตุการณ์ และส่วนขยายเคอร์เนลตามความเหมาะสม

การตั้งค่า AIX Event Infrastructure

ขั้นตอนที่จำเป็นสำหรับการตั้งค่า AIX Event Infrastructure

เฉพาะขั้นตอนที่จำเป็นต่อการติดตั้ง AIX Event Infrastructure เท่านั้นคือ:

1. ติดตั้งชุดไฟล์ bos.ahafs
2. สร้างไดเรกทอรีสำหรับจุดต่อที่ต้องการ
3. รันคำสั่งต่อไปนี้:

```
mount -v ahafs <mount point> <mount point>
```

ตัวอย่าง

```
mkdir /aha  
mount -v ahafs /aha /aha
```

การต่อระบบไฟล์ AIX Event Infrastructure จะโหลดส่วนขยายเคอร์เนลแบบอัตโนมัติและสร้างแฟกทอรีการมอนิเตอร์ทั้งหมด เฉพาะอินสแตนซ์ของ ระบบไฟล์ AIX Event Infrastructure เท่านั้นอาจถูกต้อง ณ เวลานั้น ระบบไฟล์ AIX Event Infrastructure อาจต่อกับไดเรกทอรีปกติ แต่แนะนำว่า ผู้ใช้จะใช้ /aha

มุมมองระดับสูงของวิธีการทำงานของ AIX Event Infrastructure

ผู้บริโภคมอนิเตอร์เหตุการณ์จำนวนมากและผู้บริโภคจำนวนมาก อาจมอนิเตอร์เหตุการณ์เดียวกัน ผู้บริโภคแต่ละรายอาจมอนิเตอร์เหตุการณ์แบบอิงค่า ด้วยค่า threshold ที่แตกต่างกัน หากต้องการจัดการกับสิ่งนี้ ส่วนขยายเคอร์เนล AIX Event Infrastructure คงรายการของข้อมูลของผู้บริโภคที่สอดคล้อง:

- ชนิดการรอที่ระบุไว้ (WAIT_IN_READ หรือ WAIT_IN_SELECT)
- ระดับของข้อมูลที่ร้องขอ
- Threshold (s) ที่มอนิเตอร์ (หากกำลังมอนิเตอร์เหตุการณ์ของค่า threshold)
- บัฟเฟอร์ที่ใช้เพื่อจัดการข้อมูลเกี่ยวกับเหตุการณ์ที่เกิดขึ้น

ข้อมูลเหตุการณ์ถูกเก็บไว้ต่อกระบวนการดั่งนั้นกระบวนการอื่น มอนิเตอร์เหตุการณ์เดียวกันไม่ได้เปลี่ยนข้อมูลเหตุการณ์ เมื่อกระบวนการของผู้บริโภค อ่านจากไฟล์มอนิเตอร์ กระบวนการนั้นจะอ่านสำเนาของ ข้อมูลเหตุการณ์ของตนเองเท่านั้น

โพล์ของการมอนิเตอร์เหตุการณ์ทั่วไป

ขั้นตอนต่างๆ ในการมอนิเตอร์เหตุการณ์ที่กล่าวถึงในหัวข้อนี้

1. กระบวนการพยายามเปิดหรือสร้างไฟล์มอนิเตอร์
2. AIX Event Infrastructure ส่งผ่านชื่อพารของไฟล์มอนิเตอร์ไปยังผู้สร้างเหตุการณ์ตามความเหมาะสม ผู้สร้างเหตุการณ์ตรวจสอบว่าไฟล์มอนิเตอร์แทนค่าเหตุการณ์ที่ถูกต้อง และกระบวนการมีสิทธิ์ในการมอนิเตอร์เหตุการณ์
3. กระบวนการเขียนข้อมูลไปยังไฟล์โดยระบุ:
 - a. ชนิดของการรอ (`WAIT_TYPE=WAIT_IN_READ` หรือ `WAIT_TYPE=WAIT_IN_SELECT`) ชนิดของการรอที่เป็นค่าดีฟอลต์คือ `WAIT_IN_SELECT`
 - b. เมื่อต้องถูกแจ้งเตือน สำหรับการเปลี่ยนสถานะของเหตุการณ์ ผู้ใช้ต้องระบุ `CHANGED=YES` สำหรับเหตุการณ์ ค่า `threshold` ผู้ใช้อาจระบุ `THRESH_HI=<value>`, `THRESH_LO=<value>` หรือทั้งสองแบบขึ้นอยู่กับความสามารถของผู้สร้างเหตุการณ์ที่เชื่อมโยง ไม่มีค่าดีฟอลต์สำหรับข้อกำหนดคุณสมบัตินี้ และ `CHANGED=YES` และ `THRESH_*=<value>` อาจไม่ได้ระบุไว้ทั้งสองแบบ
4. AIX Event Infrastructure จะจัดสรรตอบล็อกกระบวนการเพื่อเก็บข้อมูลนี้ หากไม่มีอยู่สำหรับกระบวนการนี้และเติมข้อมูลด้วยข้อมูลที่เขียนโดยผู้ใช้
5. กระบวนการใช้ `select()` หรือการบล็อก `read()` บนไฟล์มอนิเตอร์
6. AIX Event Infrastructure จะเรียก `ahafs_evprods` เพื่อตรวจสอบว่า `thresholds` ที่ระบุไว้ถูกต้องสำหรับเหตุการณ์เฉพาะ ตัวอย่างเช่น ผู้สร้างเหตุการณ์ `utilFs` ไม่ยอมให้ใช้ค่า `> 100%` หาก `threshold` ไม่ถูกต้อง การเรียก `select()` หรือ `read()` จะส่งคืน `RC_FROM_EVPROD` และขึ้นอยู่กับสถานะการอ่านของไฟล์มอนิเตอร์ที่จะส่งคืน `EINVAL`
7. สำหรับผู้สร้างเหตุการณ์ค่า `threshold` เฉพาะค่าหนึ่งค่าเท่านั้น ที่ถูกส่งไปยังผู้สร้างเหตุการณ์สำหรับแต่ละ `threshold` (`hi` หรือ `lo`) สำหรับการมอนิเตอร์ ณ เวลา `select()` หรือการบล็อก `read()` AIX Event Infrastructure จะลงทะเบียน ค่า `threshold` ใหม่ด้วยผู้สร้างเหตุการณ์หากหนึ่งในสิ่งต่อไปนี้ เป็นจริง:
 - a. หากไม่มีกระบวนการที่กำลังมอนิเตอร์เหตุการณ์นี้ `threshold` ที่ระบุด้วยผู้บริโภคนี้จะถูกลงทะเบียนด้วยผู้สร้างเหตุการณ์
 - b. หากมีกระบวนการอื่นที่มอนิเตอร์เหตุการณ์นี้ หาก `THRESH_LO` ที่ระบุไว้โดยผู้บริโภคนี้ค่าสูงกว่า `threshold` ระดับต่ำที่มอนิเตอร์ หรือหาก `THRESH_HI` ที่ระบุโดยผู้บริโภคนี้ค่าต่ำกว่า `threshold` ระดับสูงที่ถูกมอนิเตอร์ในปัจจุบันของการเรียก AIX Event Infrastructure ภายในเซอร์วิสเคอร์เนล `ahafs_evprods` เพื่ออัปเดต `threshold` ที่ถูกมอนิเตอร์ในปัจจุบัน
8. อ้างอิงตามการส่งคืนจากเซอร์วิสเคอร์เนล `ahafs_evprods` ค่าจริง ของเหตุการณ์ที่ถูกส่งคืน (ในกรณีบางอย่าง) หากค่าจริงที่ส่งคืนตรงกันหรือเกินค่า `threshold` อย่างใดอย่างหนึ่ง การเรียก `read()` หรือ `select()` จะส่งคืนในทันทีและ `RC_FROM_EVPROD` ที่บันทึกไว้ในบัฟเฟอร์เหตุการณ์ จะเป็น `EALREADY` การเรียก `read()` หรือ `select()` จะส่งคืนค่า 0
9. สำหรับผู้สร้างเหตุการณ์การเปลี่ยนสถานะ ฟังก์ชัน `ahafs_evprods` จะถูกเรียกเพื่อลงทะเบียนเหตุการณ์
10. อ้างอิงตามการลงทะเบียนที่เป็นผลสำเร็จ AIX Event Infrastructure ตั้งค่าการแจ้งเตือน สำหรับผู้บริโภครอ `select()` การแจ้งเตือนจะถูกตั้งค่าผ่าน `selreg()` สำหรับผู้บริโภครอ การบล็อกในการเรียก `read()` เธรดจะถูกวางเพื่อ sleep ด้วย `e_sleep_thread()`
11. หากผู้สร้างเหตุการณ์ตรวจพบว่ามีเหตุการณ์เกิดขึ้น ผู้สร้างเหตุการณ์จะแจ้งเตือน AIX Event Infrastructure ด้วยข้อมูลที่เกี่ยวข้องกับเหตุการณ์ (เช่น ข้อมูลเกี่ยวกับกระบวนการที่ทริกเกอร์เหตุการณ์ ค่าปัจจุบัน โค้ดส่งคืน เป็นต้น).
12. ในระหว่างการ `callback` จากผู้สร้างเหตุการณ์นี้ AIX Event Infrastructure จะ:
 - a. กำหนด `ahaNode` ที่สอดคล้องกับเหตุการณ์

- b. คำนวณรายการของการรอผู้บริโภครีโอดเพื่อกำหนดค่า thresholds ที่ตรงกันหรือมีค่าเกินเพื่อแจ้งเตือนกับการเรียก `selnotify()` หรือ `e_wakeup()` ผู้บริโภครีโอดทั้งหมดที่รอเหตุการณ์ที่เปลี่ยนสถานะจะถูกแจ้งเตือน
13. หากกระบวนการได้ถูกแจ้งเตือนของเหตุการณ์ กระบวนการจะอ่านจากไฟล์มอนิเตอร์ เพื่อขอรับข้อมูลเหตุการณ์ ตัวอย่างของเอาต์พุตจากเหตุการณ์จะอยู่ด้านล่าง
- ตัวอย่าง เอาต์พุตสำหรับผู้สร้างเหตุการณ์การเปลี่ยนสถานะผู้ที่ระบุว่า การติดตามแบบสแต็กควรถูกนำมาใช้:

```
BEGIN_EVENT_INFO
TIME_tvsec=1269377315
TIME_tv_nsec=955475223
SEQUENCE_NUM=0
PID=2490594
UID=0
UID_LOGIN=0
GID=0
PROG_NAME=cat
RC_FROM_EVPROD=1000
END_EVENT_INFO
```

ตัวอย่างสำหรับเหตุการณ์ค่า threshold:

```
BEGIN_EVENT_INFO
TIME_tvsec=1269378095
TIME_tv_nsec=959865951
SEQUENCE_NUM=0
CURRENT_VALUE=2
RC_FROM_EVPROD=1000
END_EVENT_INFO
```

หมายเหตุ: เนื่องจากลักษณะของกระบวนการแจ้งเตือน ค่าปัจจุบันที่ส่งคืนอาจเก่าตามเวลาที่กระบวนการอ่าน ไฟล์มอนิเตอร์ ผู้ใช้ได้รับการแจ้งเตือนเมื่อค่า threshold ตรงกัน หรือมีค่าเกิน แต่การดำเนินการอื่นอาจเปลี่ยนค่าที่ต้องถูกมอนิเตอร์ ซึ่งจะไม่ถูกบล็อก

การใช้ AIX Event Infrastructure

ไดเรกทอรีทั้งหมดในระบบไฟล์ AIX Event Infrastructure มีโหมดการเข้าถึง 1777 และไฟล์ทั้งหมด มีโหมดการเข้าถึง 0666

ในปัจจุบัน ไดเรกทอรีทั้งหมดในระบบไฟล์ AIX Event Infrastructure มีโหมด 1777 และไฟล์มีโหมด 0666 โหมดเหล่านี้ไม่สามารถเปลี่ยนแปลงได้ แต่ความเป็นเจ้าของ ไฟล์และไดเรกทอรีอาจเปลี่ยนแปลงไป การควบคุมการเข้าถึงสำหรับเหตุการณ์การมอนิเตอร์ ถูกทำที่ระดับของผู้สร้างเหตุการณ์ เวลาที่สร้าง / เวลาที่แก้ไข ไม่ได้ถูกคงไว้และถูกส่งคืนเป็นเวลาปัจจุบันเสมอ เมื่อใช้ `stat()` บนอ็อบเจกต์ไฟล์ภายในระบบไฟล์ pseudo ความพยายามในการแก้ไขเวลาเหล่านี้จะถูกส่งคืนข้อผิดพลาด

การมอนิเตอร์เหตุการณ์

การสร้างไฟล์มอนิเตอร์

ไฟล์มอนิเตอร์ที่สอดคล้องกับเหตุการณ์ต้องถูกสร้าง เพื่อมอนิเตอร์เหตุการณ์

ก่อนการมอนิเตอร์เหตุการณ์ ไฟล์มอนิเตอร์ที่สอดคล้องกับเหตุการณ์ ต้องถูกสร้างขึ้น AIX Event Infrastructure สนับสนุน `open()` ที่มีแฟล็ก `O_CREAT` ดังตัวอย่าง เราจะทำตามขั้นตอนที่จำเป็นต่อการมอนิเตอร์ระบบไฟล์ /filesys/clj-fs สำหรับการใช้งาน 90%

- ไดรฟ์ที่จำเป็นต้องถูกสร้างขึ้น:
mkdir /aha/fs/utilFs.monFactory/filesys
- เปิดไฟล์ /aha/fs/utilFs.monFactory/filesys/clj-fs.mon

ก่อนที่ไฟล์มอนิเตอร์จะถูกสร้างขึ้น ส่วนขยายเคอร์เนล AIX Event Infrastructure จะเรียกผู้สร้างเหตุการณ์เพื่อกำหนดเหตุการณ์ที่ถูกร้องขอ ให้ถูกต้องและเพื่อกำหนดผู้ใช้ให้มีสิทธิ์เพียงพอ ต่อการมอนิเตอร์เหตุการณ์ที่ระบุไว้ นี่คือการผิดพลาดทั่วไป ที่ส่งคืนจากไฟล์มอนิเตอร์ที่สร้างหรือเปิด:

ตารางที่ 71. โค้ดสำหรับส่งคืน

รหัสส่งคืน	รายละเอียด
ENODEV	ไม่มีเหตุการณ์ที่สอดคล้องกับพาร ที่ระบุไว้ หมายเหตุ: ข้อผิดพลาด ENODEV อาจยังถูกส่งคืน เมื่อพยายามเปิดไฟล์มอนิเตอร์ที่มีอยู่เมื่อไม่มี เหตุการณ์อยู่อีกต่อไป
EPERM	ผู้ใช้ไม่มีสิทธิ์ ในการมอนิเตอร์เหตุการณ์ที่ระบุไว้
ENOTSUP	เหตุการณ์ที่ระบุไม่สนับสนุนการมอนิเตอร์ AIX Event Infrastructure

การเขียนลงในไฟล์มอนิเตอร์

กระบวนการของผู้บริโภคมอนิเตอร์จะเขียนลงในไฟล์มอนิเตอร์ เพื่อระบุวิธีและเวลาที่ควรถูกแจ้งเตือนเหตุการณ์

หากไฟล์มอนิเตอร์ที่ต้องการถูกสร้างและเปิดอยู่ กระบวนการของผู้บริโภคมอนิเตอร์จะเขียนลงในไฟล์มอนิเตอร์เพื่อระบุวิธีและเวลาที่ควรถูกแจ้งเตือนเหตุการณ์ ข้อมูลนี้ถูกเขียนลงในคู่ <key>=<value> ซึ่งอาจคั่นด้วย ; หรือช่องว่าง คู่ <key>=<value> ที่สามารถยอมรับได้มีดังต่อไปนี้:

ตารางที่ 72. คู่ <key>=<value> ที่สามารถยอมรับได้

ปุ่ม	ค่าที่สามารถยอมรับได้	รายละเอียด
CHANGED	YES	ระบุว่า เหตุการณ์ที่ต้องถูกมอนิเตอร์คือชนิด AHAFS_THRESHOLD_STATE และที่ผู้บริโภคมอนิเตอร์ควรถูกแจ้งเตือน เมื่อสถานะของเหตุการณ์เปลี่ยนแปลงไป
THRESH_HI	เลขจำนวนเต็มไม่ได้ลงนามขนาด 64 บิต ที่ระบุในรูปแบบเลขฐานสิบ	คีย์นี้ระบุค่า threshold สูงสำหรับเหตุการณ์ หากเหตุการณ์เข้าถึง threshold นี้ (เท่ากับ หรือมากกว่า) ผู้บริโภคมอนิเตอร์จะได้รับการแจ้งเตือน หมายเหตุ: ขณะที่สิ่งนี้คือเลขจำนวนเต็มขนาด 64 บิต ผู้สร้างเหตุการณ์บางรายอาจจำกัดค่าที่สามารถมอนิเตอร์ได้จริง ตัวอย่างเช่น ค่าที่สามารถยอมรับได้สำหรับ THRESH_HI สำหรับผู้สร้างเหตุการณ์ utilFs อยู่ระหว่าง 1 ถึง 100 แบบกว้าง ความถูกต้องของ threshold สำหรับผู้สร้างเหตุการณ์ไม่ได้ตรวจสอบในขณะที่เขียน แต่ตรวจสอบในขณะที่ select() หรือบล็อก read()

ตารางที่ 72. คู่ <key>=<value> ที่สามารถยอมรับได้ (ต่อ)

ปุ่ม	ค่าที่สามารถยอมรับได้	รายละเอียด
THRESH_LO	เลขจำนวนเต็มไม่ได้ลงนามขนาด 64 บิต ที่ระบุในรูปแบบเลขฐานสิบ	คีย์นี้ระบุค่า threshold ต่ำ สำหรับเหตุการณ์ หากเหตุการณ์เข้าถึง threshold นี้ (เท่ากับ หรือน้อยกว่า) ผู้บริโภครจะได้รับแจ้งเตือน หมายเหตุ: ขณะที่สิ่งนี้คือ เลขจำนวนเต็มขนาด 64 บิต ผู้สร้างเหตุการณ์บางรายอาจจำกัดค่าที่สามารถมอนิเตอร์ได้จริง ตัวอย่างเช่น ค่าที่สามารถยอมรับได้สำหรับ THRESH_LO สำหรับผู้สร้าง utilFs อยู่ระหว่าง 1 ถึง 100 แบบกว้าง ความถูกต้องของ threshold สำหรับผู้สร้างเหตุการณ์ไม่ได้ตรวจสอบในขณะที่เขียน แต่ตรวจสอบในขณะที่ select() หรือบล็อก read()
WAIT_TYPE	WAIT_IN_SELECT (ดีฟอลต์), WAIT_IN_READ	ระบุวิธีที่ผู้บริโภครจะรอ เหตุการณ์ หากผู้บริโภครต้องการบล็อกเหตุการณ์ในการเรียก select() ผู้บริโภครจะระบุ WAIT_IN_SELECT หากผู้บริโภครต้องการบล็อกเหตุการณ์ในการเรียก read() ผู้บริโภครจะระบุ WAIT_IN_READ
INFO_LVL	1, 2 (ดีฟอลต์) หรือ 3	ระบุข้อมูลเหตุการณ์ที่ควรบันทึกอยู่ใน บัฟเฟอร์ของผู้ใช้: <ul style="list-style-type: none"> • INFO_LVL=1 จะบันทึกการประทับเวลาของเหตุการณ์ หมายเลขลำดับ โค้ดส่งคืนของผู้สร้างเหตุการณ์ ข้อมูลผู้ใช้* ข้อมูลกระบวนการ* ชื่อโปรแกรม* และค่าปัจจุบันของเหตุการณ์ (หากมี) • INFO_LVL=2 จะบันทึกข้อมูลทั้งหมดจากระดับ 1 บวกกับข้อความจากผู้สร้างเหตุการณ์ หากมี • INFO_LVL=3 จะบันทึกข้อมูลทั้งหมดจากระดับ 2 บวกกับสแต็กของเหตุการณ์ หากมี หมายเหตุ: ข้อมูลผู้ใช้ ข้อมูลกระบวนการ ชื่อโปรแกรม และการติดตามแบบสแต็กพร้อมใช้งานสำหรับผู้สร้างเหตุการณ์ที่ระบุแฟล็ก AHAFS_STKTRACE_AVAILABLE ไม่ใช่ผู้สร้างเหตุการณ์ทั้งหมด ที่ส่งผ่านข้อความ โปรดดูเอกสารคู่มือผู้สร้างเหตุการณ์ เพื่อกำหนดข้อมูลที่พร้อมใช้งาน ตัวอย่างของเอาต์พุตเหตุการณ์ถูกแสดงในส่วนของ “การอ่านข้อมูลเหตุการณ์” ในหน้า 684

ตารางที่ 72. คู่ <key>=<value> ที่สามารถยอมรับได้ (ต่อ)

ปุ่ม	ค่าที่สามารถยอมรับได้	รายละเอียด
NOTIFY_CNT	-1 (ดีฟอลต์) หรือค่าใดๆ ระหว่าง 1 และ 32767 แบบกว้างๆ	NOTIFY_CNT ระบุจำนวนครั้งที่เหตุการณ์ควรเกิดขึ้นก่อนที่กระบวนการจะได้รับการแจ้งเตือน หากระบุค่า -1 ไว้ ผู้บริโภคระบุจะได้รับการแจ้งเตือนตามเหตุการณ์ที่เกิดขึ้นทุกครั้ง และทุกเหตุการณ์ที่เกิดขึ้นจะถูกบันทึกใน บัฟเฟอร์ผู้ใช้ หากผู้บริโภคระบุค่าบวกที่ไม่ใช่ศูนย์ ผู้บริโภคระบุจะถูกล็อกจนกว่าเหตุการณ์จะเกิดขึ้นตาม จำนวนครั้งที่ระบุไว้ หากเหตุการณ์เกิดขึ้นตามจำนวนครั้งที่ระบุไว้ ไม่มีเหตุการณ์เพิ่มเติมที่ถูกบันทึกไว้จนกว่าผู้บริโภคระบุจะบล็อก การเรียก select() หรือบล็อกการเรียก read() อื่นๆ โปรดดูส่วนของ “การรอเหตุการณ์” ในหน้า 682 สำหรับข้อมูลเพิ่มเติม
CLUSTER	YES	หากระบบคือส่วนของคลัสเตอร์ และคลัสเตอร์แอ็คทีฟอยู่ ผู้บริโภคระบุจะบุคิยที่ต้องได้รับการแจ้งเตือน ของเหตุการณ์นั้นในโหนดอื่นในคลัสเตอร์ ไม่ใช่ผู้สร้างเหตุการณ์ทั้งหมด ที่สนับสนุนการมอนิเตอร์คลัสเตอร์แบบกว้าง คุณลักษณะนี้ปิด ตามค่าดีฟอลต์ โปรดดูส่วนของ “เหตุการณ์คลัสเตอร์” ในหน้า 707 สำหรับข้อมูลเพิ่มเติม
BUF_SIZE	จำนวนเต็มบวก มากสุด 1048576	คิยนี้ระบุขนาดของบัฟเฟอร์ที่ควรใช้ เพื่อบันทึกข้อมูลเหตุการณ์ที่ระบุไว้ในหน่วยไบต์ ขนาดดีฟอลต์คือ 2048 และขนาดที่น้อยที่สุดที่จัดสรรไว้คือ 1024 ไบต์ แม้ว่าผู้บริโภคระบุจะรองขอขนาดที่เล็กกว่า

การเขียนข้อมูลลงในไฟล์มอนิเตอร์จะจัดเตรียมการเรียก select() หรือการบล็อก read() ตามลำดับ การมอนิเตอร์ไม่ได้เริ่มต้นจนกว่า select() หรือการบล็อก read() จะเสร็จสิ้น

ตัวอย่างเช่น หากต้องการมอนิเตอร์ระบบไฟล์ /fileys/clj-fs สำหรับเหตุการณ์ที่เกิดขึ้นในครั้งแรกของการใช้งาน 90% ในการบล็อกการเรียก read() สดริงต่อไปนี้จะถูกเขียนลงในไฟล์ /aha/fs/utlFs.monFactory/fileys/clj-fs.mon:

```
WAIT_TYPE=WAIT_IN_READ THRESH_HI=90 NOTIFY_CNT=1
```

โค้ดส่งคืนที่อาจเป็นไปได้จากการเรียก write() ไปยังไฟล์มอนิเตอร์:

ตารางที่ 73. โค้ดสำหรับส่งคืน

รหัสส่งคืน	รายละเอียด
EINVAL	หากค่าที่ไม่ถูกต้องถูกกำหนดไว้สำหรับคิยใดๆ ข้างต้น การเขียนไปยังไฟล์มอนิเตอร์จะล้มเหลวพร้อมกับ EINVAL นอกจากนี้ หากพารามิเตอร์การแจ้งเตือน (CHANGED หรือ THRESH_HI/LO) ที่ระบุไว้ไม่ตรงกับความสามารถของผู้สร้างเหตุการณ์ การเขียนจะล้มเหลวพร้อมกับ EINVAL ตัวอย่างเช่น หากผู้บริโภคระบุ CHANGED=YES สำหรับผู้สร้างเหตุการณ์ utlFs (ซึ่งมอนิเตอร์เฉพาะ THRESH_HI/LO) การเรียกการเขียน จะส่งคืน EINVAL การระบุ CLUSTER=YES โดยไม่มีคลัสเตอร์ที่แอ็คทีฟ จะยังส่งผลถึง EINVAL

ตารางที่ 73. โค้ดสำหรับส่งคืน (ต่อ)

รหัสส่งคืน	รายละเอียด
EBUSY	หากมีเรดอื่นในกระบวนการที่กำลังรอเหตุการณ์ การเขียนไปยังไฟล์มอเนเตอร์โดยเรดอื่นใดจะส่งคืน EBUSY
ESTALE	ไฟล์มอเนเตอร์ได้ถูกลบทิ้ง หากต้องการ มอเนเตอร์เหตุการณ์นี้ file descriptor จำเป็นต้องถูกปิด จากนั้นเปิดด้วย O_CREAT อีกครั้ง
ENOMEM	ไม่สามารถจัดสรรหน่วยความจำชั่วคราว หรือหน่วยความจำสำหรับบัฟเฟอร์เหตุการณ์
ENOSPC	โปรเซสสูงสุด 512 โปรเซสสามารถติดตามไฟล์การตรวจสอบ หากมี 512 โปรเซสอยู่ก่อนแล้วพร้อมกับไฟล์ที่เปิดโดยผู้ที่เขียน การเขียนจะล้มเหลวด้วย ENOSPC

การรอเหตุการณ์

ข้อกำหนดคุณสมบัติการมอเนเตอร์ถูกเขียนลงในไฟล์มอเนเตอร์

เมื่อข้อกำหนดคุณสมบัติการมอเนเตอร์ถูกเขียนลงในไฟล์มอเนเตอร์ได้เป็นผลสำเร็จ กระบวนการของผู้บริโภคมจะบล็อกเหตุการณ์ที่เกิดขึ้น โดยใช้ `select()` หรือ `read()` ผู้บริโภคจะได้รับแจ้งเตือน เหตุการณ์ที่เกิดขึ้นหากผู้บริโภคบล็อก `select()` หรือ `read()` เท่านั้น มีสามวิธีที่กระบวนการสามารถส่งคืนจาก `select()` หรือการบล็อก `read()`:

- เหตุการณ์ได้เกิดขึ้นตามจำนวนครั้งที่ระบุ
 - ไม่มีข้อผิดพลาดในกรณีนี้ ผู้บริโภคควรเขียนข้อมูลเหตุการณ์เพื่อกำหนดวิธีการจัดการกับ เหตุการณ์
- มีปัญหาเมื่อตั้งค่าเหตุการณ์ภายในส่วนขยายเคอร์เนล AIX Event Infrastructure

ข้อผิดพลาดอาจเกิดขึ้นก่อนที่จะลงทะเบียนเหตุการณ์เพื่อมอเนเตอร์ ด้วยผู้สร้างเหตุการณ์:

- `read()`
 - หากมีเรดอื่นที่รอการอ่าน การอ่านจะล้มเหลว ด้วย EBUSY
 - หากไม่มีการเขียนเกิดขึ้นก่อนการอ่านนี้ การอ่านจะส่งคืน ค่า 0 ด้วยขนาดของการอ่าน 0 ไบต์
- `select()`

หมายเหตุ:

เนื่องจากการนำไปใช้งานของการเรียกระบบ `select` เพื่อให้ `select()` ส่งคืนข้อผิดพลาด การดำเนินการของระบบไฟล์ต้องส่งคืน EBADF ตามผลลัพธ์ที่ได้ หากเงื่อนไขใดๆ ต่อไปนี้ตรงกัน `select()` จะส่งคืน EBADF

- เรดอื่นพยายามใช้ `select`
- ไฟล์มอเนเตอร์ได้ถูกลบทิ้งแล้ว
- ไม่มีการเขียนที่ทำการระบุข้อกำหนดคุณสมบัติของการมอเนเตอร์ไว้
- มีข้อผิดพลาดเกิดขึ้นเมื่อลงทะเบียนกับระบบย่อย IOS

ในกรณีนี้ไม่มีข้อมูลเหตุการณ์ที่ต้องอ่าน

- มีปัญหาในการตั้งค่าเหตุการณ์ด้วยผู้สร้างเหตุการณ์

หากมีความพยายาม ในการลงทะเบียนเหตุการณ์ด้วยผู้สร้างเหตุการณ์ รายการจะถูกบันทึกลงในบัฟเฟอร์สำหรับให้ผู้ใช้บริโภครอ่าน หากต้องการกำหนดข้อผิดพลาดที่เกิดขึ้น RC_FROM_EVPROD ที่ส่งคืนข้อมูลเหตุการณ์ควรถูกอ้างถึงใน

เอกสารคู่มือของผู้สร้างเหตุการณ์ หมายถึง เหตุเอาต์พุตเหตุการณ์สำหรับกรณีนี้จะมีการประทับเวลา หมายเลขลำดับ และโค้ดส่งคืนจากผู้สร้างเหตุการณ์โดยไม่พิจารณา INFO_LVL ที่ได้ระบุไว้ โปรดดู “การอ่านข้อมูลเหตุการณ์” ในหน้า 684 สำหรับตัวอย่าง

ในกรณีนี้ `select()` จะส่งคืน EBADF แต่ `read()` จะส่งคืนโค้ดส่งคืนจากการดำเนินการ `uio_move`

หากกระบวนการของผู้บริโภคได้รับ `NOTIFY_CNT` ที่มีค่ามากกว่า 1 ข้อมูลเกี่ยวกับแต่ละเหตุการณ์ที่เกิดขึ้นจะถูกบันทึกไว้ในบัฟเฟอร์ของผู้บริโภคจนกว่าจำนวนของคำร้องขอเหตุการณ์ จะเกิดขึ้น กระบวนการของผู้บริโภคจะถูกปลุกหากเหตุการณ์เกิดขึ้นตามจำนวนครั้งที่ร้องขอ หรือเกิดเหตุการณ์ที่ไม่พร้อมใช้งาน หากกระบวนการของผู้บริโภคถูกปลุก จะไม่มีการมอนิเตอร์เหตุการณ์จนกว่าจะเรียกใช้การเรียก `select()` หรือการบล็อก `read()` อีกครั้งสำหรับไฟล์มอนิเตอร์

หากผู้บริโภคระบุ `NOTIFY_CNT` ด้วยค่า -1 กระบวนการของผู้บริโภคจะถูกปลุกหลังจากเกิดเหตุการณ์ขึ้น และเหตุการณ์ที่เกิดขึ้นหลังจากที่เริ่มต้น `select()` หรือการบล็อก `read()` ที่เป็นผลสำเร็จจะถูกบันทึกลงในบัฟเฟอร์ของผู้บริโภค

การเรียก `select()` และ `read()` จะไม่บล็อก หากมีข้อมูลเหตุการณ์ที่ไม่ได้อ่านในบัฟเฟอร์

เหตุการณ์ที่เกิดขึ้นซึ่งไม่พร้อมใช้งาน

สำหรับผู้สร้างเหตุการณ์บางราย อาจมีเหตุการณ์ที่เกิดขึ้น ซึ่งทำให้เหตุการณ์ที่ถูกมอนิเตอร์ไม่ถูกต้อง

ตัวอย่างมีดังต่อไปนี้:

- การหยุดทำงานของกระบวนการ `processMon` และ `pidProcessMon`
- การถอดระบบไฟล์ที่มีไฟล์ที่ถูกมอนิเตอร์สำหรับ `modDir` และ `modFile`
- การถอดระบบไฟล์ที่ถูกมอนิเตอร์โดย `utilFs`
- การถอดหรือเปลี่ยนชื่อไฟล์ที่ต้องถูกมอนิเตอร์โดย `modDir` หรือ `modFile`
- การถอดผู้สร้างเหตุการณ์ที่ต้องถูกใช้เพื่อมอนิเตอร์เหตุการณ์ (`RC_FROM_EVPROD` จะเป็น `ENODEV` ในกรณีนี้)

หากเหตุการณ์ที่เกิดขึ้นซึ่งไม่พร้อมใช้งานได้ถูกทริกเกอร์ ผู้บริโภคอาจไม่มอนิเตอร์เหตุการณ์นั้นต่อไปจนกว่าเหตุการณ์นั้นจะกลับมาใช้งานได้ ตัวอย่างของเหตุการณ์ที่กลับมาใช้งานได้อีกครั้ง:

- การเมาท์ระบบไฟล์ที่ถูกมอนิเตอร์อีกครั้ง
- การสร้างระบบไฟล์ที่ถูกลบทิ้งอีกครั้ง
- การสร้างโปรเซสที่ถูกมอนิเตอร์อีกครั้ง

เมื่อเหตุการณ์โลคัลที่ไม่พร้อมใช้งานถูกกระตุ้น ส่วนขยายเคอร์เนล AIX Event Infrastructure จะ ลบไฟล์มอนิเตอร์ที่มีผลกระทบ เมื่อไฟล์มอนิเตอร์ ถูกลบทิ้ง ผู้บริโภคที่ยังคงมีไฟล์ที่เปิดอยู่ จะสามารถอ่านข้อมูลเหตุการณ์ แต่ไม่สามารถเขียนหรือบล็อก การรอสัญญาณสำหรับเหตุการณ์ที่เกิดขึ้นบนไฟล์ที่ถูกมอนิเตอร์ เมื่อเหตุการณ์ ที่เกิดขึ้นถูกพบโดยผู้บริโภค เหตุการณ์เหล่านั้นควรใช้การดำเนินการที่เหมาะสม (ซึ่งจะทำให้เหตุการณ์ กลับมาใช้งานได้อีกครั้ง) ปิด file descriptor สำหรับไฟล์มอนิเตอร์ และเปิดไฟล์มอนิเตอร์อีกครั้งด้วยแฟล็ก `O_CREAT`

เหตุการณ์ที่ไม่พร้อมใช้งานบนโลคัลจะทำให้ `select()` และ `read()` ถูกปลุกบล็อกก่อนที่จำนวนที่ร้องขอของเหตุการณ์ที่เกิดขึ้น จะถูกทริกเกอร์หากผู้บริโภคระบุ `NOTIFY_CNT > 1` ตัวอย่างเช่น หากผู้บริโภคกำลังมอนิเตอร์ไฟล์ `/foo` ด้วย `NOTIFY_CNT=3` ผู้บริโภคจะกลับคืนจาก `select()` หรือ `read()` หาก `/foo` ถูกถอนออกแม้ว่าจะเป็นเหตุการณ์ที่เกิดขึ้นในครั้งแรกด้วย `/foo`

การใช้ AIX Event Infrastructure สำหรับการโพล

AIX Event Infrastructure ไม่ต้องการให้ผู้สร้างเหตุการณ์คงไว้ซึ่งค่าปัจจุบันของเหตุการณ์ที่อาจถูกมอนิเตอร์เสมอ

สิ่งนี้คือการอนุญาตให้ใช้ประสิทธิภาพการทำงานที่ดีกว่าเนื่องจากผู้สร้างเหตุการณ์ ไม่มีค่าใช้จ่ายในการรักษาค่านี้หากไม่มีใครมอนิเตอร์ เหตุการณ์ที่เกิดขึ้นของเหตุการณ์

สิ่งนี้จะสร้างปัญหาเมื่อใช้การโพลแบบซิงโครนัส เนื่องจาก ไม่มีความเป็นไปได้ในการขอรับค่าปัจจุบันของเหตุการณ์ในทุกจุดของเวลา การเรียก poll() หรือ select() แบบซิงโครนัสจะถูกจัดการด้วยวิธีการต่อไปนี้:

- เมื่อกระบวนการใช้ select() หรือ poll() บนไฟล์การมอนิเตอร์ในครั้งแรก ส่วนขยายเคอร์เนล AIX Event Infrastructure จะลงทะเบียนเหตุการณ์นั้นสำหรับการมอนิเตอร์ ด้วยผู้สร้างเหตุการณ์
 - สำหรับผู้สร้างเหตุการณ์ค่า threshold ผู้ที่รักษาค่าปัจจุบันไว้ ค่าปัจจุบันจะถูกส่งคืนไปยังส่วนขยายเคอร์เนล AIX Event Infrastructure ตามการลงทะเบียนเหตุการณ์ ค่านี้จะถูกตรวจสอบกับค่า threshold ของผู้บริโภคนั้น หาก threshold ของผู้บริโภค มีค่าเกิน select() หรือ poll() จะบ่งชี้ว่ามีเหตุการณ์เกิดขึ้นและจะมี RC_FROM_EVPROD ของ EALREADY
- แฟล็ก POLLSYNC จะถูกข้าม เหตุการณ์ยังคงลงทะเบียนด้วย ผู้สร้างเหตุการณ์จนกว่าเหตุการณ์จะเกิดขึ้นตามจำนวนครั้งที่ระบุไว้ หรือจนกว่าผู้ใช้จะปิดไฟล์
- การเรียก poll() ลำดับต่อมาจะมีลักษณะการทำงานต่อไปนี้:
 - หากเหตุการณ์ยังไม่เกิดขึ้น การเรียกจะส่งคืน โดยไม่มีเหตุการณ์ที่ส่งคืน
 - หากเกิดเหตุการณ์เกิดขึ้นตามจำนวนครั้งที่ระบุไว้ ตั้งแต่การเรียก poll() ครั้งสุดท้าย เหตุการณ์ส่งคืนจะถูกตั้งค่าไว้เพื่อบ่งชี้ว่าเกิดเหตุการณ์ขึ้น

การอ่านข้อมูลเหตุการณ์

ข้อมูลเหตุการณ์ใน AIX Event Infrastructure ประกอบด้วยคู่ของคีย์เวิร์ด-ค่า

ข้อมูลเหตุการณ์อาจถูกอ่านเพียงครั้งเดียวและไม่มีเหตุการณ์เกิดขึ้นเพิ่มเติม ซึ่งข้อมูลที่มีค่าจะถูกส่งคืนในการเรียก read() ครั้งเดียว ตัวอย่างเช่น อาจพูดได้ว่า สองเหตุการณ์เกิดขึ้นก่อนที่ผู้บริโภครู้จักจากไฟล์การมอนิเตอร์ และแต่ละเหตุการณ์มี 256 ไบต์ของข้อมูลที่มีค่า หากผู้บริโภครีกรู้จัก read() สำหรับ 4096 ไบต์ เฉพาะ 256 ไบต์ ของเหตุการณ์แรกเท่านั้นจะถูกส่งคืนไปยังผู้ใช้ การเรียก read() ครั้งที่สองอาจจำเป็นต้องดำเนินการเพื่อขอรับข้อมูลจากเหตุการณ์ที่สอง อีออฟเซตใดๆ ที่กำหนดไว้จะถูกข้ามและข้อมูลจะถูกอ่านโดยเริ่มต้นจาก ไบต์สุดท้ายที่ไม่ได้อ่าน

ข้อมูลเหตุการณ์จะอยู่ที่ 4096 ไบต์โดยส่วนใหญ่ แม้ว่าเหตุการณ์ส่วนใหญ่ จะมีขนาดเล็กกว่ามาก (< 512 ไบต์) ซึ่งขอแนะนำว่า เมื่อการอ่านเหตุการณ์ที่มีบัฟเฟอร์ที่มีขนาดใหญ่เพียงพอควรถูกใช้เพื่อหลีกเลี่ยง การอ่านส่วนของเหตุการณ์เท่านั้น

ข้อมูลเหตุการณ์ใน AHAFS ประกอบขึ้นด้วยคู่ของ คีย์เวิร์ด = ค่า พร้อมกับข้อยกเว้นของ BUF_WRAP, EVENT_OVERFLOW, BEGIN_EVENT_INFO, END_EVENT_INFO, BEGIN_EVPROD_INFO, END_EVPROD_INFO และ STACK_TRACE ที่เป็นคีย์เวิร์ดพิเศษที่ไม่มีค่าใดๆ ที่นี่คือคีย์เวิร์ด ที่คุณอาจมองเห็นในข้อมูลเหตุการณ์:

ตารางที่ 74. คีย์เวิร์ด

ปุ่ม	ค่า	รายละเอียด
BUF_WRAP	ไม่มี	บัฟเฟอร์ผู้บริโภครีโกลถูกจัดการเช่นเดียวกับบัฟเฟอร์ที่เป็นวงเวียน หากข้อมูลที่ไม่ได้อ่านใดๆ ถูกเขียนทับโดยข้อมูลเหตุการณ์ล่าสุด คีย์เวิร์ดนี้จะอยู่ถัดจากสตริงที่ส่งคืนโดยการเรียก read() แม้ว่าผู้บริโภครีโกลได้อ่านรายการก่อนหน้านี้แล้วเป็นบางส่วน การเรียก read() ในลำดับถัดมาจะส่งคืนเหตุการณ์ถัดไปทั้งหมด
EVENT_OVERFLOW	ไม่มี	หากข้อมูลเหตุการณ์มีขนาดใหญ่เกินกว่าจะอยู่ภายในบัฟเฟอร์ข้อมูลเหตุการณ์ของผู้บริโภครีโกล คีย์เวิร์ดนี้จะถูกส่งคืนจาก read() ในครั้งแรก read() ถัดไปจะส่งคืนข้อมูลเหตุการณ์ที่ไม่สามารถอยู่ภายในบัฟเฟอร์ได้พอดี หมายเหตุ: หาก EVENT_OVERFLOW ถูกพบ สตริงสุดท้าย END_EVENT_INFO จะไม่มีอยู่
BEGIN_EVENT_INFO	ไม่มี	คีย์เวิร์ดจะหมายถึงจุดเริ่มต้นของข้อมูลสำหรับเหตุการณ์ที่เกิดขึ้น
END_EVENT_INFO	ไม่มี	คีย์เวิร์ดจะหมายถึงจุดสิ้นสุดของข้อมูลสำหรับเหตุการณ์ที่เกิดขึ้น
TIME_tvsec TIME_tvnsec	จำนวนเต็ม	ฟิลด์สองฟิลด์เหล่านี้บันทึกการประทับเวลาของเหตุการณ์ที่เกิดขึ้นในหน่วยวินาทีและนาโนวินาทีตั้งแต่ Epoch
SEQUENCE_NUM	จำนวนเต็ม	ฟิลด์นี้บันทึกจำนวนครั้งที่เหตุการณ์เกิดขึ้น ตั้งแต่ select() หรือบล็อก read() เป็นผลสำเร็จในครั้งแรก จำนวนนี้ถูกรีเซ็ตไปเป็น 0 หาก select() หรือการบล็อกการเรียก read() ล้มเหลวหรือหากผู้บริโภครีโกลหยุดการมอนิเตอร์เหตุการณ์ (โดยเขียนทับข้อกำหนดคุณลักษณะของการมอนิเตอร์เหตุการณ์ หรือผ่านการมาถึงเหตุการณ์ ที่เกิดขึ้นซึ่งจำนวนนับเท่ากับ NOTIFY_CNT ที่ระบุไว้)
PID	จำนวนเต็ม	ID กระบวนการของกระบวนการที่ทริกเกอร์เหตุการณ์ที่เกิดขึ้น พร้อมใช้งานกับผู้สร้างเหตุการณ์ ผู้ที่ระบุความสามารถ AHAFS_STKTRACE_AVAILABLE ไว้เท่านั้น แต่ไม่ใช้ความสามารถ AHAFS_CALLBACK_INTRCNTX
UID	จำนวนเต็ม	ID ผู้ใช้ที่ได้รับผลของผู้ใช้ที่ทริกเกอร์เหตุการณ์ที่เกิดขึ้น พร้อมใช้งานกับผู้สร้างเหตุการณ์ ผู้ที่ระบุความสามารถ AHAFS_STKTRACE_AVAILABLE ไว้เท่านั้น แต่ไม่ใช้ความสามารถ AHAFS_CALLBACK_INTRCNTX
UID_LOGIN	จำนวนเต็ม	ID ผู้ใช้ของผู้ใช้ที่ทริกเกอร์เหตุการณ์ที่เกิดขึ้น พร้อมใช้งานกับผู้สร้างเหตุการณ์ ผู้ที่ระบุความสามารถ AHAFS_STKTRACE_AVAILABLE ไว้เท่านั้น แต่ไม่ใช้ความสามารถ AHAFS_CALLBACK_INTRCNTX

ตารางที่ 74. คีย์เวิร์ด (ต่อ)

ปุ่ม	ค่า	รายละเอียด
GID	จำนวนเต็ม	ID กลุ่มที่ได้รับผลของผู้ใช้ที่ทริกเกอร์ เหตุการณ์ที่เกิดขึ้น พร้อมใช้งานกับผู้สร้างเหตุการณ์ ผู้ที่ระบุความสามารถ AHAFS_STKTRACE_AVAILABLE ไว้เท่านั้น แต่ไม่ใช่ความสามารถ AHAFS_CALLBACK_INTRCNTX
PROG_NAME	สตริง	ชื่อของกระบวนการที่ทริกเกอร์ เหตุการณ์ที่เกิดขึ้น พร้อมใช้งานกับผู้สร้างเหตุการณ์ ผู้ที่ระบุความสามารถ AHAFS_STKTRACE_AVAILABLE ไว้เท่านั้น แต่ไม่ใช่ความสามารถ AHAFS_CALLBACK_INTRCNTX
CURRENT_VALUE	เลขจำนวนเต็ม 64 บิตที่ไม่ลงนามในรูปแบบเลขฐานสิบ	คีย์นี้ใช้สำหรับผู้สร้างเหตุการณ์ AHAFS_THRESHOLD_VALUE เท่านั้นและจะส่งคืนค่าปัจจุบันของเหตุการณ์ในเวลาที่จะพบเหตุการณ์ที่เกิดขึ้น หมายถึง เนื่องจากหน่วงเวลา ระหว่าง เวลาที่กระบวนการแจ้งเตือนและเวลาที่อ่านข้อมูลเหตุการณ์ ค่าปัจจุบันที่เกิดขึ้นจริงของเหตุการณ์อาจเปลี่ยนแปลงไป
RC_FROM_EVPROD	เลขจำนวนเต็ม 32 บิตในรูปแบบเลขฐานสิบ	โค้ดส่งคืนนี้มาจากผู้สร้างเหตุการณ์ ซึ่งเป็นผลลัพธ์ของข้อผิดพลาดเมื่อพยายามตั้งค่าเหตุการณ์ หรือเป็นผลลัพธ์ของเหตุการณ์ที่เกิดขึ้น อย่างไรก็ตาม โดยทั่วไป โค้ดส่งคืน ที่น้อยกว่า 256 บ่งชี้ถึงข้อผิดพลาดเมื่อพยายามลงทะเบียนเหตุการณ์ ด้วยผู้สร้างเหตุการณ์ ผู้สร้างเหตุการณ์บางรายจะส่งคืนโค้ดที่มากกว่า 256 เพื่อจัดเตรียมข้อมูลเพิ่มเติมเกี่ยวกับเหตุการณ์ที่เกิดขึ้น โค้ดส่งคืนเหล่านี้ จะถูกจัดทำเป็นเอกสารอยู่ใน <code>sys/ahafs_evProds.h</code>
BEGIN_EVPROD_INFO END_EVPROD_INFO	สตริง*	คีย์เวิร์ดสองตัวนี้ทำเครื่องหมายที่จุดเริ่มต้น และจุดสิ้นสุดของสตริงที่ส่งคืนโดยผู้สร้างเหตุการณ์ ซึ่งจะเป็นบรรทัดใหม่เสมอ และอยู่หลัง BEGIN_EVPROD_INFO และก่อน END_EVPROD_INFO สำหรับผู้บริโคที่ระบุ CLUSTER=YES สิ่งนี้คือข้อมูลโหนดที่จะถูกกำหนดไว้
STACK_TRACE	สตริง*	สำหรับผู้บริโคที่ระบุ INFO_LVL=3 พร้อมกับผู้สร้างเหตุการณ์ที่ระบุความสามารถ AHAFS_STKTRACE_AVAILABLE แต่ไม่ใช่ความสามารถ AHAFS_CALLBACK_INTRCNTX การติดตามแบบสแต็กของเหตุการณ์ที่เกิดขึ้นจะถูกจัดเตรียมไว้ คีย์เวิร์ด STACK_TRACE บ่งชี้ว่า มีข้อมูลเหตุการณ์ที่เหลืออยู่จนกว่าสตริง END_EVENT_INFO จะเป็นสแต็กของเหตุการณ์ที่เกิดขึ้น

ตารางที่ 74. คีย์เวิร์ด (ต่อ)

ปุ่ม	ค่า	รายละเอียด
NUM_EVDROPS_INTRCNTX	จำนวนเต็ม	คีย์เวิร์ดนี้แสดงจำนวนของเหตุการณ์ interrupt-context ที่เกิดขึ้น ซึ่งลดลงนับตั้งแต่เวลาที่ระบุไว้ โดยคีย์เวิร์ด TIMEO_tvsec และ TIMEO_tvnsec ในรายงาน เหตุการณ์ที่เกิดขึ้นลดลงก็ต่อเมื่อ ความถี่ของเหตุการณ์ที่เกิดขึ้นสูง
TIMEO_tvsec TIMEO_tvnsec	จำนวนเต็ม	คีย์เวิร์ดเหล่านี้บันทึกเวลาประทับของเหตุการณ์ ครั้งแรกที่เกิดขึ้นลดลงเป็นวินาทีและนาโนวินาที ตั้งแต่ Epoch คีย์เวิร์ดเหล่านี้ถูกแจ่งพร้อมคีย์เวิร์ด NUM_EVDROPS_INTRCNTX

ทำซ้ำการรวบรวมเหตุการณ์:

หากเหตุการณ์เดียวกันเกิดขึ้นหลายครั้งก่อนที่ลูกค้าจะอ่านข้อมูล รายการทำซ้ำจะถูกรวบรวมไปเป็น หนึ่งรายการ การรวบรวมนี้ถูกบ่งชี้โดยหมายเลขลำดับที่ไม่ได้เรียงลำดับ โดยไม่มีคีย์เวิร์ด BUF_WRAP ที่สอดคล้องกัน การประทับเวลา และหมายเลขลำดับจะสะท้อนถึงเหตุการณ์ที่เกิดขึ้นเมื่อเร็ว ๆ นี้ ของเหตุการณ์

ตัวอย่างข้อมูลเหตุการณ์

สำหรับผู้สร้างเหตุการณ์ที่ระบุ AHAFS_THRESHOLD_STATE และ AHAFS_STKTRACE_AVAILABLE และจะส่งผ่านไปยังผู้บริโภคเหตุการณ์ ระดับทั้งสามระดับของเอาต์พุต จะดูคล้ายกับสิ่งนี้:

INFO_LVL=1	INFO_LVL=2	INFO_LVL=3
BEGIN_EVENT_INFO TIME_tvsec=1269863383 TIME_tvnsec=455993143 SEQUENCE_NUM=0 PID=6947038 UID=0 UID_LOGIN=0 GID=0 PROG_NAME=cat RC_FROM_EVPROD=1000 END_EVENT_INFO	BEGIN_EVENT_INFO TIME_tvsec=1269863383 TIME_tvnsec=455993143 SEQUENCE_NUM=0 PID=6947038 UID=0 UID_LOGIN=0 GID=0 PROG_NAME=cat RC_FROM_EVPROD=1000 BEGIN_EVPROD_INFO event producer message here END_EVPROD_INFO END_EVENT_INFO	BEGIN_EVENT_INFO TIME_tvsec=1269863383 TIME_tvnsec=455993143 SEQUENCE_NUM=0 PID=6947038 UID=0 UID_LOGIN=0 GID=0 PROG_NAME=cat RC_FROM_EVPROD=1000 BEGIN_EVPROD_INFO event producer message here END_EVPROD_INFO STACK_TRACE ahafs_prod_callback+3C4 ahafs_cbfm_wrapper+30 ahafs_vn_write+204 vnop_rdw+7E4 vno_rw+B4 rwuio+12C rdwr+184 kewrite+16C .svc_instr write+1A4 _xwrite+6C _xf1sbuf+B0 __f1sbuf+9C copyopt_ascii+2C0 scat+388 main+11C __start+68 END_EVENT_INFO

สำหรับผู้สร้างเหตุการณ์ที่ระบุ AHAFS_THRESHOLD_VALUE_HI และไม่ได้ระบุ AHAFS_STKTRACE_AVAILABLE และจะส่งผ่านข้อความ ไปยังผู้บริโภคเหตุการณ์ ระดับทั้งสามระดับของเอาต์พุตจะดูคล้ายกับสิ่งนี้:

INFO_LVL=1	INFO_LVL=2	INFO_LVL=3
BEGIN_EVENT_INFO TIME_tvsec=1269866715 TIME_tvnsec=16678418 SEQUENCE_NUM=0 CURRENT_VALUE=3 RC_FROM_EVPROD=1000 END_EVENT_INFO	BEGIN_EVENT_INFO TIME_tvsec=1269866715 TIME_tvnsec=16678418 SEQUENCE_NUM=0 CURRENT_VALUE=3 RC_FROM_EVPROD=1000 BEGIN_EVPROD_INFO event producer message here END_EVPROD_INFO END_EVENT_INFO	BEGIN_EVENT_INFO TIME_tvsec=1269866715 TIME_tvnsec=16678418 SEQUENCE_NUM=0 CURRENT_VALUE=3 RC_FROM_EVPROD=1000 BEGIN_EVPROD_INFO event producer message here END_EVPROD_INFO END_EVENT_INFO

รูปแบบข้อผิดพลาด:

หากมีข้อผิดพลาดจากผู้สร้างเหตุการณ์ ผู้สร้างเหตุการณ์ทั้งหมด มีรูปแบบต่อไปนี้สำหรับ INFO_LVL ทั้งหมด:

```
BEGIN_EVENT_INFO
TIME_tvsec=1269868036
TIME_tvnsec=966708948
SEQUENCE_NUM=0
RC_FROM_EVPROD=20
END_EVENT_INFO
```

หากผู้บริโภครอค่าล้นมอนิเตอร์เหตุการณ์ **AHAFS_THRESHOLD_VALUE** และค่าปัจจุบันยังคงเกิน **threshold** ที่ร้องขอ รูปแบบขอผิดพลาดจะถูกใช้เพื่อบันทึกเหตุการณ์ **EALREADY** นี้:

```
BEGIN_EVENT_INFO
TIME_tvsec=1269868036
TIME_tvnsec=966708948
SEQUENCE_NUM=0
CURRENT_VALUE=1
RC_FROM_EVPROD=56
END_EVENT_INFO
```

BUF_WRAP และ EVENT_OVERFLOW:

หากข้อมูลที่ไม่ได้อ่านถูกเขียนทับโดยข้อมูลจากเหตุการณ์ใหม่ที่เกิดขึ้น คีย์เวิร์ด **BUF_WRAP** จะถูกเอาต์พุตในครั้งแรกจาก **read()** บนไฟล์การมอนิเตอร์ หากมีการตัดบัพเฟอร์ **AND** ที่โอเวอร์โฟลว์เหตุการณ์ **BUF_WRAP** จะมาก่อนเสมอ แล้วตามด้วย **EVENT_OVERFLOW** นี่คือการตัวอย่างเอาต์พุตจาก **read()** ในกรณีที่เรามีทั้งการตัดบัพเฟอร์และโอเวอร์โฟลว์เหตุการณ์:

read() อันดับแรกจะส่งคืน:

```
BUF_WRAP
```

read() อันดับสองจะส่งคืน:

```
EVENT_OVERFLOW
```

read() อันดับสามจะส่งคืนข้อมูลเหตุการณ์ที่สามารถอยู่ภายในบัพเฟอร์ได้พอดี:

```
BEGIN_EVENT_INFO
TIME_tvsec=1269863383
TIME_tvnsec=455993143
SEQUENCE_NUM=0
PID=6947038
UID=0
UID_LOGIN=0
GID=0
PROG_NAME=cat
RC_FROM_EVPROD=1000
BEGIN_EVPROD_INFO
event producer message here
END_EVPROD_INFO
STACK_TRACE
ahafs_prod_callback+3C4
ahafs_cbfn_wrapper+30
ahafs_vn_write+204
vnode_rdw+7E4
vno_rw+B4
```

```
rwuio+12C
rdwr+184
kewrite+16C
.svc_instr
write+1A4
_xwri
```

หากข้อมูลเหตุการณ์กำลังมาอย่างรวดเร็วเพียงพอ จึงมีความเป็นไปได้ในการรับรายการ **BUF_WRAP** สองรายการในแถว หากคุณกำลังดู **BUF_WRAP** ให้เพิ่มขนาดของบัฟเฟอร์ (โดยใช้ **BUF_SIZE** เมื่อเขียนลงในไฟล์มอนิเตอร์)

NUM_EVDROPS_INTRCNTX:

ถ้าเหตุการณ์ `interrupt-context` ที่เกิดขึ้นถูกรีเซ็ต เนื่องจากความถี่ของเหตุการณ์ที่เกิดขึ้น เอาต์พุตจากการเรียก `read()` บนไฟล์เหตุการณ์ แสดงถึงเหตุการณ์นั้น มีคีย์เวิร์ด **NUM_EVDROPS_INTRCNTX** หลังบรรทัดที่มีคีย์เวิร์ด

BEGIN_EVENT_INFO

ตัวอย่างต่อไปนี้แสดงเอาต์พุตจากการเรียก `read()`:

```
BEGIN_EVENT_INFO
BEGIN_EVENT_INFO
NUM_EVDROPS_INTRCNTX=5508
TIMEO_tvsec=1353437661
TIMEO_tvnssec=75494625
TIME_tvsec=1353437661
TIME_tvnssec=741365037
SEQUENCE_NUM=6663
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
...msg from event-producer...
END_EVPROD_INFO
END_EVENT_INFO
```

เอาต์พุตตัวอย่างนี้มีชุดข้อมูลต่อไปนี้:

- ค่า **NUM_EVDROPS_INTRCNTX=5508** เป็นตัวเลข ของเหตุการณ์ `interrupt-context` ที่เกิดขึ้นที่รีเซ็ตตั้งแต่เวลาที่ระบุไว้โดยฟิลด์ **TIMEO_tvsec** และ **TIMEO_tvnssec**
- ข้อมูลที่เหลือ (นั่นคือ **SEQUENCE_NUM=6663**, **RC_FROM_EVPROD=0**, **...msg from event-producer...** และอื่นๆ) จะเกี่ยวกับเหตุการณ์ที่เกิดขึ้นตามเวลาที่ระบุไว้โดยฟิลด์ **TIME_tvsec** และ **TIME_tvnssec**

โปรดิวเซอร์เหตุการณ์ที่กำหนดไว้ล่วงหน้า

modFile

ผู้สร้างเหตุการณ์ **modFile** มอนิเตอร์การแก้ไข เนื้อหาของไฟล์

ภาพรวม

ผู้สร้างเหตุการณ์ **modFile** อยู่ภายใต้ไต่เดเร็ททอรี **fs** และมอนิเตอร์การแก้ไขไปยังไฟล์ การดำเนินการ **vnode** ต่อไปนี้ ถูกมอนิเตอร์: **vnop_rdwr()**, **vnop_map_lloff()**, **vnop_remove()**, **vnop_ftrunc()**, **vnop_fclear()** และ **vnop_rename()** การแก้ไขที่ไม่ได้ไปยังเลเยอร์ **LFS** ไม่สามารถมอนิเตอร์ได้ (ซึ่งคือการเขียนไปยังไฟล์ที่แม็ปแล้ว)

ไฟล์อาจไม่ถูกมอนิเตอร์ หาก:

- ไฟล์เหล่านั้นอยู่ในระบบไฟล์แบบรีโมต
- ไฟล์เหล่านั้นอยู่ในระบบไฟล์ที่มีชนิด **ahafs**, **prodfs** หรือ **namefs**
- ไฟล์เหล่านั้นคือลิงก์สัญลักษณ์
- ไฟล์เหล่านั้นอยู่ภายใต้ไดเรกทอรีที่ลงท้ายด้วยส่วนขยาย AIX Event Infrastructure (**.mon**, **.list**, **.monFactory**)
- ไฟล์มอนิเตอร์ที่มีชื่อพาธเต็มซึ่งมีขนาดใหญ่กว่า **MAXPATHLEN** ในระบบไฟล์ pseudo ของ AIX Event Infrastructure ไม่สามารถมอนิเตอร์ได้

ความสามารถ

AHAFS_THRESHOLD_STATE
AHAFS_STKTRACE_AVAILABLE
AHAFS_REMOTE_EVENT_ENABLED

โค้ดส่งคืน

ผู้สร้างเหตุการณ์ **modFile** ใช้รหัสส่งคืนที่ถูกกำหนดอยู่ใน `<sys/ahafs_evProds.h>`

รหัสส่งคืนเหล่านี้ถูกใช้เพื่อระบุว่าเนื้อหาของไดเรกทอรีที่ถูกตรวจสอบจะได้รับการแก้ไขอย่างไร:

AHAFS_MODFILE_WRITE

ไฟล์ที่ถูกมอนิเตอร์ถูกเขียน

AHAFS_MODFILE_UNMOUNT

ระบบไฟล์ที่มีไฟล์ที่ถูกมอนิเตอร์จะถูกถอดออก นี่เป็นเหตุการณ์ที่ไม่พร้อมใช้งาน

AHAFS_MODFILE_MAP

กระบวนการแมปส่วนของไฟล์ที่ถูกมอนิเตอร์สำหรับการเขียน

AHAFS_MODFILE_REMOVE

ไฟล์ที่ถูกมอนิเตอร์ได้ถูกลบออกแล้ว นี่เป็นเหตุการณ์ที่ไม่พร้อมใช้งาน

AHAFS_MODFILE_RENAME

ไฟล์ที่ถูกมอนิเตอร์ได้ถูกเปลี่ยนชื่อแล้ว นี่เป็นเหตุการณ์ที่ไม่พร้อมใช้งาน

AHAFS_MODFILE_FCLEAR

กระบวนการได้ใช้ **fclear** สำหรับไฟล์ที่ถูกมอนิเตอร์

AHAFS_MODFILE_FTRUNC

กระบวนการได้ใช้ **ftrunc** สำหรับไฟล์ที่ถูกมอนิเตอร์

AHAFS_MODFILE_OVERMOUNT

ไฟล์ที่ถูกมอนิเตอร์ได้ถูกประกอบแล้ว

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ไม่ส่งผ่านข้อความใดๆ เป็นส่วนหนึ่งของ ข้อมูลเหตุการณ์

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์การแก้ไขไฟล์ ไฟล์มอนิเตอร์ที่มีพารามิเตอร์ที่เหมือนกับไฟล์ที่คุณต้องการมอนิเตอร์ควรถูกสร้างภายใต้ไดเรกทอรี `modFile.monFactory` ตัวอย่างเช่น หากต้องการมอนิเตอร์ `/etc/passwd` ไฟล์มอนิเตอร์ `/aha/fs/modFile.monFactory/etc/passwd.mon` จะถูกใช้

ตัวอย่างข้อมูลเหตุการณ์

ข้อมูลเหตุการณ์ต่อไปนี้ถูกสร้างขึ้นจากกระบวนการที่เขียนลงในไฟล์ที่ถูกมอนิเตอร์นี้คือเอาต์พุตที่สามารถมองเห็นได้ด้วย `INFO_LVL` ของ 3:

```
BEGIN_EVENT_INFO
TIME_tvsec=1271703118
TIME_tv_nsec=409201093
SEQUENCE_NUM=0
PID=5701678
UID=0
UID_LOGIN=0
GID=0
PROG_NAME=cat
RC_FROM_EVPROD=1000
STACK_TRACE
aha_cbfn_wrapper+30
ahafs_evprods+510
aha_vn_write+154
vnop_rdwr+7E8
vno_rw+B4
rwuio+100
rdwr+188
kewrite+104
.svc_instr
write+1A4
_xwrite+6C
_xflsbuf+A8
__flsbuf+C0
copyopt+2E8
scat+22C
main+11C
__start+68
END_EVENT_INFO
```

modFileAttr

ผู้สร้างเหตุการณ์ `modFileAttr` มอนิเตอร์หาการแก้ไขที่ทำกับแอตทริบิวต์ของไฟล์

ภาพรวม

ผู้สร้างเหตุการณ์ `modFileAttr` อยู่ภายใต้ไดเรกทอรี `fs` และมอนิเตอร์หาการแก้ไขที่ดำเนินการกับแอตทริบิวต์ของไฟล์หรือไดเรกทอรี (โหนด, ความเป็นเจ้าของ และ ACL) การดำเนินการ `vnode` ดังต่อไปนี้ได้รับการตรวจสอบ:

`vnop_setattr()` (เฉพาะสำหรับการดำเนินการ `V_OWN` และ `V_MODE`), `vnop_setacl()`, `vnop_setxacl()`, `vnop_remove()`, `vnop_rename()` และ `vnop_rmdir()`

ไฟล์หรือไดเรกทอรีอาจไม่ได้รับการตรวจสอบถ้า:

- ไฟล์หรือไดเรกทอรีอยู่ในระบบไฟล์แบบรีโมต

- ไฟล์หรือไดเรกทอรีอยู่ในระบบไฟล์ที่มีชนิด **ahafs**, **prodfs** หรือ **namefs**
- ไฟล์หรือไดเรกทอรีอยู่ภายใต้ไดเรกทอรีที่ลงท้ายด้วยส่วนขยาย **AIX Event Infrastructure (.mon, .list, .monFactory)**
- ไฟล์มอนิเตอร์ที่มีชื่อพาธเต็มซึ่งมีขนาดใหญ่กว่า **MAXPATHLEN** ในระบบไฟล์ **pseudo** ของ **AIX Event Infrastructure** ไม่สามารถมอนิเตอร์ได้

ความสามารถ

AHAFS_THRESHOLD_STATE

AHAFS_STKTRACE_AVAILABLE

AHAFS_REMOTE_EVENT_ENABLED

โค้ดส่งคืน

ผู้สร้างเหตุการณ์ **modFileAttr** ใช้รหัสส่งคืนที่ถูกกำหนดใน **<sys/ahafs_evProds.h>**

รหัสส่งคืนเหล่านี้ถูกใช้เพื่อระบุว่าการกระทำของไดเรกทอรีที่ถูกรวบรวมจะได้รับการแก้ไขอย่างไร:

AHAFS_MODFILEATTR_UNMOUNT

ระบบไฟล์ที่มีไฟล์หรือไดเรกทอรีที่ถูกลบออกโดยการเฝ้าระวังนี้เป็นเหตุการณ์ที่ไม่พร้อมใช้งาน

AHAFS_MODFILEATTR_REMOVE

ไฟล์หรือไดเรกทอรีที่ถูกลบออกแล้วนี้เป็น เหตุการณ์ที่ไม่พร้อมใช้งาน

AHAFS_MODFILEATTR_RENAME

ไฟล์หรือไดเรกทอรีที่ถูกลบออกแล้วตั้งชื่อใหม่นี้เป็น เหตุการณ์ที่ไม่พร้อมใช้งาน

AHAFS_MODFILEATTR_OVERMOUNT

ไฟล์หรือไดเรกทอรีที่ถูกเฝ้าระวังเกินนี้เป็น เหตุการณ์ที่ไม่พร้อมใช้งาน

AHAFS_MODFILEATTR_SETACL

ACL ของไฟล์หรือไดเรกทอรีที่ถูกลบออกแล้วได้รับการแก้ไขแล้ว

AHAFS_MODFILEATTR_SETOWN

ความเป็นเจ้าของของไฟล์หรือไดเรกทอรีที่ถูกลบออกแล้วได้รับการแก้ไข

AHAFS_MODFILEATTR_SETMODE

โหมดของไฟล์หรือไดเรกทอรีที่ถูกลบออกแล้วได้รับการแก้ไข

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ไม่ส่งผ่านข้อความใดๆ เป็นส่วนหนึ่งของ ข้อมูลเหตุการณ์

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์การแก้ไขไฟล์ ไฟล์มอนิเตอร์ที่มีพาธที่เหมือนกับไฟล์ที่คุณต้องการมอนิเตอร์ควรถูกสร้างภายใต้ไดเรกทอรี **modFileAttr.monFactory** ตัวอย่างเช่น หากต้องการมอนิเตอร์ **/etc/passwd**, ไฟล์มอนิเตอร์ **/aha/fs/modFileAttr.monFactory/etc/passwd.mon** ควรจะถูกสร้างขึ้น

ตัวอย่างข้อมูลเหตุการณ์

ข้อมูลเหตุการณ์ดังต่อไปนี้ถูกสร้างขึ้นจากกระบวนการเปลี่ยนโหมดของไฟล์ที่ถูกมอนิเตอร์นี้คือเอาต์พุตที่สามารถมองเห็นได้ด้วย `INFO_LVL` ของ 3:

```
BEGIN_EVENT_INFO
TIME_tvsec=1291994430
TIME_tv_nsec=760097298
SEQUENCE_NUM=0
PID=5767216
UID=0
UID_LOGIN=0
GID=0
PROG_NAME=chmod
RC_FROM_EVPROD=1010
STACK_TRACE
ahafs_evprods+70C
aha_process_attr+120
vnode_setattr+21C
vsetattr@AF13_1+20
setnameattr+B4
chmod+110
.svc_instr
change+3C8
main+190
__start+68
END_EVENT_INFO
```

modDir

ผู้สร้างเหตุการณ์ `modDir` มอนิเตอร์การแก้ไขเนื้อหาของไฟล์

ภาพรวม

ผู้สร้างเหตุการณ์ `modDir` อยู่ภายใต้ไต่เร็กทอรี `fs` และมอนิเตอร์สำหรับการแก้ไขเนื้อหาของไต่เร็กทอรี การดำเนินการ `vnode` ต่อไปนี้จะถูกมอนิเตอร์ `vnode_create()`, `vnode_link()`, `vnode_symlink()`, `vnode_remove()`, `vnode_rename()`, `vnode_mkdir()` และ `vnode_rmdir()`

ไต่เร็กทอรีอาจไม่ถูกมอนิเตอร์ หาก:

- ไต่เร็กทอรีเหล่านี้อยู่ในระบบไฟล์แบบรีโมต
- ไต่เร็กทอรีเหล่านี้อยู่ในระบบไฟล์ที่มีชนิด `ahafs`, `procfs` หรือ `namefs`
- ไต่เร็กทอรีเหล่านี้คือลิงก์สัญลักษณ์
- ไฟล์เหล่านี้ที่อยู่ภายใต้ไต่เร็กทอรีที่ส่งท้ายด้วยส่วนขยาย AIX Event Infrastructure (`.mon`, `.list`, `.monFactory`)
- ไฟล์มอนิเตอร์ที่มีชื่อพาธเต็มซึ่งมีขนาดใหญ่กว่า `MAXPATHLEN` ในระบบไฟล์ `pseudo` ของ AIX Event Infrastructure ไม่สามารถมอนิเตอร์ได้

ผู้สร้างเหตุการณ์ `modDir` ไม่ได้มอนิเตอร์แบบวนซ้ำ สำหรับการแก้ไขไต่เร็กทอรี เฉพาะการแก้ไขไปยังไต่เร็กทอรีที่ระบุไว้เท่านั้น ที่ถูกมอนิเตอร์

ความสามารถ

AHAFS_THRESHOLD_STATE
AHAFS_STKTRACE_AVAILABLE
AHAFS_REMOTE_EVENT_ENABLED

โค้ดส่งคืน

ผู้สร้างเหตุการณ์ `modDir` ใช้รหัสส่งคืนที่ถูกกำหนดอยู่ใน `<sys/ahafs_evProds.h>`

รหัสส่งคืนเหล่านี้ถูกใช้เพื่อระบุว่าเนื้อหาของไดเรกทอรีที่ถูกตรวจสอบจะได้รับการแก้ไขอย่างไร:

AHAFS_MODDIR_CREATE

อ็อบเจกต์ระบบไฟล์ใหม่ถูกสร้างขึ้นภายใต้ไดเรกทอรีที่ถูกมอนิเตอร์

AHAFS_MODDIR_UNMOUNT

ระบบไฟล์ที่มีไดเรกทอรีที่ถูกมอนิเตอร์ได้ถูกถอดออกแล้ว นี่เป็นเหตุการณ์ที่ไม่พร้อมใช้งาน

AHAFS_MODDIR_REMOVE

อ็อบเจกต์ระบบไฟล์ภายในไดเรกทอรีที่ถูกมอนิเตอร์ได้ถูกถอดออกแล้ว

AHAFS_MODDIR_REMOVE_SELF

ไดเรกทอรีที่ถูกมอนิเตอร์ด้วยตนเองได้ถูกถอดหรือเปลี่ยนชื่อแล้ว นี่เป็น เหตุการณ์ที่ไม่พร้อมใช้งาน

ข้อความผู้สร้างเหตุการณ์

ชื่อของอ็อบเจกต์ระบบไฟล์ที่ทริกเกอร์เหตุการณ์ จะถูกสอดแทรกอยู่ในข้อมูลเหตุการณ์

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์การแก้ไขเนื้อหาของไดเรกทอรี ไฟล์มอนิเตอร์ที่มีพารามิเตอร์ที่เหมือนกับไดเรกทอรีที่คุณต้องการมอนิเตอร์ ควรถูกสร้างอยู่ภายใต้ไดเรกทอรี `modDir.monFactory` ตัวอย่างเช่น หากต้องการมอนิเตอร์ไดเรกทอรี `/home/cheryl` สำหรับการแก้ไข ไฟล์มอนิเตอร์ `/aha/fs/modDir.monFactory/home/cheryl.mon` จะถูกใช้

ตัวอย่างข้อมูลเหตุการณ์

ข้อมูลเหตุการณ์ต่อไปนี้ถูกสร้างขึ้นจากไฟล์ใหม่ที่มีชื่อ `file1` ซึ่งถูกสร้างขึ้นในไดเรกทอรีที่ถูกมอนิเตอร์ นี่คือการเอาต์พุตที่สามารถมองเห็นได้ด้วย `INFO_LVL` ของ 3:

```
BEGIN_EVENT_INFO
TIME_tvsec=1271780397
TIME_tvnsec=24369022
SEQUENCE_NUM=0
PID=6095102
UID=0
UID_LOGIN=0
GID=0
PROG_NAME=touch
RC_FROM_EVPROD=1000
BEGIN_EVPROD_INFO
file1
```

```
END_EVPROD_INFO
STACK_TRACE
aha_cbfm_wrapper+30
ahafs_evprods+510
aha_process_vnop+138
vnop_create_attr+4AC
openpnp+418
openpath+100
copen+294
kopen+1C
.svc_instr
open+F8
creat64+1C
main+1EC
__start+68
END_EVENT_INFO
```

utilFs

ผู้สร้างเหตุการณ์ utilFs มอนิเตอร์การใช้ระบบไฟล์

ภาพรวม

ผู้สร้างเหตุการณ์ utilFs มอนิเตอร์การใช้ของระบบไฟล์ เป็นเปอร์เซ็นต์ ซึ่งอยู่ภายใต้ไดเรกทอรี fs ในปัจจุบัน เฉพาะระบบไฟล์ JFS2 เท่านั้นที่สนับสนุนการมอนิเตอร์ utilFs เมื่ออ้างถึงการเขียนไฟล์ การสร้างไฟล์และการลบไฟล์ การใช้ระบบไฟล์จะถูกตรวจสอบเพื่อดูว่าตรงกันหรือเกินค่า threshold ที่กำหนดไว้ ซึ่งอาจมีระบบไฟล์บางตัวที่ระบุเฉพาะการดำเนินการ ซึ่งอาจมีผลต่อการใช้ระบบไฟล์ แต่ utilFs อาจไม่สามารถตรวจพบจนกว่าเขียนไฟล์ สร้างหรือลบไฟล์ถัดไป Thresholds ที่มีค่าเกินเนื่องจากผลลัพธ์ของการลบอ็อบเจกต์ไฟล์ จะถูกแจ้งเตือนจนกว่าจะเขียนไฟล์ สร้าง หรือลบไฟล์ถัดไป

ระบบไฟล์ที่มีชื่อพารของไฟล์มอนิเตอร์ที่มีขนาดใหญ่กว่า MAXPATHLEN ใน AHAFS จะไม่สามารถมอนิเตอร์ได้ หากต้องการหลีกเลี่ยง การแจ้งเตือนเหตุการณ์และผลกระทบกับผลการทำงานที่สำคัญ ขอแนะนำว่า เหตุการณ์ utilFs จะถูกมอนิเตอร์ด้วย NOTIFY_CNT ของ 1

ความสามารถ

```
AHAFS_THRESHOLD_VALUE_HIGH
AHAFS_THRESHOLD_VALUE_LOW
AHAFS_REMOTE_EVENT_ENABLED
```

Thresholds ที่ระบุ ต้องอยู่ระหว่าง 1 ถึง 100 แบบกว้าง

โค้ดส่งคืน

ผู้สร้างเหตุการณ์ utilFs ใช้รหัสส่งคืนซึ่งถูกกำหนดใน <sys/ahafs_evProds.h>

รหัสส่งคืนเหล่านี้ถูกใช้เพื่อระบุว่าเนื้อหาของไดเรกทอรีที่ถูกตรวจสอบจะได้รับการแก้ไขอย่างไร:

AHAFS_UTILFS_THRESH_HIT

ระบบไฟล์ที่ถูกมอนิเตอร์ได้เข้าถึงค่า threshold ที่ระบุไว้

AHAFS_UTILFS_UNMOUNT

ระบบไฟล์ที่ถูกมอนิเตอร์ได้ถูกถอดออกแล้ว นี่เป็น เหตุการณ์ที่ไม่พร้อมใช้งาน

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ไม่ส่งผ่านข้อความใดๆ เป็นส่วนหนึ่งของ ข้อมูลเหตุการณ์

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์การใช้ระบบไฟล์ ไฟล์มอนิเตอร์ที่มี พาสเดียวกับจุดประกอบของระบบไฟล์ที่ต้องถูกมอนิเตอร์ ควรถูกสร้างภายใต้ไอดีเรียกทอริ `utilFs.monFactory` ตัวอย่างเช่น หากต้องการมอนิเตอร์ระบบไฟล์ `/data/fs1` ไฟล์มอนิเตอร์ `/aha/fs/utilFs.monFactory/data/fs1.mon` จะถูกนำมาใช้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ `AHAFS_UTILFS_THRESH_HIT` สำหรับ `INFO_LVL` ของ 3:

```
BEGIN_EVENT_INFO
TIME_tvsec=1271705858
TIME_tvnsec=704241888
SEQUENCE_NUM=0
CURRENT_VALUE=10
RC_FROM_EVPROD=1000
END_EVENT_INFO
```

waitTmCPU

ผู้สร้างเหตุการณ์ `waitTmCPU` มอนิเตอร์เวลารอเฉลี่ย ของเซตที่สามารถรันได้

ภาพรวม

ผู้สร้างเหตุการณ์ `waitTmCPU` มอนิเตอร์เวลารอเฉลี่ย ของเซตที่สามารถรันได้ซึ่งรอเพื่อขอรับเวลา CPU ในช่วงเวลาหนึ่งวินาที ที่วัดได้ในหน่วยมิลลิวินาที `waitTmCPU` อยู่ภายใต้ไอดีเรียกทอริ `cpu`

ความสามารถ

```
AHAFS_THRESHOLD_VALUE_HIGH
AHAFS_CALLBACK_INTRCNTX
AHAFS_REMOTE_EVENT_ENABLED
```

Thresholds ที่ระบุ ต้องมากกว่า 0

โค้ดส่งคืน

ผู้สร้างเหตุการณ์นี้ส่งคืนค่า 0 เสมอเมื่อเหตุการณ์เกิดขึ้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ไม่ส่งผ่านข้อความใดๆ เป็นส่วนหนึ่งของ ข้อมูลเหตุการณ์

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์เหตุการณ์นี้ ไฟล์มอนิเตอร์ต่อไปนี้ควรถูกนำมาใช้:

```
/aha/cpu/waitTmCPU.monFactory/waitTmCPU.mon
```

ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไดเร็กทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ waitTmCPU ที่มี INFO_LVL ของ 3:

```
BEGIN_EVENT_INFO  
TIME_tvsec=1271779504  
TIME_tvnsec=18056777  
SEQUENCE_NUM=0  
CURRENT_VALUE=4  
RC_FROM_EVPROD=0  
END_EVENT_INFO
```

waitersFreePg

ผู้สร้างเหตุการณ์ waitersFreePg มอนิเตอร์จำนวน threads ที่รอกรอบที่ว่าง

ภาพรวม

ผู้สร้างเหตุการณ์ waitersFreePg มอนิเตอร์จำนวนของเซรต ที่รอกรอบที่ว่างมากกว่าช่วงเวลาหนึ่งวินาที waitersFreePg อยู่ภายใต้ไดเร็กทอรีย่อย mem

ความสามารถ

```
AHAFS_THRESHOLD_VALUE_HIGH  
AHAFS_CALLBACK_INTRCNTX  
AHAFS_REMOTE_EVENT_ENABLED
```

Thresholds ที่ระบุ ต้องมากกว่า 0

โค้ดส่งคืน

ผู้สร้างเหตุการณ์นี้ส่งคืนค่า 0 เสมอเมื่อเหตุการณ์เกิดขึ้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ไม่ส่งผ่านข้อความใดๆ เป็นส่วนหนึ่งของ ข้อมูลเหตุการณ์

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์เหตุการณ์นี้ไฟล์มอนิเตอร์ต่อไปนี้ ควรถูกนำมาใช้:

```
/aha/mem/waitersFreePg.monFactory/waitersFreePg.mon
```

ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไดเร็กทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ waitersFreePg ที่มี INFO_LVL ของ 3:

```
BEGIN_EVENT_INFO  
TIME_tvsec=1271779680  
TIME_tvnsec=347233732
```

SEQUENCE_NUM=0
CURRENT_VALUE=19843
RC_FROM_EVPROD=0
END_EVENT_INFO

waitTmPgInOut

ผู้สร้างเหตุการณ์ **waitTmPgInOut** มอนิเตอร์เวลารอเฉลี่ยในหน่วยมิลลิวินาทีสำหรับเหตุการณ์ดำเนินการเพจอิน หรือ เพจเอาต์

ภาพรวม

ผู้สร้างเหตุการณ์ **waitTmPgInOut** มอนิเตอร์เวลารอเฉลี่ยในหน่วยมิลลิวินาทีสำหรับเหตุการณ์ดำเนินการเพจอิน หรือ เพจเอาต์เพื่อให้เสร็จสิ้นในระยะเวลาที่มากกว่าหนึ่งวินาที ผู้สร้างเหตุการณ์ **waitTmPgInOut** ตั้งอยู่ภายใต้ไอดีเรียกทอรี **mem**

ความสามารถ

AHAFS_THRESHOLD_VALUE_HIGH
AHAFS_CALLBACK_INTRCNTX
AHAFS_REMOTE_EVENT_ENABLED

Thresholds ที่ระบุ ต้องมากกว่า 0

โคัดส่งคืน

ผู้สร้างเหตุการณ์นี้ส่งคืนค่า 0 เสมอเมื่อเหตุการณ์เกิดขึ้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ไม่ส่งผ่านข้อความใดๆ เป็นส่วนหนึ่งของ ข้อมูลเหตุการณ์

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์เหตุการณ์นี้ ไฟล์มอนิเตอร์ต่อไปนี้ ควรถูกนำมาใช้:

/aha/mem/waitTmPgInOut.monFactory/waitTmPgInOut.mon

ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไอดีเรียกทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้คือข้อมูลเหตุการณ์จากเหตุการณ์ **waitTmPgInOut** ที่มี **INFO_LVL** ของ 3:

BEGIN_EVENT_INFO
TIME_tvsec=1271779359
TIME_tvnsec=941699413
SEQUENCE_NUM=0
CURRENT_VALUE=12
RC_FROM_EVPROD=0
END_EVENT_INFO

vmo

ผู้สร้างเหตุการณ์ **vmo** จะมอนิเตอร์การเปลี่ยนแปลงกับพารามิเตอร์ **vmo** ที่ปรับได้

ภาพรวม

ผู้สร้างเหตุการณ์ vmo อยู่ในไดเรกทอรี mem และมอนิเตอร์การเปลี่ยนแปลงกับ พารามิเตอร์ vmo ที่ปรับได้ต่อไปนี้

หมายเหตุ: คำสั่ง vmo เป็นคำสั่งที่อธิบายตัวเอง พารามิเตอร์ที่ปรับได้บางพารามิเตอร์ที่แสดงในรายการต่อไปนี้ อาจไม่ได้รับการสนับสนุน

- npskill
- npswarn
- force_realias_lite
- low_ps_handling
- maxpin% (ควรมอนิเตอร์เป็นไฟล์ maxpin_pct.mon)
- nokilluid
- relalias_percentage
- vmm_default_pspa
- npsrpgmin
- npsrpgmax
- npsscrubmin
- npsscrubmax
- scrubclean
- rpgcontrol
- rpgclean
- vm_modlist_threshold
- vmm_fork_policy
- lru_poll_interval

ความสามารถ

AHAFS_THRESHOLD_STATE
AHAFS_STKTRACE_AVAILABLE
AHAFS_REMOTE_EVENT_ENABLED

โค้ดส่งคืน

ผู้สร้างเหตุการณ์นี้ส่งคืนค่า 0 เสมอเมื่อเหตุการณ์เกิดขึ้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ไม่ส่งผ่านข้อความใดๆ เป็นส่วนหนึ่งของ ข้อมูลเหตุการณ์

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์ความสามารถในปรับข้างต้น ไฟล์มอนิเตอร์ในรูปแบบต่อไปนี้ควรถูกนำมาใช้:

```
/aha/mem/vmo.monFactory/<tunable>.mon
```

ไฟล์ที่ไม่สอดคล้องกับเหตุการณ์ข้างต้นไม่สามารถสร้างภายใต้ไอดีเรียกทอร์นี้ได้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากการแก้ไขความสามารถในการปรับที่ถูกรับมือได้ด้วย INFO_LVL ของ 3

```
BEGIN_EVENT_INFO
TIME_tvsec=1271770698
TIME_tvnsec=787565808
SEQUENCE_NUM=0
PID=5701808
UID=0
UID_LOGIN=0
GID=0
PROG_NAME=vmo
RC_FROM_EVPROD=0
STACK_TRACE
aha_cbfn_wrapper+30
ahafs_evprods+510
vm_mon_tunable+B0
vm_chk_mod_tun+5CC
_vmgetinfo+53C
vmgetinfo+48
.svc_instr
vmo_write_vmsetkervars+134
vmo_write_dynamic_values+404
main+BC
__start+70
END_EVENT_INFO
```

schedo

ผู้สร้างเหตุการณ์นี้มอดิเตอร์การเปลี่ยนแปลงไปเป็น schedo ที่สามารถปรับได้

ภาพรวม

ในปัจจุบัน เฉพาะ `vpm_xvcpus` ที่สามารถปรับได้เท่านั้นที่อาจถูกมอดิเตอร์ ผู้สร้างเหตุการณ์นี้จะส่งคืนการติดตามแบบสแต็กและข้อมูลผู้ใช้ เมื่อเหตุการณ์เกิดขึ้น ผู้สร้างเหตุการณ์นี้อยู่ภายใต้ไอดีเรียกทอร์ `cpu`

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_STKTRACE_AVAILABLE
AHAFS_REMOTE_EVENT_ENABLED
```

โค้ดส่งคืน

ผู้สร้างเหตุการณ์นี้ส่งคืนค่า 0 เสมอเมื่อเหตุการณ์เกิดขึ้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ไม่ส่งผ่านข้อความใดๆ เป็นส่วนหนึ่งของ ข้อมูลเหตุการณ์

ไฟล์มอดิเตอร์ที่สามารถยอมรับได้

ไฟล์มอดิเตอร์ที่ใช้เพื่อมอดิเตอร์ความสามารถในการปรับนี้คือ:

```
/aha/cpu/schedo.monFactory/vpm_xvcpus.mon
```

ไม่มีไฟล์มอนิเตอร์อื่น ที่อาจถูกสร้างขึ้นในไดเรกทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากการแก้ไข `vpm_xvcpus` ที่สามารถปรับได้ด้วย `INFO_LVL` ของ 3:

```
BEGIN_EVENT_INFO
TIME_tvsec=1271771009
TIME_tvnsec=251723285
SEQUENCE_NUM=0
PID=7143474
UID=0
UID_LOGIN=0
GID=0
PROG_NAME=schedo
RC_FROM_EVPROD=0
STACK_TRACE
aha_cbfn_wrapper+30
ahafs_evprods+510
schedtune+394
.svc_instr
schedo_write_schedparams+94
schedo_write_dynamic_values+6F0
main+1B0
__start+68
END_EVENT_INFO
```

pidProcessMon

ผู้สร้างเหตุการณ์ `pidProcessMon` มอนิเตอร์กระบวนการที่หยุดทำงาน อ้างอิงตาม PID

ภาพรวม

ผู้สร้างเหตุการณ์ `pidProcessMon` อยู่ภายใต้ไดเรกทอรี `cpu` และมอนิเตอร์กระบวนการที่หยุดทำงาน อ้างอิงตาม PID

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_CALLBACK_INTRCNTX
```

โค้ดส่งคืน

ผู้สร้างเหตุการณ์ `pidProcessMon` แสดงผลเฉพาะรหัสส่งคืนเดียว 0

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ `PROCESS_DOWN` เป็นส่วนหนึ่งของข้อมูลเหตุการณ์

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์กระบวนการที่หยุดทำงาน ไฟล์มอนิเตอร์ควรถูกสร้างขึ้นภายใต้ไดเรกทอรี `pidProcessMon.monFactory` ชื่อไฟล์มอนิเตอร์ที่มีรูปแบบ

```
/aha/cpu/pidProcessMon.monFactory/<process_PID>.mon
```

ได้ถูกนำมาใช้

ตัวอย่างข้อมูลเหตุการณ์

ข้อมูลเหตุการณ์ต่อไปนี้ถูกสร้างขึ้นจากการหยุดทำงานของ กระบวนการที่ถูกมอนิเตอร์ นี่คือเอาต์พุตที่สามารถมองเห็นได้ด้วยค่าดีฟอลต์ **INFO_LVL**

```
BEGIN_EVENT_INFO
TIME_tvsec=1272348759
TIME_tv_nsec=379259175
SEQUENCE_NUM=0
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=PROCESS_DOWN
END_EVPROD_INFO
END_EVENT_INFO
```

processMon

ผู้สร้างเหตุการณ์ **processMon** มอนิเตอร์กระบวนการที่หยุดทำงาน

ภาพรวม

ผู้สร้างเหตุการณ์ **processMon** อยู่ภายใต้ไดเรกทอรี `cpu` และมอนิเตอร์กระบวนการที่หยุดทำงาน อ้างอิงตามชื่อกระบวนการ เฉพาะกระบวนการหลัก สำหรับกระบวนการที่กำหนดไว้ซึ่งมีชื่อเดียวกันเท่านั้นที่จะถูกมอนิเตอร์ นั่นหมายความว่าเรามีแผนผังกระบวนการ `abc (pid 123) -> xyz (pid 345) -> xyz (pid 567)` และการร้องขอบางอย่างที่ต้องการมอนิเตอร์กระบวนการ `xyz (pid = 345)` จะขอรับการมอนิเตอร์จริง

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_REMOTE_EVENT_ENABLED
AHAFS_CALLBACK_INTRCNTX
```

โค้ดส่งคืน

ผู้สร้างเหตุการณ์ **processMon** แสดงผลเฉพาะรหัสส่งคืนเดียว 0

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ **PROCESS_DOWN** เป็นส่วนหนึ่งของข้อมูลเหตุการณ์

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์กระบวนการที่หยุดทำงาน ไฟล์มอนิเตอร์ที่มีพาทเดียวกัน ซึ่งใช้ในการเริ่มต้นกระบวนการควรถูกสร้างภายใต้ไดเรกทอรี `processMon.monFactory` ตัวอย่างเช่น หากต้องการมอนิเตอร์กระบวนการที่ชื่อ `test` ซึ่งวางอยู่ภายใต้ไดเรกทอรี `/usr/samples/ahafs` ไฟล์มอนิเตอร์ `/aha/cpu/processMon.monFactory/usr/samples/ahafs/test.mon` ควรถูกนำมาใช้

ตัวอย่างข้อมูลเหตุการณ์

ข้อมูลเหตุการณ์ต่อไปนี้ถูกสร้างขึ้นจากการหยุดทำงานของ กระบวนการที่ถูกมอนิเตอร์ นี่คือเอาต์พุตที่สามารถมองเห็นได้ด้วยค่าดีฟอลต์ **INFO_LVL**

```
BEGIN_EVENT_INFO
TIME_tvsec=1272348909
TIME_tv_nsec=482502597
SEQUENCE_NUM=0
```

```
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=PROCESS_DOWN
END_EVPROD_INFO
END_EVENT_INFO
```

inetsock

โปรโตคอลเหตุการณ์ **inetsock** จะมอนิเตอร์การดำเนินการซ็อกเก็ต Transmission Control Protocol (TCP) และ User Datagram Protocol (UDP)

ภาพรวม

โปรโตคอลเหตุการณ์ **inetsock** มีการวางไว้ ภายใต้ไดเรกทอรี net และจะมอนิเตอร์การดำเนินการซ็อกเก็ต

มีการมอนิเตอร์ การดำเนินการซ็อกเก็ตต่อไปนีสำหรับ TCP:

- การสร้างซ็อกเก็ต
- การโยงพอร์ตหรือแอดเดรสเข้ากับซ็อกเก็ต
- การรับฟังบนซ็อกเก็ต
- การยอมรับและการสร้างการเชื่อมต่อบนซ็อกเก็ต
- การเชื่อมต่อกับซ็อกเก็ต
- การตัดการเชื่อมต่อจากซ็อกเก็ต
- การปิดซ็อกเก็ต

มีการมอนิเตอร์การดำเนินการซ็อกเก็ตต่อไปนีสำหรับ UDP:

- การสร้างซ็อกเก็ต
- การโยงพอร์ตหรือแอดเดรสเข้ากับซ็อกเก็ต
- การปิดซ็อกเก็ต

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_CALLBACK_INTRCNTX
AHAFS_REMOTE_EVENT_ENABLED
```

ข้อความผู้สร้างเหตุการณ์

โปรโตคอลเหตุการณ์นี้จะส่งผ่านข้อมูลที่มีอยู่ใน บล็อกควบคุมโปรโตคอลและซ็อกเก็ตเป็นส่วนหนึ่งของข้อมูลเหตุการณ์

มีการส่งผ่าน ข้อมูลต่อไปนีสำหรับการดำเนินการซ็อกเก็ต TCP:

การดำเนินการซ็อกเก็ต	ข้อมูล
PRU_ATTACH	ข้อมูลทั่วไป: <ul style="list-style-type: none"> • PROG_NAME • SO_FAMILY • SO_PID • SO_PROTO • SO_TYPE • SO_UID
PRU_BIND	ข้อมูลทั่วไปบวกไอเอ็มทีโอไปนี้: <ul style="list-style-type: none"> • SO_LADDR • SO_LPORT
PRU_LISTEN	ข้อมูลทั่วไปบวกไอเอ็มทีโอไปนี้: <ul style="list-style-type: none"> • SO_LADDR • SO_LPORT
PRU_ACCEPT	ข้อมูลทั่วไปบวกไอเอ็มทีโอไปนี้: <ul style="list-style-type: none"> • SO_FADDR • SO_FPORT • SO_LADDR • SO_LPORT
PRU_CONNECT	ข้อมูลทั่วไปบวกไอเอ็มทีโอไปนี้: <ul style="list-style-type: none"> • SO_FADDR • SO_FPORT • SO_LADDR • SO_LPORT
PRU_DISCONNECT	ข้อมูลทั่วไปบวกไอเอ็มทีโอไปนี้: <ul style="list-style-type: none"> • SO_FADDR • SO_FPORT • SO_LADDR • SO_LPORT
PRU_DETACH, PRU_ABORT	ข้อมูลทั่วไปบวกไอเอ็มทีโอไปนี้: <ul style="list-style-type: none"> • SO_LADDR • SO_LPORT • SO_FADDR • SO_FPORT

มีการส่งผ่านข้อมูลต่อไปนี้สำหรับการดำเนินการซ็อกเก็ต UDP:

การดำเนินการซ็อกเก็ต	ข้อมูล
PRU_ATTACH	ข้อมูลทั่วไป: <ul style="list-style-type: none"> • PROG_NAME • SO_FAMILY • SO_PID • SO_PROTO • SO_TYPE • SO_UID
PRU_BIND, PRU_DYNBIND	ข้อมูลทั่วไปบวกไอเท็มต่อไปนี้: <ul style="list-style-type: none"> • SO_LADDR • SO_LPORT
PRU_DETACH, PRU_ABORT	ข้อมูลทั่วไปบวกไอเท็มต่อไปนี้: <ul style="list-style-type: none"> • SO_LADDR • SO_LPORT

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

ในการมอนิเตอร์การดำเนินการซ็อกเก็ต ไฟล์การมอนิเตอร์ที่มีชื่อของการดำเนินการซ็อกเก็ตซึ่งคุณต้องการมอนิเตอร์ ต้องมีการสร้างขึ้นแล้ว ในไดเรกทอรี `inetsock.monFactory` ตัวอย่างเช่น ในการมอนิเตอร์การสร้างซ็อกเก็ต TCP ระบบจะใช้ไฟล์การมอนิเตอร์ `/aha/net/inetsock.monFactory/streamCreate.mon` ในลักษณะคล้ายกัน ในการมอนิเตอร์การสร้างซ็อกเก็ต UDP ระบบจะใช้ไฟล์การมอนิเตอร์ `/aha/net/inetsock.monFactory/dgramCreate.mon`

มีการใช้ไฟล์ต่อไปนี้สำหรับการดำเนินการซ็อกเก็ต TCP ที่สามารถมอนิเตอร์ Health Advisor File System (AHAFS) ทั้งหมด:

- `/aha/net/inetsock.monFactory/streamCreate.mon`
- `/aha/net/inetsock.monFactory/streamBind.mon`
- `/aha/net/inetsock.monFactory/streamListen.mon`
- `/aha/net/inetsock.monFactory/streamAccept.mon`
- `/aha/net/inetsock.monFactory/streamConnect.mon`
- `/aha/net/inetsock.monFactory/streamDisconnect.mon`
- `/aha/net/inetsock.monFactory/streamClose.mon`

มีการใช้ไฟล์ต่อไปนี้สำหรับการดำเนินการซ็อกเก็ต UDP ที่สามารถมอนิเตอร์ AHAFS ทั้งหมด:

- `/aha/net/inetsock.monFactory/dgramCreate.mon`
- `/aha/net/inetsock.monFactory/dgramBind.mon`
- `/aha/net/inetsock.monFactory/dgramClose.mon`

ตัวอย่างข้อมูลเหตุการณ์

ข้อมูลเหตุการณ์ต่อไปนี้ถูกสร้างขึ้นจากกระบวนการเมื่อสร้าง ซ็อกเก็ต ตัวอย่างต่อไปนี้คือเอาต์พุตที่จะแสดงขึ้น ด้วยระดับเอาต์พุต 2 (INFO_LVL=2):

AHAFS event: /aha/net/inetsock.monFactory/streamCreate.mon

```
-----  
BEGIN_EVENT_INFO  
Time           : Mon Jan 23 23:04:06 2012  
Sequence Num: 1  
RC_FROM_EVPROD=0  
BEGIN_EVPROD_INFO  
PROG_NAME=xmtopas  
SO_FAMILY=2  
SO_TYPE=1  
SO_PROTO=6  
SO_UID=0  
SO_PID=5243048  
END_EVPROD_INFO  
END_EVENT_INFO
```

เหตุการณ์คลัสเตอร์

เมื่อระบบเป็นส่วนหนึ่งของคลัสเตอร์ ระบบจะสามารถรับการแจ้งเตือน เหตุการณ์ที่เกิดขึ้นบนโหนดอื่นที่เป็นส่วนหนึ่งของคลัสเตอร์เดียวกัน ผู้สร้างเหตุการณ์ที่ระบุความสามารถ AHAFS_REMOTE_EVENT_ENABLED จะสนับสนุนการมอนิเตอร์แบบกว้างๆ ของคลัสเตอร์ โพรติวเซอร์เหตุการณ์ AIX Event Infrastructure ทั้งหมดยกเว้น pidProcessMon และ diskState สามารถ มีการแจ้งเตือนแบบรีโมต

พฤติกรรมของคำสั่ง **mkcluster** พร้อม AIX Event Infrastructure:

ถ้าไม่ได้โหลด AIX Event Infrastructure บนระบบและคำสั่ง **mkcluster** รันอยู่ แล้ว AIX Event Infrastructure pseudo filesystem จะถูกเมาท์บนไดเรกทอรี /aha และ ชื่อไฟล์มอนิเตอร์จะขึ้นต้นด้วยไดเรกทอรี /aha ถ้า AIX Event Infrastructure ถูกโหลดไว้แล้วบนระบบและคำสั่ง **mkcluster** รันอยู่ แล้ว AIX Event Infrastructure pseudo filesystem จะไม่ยกเลิกการเมาท์ และชื่อไฟล์มอนิเตอร์ จะขึ้นต้นด้วยไดเรกทอรีที่ AIX Event Infrastructure pseudo filesystem ถูกเมาท์ไว้ แอ็พพลิเคชันของผู้บริโภคต้องตรวจสอบตำแหน่งของ AIX Event Infrastructure pseudo filesystem ที่เมาท์ไว้เพื่อรับพาทของไฟล์มอนิเตอร์

หากต้องการขอรับเหตุการณ์คลัสเตอร์ กระบวนการของผู้บริโภคต้องระบุ **CLUSTER=YES** เมื่อเขียนไปยังไฟล์มอนิเตอร์ที่แทนค่าเหตุการณ์ที่ต้องมอนิเตอร์ใน คลัสเตอร์ หากต้องการให้เหตุการณ์แบบรีโมตต้องถูกตรวจสอบ กระบวนการของผู้บริโภคต้องถูกมอนิเตอร์เหตุการณ์บนแต่ละโหนดด้วย **CLUSTER=YES** ที่ระบุไว้

เหตุการณ์ที่ได้รับจากโหนดแบบรีโมตไม่ต้องสอดแทรกข้อมูลผู้ใช้ หรือกระบวนการ หรือการติดตามแบบสแต็ก แม้ว่าผู้สร้างเหตุการณ์จะสนับสนุนก็ตาม

สำหรับเหตุการณ์ที่รับบนการติดตามแบบสแต็กของโหนดแบบรีโมตไม่ได้ถูกจัดเตรียมไว้ แม้ว่าผู้สร้างเหตุการณ์ที่สนับสนุนก็ตาม

ข้อมูลคลัสเตอร์ **NODE_NUMBER**, **NODE_ID** and **CLUSTER_ID** จะพร้อมใช้งานระหว่างตัวค้น

BEGIN_EVPROD_INFO และ **END_EVPROD_INFO** สำหรับเหตุการณ์คลัสเตอร์ทั้งหมด ซึ่งจะช่วยให้โปรแกรมการมอนิเตอร์ เพื่อกำหนดโหนดที่เกิดเหตุการณ์ ข้อมูลที่คืนค่าจาก เอาต์พุตของคำสั่ง **lscluster -m** ในฟิลด์: Cluster shorthand id สำหรับโหนด, **uuid** สำหรับโหนดและคลัสเตอร์ **uuids** ถูกคืนค่า ในเอาต์พุตเหตุการณ์ของ AIX Event Infrastructure ในฟิลด์ **NODE_NUMBER**, **NODE_ID**, **CLUSTER_ID** ตามลำดับ

ด้านล่างคือตัวอย่างเอาต์พุตจากทั้งเหตุการณ์ที่เกิดขึ้นบนโลคัล และรีโมตพร้อมกับ INFO_LVL ที่มีค่า 2 และผู้สร้างเหตุการณ์ที่ระบุความสามารถ AHAFS_STKTRACE_AVAILABLE

ข้อมูลเหตุการณ์โลคัล	ข้อมูลเหตุการณ์รีโมต
BEGIN_EVENT_INFO TIME_tvsec=1262670289 TIME_tvnsec=453840229 SEQUENCE_NUM=0 PID=4194474 UID=0 UID_LOGIN=0 GID=0 PROG_NAME=rpc.statd RC_FROM_EVPROD=0 BEGIN_EVPROD_INFO NODE_NUMBER=1 NODE_ID=0xF079E8C801C11DF CLUSTER_ID=0x6EA7B08888D811DFB918BEB25635B404 EVENT_TYPE=PROCESS_DOWN END_EVPROD_INFO END_EVENT_INFO	BEGIN_EVENT_INFO TIME_tvsec=1262670289 TIME_tvnsec=248144872 SEQUENCE_NUM=0 RC_FROM_EVPROD=0 BEGIN_EVPROD_INFO EVENT_TYPE=PROCESS_DOWN NODE_NUMBER=1 NODE_ID=0xF079E8C801C11DF CLUSTER_ID=0x6EA7B08888D811DFB918BEB25635B404 END_EVPROD_INFO END_EVENT_INFO

โปรดิวเซอร์เหตุการณ์ที่กำหนดไว้ล่วงหน้า producers สำหรับอินสแตนซ์ Cluster Aware AIX

โปรดิวเซอร์เหตุการณ์เหล่านี้พร้อมใช้งานเมื่อระบบ เป็นส่วนหนึ่งของคลัสเตอร์ที่แอ็คทีฟ

nodeList

ผู้สร้างเหตุการณ์ nodeList มอนิเตอร์การเปลี่ยนแปลง ในความเป็นสมาชิกคลัสเตอร์

ภาพรวม

ผู้สร้างเหตุการณ์ nodeList อยู่ภายใต้ไอดีเร็กทอรีคลัสเตอร์ และมอนิเตอร์โหนดที่เพิ่มหรือถอนออกจากคลัสเตอร์ ผู้สร้าง เหตุการณ์นี้พร้อมใช้งานเมื่อระบบเป็นส่วนหนึ่งของคลัสเตอร์เท่านั้น เหตุการณ์นี้ ถูกสร้างเมื่อเพิ่มโหนดหรือ ถอนออกจากคลัสเตอร์ (ตัวอย่างเช่น ผ่านคำสั่ง chcluster)

ความสามารถ

AHAFS_THRESHOLD_STATE
 AHAFS_REMOTE_EVENT_ENABLED
 AHAFS_CALLBACK_INTRCNTX

โค้ดส่งคืน

nodeList ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน หากถอนคลัสเตอร์ ออก AHAFS_CLUSTER_REMOVE (-1) จะถูกส่งคืน เท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ NODE_ADD และ NODE_DELETE เป็นส่วนหนึ่งของข้อมูลเหตุการณ์ และเนื่องจากเป็นผู้สร้างเหตุการณ์คลัสเตอร์ จึงส่งผ่านข้อมูล NODE_NUMBER, NODE_ID และ CLUSTER_ID

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์สำหรับการเปลี่ยนแปลงในรายการโหนด ไฟล์มอนิเตอร์ควร ถูกสร้างขึ้นภายใต้ไอดีเร็กทอรี nodeList.monFactory ชื่อไฟล์ มอนิเตอร์

/aha/cluster/nodeList.monFactory/nodeListEvent.mon

ได้ถูกนำมาใช้ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไดเรกทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ `nodeList` ที่มีค่าดีฟอลต์ `INFO_LVL`

```
BEGIN_EVENT_INFO
TIME_tvsec=1271922590
TIME_tvnsec=886742634
SEQUENCE_NUM=1
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=NODE_ADD
NODE_NUMBER=1
NODE_ID=0xF079E8C801C11DF
CLUSTER_ID=0x6EA7B0888D811DFB918BEB25635B404
END_EVPROD_INFO
END_EVENT_INFO
```

clDiskList

ผู้สร้างเหตุการณ์ `clDiskList` มอนิเตอร์การเปลี่ยนแปลง ในความเป็นสมาชิกคลัสเตอร์

ภาพรวม

ผู้สร้างเหตุการณ์ `clDiskList` อยู่ภายใต้ไดเรกทอรีดิสก์ และมอนิเตอร์ดิสก์ที่เพิ่มหรือถอนออกจากคลัสเตอร์ ผู้สร้างเหตุการณ์นี้พร้อมใช้งานเมื่อระบุเป็นส่วนหนึ่งของคลัสเตอร์เท่านั้น เหตุการณ์นี้ถูกสร้างเมื่อเพิ่มโหนดหรือถอนออกจากคลัสเตอร์ (ตัวอย่างเช่น ใช้คำสั่ง `chcluster`)

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_REMOTE_EVENT_ENABLED
AHAFS_CALLBACK_INTRCNTX
```

โค้ดส่งคืน

`clDiskList` ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน หากถอนคลัสเตอร์ออก `AHAFS_CLUSTER_REMOVE (-1)` จะถูกส่งคืนเท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ `DISK_ADD` และ `DISK_DELETE` เป็นส่วนหนึ่งของข้อมูลเหตุการณ์ในฟิลด์ `EVENT_TYPE` ซึ่งจะส่งผ่าน `DISK_NAME` และ `DISK_UID` ของดิสก์ที่เกี่ยวข้อง และเนื่องจากเป็นส่วนหนึ่งของคลัสเตอร์ ผู้สร้างเหตุการณ์จะส่งผ่านข้อมูล `NODE_NUMBER`, `NODE_ID` และ `CLUSTER_ID`

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์สำหรับการเปลี่ยนแปลงในรายการดิสก์ ไฟล์มอนิเตอร์ควร ถูกสร้างขึ้นภายใต้ไดเรกทอรี `clDiskList.monFactory` ชื่อไฟล์ มอนิเตอร์

/aha/disk/clDiskList.monFactory/clDiskListEvent.mon

ได้ถูกนำมาใช้ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไดเรกทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ `clDiskList` ที่มีค่าดีฟอลต์ `INFO_LVL`

```
BEGIN_EVENT_INFO
TIME_tvsec=1271927983
TIME_tvnsec=696543410
SEQUENCE_NUM=0
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=DISK_ADD
DISK_NAME=cldisk1
DISK_UID=3E213600A0B800016726C00000FF4B8677C80F1724-100 FAST03IBMfcp
NODE_NUMBER=2
NODE_ID=0xF079E8C801C11DF
CLUSTER_ID=0x6EA7B08888D811DFB918BEB25635B404
END_EVPROD_INFO
END_EVENT_INFO
```

linkedCI

ผู้สร้างเหตุการณ์ `linkedCI` ถูกสร้างขึ้น เมื่อคลัสเตอร์ถูกลิงก์หรือถอนการลิงก์กับคลัสเตอร์อื่น

ภาพรวม

ผู้สร้างเหตุการณ์ `linkedCI` อยู่ภายใต้ไต่เร็กทอรีคลัสเตอร์ และมอนิเตอร์การเปลี่ยนสถานะลิงก์ ผู้สร้างเหตุการณ์นี้พร้อมใช้งาน เมื่อระบุเป็นส่วนหนึ่งของคลัสเตอร์เท่านั้น เหตุการณ์นี้ถูกสร้างขึ้น เมื่อคลัสเตอร์ถูกลิงก์หรือถอนการลิงก์กับคลัสเตอร์อื่น

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_REMOTE_EVENT_ENABLED
AHAFS_CALLBACK_INTRCNTX
```

โค้ดส่งคืน

`linkedCI` ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน หากถอนคลัสเตอร์ออก `AHAFS_CLUSTER_REMOVE (-1)` จะถูกส่งคืนเท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ `LINK_UP` หรือ `LINK_DOWN` เป็นส่วนหนึ่งของข้อมูลเหตุการณ์ ซึ่งจะส่งผ่านข้อมูล `LINK_ID` และเนื่องจากเป็นผู้สร้างเหตุการณ์คลัสเตอร์ `NODE_NUMBER`, `NODE_ID` และ `CLUSTER_ID`

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์การเปลี่ยนแปลงในรายการโหนด ไฟล์มอนิเตอร์ ควรถูกสร้างภายใต้ไต่เร็กทอรี `linkedCI.monFactory` ชื่อไฟล์ มอนิเตอร์

```
/aha/cluster/linkedCI.monFactory/linkedCIEvent.mon
```

ได้ ถูกนำมาใช้ ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไต่เร็กทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ `linkedCI` ที่มีค่าดีฟอลต์ `INFO_LVL`

```
BEGIN_EVENT_INFO
TIME_tvsec=1271224025
TIME_tvnsec=795042625
SEQUENCE_NUM=0
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=LINK_DOWN
LINK_ID=0x7BE9C1BD
NODE_NUMBER=1
NODE_ID=0xF079E8C801C11DF
CLUSTER_ID=0x6EA7B08888D811DFB918BEB25635B404
END_EVPROD_INFO
END_EVENT_INFO
```

nodeContact

ผู้สร้างเหตุการณ์ **nodeContact** มอนิเตอร์สถานะการติดต่อล่าสุดของโหนดในคลัสเตอร์

ภาพรวม

ผู้สร้างเหตุการณ์ **nodeContact** อยู่ภายใต้ไต่เร็กทอรีคลัสเตอร์ และมอนิเตอร์สถานะผู้ติดต่อล่าสุดของโหนดใน คลัสเตอร์ ผู้สร้างเหตุการณ์นี้พร้อมใช้งานเมื่อระบุเป็นส่วนหนึ่งของ คลัสเตอร์เท่านั้น

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_REMOTE_EVENT_ENABLED
AHAFS_CALLBACK_INTRCNTX
```

โค้ดส่งคืน

nodeContact ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน หากถอนคลัสเตอร์ออก **AHAFS_CLUSTER_REMOVE (-1)** จะถูกส่งคืนเท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ **CONNECT_UP** และ **CONNECT_DOWN** เป็นส่วนหนึ่งของข้อมูลเหตุการณ์ ซึ่งจะส่งผ่าน **INTERFACE_NAME** ที่เกี่ยวข้อง และเนื่องจากผู้สร้างเหตุการณ์คลัสเตอร์จะส่งผ่านข้อมูล **NODE_NUMBER, NODE_ID** และ **CLUSTER_ID** เพิ่มเติม

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์การเปลี่ยนแปลงในรายการโหนด ไฟล์มอนิเตอร์ ควรถูกสร้างภายใต้ไต่เร็กทอรี

nodeContact.monFactory ชื่อไฟล์ มอนิเตอร์

```
/aha/cluster/nodeContact.monFactory/nodeContactEvent.mon
```

ได้ ถูกนำมาใช้ ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไต่เร็กทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ **nodeContact** ที่มีค่าดีฟอลต์ **INFO_LVL**

```
BEGIN_EVENT_INFO
TIME_tvsec=1271921874
TIME_tvnsec=666770128
SEQUENCE_NUM=0
RC_FROM_EVPROD=0
```

```
BEGIN_EVPROD_INFO
EVENT_TYPE=CONNECT_DOWN
INTERFACE_NAME=en0
NODE_NUMBER=2
NODE_ID=0xF079E8C801C11DF
CLUSTER_ID=0x6EA7B08888D811DFB918BEB25635B404
END_EVPROD_INFO
END_EVENT_INFO
```

nodeState

ผู้สร้างเหตุการณ์ **nodeState** มอนิเตอร์สถานะของโหนด ในคลัสเตอร์

ภาพรวม

ผู้สร้างเหตุการณ์ **nodeState** อยู่ภายใต้ไอดีเร็กทอรีคลัสเตอร์ และมอนิเตอร์สถานะของโหนดในคลัสเตอร์ ผู้สร้างเหตุการณ์นี้พร้อมใช้งานเมื่อระบุเป็นส่วนหนึ่งของคลัสเตอร์เท่านั้น ตัวอย่างเช่น เหตุการณ์นี้ถูกสร้างขึ้นเมื่อโหนดหยุดทำงานหรือปิดระบบ

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_REMOTE_EVENT_ENABLED
AHAFS_CALLBACK_INTRCNTX
```

โค้ดส่งคืน

nodeState ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน หากถอนคลัสเตอร์ออก **AHAFS_CLUSTER_REMOVE (-1)** จะถูกส่งคืนเท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ **NODE_UP** และ **NODE_DOWN** เป็นส่วนหนึ่งของข้อมูลเหตุการณ์ และเนื่องจากเป็นผู้สร้างเหตุการณ์คลัสเตอร์ จึงส่งผ่านข้อมูล **NODE_NUMBER**, **NODE_ID** และ **CLUSTER_ID** เพิ่มเติม

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์สำหรับการเปลี่ยนแปลงในสถานะของโหนด ไฟล์มอนิเตอร์ควรถูกสร้างขึ้นภายใต้ไอดีเร็กทอรี **nodeState.monFactory** ชื่อไฟล์มอนิเตอร์

```
/aha/c/cluster/nodeState.monFactory/nodeStateEvent.mon
```

ได้ ถูกนำมาใช้ ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไอดีเร็กทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ **nodeState** ที่มีค่าดีฟอลต์ **INFO_LVL**

```
BEGIN_EVENT_INFO
TIME_tvsec=1271921536
TIME_tvnsec=68254861
SEQUENCE_NUM=1
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=NODE_UP
NODE_NUMBER=2
```

```
NODE_ID=0xF079E8C801C11DF
CLUSTER_ID=0x6EA7B08888D811DFB918BEB25635B404
END_EVPROD_INFO
END_EVENT_INFO
```

nodeAddress

ผู้สร้างเหตุการณ์ **nodeAddress** มอนิเตอร์ที่อยู่เครือข่ายของโหนด

ภาพรวม

ผู้สร้างเหตุการณ์ **nodeAddress** อยู่ภายใต้ไต่เรียกทอริคลาสเตอร์และมอนิเตอร์เน็ตเวิร์กแอดเดรสของโหนด ผู้สร้างเหตุการณ์นี้พร้อมใช้งานเมื่อระบุเป็นส่วนหนึ่งของคลาสเตอร์เท่านั้น ตัวอย่างเช่น เหตุการณ์นี้ ถูกสร้างขึ้นเมื่อ alias ถูกเพิ่มหรือถอนออกจาก เน็ตเวิร์กอินเทอร์เน็ตเฟส

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_REMOTE_EVENT_ENABLED
AHAFS_CALLBACK_INTRCNTX
```

โค้ดส่งคืน

nodeAddress ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน หากถอนคลาสเตอร์ออก **AHAFS_CLUSTER_REMOVE (-1)** จะถูกส่งคืนเท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ **ADDRESS_ADD** และ **ADDRESS_DELETE** เป็นส่วนหนึ่งของข้อมูลเหตุการณ์ ซึ่งจะส่งผ่าน **INTERFACE_NAME** ของอินเทอร์เน็ตเฟสที่เกี่ยวข้องและ **FAMILY**, **ADDRESS** และ **NETMASK** ของ IP แอดเดรส และเนื่องจากผู้สร้างเหตุการณ์คลาสเตอร์จะส่งผ่านข้อมูล **NODE_NUMBER**, **NODE_ID** และ **CLUSTER_ID** เพิ่มเติม

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์สำหรับการเปลี่ยนแปลงในรายการโหนด ไฟล์มอนิเตอร์ควร ถูกสร้างขึ้นภายใต้ไต่เรียกทอริ **nodeAddress.monFactory** ชื่อไฟล์ มอนิเตอร์

```
/aha/cluster/nodeAddress.monFactory/nodeAddressEvent.mon
```

ได้ ถูกนำมาใช้ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไต่เรียกทอรินี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ **nodeAddress** ที่มีค่าตีฟอลด์ **INFO_LVL**

```
BEGIN_EVENT_INFO
TIME_tvsec=1271922254
TIME_tvnsec=9053410
SEQUENCE_NUM=0
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=ADDRESS_ADD
INTERFACE_NAME=et0
FAMILY=2
ADDRESS=0x0A0A0A0A
NETMASK=0xFF000000
```

```
NODE_NUMBER=2
NODE_ID=0xF079E8C801C11DF
CLUSTER_ID=0x6EA7B08888D811DFB918BEB25635B404
END_EVPROD_INFO
END_EVENT_INFO
```

networkAdapterState

ผู้สร้างเหตุการณ์ **networkAdapterState** มอนิเตอร์เน็ตเวิร์กอินเตอร์เฟซของโหนดในคลัสเตอร์

ภาพรวม

ผู้สร้างเหตุการณ์ **networkAdapterState** อยู่ภายใต้ไต่เรียกทอริคลัสเตอร์และมอนิเตอร์เน็ตเวิร์กอินเตอร์เฟซของโหนดในคลัสเตอร์ ผู้สร้างเหตุการณ์นี้พร้อมใช้งาน เมื่อระบบคือส่วนหนึ่งของคลัสเตอร์เท่านั้น เหตุการณ์นี้ถูกสร้างขึ้นเมื่อเน็ตเวิร์กอินเตอร์เฟซ หยุดทำงานหรือเริ่มทำงาน

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_REMOTE_EVENT_ENABLED
AHAFS_CALLBACK_INTRCNTX
```

โค้ดส่งคืน

networkAdapterState ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน หากถอนคลัสเตอร์ออก AHAFS_CLUSTER_REMOVE (-1) จะถูกส่งคืนเท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ ADAPTER_UP, ADAPTER_DOWN, ADAPTER_ADD และ ADAPTER_DEL เป็นส่วนหนึ่งของข้อมูลเหตุการณ์ ซึ่งจะส่งผ่าน INTERFACE_NAME และเนื่องจากเป็นผู้สร้างเหตุการณ์คลัสเตอร์ NODE_NUMBER, NODE_ID และ CLUSTER_ID

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์สำหรับการเปลี่ยนแปลงในรายการโหนด ไฟล์มอนิเตอร์ควร ถูกสร้างขึ้นภายใต้ไต่เรียกทอริ **networkAdapterState.monFactory** ชื่อไฟล์มอนิเตอร์

```
/aha/cluster/networkAdapterState.monFactory/networkAdapterStateEvent.mon
```

ได้ ถูกนำมาใช้ ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไต่เรียกทอรินี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ **networkAdapterState** ที่มีค่าดีฟอลต์ INFO_LVL

```
BEGIN_EVENT_INFO
TIME_tvsec=1271920539
TIME_tvnsec=399378269
SEQUENCE_NUM=1
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=ADAPTER_UP
INTERFACE_NAME=en0
NODE_NUMBER=2
```

NODE_ID=0xF079E8C801C11DF
CLUSTER_ID=0x6EA7B08888D811DFB918BEB25635B404
END_EVPROD_INFO
END_EVENT_INFO

cIDiskState

ผู้สร้างเหตุการณ์ cIDiskState มอนิเตอร์ดิสก์คลัสเตอร์

ภาพรวม

ผู้สร้างเหตุการณ์ cIDiskState อยู่ภายใต้ไดเรกทอรีดิสก์และมอนิเตอร์ดิสก์คลัสเตอร์ ผู้สร้างเหตุการณ์นี้พร้อมใช้งานเมื่อระบบคือส่วนของคลัสเตอร์เท่านั้น เหตุการณ์นี้ถูกสร้างขึ้นเมื่อดิสก์คลัสเตอร์หยุดทำงานหรือเริ่มทำงาน

ความสามารถ

AHAFS_THRESHOLD_STATE
AHAFS_REMOTE_EVENT_ENABLED
AHAFS_CALLBACK_INTRCNTX

โค้ดส่งคืน

cIDiskState ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน หากถอนคลัสเตอร์ออก AHAFS_CLUSTER_REMOVE (-1) จะถูกส่งคืนเท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งข้อความ DISK_UP และ DISK_DOWN ในฐานะที่เป็นส่วนหนึ่งของข้อมูลเหตุการณ์ในฟิลด์ EVENT_TYPE พร้อมกับชื่อดิสก์ของคลัสเตอร์ที่เกี่ยวข้อง และเนื่องจากผู้สร้างเหตุการณ์คลัสเตอร์จะส่งผ่านข้อมูล NODE_NUMBER, NODE_ID และ CLUSTER_ID เพิ่มเติม

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์ดิสก์คลัสเตอร์ ไฟล์มอนิเตอร์ควรถูกสร้างภายใต้ไดเรกทอรี cIDiskState.monFactory ชื่อไฟล์มอนิเตอร์

/aha/disk/cIDiskState.monFactory/cIDiskStateEvent.mon

ได้ถูกนำมาใช้ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไดเรกทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ cIDiskState ที่มีค่าดีฟอลต์ INFO_LVL

BEGIN_EVENT_INFO
TIME_tvsec=1271935734
TIME_tvnsec=265210314
SEQUENCE_NUM=1
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=DISK_DOWN
DISK_NAME=cldisk1
NODE_NUMBER=2
NODE_ID=0xF079E8C801C11DF
CLUSTER_ID=0x6EA7B08888D811DFB918BEB25635B404
END_EVPROD_INFO
END_EVENT_INFO

repDiskState

ผู้สร้างเหตุการณ์ repDiskState มอนิเตอร์ที่เก็บดิสก์

ภาพรวม

ผู้สร้างเหตุการณ์ repDiskState อยู่ภายใต้ไอดีเร็กทอรีดิสก์ และมอนิเตอร์ที่เก็บดิสก์ ผู้สร้างเหตุการณ์นี้พร้อมใช้งานเมื่อระบบคือส่วนหนึ่งของคลัสเตอร์เท่านั้น เหตุการณ์นี้ถูกสร้างขึ้นเมื่อที่เก็บดิสก์หยุดทำงานหรือเริ่มทำงาน

ความสามารถ

AHAFS_THRESHOLD_STATE
AHAFS_REMOTE_EVENT_ENABLED
AHAFS_CALLBACK_INTRCNTX

โค้ดส่งคืน

repDiskState ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน หากถอนคลัสเตอร์ ออก AHAFS_CLUSTER_REMOVE (-1) จะถูกส่งคืนเท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ REP_UP และ REP_DOWN เป็นส่วนหนึ่งของข้อมูลเหตุการณ์ในฟิลด์ EVENT_TYPE พร้อมกับชื่อดิสก์ของที่เก็บดิสก์ที่เกี่ยวข้อง และเนื่องจากเป็นผู้สร้าง เหตุการณ์คลัสเตอร์ จึงส่งผ่านข้อมูล NODE_NUMBER, NODE_ID และ CLUSTER_ID เพิ่มเติม

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์ที่เก็บดิสก์ ไฟล์มอนิเตอร์ควรถูกสร้างภายใต้ไอดีเร็กทอรี repDiskState.monFactory ชื่อไฟล์มอนิเตอร์

```
/aha/disk/ repDiskState.monFactory/repDiskStateEvent.mon
```

ได้ถูกนำมาใช้ ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไอดีเร็กทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ repDiskState ที่มีค่าดีฟอลต์ INFO_LVL

```
BEGIN_EVENT_INFO  
TIME_tvsec=1271933757  
TIME_tvnsec=134003703  
SEQUENCE_NUM=1  
RC_FROM_EVPROD=0  
BEGIN_EVPROD_INFO  
EVENT_TYPE=REP_UP  
DISK_NAME=hdisk2  
NODE_NUMBER=2  
NODE_ID=0xF079E8C801C11DF  
CLUSTER_ID=0x6EA7B0888D811DFB9188EB25635B404  
END_EVPROD_INFO  
END_EVENT_INFO
```

diskState

ผู้สร้างเหตุการณ์ diskstate มอนิเตอร์ การเปลี่ยนโลคัลดิสก์

ภาพรวม

ผู้สร้างเหตุการณ์ `diskState` อยู่ภายใต้ไอดีเร็กทอรีดิสก์ และมอนิเตอร์การเปลี่ยนโลคัลดิสก์ ผู้สร้างเหตุการณ์นี้พร้อมใช้งานเมื่อระบบเป็นส่วนหนึ่งของคลัสเตอร์เท่านั้น เหตุการณ์นี้ถูกสร้างขึ้นเมื่อโลคัลดิสก์หยุดทำงานหรือเริ่มทำงาน เหตุการณ์นี้จะได้รับแจ้งเฉพาะสำหรับดิสก์ที่สนับสนุนโดยเฟรมเวิร์กที่จัดเก็บ

ความสามารถ

AHAFS_THRESHOLD_STATE
AHAFS_CALLBACK_INTRCNTX

โค้ดส่งคืน

`diskState` ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน `AHAFS_CLUSTER_REMOVE (-1)` ถูกส่งคืนหากถอนคลัสเตอร์ออกเท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ `LOCAL_UP` และ `LOCAL_DOWN` พร้อมกับชื่อโลคัลดิสก์ที่เกี่ยวข้องซึ่งเป็นส่วนหนึ่งของข้อมูลเหตุการณ์ และเนื่องจากเป็นผู้สร้างเหตุการณ์คลัสเตอร์ จึงส่งผ่านข้อมูล `NODE_NUMBER`, `NODE_ID` และ `CLUSTER_ID` เพิ่มเติม

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์โลคัลดิสก์ ไฟล์มอนิเตอร์ควรถูกสร้างภายใต้ไอดีเร็กทอรี `diskState.monFactory` ชื่อไฟล์มอนิเตอร์ของรูปแบบ

```
/aha/disk/diskState.monFactory/<hdiskn>.mon
```

ต้องถูกใช้ พร้อมกับชื่อของโลคัลดิสก์ที่ได้ถูกมอนิเตอร์

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ `diskState` ที่มีค่าดีฟอลต์ `INFO_LVL`

```
BEGIN_EVENT_INFO  
TIME_tvsec=1271935029  
TIME_tvnsec=958362343  
SEQUENCE_NUM=1  
RC_FROM_EVPROD=0  
BEGIN_EVPROD_INFO  
EVENT_TYPE=LOCAL_UP  
DISK_NAME=hdisk4  
NODE_NUMBER=2  
NODE_ID=0xF079E8C801C11DF  
CLUSTER_ID=0x6EA7B08888D811DFB918BEB25635B404  
END_EVPROD_INFO  
END_EVENT_INFO
```

vgState

ผู้สร้างเหตุการณ์ `vgstate` สามารถตรวจสอบสถานะของ VG บนดิสก์

ภาพรวม

ผู้สร้างเหตุการณ์ `vgState` อยู่ภายใต้ไอดีเร็กทอรีดิสก์ ผู้สร้างเหตุการณ์นี้พร้อมใช้งานเมื่อระบบเป็นส่วนหนึ่งของคลัสเตอร์เท่านั้น เมื่อใดก็ตามที่โลคัลดิสก์ (ลงทะเบียนด้วย `diskState`) หรือดิสก์คลัสเตอร์ที่เกิดเหตุการณ์การทำงานหรือหยุด

ทำงาน เหตุการณ์ที่สอดคล้องกับ VG_UP และ VG_DOWN จะถูกทริกเกอร์สำหรับกลุ่มวอลุ่มที่อยู่บนดิสก์นั้น การใช้ผู้สร้างเหตุการณ์นี้ แอปพลิเคชันสามารถตรวจสอบสถานะของ VG บนดิสก์ ด้วยระบบย่อย LVM

ความสามารถ

```
AHAFS_THRESHOLD_STATE
AHAFS_REMOTE_EVENT_ENABLED
AHAFS_CALLBACK_INTRCNTX
```

โค้ดส่งคืน

vgState ส่งคืนค่า 0 ซึ่งเป็นโค้ดส่งคืน AHAFS_CLUSTER_REMOVE (-1) ถูกส่งคืนหากถอนคลัสเตอร์ออกเท่านั้น

ข้อความผู้สร้างเหตุการณ์

ผู้สร้างเหตุการณ์นี้ส่งผ่านข้อความ VG_UP และ VG_DOWN เป็นส่วนหนึ่งของข้อมูลเหตุการณ์ ซึ่งจะส่งผ่านชื่อดิสก์และชื่อกลุ่มวอลุ่มที่เกี่ยวข้อง และเนื่องจากเป็นผู้สร้างเหตุการณ์คลัสเตอร์จึงส่งผ่านข้อมูล NODE_NUMBER, NODE_ID และ CLUSTER_ID

ไฟล์มอนิเตอร์ที่สามารถยอมรับได้

หากต้องการมอนิเตอร์การเปลี่ยนแปลงในรายการโหนด ไฟล์มอนิเตอร์ ควรถูกสร้างภายใต้ไดเรกทอรี vgState.monFactory ชื่อไฟล์ มอนิเตอร์

```
/aha/disk/vgState.monFactory/vgStateEvent.mon
```

ได้ ถูกนำมาใช้ ไม่มีไฟล์มอนิเตอร์อื่นที่อาจถูกสร้างขึ้นในไดเรกทอรีนี้

ตัวอย่างข้อมูลเหตุการณ์

ต่อไปนี้เป็นข้อมูลเหตุการณ์จากเหตุการณ์ vgstate ที่มีค่าตีฟอลด์ INFO_LVL

```
BEGIN_EVENT_INFO
TIME_tvsec=1271915408
TIME_tvnsec=699408296
SEQUENCE_NUM=0
RC_FROM_EVPROD=0
BEGIN_EVPROD_INFO
EVENT_TYPE=VG_UP
DISK_NAME=hdisk3
VG_NAME=myvg
NODE_NUMBER=2
NODE_ID=0xF079E8C801C11DF
CLUSTER_ID=0x6EA7B0888D811DFB918BEB25635B404
END_EVPROD_INFO
END_EVENT_INFO
```

คำประกาศ

ข้อมูลนี้พัฒนาขึ้นสำหรับผลิตภัณฑ์และบริการที่มีในประเทศสหรัฐอเมริกา

IBM อาจไม่นำเสนอผลิตภัณฑ์ เซอร์วิส หรือคุณลักษณะที่อธิบายในเอกสารนี้ในประเทศอื่น โปรดปรึกษาตัวแทน IBM ในท้องถิ่นของคุณสำหรับข้อมูลเกี่ยวกับผลิตภัณฑ์ และเซอร์วิส ที่มีอยู่ในพื้นที่ของคุณในปัจจุบัน การอ้างอิงใดๆ ถึงผลิตภัณฑ์ โปรแกรม หรือเซอร์วิสของ IBM ไม่ได้มีวัตถุประสงค์ที่จะระบุหรือตีความว่า สามารถใช้ได้เฉพาะผลิตภัณฑ์ โปรแกรม หรือ เซอร์วิสของ IBM เพียงอย่างเดียว เท่านั้น ผลิตภัณฑ์ โปรแกรม หรือเซอร์วิสใดๆ ที่สามารถทำงานได้เท่าเทียมกัน และไม่ละเมิดสิทธิทรัพย์สินทางปัญญาของ IBM อาจนำมาใช้แทนได้ อย่างไรก็ตาม ถือเป็นความรับผิดชอบของผู้ใช้ที่จะประเมิน และตรวจสอบการดำเนินการของ ผลิตภัณฑ์ โปรแกรม หรือเซอร์วิสใดๆ ที่ไม่ใช่ของ IBM

IBM อาจมีสิทธิบัตร หรืออยู่ระหว่างดำเนินการขอ สิทธิบัตรที่ครอบคลุมถึงหัวข้อซึ่งอธิบายในเอกสารนี้ การนำเสนอเอกสารนี้ ไม่ได้เป็นการให้ไลเซนส์ใดๆ ในสิทธิบัตรเหล่านี้แก่คุณ คุณสามารถส่งการสอบถามเกี่ยวกับไลเซนส์ เป็นลายลักษณ์อักษรไปยัง:

IBM Director of Licensing

IBM Corporation

North Castle Drive, MD-NC119

Armonk, NY 10504-1785

US

หากมีคำถามเกี่ยวกับข้อมูลชุดอักขระไบต์คู่ (DBCS) โปรดติดต่อแผนกทรัพย์สินทางปัญญาของ IBM ในประเทศของคุณ หรือส่งคำถาม เป็นลายลักษณ์อักษร ไปยัง:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION จัดเตรียมเอกสาร "ตามสภาพที่เป็น" โดยไม่มีการรับประกันใดๆ ทั้งโดยชัดแจ้งหรือโดยนัย ซึ่งรวมถึง แต่ไม่จำกัดถึงการรับประกันโดยนัยที่ไม่ละเมิดความสามารถในการจัดจำหน่าย หรือตามความเหมาะสมสำหรับวัตถุประสงค์อย่างใดอย่างหนึ่ง ในบางรัฐไม่อนุญาตให้ ปฏิเสธการรับประกันทั้งโดยชัดแจ้งหรือโดยนัยในธุรกรรมบางอย่าง ดังนั้น ข้อความนี้จึงอาจใช้ไม่ได้กับคุณ

ข้อมูลนี้อาจมีความไม่ถูกต้องด้านเทคนิคหรือข้อผิดพลาดจากการพิมพ์ มีการเปลี่ยนแปลง ข้อมูลในเอกสารนี้เป็นระยะ และการเปลี่ยนแปลงเหล่านี้จะรวมอยู่ในเอ็ดชันใหม่ของ สิ่งพิมพ์ IBM อาจปรับปรุง และ/หรือเปลี่ยนแปลงในผลิตภัณฑ์ และ/หรือโปรแกรมที่อธิบายในสิ่งพิมพ์นี้ได้ตลอดเวลา โดยไม่ต้องแจ้งให้ทราบ

การอ้างอิงใดๆ ในข้อมูลนี้ถึงเว็บไซต์ไม่ใช่ของ IBM มีการจัดเตรียมเพื่อความสะดวกเท่านั้น และ ไม่ได้เป็นการรับรองเว็บไซต์เหล่านั้นในลักษณะใดๆ เอกสารประกอบที่เว็บไซต์เหล่านั้นไม่ได้เป็นส่วนหนึ่งของเอกสารประกอบสำหรับผลิตภัณฑ์ IBM นี้ และการใช้เว็บไซต์เหล่านั้นถือเป็นความเสี่ยงของคุณเอง

IBM อาจใช้หรือแจกจ่ายข้อมูลที่คุณจัดหาให้ในลักษณะใดก็ตามที่พิจารณาแล้วว่าเหมาะสม โดยไม่ทำให้เกิดข้อผูกมัดต่อคุณแต่อย่างใด

ผู้รับไลเซนส์ของโปรแกรมนี้ที่ต้องการข้อมูลเกี่ยวกับโปรแกรมสำหรับวัตถุประสงค์ในการเปิดใช้งาน: (i) การแลกเปลี่ยนข้อมูลระหว่างโปรแกรมที่สร้างขึ้นอย่างอิสระกับโปรแกรมอื่น (รวมถึง โปรแกรมนี้) และ (ii) การใช้ข้อมูลซึ่งแลกเปลี่ยนร่วมกัน ควรติดต่อ:

IBM Director of Licensing

IBM Corporation

North Castle Drive, MD-NC119

Armonk, NY 10504-1785

US

ข้อมูลดังกล่าวอาจพร้อมใช้งาน ภายใต้ข้อตกลงและเงื่อนไขที่เหมาะสม รวมถึง การชำระค่าธรรมเนียมในบางกรณี

โปรแกรมที่มีไลเซนส์ซึ่งอธิบายในเอกสารนี้ และเอกสารประกอบที่มีไลเซนส์ทั้งหมดสำหรับโปรแกรม นั้น มีการจัดเตรียมโดย IBM ภายใต้ข้อตกลงของข้อตกลงกับลูกค้าของ IBM, ข้อตกลงไลเซนส์โปรแกรมระหว่างประเทศของ IBM หรือข้อตกลงที่เท่าเทียมกันใดๆ ระหว่างเรา

ข้อมูลประสิทธิภาพ และตัวอย่างลูกค้าที่ระบุมีการนำเสนอสำหรับวัตถุประสงค์การสาธิตเท่านั้น ผลลัพธ์ของประสิทธิภาพการทำงานจริงอาจขึ้นอยู่กับคอนฟิกูเรชันและเกณฑ์การทำงานที่ ระบุเฉพาะ

ข้อมูลเกี่ยวกับผลิตภัณฑ์ที่ไม่ใช่ของ IBM ได้รับมาจากซัพพลายเออร์ของผลิตภัณฑ์เหล่านั้น ประกาศที่เผยแพร่ หรือแหล่งข้อมูลที่เปิดเผยต่อสาธารณะ IBM ไม่ได้ทดสอบผลิตภัณฑ์ดังกล่าว และไม่สามารถยืนยันความถูกต้องของ ประสิทธิภาพ ความเข้ากันได้ หรือการเรียกร้องอื่นใดที่เกี่ยวข้องกับผลิตภัณฑ์ที่ไม่ใช่ของ IBM คำถามเกี่ยวกับ ความสามารถของผลิตภัณฑ์ที่ไม่ใช่ของ IBM ควรส่งไปยังซัพพลายเออร์ของผลิตภัณฑ์เหล่านั้น

ข้อความใดๆ ที่เกี่ยวข้องกับทิศทางในอนาคตและเจตจำนงค์ของ IBM อาจมีการเปลี่ยนแปลง หรือเพิกถอนได้โดยไม่ต้องแจ้งล่วงหน้า และ นำเสนอเฉพาะเป้าหมาย และวัตถุประสงค์เท่านั้น

ราคาของ IBM ทั้งหมดที่แสดงเป็นราคาขายปลีกที่แนะนำของ IBM ซึ่งเป็นราคาปัจจุบัน และอาจเปลี่ยนแปลงได้โดยไม่ต้องแจ้งให้ทราบ ราคาของผู้แทนจำหน่ายอาจแตกต่างกันไป

ข้อมูลนี้ใช้สำหรับวัตถุประสงค์ของการวางแผนเท่านั้น ข้อมูลในเอกสารนี้อาจมีการเปลี่ยนแปลง ก่อนผลิตภัณฑ์ที่อธิบายจะวางจำหน่าย

ข้อมูลนี้มีตัวอย่างของข้อมูลและรายงานที่ใช้ในการดำเนินการทางธุรกิจรายวัน เพื่อ สาธิตข้อมูลให้สมบูรณ์ที่สุดเท่าที่จะเป็นไปได้ ตัวอย่างจึงมีชื่อของแต่ละบุคคล บริษัท ยี่ห้อ และผลิตภัณฑ์ ชื่อเหล่านี้ทั้งหมดเป็นชื่อสมมติ และความคลายคลึงใด ๆ กับบุคคล หรือ องค์กรธุรกิจที่มีอยู่จริง ถือเป็นเหตุบังเอิญ

ไลเซนส์สิทธิ์:

ข้อมูลนี้มีตัวอย่างแอฟพลิเคชันโปรแกรมในภาษาต้นฉบับ ซึ่งแสดงถึง เทคนิคด้านโปรแกรมในหลากหลายแพลตฟอร์ม คุณอาจคัดลอก ปรับเปลี่ยน และแจกจ่าย โปรแกรมตัวอย่างเหล่านี้ในรูปแบบใดๆ โดยไม่ต้องชำระเงินให้แก่ IBM สำหรับวัตถุประสงค์ในการพัฒนา การใช้ การตลาด หรือการแจกจ่ายโปรแกรมแอฟพลิเคชัน ที่สอดคล้องกับอินเทอร์เฟซการเขียน

โปรแกรมแอปพลิเคชันสำหรับแพลตฟอร์มปฏิบัติการ ซึ่งเขียน โปรแกรมตัวอย่าง ตัวอย่างเหล่านี้ยังไม่ได้ผ่านการทดสอบใน ทุกสภาพ ดังนั้น IBM จึงไม่สามารถรับประกัน หรือบอกเป็นนัยถึง ความน่าเชื่อถือ ความสามารถบริการได้ หรือฟังก์ชันของ โปรแกรมเหล่านี้ โปรแกรมตัวอย่างมีการนำเสนอ "ตาม สภาพ" โดยไม่มีการรับประกันประเภทใดๆ IBM ไม่รับผิดชอบ ต่อ ความเสียหายใดๆ ที่เกิดขึ้นเนื่องจากการใช้โปรแกรมตัวอย่างของคุณ

แต่ละสำเนาหรือส่วนใดๆ ของโปรแกรมตัวอย่างเหล่านี้ หรืองานที่สืบเนื่องใดๆ ต้องมีคำประกาศ ลิขสิทธิ์ดังนี้:

© (ชื่อบริษัทของคุณ) (ปี)

ส่วนของโค้ดนี้ได้มาจากโปรแกรมตัวอย่างของ IBM Corp.

© Copyright IBM Corp. (C) ลิขสิทธิ์ IBM Corp. _ป้อน ปี_

ข้อควรพิจารณาเกี่ยวกับนโยบายความเป็นส่วนตัว

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as the customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at <http://www.ibm.com/privacy> and IBM’s Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at <http://www.ibm.com/software/info/product-privacy>.

เครื่องหมายการค้า

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Windows เป็นเครื่องหมายการค้าของ Microsoft Corporation ในสหรัฐอเมริกา ประเทศอื่น หรือทั้งสองกรณี

ดัชนี

อักขระพิเศษ

: คำสั่งในตัว

Korn หรือ POSIX เชลล์ 259

เชลล์เป้าหมาย 292

/etc/hosts 20

/etc/passwd file 230

/usr/bin/ksh93 246

. คำสั่งในตัว

Korn หรือ POSIX เชลล์ 259

เชลล์เป้าหมาย 292

A

ACLs 328

การคงไว้ 328

ตัวอย่าง 334

ตัวอย่างของ AIXC ACL 331

สำหรับอ็อบเจ็กต์ระบบไฟล์ 328

ahafs_evprods

นิยาม 676

AIX

ภาพรวมของผู้จัดการระบบ BSD

พื้นที่การเลื่อนหน้า 362

AIX Event Infrastructure (AHAFS) 673

AIX Runtime Expert 70

AIXwindows

การเริ่มต้นตัวจัดการหน้าต่าง 346

ไฟล์สตาร์ทอัพ 346

API

Workload Manager (WLM) 569

ASCII เป็น PostScript

printing 666

การทำการแปลงให้เป็นระบบอัตโนมัติ 666, 667

การแปลงไฟล์ 666, 667

B

BSD 358, 359, 360, 367, 368, 373, 374

การเปรียบเทียบ AIX for system managers

พื้นที่การเลื่อนหน้า 362

การเปรียบเทียบผู้จัดการระบบ 353

cron 371

NFS และ NIS (เรียกอย่างเป็นทางการว่า Yellow Pages) 359

UUCP 373

การค้นหาและการตรวจสอบไฟล์ 362

การบัญชี 360

BSD (ต่อ)

การเปรียบเทียบผู้จัดการระบบ (ต่อ)

การเปรียบเทียบไฟล์ 357

คำสั่ง 369

เครื่องพิมพ์ 368

บูตและสตาร์ทอัพ 361

ประสิทธิภาพ 367

ระบบไฟล์ 373

ระบบเครือข่าย 353, 358, 363

สำเนาสำรอง 360

อุปกรณ์ 372

เอกสารคู่มือทางออนไลน์และคำสั่ง man 358

buf_wrap 689

C

C เชลล์ 230

การควบคุมงาน 307

การจัดการสัญญาณ 309

การดำเนินการคำสั่ง 317

การทดแทนคำสั่ง 316

การทดแทนชื่อไฟล์ 303

การทดแทนตัวแปร 301

การทดแทนประวัติ 317

การทดแทนสมนาม 300

การเปลี่ยนทิศทางอินพุตและเอาต์พุต 320

การเริ่มต้น 299

ขีดจำกัด 300

คำสั่ง 309

คำสั่งในตัว 310

ตัวดำเนินการ 315

ตัวแปรสภาวะแวดล้อม 305

นิพจน์ 315

ไฟล์สตาร์ทอัพ 299

รายการของคำสั่งในตัว 307

รายการประวัติ 318

cfgmgr 616

clDiskList 709

clDiskState 715

commands

pr 665

qpri 659

qprt 666

smit 656, 660

Common Desktop Environment 649

การกำหนดอุปกรณ์แสดงผลเอง 652

การแก้ไขโปรไฟล์ 650

การเพิ่มจอแสดงผล และเทอร์มินัล 650

Common Desktop Environment (ต่อ)
การลบจอแสดงผลและเทอร์มินัล 650
cron
สำหรับ BSD System Managers 371
cron daemon
การสร้างข้อมูลด้วย 194

D

diskState 717

E

EFS
ระบบไฟล์ที่เข้ารหัส 473

F

file
คำสั่ง 216
files
การจัดรูปแบบสำหรับการพิมพ์ 665
การพิมพ์ ASCII บนเครื่องพิมพ์ PostScript 666
ภาพรวม 209

H

HISTFILE
ตัวแปร 236
ไฟล์ 281
hot removability 416, 435, 436
hot spots ในโลจิสติกส์ 451

I

ID ผู้ใช้
การเปลี่ยนเป็นผู้ใช้อื่น 643
ล็อกอิน 322
idbggen 58
IDs
ผู้ใช้ 323
inetsock 704

J

JFS
ตัดลอกไปยังฟิลิคัลวอลุ่มอื่น 514
JFS (journaled file system)
ด้วยหมายเลขตัวแปรของไอ-โหนด 501
แฟร็กเมนต์ 501

JFS (ระบบไฟล์ที่เจอร์นัล)
การบีบอัดข้อมูล 509
ขนาดสูงสุดของ 506
ขีดจำกัดขนาด 505
บนสื่อบันทึกออปติคัล การอ่าน/เขียน 479
JFS2 (ระบบไฟล์ที่เจอร์นัลที่เพิ่มประสิทธิภาพ)
ขีดจำกัดขนาด 505, 506
Journaled File System (JFS) 464

K

Korn shell ที่จำกัด
การเริ่มต้น 245
Korn shell ที่เพิ่มประสิทธิภาพ
กฎการค้นหา PATH 246
การกำหนดผสม 246
การขยายพารามิเตอร์ 246
การเพิ่มประสิทธิภาพทางคณิตศาสตร์ 246
การอ้างอิงชื่อตัวแปร 246
คำสั่งคืนของคำสั่ง 246
คำสั่งในตัว 246
คำอธิบาย 246
ตัวแปร 246
ตัวแปรผสม 246
ประวัติเซลล์ 246
ฟังก์ชันระเบียบวินัย 246
สถานะแวดล้อมฟังก์ชัน 246
อาร์เรย์เชื่อมโยง 246
Korn shell หรือ POSIX shell 230
การกำหนดสมนามคำสั่ง 281
การแก้ไข 269
การควบคุมงาน 268
การจัดการสัญญาณ 269
การจัดรูปแบบ 254
การใช้คำสั่ง 276
การทดแทนคำสั่ง 238
การทดแทนเครื่องหมายการออกเสียง 283
การทดแทนชื่อไฟล์ 254
การทดแทนพารามิเตอร์ 251, 252
การแบ่งฟิลด์ 241
การประเมินทางคณิตศาสตร์ 239
การเปลี่ยนทิศทางอินพุตและเอาต์พุต 255
การเปลี่ยนทิศทางอินพุตและเอาต์พุตจากโปรเซสรวม 258
การเริ่มต้น 278
การลบอัญประกาศ 255
การใส่ไว้ในเครื่องหมายอัญประกาศ 243
คำที่สงวนไว้ 246
คำสั่งในตัว 258
คำสั่งผสม 277
ตัวแปรที่กำหนดไว้ล่วงหน้า 236
ตัวแปรผู้ใช้กำหนด 236
ที่เพิ่มประสิทธิภาพ 246

Korn เซลล์หรือ POSIX เซลล์ (ต่อ)
นิพจน์เงื่อนไข 242
ประวัติคำสั่ง 281
พารามิเตอร์ที่กำหนดไว้ล่วงหน้า 253
ฟังก์ชัน 279
รายการของคำสั่งในตัวปกติ 242
รายการของคำสั่งในตัวพิเศษ 241
สถานะการออก 250
สภาวะแวดล้อม 279
สิ่งอำนวยความสะดวกโปรเซสรวม 257

ksh93

กฎการค้นหาPATH 246
การกำหนดผสม 246
การขยายพารามิเตอร์ 246
การเพิ่มประสิทธิภาพทางคณิตศาสตร์ 246
การอ้างอิงชื่อตัวแปร 246
คำสั่งคืนของคำสั่ง 246
คำสั่งในตัว 246
คำอธิบาย 246
ตัวแปร 246
ตัวแปรผสม 246
ประวัติเซลล์ 246
ฟังก์ชันระเบียบวินัย 246
สภาวะแวดล้อมฟังก์ชัน 246
อาร์เรย์เชื่อมโยง 246

L

linkedCl 710
Logical Volume Manager 389
Logical Volume Manager (LVM) 421
การซิงโครไนซ์กับฐานข้อมูลคอนฟิกูเรชันอุปกรณ์ 419
นิยาม 384
LVCB (logical-volume control block)
ไม่ได้ป้องกันจากการเข้าถึงโลจิคัลวอลุ่มแบบดิบ 433
LVM 389, 421

M

Mirror Write Consistency (MWC) 444
moddir 694
modDir 704
modfile 690, 692
modFile 704
MPIO 591
การจัดการ 593
Multi-path I/O 591

N

NBPI 504
networkAdapterState 714
NFS และ NIS
BSD System Managers 359
NIS 359
nodeAddress 713
nodeContact 711
nodeList 708
nodeState 712
NUM_EVDROPS_INTRCNTX 690

P

pidProcessMon 702, 704
printing
การจัดระดับความสำคัญงานพิมพ์ 659, 660
การจัดรูปแบบไฟล์สำหรับ 665
ไฟล์ ASCII บนเครื่องพิมพ์ PostScript 666
processMon 703, 704

R

repDiskState 716
RSET ที่ใช้เฉพาะตัว
ชุดรีซอร์สตัวประมวลผลที่ใช้เฉพาะตัว 562

S

schedo 701, 704
setenv built-in command
Korn หรือ POSIX เซลล์ 265
shutdown
กรณีฉุกเฉิน 56
การทำความเข้าใจ 55
โดยไม่ต้องรีบูต 56
ในโหมดผู้ใช้เดียว 56
smit command
การจัดระดับความสำคัญงานพิมพ์ 660
การเริ่มต้นงานพิมพ์ 656
srcmstr daemon 205
stderr 375
stdin 375
stdout 375
switches
ในคำสั่ง 142
System Resource Controller
การเริ่มต้น 205
คำสั่ง
รายการของ 205

System Resource Controller (ต่อ)
ฟังก์ชันของ 203

T

TCP/IP

/etc/hosts 20
การตั้งชื่อ
เน็ตเวิร์กเชิงลำดับชั้น 20
เน็ตเวิร์กแบบราบ 20
การวางแผนเครือข่าย 20

text formatting

การสนับสนุนอักขระสากล 149
คำสั่ง 149

Transmission Control Protocol/Internet Protocol 20

tty (เครื่องพิมพ์ดีดโทรพิมพ์)
โค้ดที่ตั้ง 582

U

USB flash ไดรฟ์ 632

utilFs 696, 704

UUCP

BSD System Managers 373

V

VGDA (volume group descriptor area) 384

VGSA (volume group status area) 384

vgState 717

Virtual Memory Manager 462

Virtual Memory Manager (VMM)

ภาพรวม 453

VMM 462

vmo 700, 704

volume group descriptor area (VGDA) 384

volume group status area (VGSA) 384

W

waitersFreePg 698, 704

waitTmCPU 697, 704

waitTmPgInOut 699, 704

WLM

API 569

Workload Manager

API 569

การเริ่มต้น และการหยุด 533

Y

Yellow Pages 359

BSD System Managers 359

ก

กรณีฉุกเฉิน

การปิดใน 56

กระบวนการ

การลบกระบวนการส่วนหลัง 162

กระบวนการ zombie 154

กระบวนการพื้นหน้า 154

กระบวนการส่วนหลัง 154

กลุ่มระบบย่อย

การแสดงผลสถานะ 207

คำอธิบายของ 204

กลุ่มระบบย่อย

การปิด การติดตาม 208

การเปิดการติดตาม 208

การรีเฟรช 207

การเริ่มต้น 206

การหยุด 206

กลุ่มวอลุ่ม

กลยุทธ์สำหรับ 439

การนำนโยบายไปใช้ 453

การแบ่งดิสก์ที่มีรีเรอร์จิก 407

การเปลี่ยนดิสก์ 416

การเปลี่ยนเป็นสถานะไม่ใช่องค์ประกอบ 387

การมีรีเรอร์ 431

การมีอยู่สูง 439

การยกเลิกมีรีเรอร์ 434

การย้าย 395

การอิมพอร์ต 395

การเอ็กซ์พอร์ต 395

ควรวางแยกต่างหากเมื่อไร 440

ที่ผู้ใช้กำหนด

การอิมพอร์ต 482

ที่มีรีเรอร์

การเปลี่ยนฟิสิคัลวอลุ่ม 401

นิยามของ 427

โปรเซสแปรผัน 384

ไม่ใช่องค์ประกอบ 387

ราก

การมีรีเรอร์ 431

องค์ประกอบ 385

กลุ่มวอลุ่มที่ทำสำเนา

การเปลี่ยนฟิสิคัลวอลุ่ม 401

กลุ่มวอลุ่มที่ผู้ใช้กำหนด

การอิมพอร์ต 482

กลุ่มวอลุ่มที่ไม่ใช่องค์ประกอบ 387

- กลุ่มวอล์มราก (rootvg)
 - การมีร์เรอร์ 431
 - การลดขนาดของระบบไฟล์ 482
- การกำหนด
 - ค่าและแอ็ททริบิวต์ 251
- การกำหนดคอนฟิกอุปกรณ์ปลายทาง 616
- การกำหนดสมนาม
 - Korn หรือ POSIX เซลล์ 281
- การกำหนดสมนามคำสั่ง
 - Korn หรือ POSIX เซลล์ 281
 - การทดแทนเครื่องหมายการออกเสียง 283
- การกำหนดหมายเลข
 - บรรทัดในไฟล์ข้อความ 223
- การกำหนดเอง
 - การผูก ปุ่มเมาส์ 348
 - การผูกคีย์ 348
 - นิยามเมนู 348
 - สภาวะแวดล้อมของระบบ 350, 351, 352
 - สีและฟอนต์ 347
 - อุปกรณ์แสดงผล 652
- การกู้คืนข้อมูลจากดิสก์โดยไม่ต้องจัดรูปแบบใหม่ 410
- การแก้ไข
 - ข้อมูลควบคุมการเข้าถึง 335
 - เดสก์ทอปโปรไฟล์ 650
 - แบบแทรกใน Korn หรือ POSIX เซลล์ 269
 - ประวัติคำสั่ง 148
- การแก้ไขแบบแทรก
 - Korn หรือ POSIX เซลล์ 269
 - โหมดการแก้ไข emacs 270
 - โหมดการแก้ไข gmacs 270
 - โหมดการแก้ไข vi 272, 273, 274, 275
- การขยาย
 - ไฟล์ 42, 43
- การเข้าถึงระบบที่จะไม่บูต 22
- การเขียนลงในไฟล์มอดิเตอร์
 - นิยาม 679
- การคงไว้
 - ACLs 328
 - รายการควบคุมการเข้าถึง 328
- การค้นหา
 - คีย์เวิร์ด 153
 - ไฟล์ 215
 - สตรีงข้อความภายในไฟล์ 218
- การค้นหาคีย์เวิร์ด
 - คำสั่ง apropos 153
- การคลายบีบอัด
 - ไฟล์ 43
- การคลายแพ็ก
 - ไฟล์ 43
- การควบคุมการเข้าถึง
 - การแก้ไขข้อมูล 335
 - การแสดงข้อมูล 333
 - ข้อมูลค่าติดตั้ง 334
- การควบคุมงาน
 - C เซลล์ 307
 - Korn หรือ POSIX เซลล์ 268
- การคัดลอก
 - ไปยังหรือจากดิสเก็ต 51
 - ไปยังหรือจากเทป 53
 - ไฟล์ 214
 - ไฟล์ DOS 226
 - ไฟล์จากเทปหรือดิสก์ 52
 - ไฟล์ไปยังเทปหรือดิสก์ 52
 - ไฟล์ระบบปฏิบัติการพื้นฐาน 227
 - หน้าจอไปยังไฟล์ 381
- การคำนวณเลขจำนวนเต็ม 239
- การแคช
 - การกำหนดคอนฟิก 636
 - การกำหนดค่าในโหมด NPIV 640
 - การกำหนดค่าในโหมดเสมือน 638
 - การกำหนดค่าโหมดที่แยกต่างหาก 636
 - การจัดการ 640
 - ข้อควรพิจารณาเกี่ยวกับความพร้อมใช้งานสูง 641
 - ข้อจำกัด 634
 - ข้อดี 634
 - ข้อมูลหน่วยเก็บข้อมูล 633
 - คอมโพเนนต์ 635
 - แนวคิด 633
- การจัดการสัญญาณ
 - C เซลล์ 309
 - Korn หรือ POSIX เซลล์ 269
 - เซลล์เป้าหมาย 290
- การจัดการฮอตปลั๊ก
 - PCI 586
- การจัดตารางเวลา
 - โปรเซส 159
- การจัดระดับความสำคัญ
 - งานพิมพ์ 659, 660
- การจัดรูปแบบ
 - ดิสเก็ต 50
 - ไฟล์สำหรับการพิมพ์ 665
- การจัดรูปแบบข้อความ
 - การสนับสนุนอักขระหลายไบต์ 150
 - อักขระไบต์เดียวแบบขยาย 150
- การจัดสรร, ไฟล์ศูนย์ (kproc) 508
- การจัดสรรไฟล์ศูนย์ 508
- การจัดรูปแบบ
 - Korn หรือ POSIX เซลล์ 254
 - เซลล์เป้าหมาย 286
- การจำกัดผู้ใช้จากไดเรกทอรีที่ระบุ 409
- การใช้ CPU
 - การแสดงผล 191
- การใช้ดิสก์
 - ผลกระทบต่อแฟร็กเมนต์บน 501
- การตรวจสอบ
 - คุณภาพของเทป 53

- การตรวจสอบ (ต่อ)
 - สถานะของงานพิมพ์ 664
 - สถานะโปรเซส 155
- การตรวจสอบระบบไฟล์ 481
- การตรวจสอบระบบไฟล์เพื่อหาความไม่สอดคล้องกัน 481
- การต่อ
 - ไฟล์ข้อความ 377
- การตั้งค่า
 - ข้อมูลควบคุมการเข้าถึง 334
 - ลำดับความสำคัญแรกเริ่มของโปรเซส 157
- การตัด
 - ส่วนของไฟล์ข้อความ 221
- การติดตั้ง
 - การใช้หลายการติดตั้ง 494
 - การติดตั้งระบบไฟล์ 494
 - การติดตั้งเวิร์กสเตชันแบบไม่มีดิสก์
 - การรักษาความปลอดภัย 496
 - คำอธิบายของ 498
 - การติดตั้งอัตโนมัติ 495
 - การติดตั้งอัตโนมัติ /etc/filesystem 495
 - ภาพรวม 493
- รีโมต
 - นิยาม 494
- โลคัล
 - นิยาม 494
- การตีความ
 - พื้นที่ว่างเปล่า 299
- การถอดออกได้ของฮอตดิสก์ 391, 434
- การทดแทนคำสั่ง
 - C เซลล์ 316
 - Korn หรือ POSIX เซลล์ 238
 - เซลล์เป้าหมาย 295
- การทดแทนเครื่องหมายการออกเสียง
 - การกำหนดสมนามคำสั่ง 283
- การทดแทนชื่อไฟล์
 - C เซลล์ 303
 - Korn หรือ POSIX เซลล์ 254
 - เซลล์เป้าหมาย 286
- การทดแทนตัวแปร
 - C เซลล์ 301
 - Korn หรือ POSIX เซลล์ 236
 - เซลล์เป้าหมาย 295
- การทดแทนแบบมีเงื่อนไข
 - เซลล์เป้าหมาย 284
- การทดแทนพารามิเตอร์
 - Korn หรือ POSIX เซลล์ 252
- การทดแทนสมนาม
 - C เซลล์ 300
- การทิ้งเอาต์พุต 377
- การนับ
 - คำ 220
 - บรรทัด 220
 - ไบต์ 220
- การบริหารเวลา
 - การเขียนข้อความบันทึกช่วยเตือน 152
 - ข้อความ บันทึกช่วยเตือน 152
 - คำสั่ง calendar 152
 - การบัญชีค่าธรรมเนียม 197
 - การบัญชีเวลาเชื่อมต่อ 195
 - การบันทึกข้อผิดพลาด
 - การตรวจสอบข้อผิดพลาดอุปกรณ์ 612
 - การบำรุงรักษา 389
 - การบีบอัด
 - ไฟล์ 41
 - การบีบอัดข้อมูล 509
 - ต้นทุนประสิทธิภาพของ 511
 - แฟร็กเมนต์ 501
- การบูต
 - BSD System Managers 361
 - การทำความเข้าใจ
 - การประมวลผลบูตระบบ 19
 - ภาพรวม 18
 - ระบบไฟล์ RAM 21
 - โหมดการบำรุงรักษา 20
 - การรีบูตระบบที่กำลังรัน 5
 - การวิเคราะห์ปัญหา 22
 - จากฮาร์ดดิสก์สำหรับการบำรุงรักษา 6
 - ระบบที่เสียหาย 7
 - การแบ่งดิสก์ที่มีเรอร์จากกลุ่มวอลุ่ม 407
 - การแบ่งพาร์ติชัน
 - Korn หรือ POSIX เซลล์ 241
 - การประมวลผลบูต
 - ระยะของ 19
 - การประเมินทางคณิตศาสตร์
 - Korn หรือ POSIX เซลล์ 239
 - การปิดระบบ 56
 - การปิดระบบปฏิบัติการ 144
 - การเปรียบเทียบไฟล์ 220
 - การเปลี่ยน
 - คีย์ควบคุม 352
 - ดีฟอลต์พอนด์ 351
 - พร้อมระบบ 352
 - ลำดับความสำคัญของโปรเซส 157
 - สิทธิอนุญาต 327
 - การเปลี่ยนชื่อ
 - ไดเรกทอรี 519
 - ไฟล์ 213
 - การเปลี่ยนทิศทาง
 - อินพุตมาตรฐาน 377
 - อินพุตและเอาต์พุตจากโปรเซสรวม 258
 - อินพุตและเอาต์พุตใน Korn หรือ POSIX เซลล์ 255
 - อินพุตและเอาต์พุตในเซลล์เป้าหมาย 287
 - เอาต์พุตข้อผิดพลาดมาตรฐาน 378
 - เอาต์พุตไปยังไฟล์ 376
 - เอาต์พุตมาตรฐาน 376

- การเปลี่ยนทิศทาง I/O
 - C เซลล์ 320
 - Korn หรือ POSIX เซลล์ 255
 - เซลล์เป้าหมาย 287
 - มาตรฐาน 375
- การเปลี่ยนแปลง
 - กลายเป็นผู้ใช้อื่น 643
 - ดีพอลต์ 347
 - เป็นไดเรกทอรีอื่น 520
- การเปิดใช้งานระบบไฟล์
 - การจัดสรรไฟล์ศูนย์ 508
- การแปลง
 - ไฟล์ ASCII เป็น PostScript 666
 - ไฟล์ DOS 226
- การผูกคีย์ 348
- การผูกปุ่มเมาส์ 348
- การผูกโปรเซสเข้ากับตัวประมวลผล 171
- การพิมพ์
 - การตรวจสอบสถานะของงานพิมพ์ 664
 - การยกเลิกงานพิมพ์ 659
 - การยกเลิกชนิดไฟล์การพิมพ์ 668
 - การย้ายงานพิมพ์ 660, 661
 - การเริ่มต้นงานพิมพ์ 656
 - เงื่อนไขสถานะของเครื่องพิมพ์ 663, 664
 - ภาพรวม 655
- การพิสูจน์ตัวตน 322
- การไฟฟ์ 141
- การมอนิเตอร์โปรเซส 167
- การมีร์เรอร์
 - กลุ่มวอลุ่ม 431
 - กลุ่มวอลุ่มราก (rootvg) 431
 - การแบ่งดิสก์ที่มิร์เรอร์จาก กลุ่มวอลุ่ม 407
- การมีอยู่
 - สำหรับความล้มเหลวดิสก์ 440
 - สำหรับความล้มเหลวอะแด็ปเตอร์หรือแหล่งจ่ายไฟ 441
- การยกเลิก
 - กระบวนการพื้นหน้า 158
 - การกำหนดชนิดไฟล์การพิมพ์โดยอัตโนมัติ 668
 - งานพิมพ์ 659
- การยกเลิกการจัดสรรตัวประมวลผลแบบไดนามิก 59, 61
- การยกเลิกมิร์เรอร์
 - กลุ่มวอลุ่ม 434
- การย้าย
 - งานพิมพ์ 660
- การย้ายที่
 - อะแด็ปเตอร์สำหรับ DLPAR 671
- การแยกตัวประกอบตัวเลข
 - ค่าสิ่ง factor 153
- การรวมค่าสิ่ง 141
- การรอดูการณ์
 - นิยาม 682
- การระบุแฟร็กเมนต์ระบบไฟล์ได้ 506
- การรักษาความปลอดภัย
 - ID ผู้ใช้สำหรับล็อกอิน 322
 - การพิสูจน์ตัวตน 322
 - การระบุ 322
 - เทอร์มินัลที่ไม่ได้ใส่ใจ 322
- ไฟล์ 321
- ไฟล์ /etc/security/passwd 322
- ระบบ 321
- การรัน
 - เซลล์สคริปต์ 235
- การริบระบบโดยใช้กราฟิก planar 8
- การรีสตาร์ท
 - โปรเซสที่หยุด 158
- การเริ่มต้น
 - AIXwindows Window Manager 346
 - C เซลล์ 299
 - Korn เซลล์ที่จำกัด 245
 - Korn หรือ POSIX เซลล์ 278
 - เซลล์ที่จำกัด 234
 - เซลล์เป้าหมาย 283
 - โปรเซส 155
- การเริ่มต้น Workload Manager 533
- การเรียกคืน
 - ไฟล์ 30, 55
- การเรียงลำดับ
 - ไฟล์ข้อความ 218
- การลบ
 - กระบวนการส่วนหลัง 162
 - คอลัมน์ในไฟล์ข้อความ 223
 - จอแสดงผลโลคัล 650
 - ไดเรกทอรี 523
 - โปรเซสจากตารางเวลา 161
 - ไฟล์ 213
 - ไฟล์ DOS 227
 - ไฟล์ที่ลิงก์ 226
 - สมนาม 281
- การลบอัญประกาศ
 - Korn หรือ POSIX เซลล์ 255
- การล็อก
 - เทอร์มินัลของคุณ 336
- การล็อกอิน
 - เข้าสู่ระบบปฏิบัติการ 642
 - เป็นผู้ใช้อื่น 643
 - มากกว่าหนึ่งครั้ง 642
- การล็อกเอาต์
 - ของระบบปฏิบัติการ 643
- การล้างหน้าจอของคุณ 380
- การลิงก์
 - ไดเรกทอรี 224
 - ไฟล์ 224, 225
- การเลื่อน
 - งานพิมพ์ 661

- การวาง
 - ส่วนของไฟล์ข้อความ 222
- การวางแผนเครือข่าย
 - TCP/IP 20
- การวินิจฉัยปัญหาการบูต
 - การเข้าถึงระบบที่จะไม่บูต 22
 - การรีบูตระบบโดยใช้กราฟิก planar 8
- การวินิจฉัยปัญหาดีสก์ไดรฟ์ 408
- การสำรอง
 - งานพิมพ์ 656
- การสนับสนุนอักขระสากล
 - text formatting 149
- การสนับสนุนอักขระหลายไบต์
 - การจัดรูปแบบข้อความ 150
 - ป้อนอักขระ 150
- การสนับสนุนอุปกรณ์ USB 631
- การสร้าง
 - เซลล์สคริปต์ 235
 - ไดเรกทอรี 519
 - สมนาม 281
 - สมนามคำสั่ง 148
- การสร้างไฟล์มอโนเตอร์
 - นิยาม 678
- การแสดงผล
 - ID ผู้ใช้ 644
 - การกำหนดคีย์คั่วคุด 341
 - ข้อความในตัวอักษรขนาดใหญ่บนหน้าจอ 382
 - ข้อมูลกลุ่ม 325
 - ข้อมูลควบคุมการเข้าถึง 333
 - ค่าของตัวแปรสภาวะแวดล้อม 342
 - คีย์บอร์ดแม่พิมพ์ 340
 - จอแสดงผลที่มีอยู่ 339
 - ชนิดไฟล์ 216
 - ชื่อคอนโซล 339
 - ชื่อเทอร์มินัล 339
 - ชื่อระบบของคุณ 646
 - ชื่อระบบปฏิบัติการ 646
 - ชื่อล็อกอิน 645
 - ไดเรกทอรีไฟล์ 520
 - ตัวแปรสภาวะแวดล้อม 341
 - เนื้อหาไดเรกทอรี DOS 228
 - เนื้อหาไดเรกทอรีไฟล์ 521
 - เนื้อหาไฟล์ 217
 - บรรทัดแรกของไฟล์ 220
 - บรรทัดสุดท้ายของไฟล์ 221
 - ปฏิทิน 151
 - ผลิตภัณฑ์ซอฟต์แวร์ 340
 - ผู้ใช้ที่ล็อกอิน 646
 - พื้นที่ว่างที่มีอยู่ 475
 - ฟอนต์ที่มีอยู่ 339
 - หน้าจอในแต่ละครั้ง 217
 - อุปกรณ์ระบบ 337

- การแสดงรายการ
 - โปรเซสที่จัดตารางเวลา 160
 - สมนาม 281
- การแสดงผลสามตำแหน่ง 642
- การใช้อักขระไว้ในเครื่องหมายอัญประกาศ
 - Kom หรือ POSIX เซลล์ 243
- การหยุด
 - กระบวนการพื้นหน้า 158
- การหยุด Workload Manager 533
- การอนุญาต 332
- การอ้างอิงอักขระ
 - เซลล์เป้าหมาย 289
- การอ่านการแสดงผลสามตำแหน่ง 642
- การอ่านข้อมูลเหตุการณ์ 684
- การอิมพอร์ตกลุ่มข้อมูลที่ผู้ใช้กำหนด 482
- การเอ็กซ์พอร์ต
 - ตัวแปรเซลล์ 350
 - สมนาม 281
- กิจกรรมระบบ
 - การติดตาม 179

ข

- ขนาดกลุ่มการจัดสรร 506
- ข้อความ
 - การผนวกเข้ากับไฟล์ 381
 - การส่งไปยังเอาต์พุตมาตรฐาน 380
 - การแสดงผลในตัวอักษรขนาดใหญ่ 382
 - การแสดงผลบนหน้าจอ 380
- ข้อความ, หน้าจอ, การตอบกลับไปยัง 173
- ข้อความของวัน
 - การเปลี่ยน 70
- ข้อความสั่งการใช้
 - สำหรับคำสั่ง 143
- ข้อความหน้าจอ, การตอบกลับไปยัง 173
- ข้อคิดเห็น
 - นิยาม 230
- ข้อผิดพลาด tacct
 - การแก้ไข 197
- ข้อผิดพลาด wtmp
 - การแก้ไข 198
- ข้อผิดพลาดมาตรฐาน 375
- ข้อมูลสรุป
 - การกำหนดสภาวะแวดล้อมของระบบเอง 336
 - คำสั่ง 382
 - ไฟล์สตาร์ทอัพ AIXwindows 336
 - ไฟล์สตาร์ทอัพระบบ 336
 - สำหรับโปรเซส 162
 - สำหรับการพิมพ์ 668
 - สำหรับคำสั่ง 162
- ข้อมูลสรุปคำสั่ง 162
 - IDs ระบบ 648

ข้อมูลสรุปคำสั่ง (ต่อ)

printing 668
การเปลี่ยนทิศทาง I/O 382
ข้อมูลระบบ 343
ความปลอดภัยของระบบ 336
ความปลอดภัยของไฟล์ 336
ชื่อล็อกอิน 648
ไต่เร็กทอรี 524
ไฟล์ 228
ไฟล์สำเนาสำรอง 28
รหัสผ่าน 648
ระบบไฟล์ 524
สภาวะแวดล้อมผู้ใช้ 343
สื่อบันทึกหน่วยเก็บ 28
ข้อมูลสรุปโปรเซส 162
ซีตจำกัด
โลจิคัลวอลุ่ม 418

ค

คณิตศาสตร์

การแยกตัวประกอบตัวเลข 153

คลาส

ผู้ใช้ 324

คลาสอักขระ

เซลล์เป้าหมาย 234

ความล้มเหลวของระบบ

การตรวจสอบโปรเซส 11
การตรวจสอบฮาร์ดแวร์ 10
การรีเซ็ตระบบ 13

คอนโซล

การแสดงชื่อ 339

คอมโพเนนต์ AIX Event Infrastructure

นิยาม 673, 684

ค่าติดตั้ง strict ดิสก์อินเตอร์ 447

ค่าติดตั้ง super strict ดิสก์อินเตอร์ 447

ค่าติดตั้งช่วง 446

คำ

การนับจำนวนของ 220

นิยาม 230

สแกนไวน์ Korn หรือ POSIX เซลล์ 246

คำที่สแกนไว้

Korn หรือ POSIX เซลล์ 246

เซลล์เป้าหมาย 290

คำศัพท์

สำหรับเซลล์ 230

คำสั่ง 140

/usr/bin/psh 230

/usr/bin/sh 230

> 376

>> 376

< 377

คำสั่ง (ต่อ)

l 379
acledit 328, 335
aclget 328, 333
aclput 328, 334
aixterm 342
alias 148
at 159, 160, 161
atq 160, 161
awk 212
banner 382
bsh 230, 234, 245, 283
C เซลล์ 309
capture 381
cat 218, 224, 377, 381
cd 517, 520
chdev 612
chfont 351
chgrp 334
chmod 325, 327, 328
chown 323, 334
chpq 667
clear 380
colrm 223
compress 41, 42
cp 214, 521
cpio 26
cpio -i 52
cpio -o 52
csh 230, 299
cut 221
date 69
del 226
df 475
diag 68
diff 220
dircmp 524
dosdel 227
dosdir 228
dosread 226
doswrite 227
echo 380, 381
env 341
exit 643
export 350
fdformat 50
file 216
find 54, 215
flcopy 51
format 50
fsck 24, 51
grep 11, 218, 379
head 220

คำสั่ง (ต่อ)

history 146
id 323, 643, 644
kill 11, 162, 172
Kom หรือ POSIX เซลล์ 276
ksh 52, 230, 278
ln 224, 225, 516
lock 336
login 332
logname 645
logout 643
ls 323, 324, 521
lsattr 612
lscfg 337
lscons 339
lsdev 612
lsdisp 339
lsfont 339
lsgroup 325
lskbd 340
lspp 340
man 144
mkdev 612
mkdir 519
more 217
mv 213
mkdir 519
mwm 346
nice 157
nl 223
pack 41, 42
page 217
passwd 647, 648
paste 222
pg 172, 217, 224
printenv 342
ps 11, 155, 172, 265
psh 230, 278
pwd 520
qcan 659
qchk 663
qmov 660
qprt 656
r 147
renice 157, 172
restore 30, 54, 55
rm 213, 226
rmdir 523
rsh 230
Rsh 230, 234, 245
script 381
setclock 69
sh 230

คำสั่ง (ต่อ)

shutdown 144
smit 30, 55, 351, 659, 661, 664, 667
smitrmat 161
sort 218
stty 341, 352
su 332, 643
tail 221
tapechk 24, 53
tar 26, 41, 53
tcopy 53
tee 380
text formatting 149
tn 11
touch 643
tsh 230
tty 339
uname 646
uncompress 41, 42, 43
unpack 41, 42, 43
wc 220
whatis 145
whereis 144
who 172, 645, 646
whoami 645
whoami 645
xinit 347
xlock 336
zcat 43
กลุ่ม 323
การทดแทนสตริง 147
การทำซ้ำที่ซ้อน 147
การบันทึกที่ซ้อนไว้ 146
การไฟฟ์ 141
การรวม 141
การสร้างชื่อช็อตคัต 148
ข้อความสั่งการใช้ 143
ชื่อ 141
เซลล์เป้าหมาย 288
นิยาม 230
ในตัว C เซลล์ 310
ในตัว Kom หรือ POSIX เซลล์ 258
ในตัวเซลล์เป้าหมาย 291
ผสม Kom เซลล์ 277
พารามิเตอร์ 143
แฟล็ก 142
ภาพรวม 141
ล็อกอิน 642
ไวยากรณ์ 141
สำเนาสำรอง 26, 54
สำหรับ BSD System Managers 369
คำสั่ง /usr/bin/psh 230
คำสั่ง /usr/bin/sh 230

คำสั่ง acledit 328, 335
คำสั่ง aclget 328, 333
คำสั่ง aclput 328, 334
คำสั่ง aixterm 342
คำสั่ง alias 148
คำสั่ง apropos 153
คำสั่ง at 159, 160, 161
คำสั่ง atq 160, 161
คำสั่ง awk 212
คำสั่ง backup 26, 54
คำสั่ง banner 382
คำสั่ง bsh 230, 234, 245, 283
คำสั่ง cal 151
คำสั่ง capture 381
คำสั่ง cat 218, 224, 377, 381
คำสั่ง cd 517, 520
คำสั่ง chdev 612
คำสั่ง chfont 351
คำสั่ง chgrp 334
คำสั่ง chmod 325, 327, 328
คำสั่ง chown 323, 334
คำสั่ง chpq 667
คำสั่ง clear 380
คำสั่ง colrm 223
คำสั่ง compress 41, 42
คำสั่ง cp 214, 521
คำสั่ง cpio 26
คำสั่ง cpio -i 52
คำสั่ง cpio -o 52
คำสั่ง csh 230, 299
คำสั่ง cut 221
คำสั่ง date 69
คำสั่ง del 226
คำสั่ง df 475
คำสั่ง diag 68
คำสั่ง diff 220
คำสั่ง dircmp 524
คำสั่ง dosdel 227
คำสั่ง dosdir 228
คำสั่ง dosread 226
คำสั่ง doswrite 227
คำสั่ง echo 380, 381
คำสั่ง env 341
คำสั่ง exit 643
คำสั่ง export 350
คำสั่ง factor 153
คำสั่ง fdformat 50
คำสั่ง find 54, 215
คำสั่ง flcopy 51
คำสั่ง format 50
คำสั่ง fsck 24, 51
คำสั่ง grep 11, 218, 379
คำสั่ง groups 323
คำสั่ง head 220
คำสั่ง history 146
คำสั่ง id 323, 643, 644
คำสั่ง In 224, 225, 516
คำสั่ง Is 323
คำสั่ง Isattr 612
คำสั่ง Iscfg 337
คำสั่ง Iscons 339
คำสั่ง Isdev 612
คำสั่ง Isdisp 339
คำสั่ง Isfont 339
คำสั่ง Isgroup 325
คำสั่ง Iskbd 340
คำสั่ง Islpp 340
คำสั่ง Issrc 207
คำสั่ง kill 11, 162, 172
คำสั่ง ksh 52, 230, 278
คำสั่ง lock 336
คำสั่ง login 332, 642
คำสั่ง logname 645
คำสั่ง logout 643
คำสั่ง ls 324, 521
คำสั่ง man 144
 BSD System Managers 358
คำสั่ง mkdev 612
คำสั่ง mkdir 519
คำสั่ง more 217
คำสั่ง mv 213
คำสั่ง mmdir 519
คำสั่ง mwm 346
คำสั่ง nice 157
คำสั่ง nl 223
คำสั่ง pack 41, 42
คำสั่ง page 217
คำสั่ง passwd 647, 648
คำสั่ง paste 222
คำสั่ง pg 172, 217, 224
คำสั่ง pr
 แฟล็ก 665
คำสั่ง printenv 342
คำสั่ง ps 11, 155, 172, 265
คำสั่ง psh 230, 278
คำสั่ง pwd 520
คำสั่ง qcan 659
คำสั่ง qchk 664
คำสั่ง qmov 660
คำสั่ง qpri 659
คำสั่ง qprt 656
 แฟล็ก 656, 666
คำสั่ง r 147
คำสั่ง refresh 207
คำสั่ง renice 157, 172
คำสั่ง restore 30, 54, 55

คำสั่ง rm 213, 226
 คำสั่ง rmdir 523
 คำสั่ง rsh 230
 คำสั่ง Rsh 230, 234, 245
 คำสั่ง runacct
 การรีสตาร์ท 183
 การเริ่มต้น 183
 คำสั่ง script 381
 คำสั่ง setclock 69
 คำสั่ง sh 230
 คำสั่ง shutdown 144
 คำสั่ง skulker 409
 คำสั่ง smit 55, 351
 การตรวจสอบสถานะของงานพิมพ์ 664
 การแปลง ASCII เป็น PostScript 667
 การยกเลิกงานพิมพ์ 659
 การย้ายงานพิมพ์ 661
 การเรียกคืนไฟล์ 30
 คำสั่ง smit rmat 161
 คำสั่ง sort 218
 คำสั่ง srcmstr 207
 คำสั่ง startsrc 206
 คำสั่ง stopsrc 206
 คำสั่ง stty 341, 352
 คำสั่ง su 332, 643
 คำสั่ง tail 221
 คำสั่ง tapechk 24, 53
 คำสั่ง tar 26, 41, 53
 คำสั่ง tcopy 53
 คำสั่ง tee 380
 คำสั่ง tn3270 353
 คำสั่ง touch 643
 คำสั่ง tracesoff 208
 คำสั่ง traceson 208
 คำสั่ง tsh 230
 คำสั่ง tty 339
 คำสั่ง uname 646
 คำสั่ง uncompress 41, 42, 43
 คำสั่ง unpack 41, 42, 43
 คำสั่ง wc 220
 คำสั่ง whatis 145
 คำสั่ง whereis 144
 คำสั่ง who 172, 645, 646
 คำสั่ง whoami 645
 คำสั่ง whoami 645
 คำสั่ง xinit 347
 คำสั่ง xlock 336
 คำสั่ง zcat 43
 คำสั่งในตัว
 cd 265
 fg 265
 kill 265
 pwd 265

คำสั่งในตัว (ต่อ)
 setgroups 265
 setenv 265
 unalias 265
 wait 265
 คำสั่งแบบง่าย
 นิยาม 230
 คำสั่งเซลล์
 fc 148
 history 146
 สมนาม r 147
 คำสั่งในตัว
 : 259, 292
 . 259, 292
 @ 310
 alias 265, 281, 310
 bg 265, 310
 break 259, 292, 310
 breaksw 310
 C เซลล์ 310
 case 310
 cd 292, 310
 chdir 310
 continue 259, 292, 310
 default 310
 dirs 310
 echo 265, 292, 310
 else 310
 end 310
 endif 310
 endsw 310
 eval 259, 292, 310
 exec 259, 278, 292, 310
 exit 259, 292, 310
 export 258, 259, 279, 292
 fc 148, 265, 281
 fg 310
 foreach 310
 getopts 265
 glob 310
 goto 310
 hangups 310
 hash 292
 hashstat 310
 history 310
 if 310
 jobs 265, 307, 310
 kill 265, 310
 Kom หรือ POSIX เซลล์ 258
 let 239, 265
 limit 310
 login 310
 logout 310

คำสั่งในตัว (ต่อ)

newgrp 259
nice 310
notify 310
onintr 310
popd 310
print 265
pushd 310
pwd 292
read 265, 292, 295
readonly 258, 259, 292
rehash 310
repeat 310
return 259, 292
set 259, 286, 292, 310
setenv 310
shift 259, 286, 292, 310
source 310
stop 310
suspend 310
switch 310
test 265, 292
time 310
times 259, 292
trap 259, 292
type 292
typeset 239, 251, 258, 259, 279
ulimit 265, 292
umask 265, 292, 310
unalias 281, 310
unhash 310
unlimit 310
unset 259, 292, 310
unsetenv 310
wait 292, 310
whence 265
while 310
คำสั่ง 265
เซลล์เป้าหมาย 287, 291
นิยาม 230
ปกติ 258, 265, 291
พิเศษ 258, 259, 291, 292
คำสั่งในตัว @
C เซลล์ 310
คำสั่งในตัว alias
C เซลล์ 310
Kom หรือ POSIX เซลล์ 265, 281
คำสั่งในตัว bg
C เซลล์ 310
Kom หรือ POSIX เซลล์ 265
คำสั่งในตัว break
C เซลล์ 310
Kom หรือ POSIX เซลล์ 259

คำสั่งในตัว break (ต่อ)

เซลล์เป้าหมาย 292
คำสั่งในตัว breaksw
C เซลล์ 310
คำสั่งในตัว case
C เซลล์ 310
คำสั่งในตัว cd
C เซลล์ 310
Kom หรือ POSIX เซลล์ 265
เซลล์เป้าหมาย 292
คำสั่งในตัว chdir
C เซลล์ 310
คำสั่งในตัว command
Kom หรือ POSIX เซลล์ 265
คำสั่งในตัว continue
C เซลล์ 310
Kom หรือ POSIX เซลล์ 259
เซลล์เป้าหมาย 292
คำสั่งในตัว default
C เซลล์ 310
คำสั่งในตัว dirs
C เซลล์ 310
คำสั่งในตัว echo
C เซลล์ 310
Kom หรือ POSIX เซลล์ 265
เซลล์เป้าหมาย 292
คำสั่งในตัว else
C เซลล์ 310
คำสั่งในตัว end
C เซลล์ 310
คำสั่งในตัว endif
C เซลล์ 310
คำสั่งในตัว endsw
C เซลล์ 310
คำสั่งในตัว eval
C เซลล์ 310
Kom หรือ POSIX เซลล์ 259
เซลล์เป้าหมาย 292
คำสั่งในตัว exec
C เซลล์ 310
Kom หรือ POSIX เซลล์ 259, 278
เซลล์เป้าหมาย 292
คำสั่งในตัว exit
C เซลล์ 310
Kom หรือ POSIX เซลล์ 259
เซลล์เป้าหมาย 292
คำสั่งในตัว export 258
Kom หรือ POSIX เซลล์ 259, 279
เซลล์เป้าหมาย 292
คำสั่งในตัว fc 148, 281
Kom หรือ POSIX เซลล์ 265
คำสั่งในตัว fg
C เซลล์ 310

คำสั่งในตัว fg (ต่อ)
 Korn หรือ POSIX เชลล์ 265

คำสั่งในตัว foreach
 C เชลล์ 310

คำสั่งในตัว getopts
 Korn หรือ POSIX เชลล์ 265

คำสั่งในตัว glob
 C เชลล์ 310

คำสั่งในตัว goto
 C เชลล์ 310

คำสั่งในตัว hangups
 C เชลล์ 310

คำสั่งในตัว hash
 เชลล์เป้าหมาย 292

คำสั่งในตัว hashstat
 C เชลล์ 310

คำสั่งในตัว history
 C เชลล์ 310

คำสั่งในตัว if
 C เชลล์ 310

คำสั่งในตัว jobs
 C เชลล์ 307, 310
 Korn หรือ POSIX เชลล์ 265

คำสั่งในตัว kill
 C เชลล์ 310
 Korn หรือ POSIX เชลล์ 265

คำสั่งในตัว let
 Korn หรือ POSIX เชลล์ 239, 265

คำสั่งในตัว limit
 C เชลล์ 310

คำสั่งในตัว login
 C เชลล์ 310

คำสั่งในตัว logout
 C เชลล์ 310

คำสั่งในตัว newgrp
 Korn หรือ POSIX เชลล์ 259

คำสั่งในตัว nice
 C เชลล์ 310

คำสั่งในตัว notify
 C เชลล์ 310

คำสั่งในตัว onintr
 C เชลล์ 310

คำสั่งในตัว popd
 C เชลล์ 310

คำสั่งในตัว print
 Korn หรือ POSIX เชลล์ 265

คำสั่งในตัว pushd
 C เชลล์ 310

คำสั่งในตัว pwd
 Korn หรือ POSIX เชลล์ 265
 เชลล์เป้าหมาย 292

คำสั่งในตัว read
 Korn หรือ POSIX เชลล์ 265

คำสั่งในตัว read (ต่อ)
 เชลล์เป้าหมาย 292, 295

คำสั่งในตัว readonly 258
 Korn หรือ POSIX เชลล์ 259
 เชลล์เป้าหมาย 292

คำสั่งในตัว rehash
 C เชลล์ 310

คำสั่งในตัว repeat
 C เชลล์ 310

คำสั่งในตัว return
 Korn หรือ POSIX เชลล์ 259
 เชลล์เป้าหมาย 292

คำสั่งในตัว set 286
 C เชลล์ 310
 Korn หรือ POSIX เชลล์ 259
 เชลล์เป้าหมาย 292

คำสั่งในตัว setenv
 C เชลล์ 310

คำสั่งในตัว setgroups
 Korn หรือ POSIX เชลล์ 265

คำสั่งในตัว shift 286
 C เชลล์ 310
 Korn หรือ POSIX เชลล์ 259
 เชลล์เป้าหมาย 292

คำสั่งในตัว source
 C เชลล์ 310

คำสั่งในตัว stop
 C เชลล์ 310

คำสั่งในตัว suspend
 C เชลล์ 310

คำสั่งในตัว switch
 C เชลล์ 310

คำสั่งในตัว test
 Korn หรือ POSIX เชลล์ 265
 เชลล์เป้าหมาย 292

คำสั่งในตัว time
 C เชลล์ 310

คำสั่งในตัว times
 Korn หรือ POSIX เชลล์ 259
 เชลล์เป้าหมาย 292

คำสั่งในตัว trap
 Korn หรือ POSIX เชลล์ 259
 เชลล์เป้าหมาย 292

คำสั่งในตัว type
 เชลล์เป้าหมาย 292

คำสั่งในตัว typeset 258
 Korn หรือ POSIX เชลล์ 239, 251, 259, 279

คำสั่งในตัว ulimit
 Korn หรือ POSIX เชลล์ 265
 เชลล์เป้าหมาย 292

คำสั่งในตัว umask
 C เชลล์ 310
 Korn หรือ POSIX เชลล์ 265

คำสั่งในตัว umask (ต่อ)
 เซลล์เป้าหมาย 292

คำสั่งในตัว unalias
 C เซลล์ 310
 Korn หรือ POSIX เซลล์ 265, 281

คำสั่งในตัว unhash
 C เซลล์ 310

คำสั่งในตัว unlimit
 C เซลล์ 310

คำสั่งในตัว unset
 C เซลล์ 310
 Korn หรือ POSIX เซลล์ 259
 เซลล์เป้าหมาย 292

คำสั่งในตัว unsetenv 310

คำสั่งในตัว wait
 C เซลล์ 310
 Korn หรือ POSIX เซลล์ 265
 เซลล์เป้าหมาย 292

คำสั่งในตัว whence
 Korn หรือ POSIX เซลล์ 265

คำสั่งในตัว while
 C เซลล์ 310

คำสั่งในตัวปกติ
 Korn หรือ POSIX เซลล์ 242, 265

คำสั่งในตัวพิเศษ
 Korn หรือ POSIX เซลล์ 241, 259
 เซลล์เป้าหมาย 292

คำสั่งผสม 277
 เซลล์เป้าหมาย 290

คำสั่งและพาด่วน 389

คิว
 เงื่อนไขสถานะ 664

คิวการพิมพ์
 เงื่อนไขสถานะ 663, 664

คีย์ควบคุม
 การเปลี่ยน 352
 การแสดงค่าติดตั้ง 341

คีย์บอร์ด
 การเปลี่ยน แอ็ททริบิวต์
 การใช้ คำสั่ง chhwkbd 369

คีย์บอร์ดแม่พิมพ์
 การแสดงรายการที่มีอยู่ในปัจจุบัน 340

เครือข่าย
 การแสดงชื่อระบบ 646
 สำหรับ BSD System Managers 353, 358, 363

เครื่องพิมพ์
 โค้ดที่ตั้ง 581
 เงื่อนไขสถานะ 664
 สำหรับ BSD System Managers 368

โค้ดที่ตั้ง 580
 dials/LPFKeys 584
 tty 582
 เครื่องพิมพ์/พล็อตเตอร์ 581

โค้ดที่ตั้ง (ต่อ)
 ดิสก์ที่ต่อพ่วงโดยตรง 583
 ดิสก์ที่ลิงก์แบบอนุกรม 583
 ดิสก์เก็ตไดรฟ์ 583
 ที่กำหนด 580
 พอร์ตหลายโปรโตคอล 584
 อะแดปเตอร์ 581
 อุปกรณ์ SCSI 582

โค้ดที่ตั้ง dials/LPFKeys 584

โค้ดที่ตั้งอะแดปเตอร์ 581

ง

งาน
 การจัดตารางเวลา 159
 การลบออกจากตารางเวลา 161
 การแสดงรายการที่จัดตารางเวลา 160

งานพิมพ์
 การจัดระดับความสำคัญ 659, 660
 การตรวจสอบสถานะ 663, 664
 การยกเลิก 659
 การย้าย 660
 การเลื่อน 661
 การสแตนท์ 656

เงื่อนไขสถานะ
 ของเครื่องพิมพ์ 663, 664

จ

จอแสดงผล
 การแสดงรายการที่มีอยู่บนระบบในปัจจุบัน 339

จำนวนไบต์ต่อไอ-โหนด (NBPI) 504

จุดต่อ 494

ช

ชนิด ACL
 AIXC 328
 NFS4 330

ชนิดไฟล์
 ข้อความ 209
 ไดร็อกทอรี 209
 โบนารี 209

ชื่อชื่อคัดสำหรับคำสั่ง
 การสร้าง 148

ชื่อพาธ
 ไดร็อกทอรี 518
 ไฟล์ 210
 สัมบูรณ์ 210, 518
 สัมพันธ์ 518

ชื่อพาธที่สัมพันธ์ 518

ชื่อพารามิเตอร์ 518

เซลล์

- C 230
- Korn 230
- POSIX 230
- การกำหนดสมนามคำสั่ง Korn หรือ POSIX 281
- การแก้ไขแบบแทรก Korn หรือ POSIX 269
- การควบคุมงาน Korn หรือ POSIX 268
- การควบคุมงานใน C เซลล์ 307
- การจัดการสัญญาณ Korn หรือ POSIX 269
- การจัดการสัญญาณใน C เซลล์ 309
- การใช้คำสั่ง Korn หรือ POSIX 276
- การดำเนินการคำสั่งใน C เซลล์ 317
- การทดแทนคำสั่ง Korn หรือ POSIX 238
- การทดแทนคำสั่งเซลล์เป้าหมาย 295
- การทดแทนคำสั่งใน C เซลล์ 316
- การทดแทนชื่อไฟล์ Korn หรือ POSIX 254
- การทดแทนชื่อไฟล์ใน C เซลล์ 303
- การทดแทนชื่อไฟล์ในเซลล์เป้าหมาย 286
- การทดแทนตัวแปรเซลล์เป้าหมาย 295
- การทดแทนตัวแปรใน C เซลล์ 301
- การทดแทนแบบมีเงื่อนไขในเซลล์เป้าหมาย 284
- การทดแทนประวัติใน C เซลล์ 317
- การทดแทนสมนามใน C เซลล์ 300
- การประเมินทางคณิตศาสตร์ Korn หรือ POSIX 239
- การเปลี่ยนทิศทาง I/O Korn หรือ POSIX 255
- การเปลี่ยนทิศทาง I/O ในเซลล์เป้าหมาย 287
- การเปลี่ยนทิศทางอินพุตและเอาต์พุตใน C เซลล์ 320
- การเริ่มต้น C เซลล์ 299
- การเริ่มต้น Korn หรือ POSIX 278
- การเริ่มต้นเซลล์เป้าหมาย 283
- การเริ่มต้นที่จำกัด 234, 245
- การสร้างเซลล์สคริปต์ 235
- การใส่ไว้ในเครื่องหมายอัฒภาคใน Korn หรือ POSIX 243
- คลาสอักขระในเซลล์เป้าหมาย 234
- คำที่สงวนไว้ Korn หรือ POSIX 246
- คำศัพท์ 230
- คำสั่งในตัว C 310
- คำสั่งในตัว Korn หรือ POSIX 258
- คำสั่งในตัวเซลล์เป้าหมาย 291
- คำสั่งผสม Korn หรือ POSIX 277
- คุณลักษณะ 232
- ชนิด 230
- ดีฟอลต์ 230
- ตัวแปรที่กำหนดไว้ล่วงหน้าในเซลล์เป้าหมาย 298
- ตัวแปรที่ใช้โดยเซลล์เป้าหมาย 297
- ตัวแปรผู้ใช้กำหนดในเซลล์เป้าหมาย 295
- ตัวแปรสถานะแวดล้อมใน C เซลล์ 305
- ที่จำกัด 230
- ที่เชื่อถือได้ 230
- นิพจน์เงื่อนไข Korn หรือ POSIX 242
- บอร์น 230
- ประวัติคำสั่ง Korn หรือ POSIX 281

เซลล์ (ต่อ)

- พารามิเตอร์ 251
- พารามิเตอร์ตำแหน่งในเซลล์เป้าหมาย 286
- ภาพรวม 229
- มาตรฐาน 230
- รายการของคำสั่งในตัวพิเศษ Korn หรือ POSIX 241
- รายการของคำสั่งในตัวปกติ Korn หรือ POSIX 242
- รายการคำสั่งในตัวเซลล์เป้าหมาย 287
- รายการประวัติใน C เซลล์ 318
- รีโมต 230
- ล็อกอิน 230
- สถานะการออก Korn หรือ POSIX 250
- สถานะแวดล้อม Korn หรือ POSIX 279
- สถานะแวดล้อมเซลล์เป้าหมาย 283
- สิ่งอำนวยความสะดวกโปรแกรม Korn หรือ POSIX 257
- เซลล์ที่จำกัด 230
- การเริ่มต้น 234
- เซลล์ที่เชื่อถือได้ 230
- เซลล์เป้าหมาย 230
- การจัดการสัญญาณ 290
- การจัดรูปแบบ 286
- การทดแทนคำสั่ง 295
- การทดแทนชื่อไฟล์ 286
- การทดแทนตัวแปร 295
- การทดแทนแบบมีเงื่อนไข 284
- การเปลี่ยนทิศทางอินพุตและเอาต์พุต 287
- การเริ่มต้น 283
- การอ้างถึงอักขระ 289
- คลาสอักขระ 234
- คำที่สงวนไว้ 290
- คำสั่ง 288
- คำสั่งในตัว 291
- คำสั่งผสม 290
- ตัวแปร 297
- ตัวแปรที่กำหนดไว้ล่วงหน้า 298
- ตัวแปรผู้ใช้กำหนด 295
- พารามิเตอร์ตำแหน่ง 286
- รายการของคำสั่งในตัว 287
- สถานะแวดล้อม 283
- เซลล์ไพธอน 141
- เซลล์มาตรฐาน
- นิพจน์เงื่อนไข 242
- เซลล์ย่อย
- นิยาม 230
- เซลล์สคริปต์ 141
- การระบุเซลล์ 232
- การสร้าง 235

ซ

ซอฟต์แวร์

- การตรวจสอบปัญหาอุปกรณ์ 612

ซีดีรอม

- ระบบไฟล์ 479
- เซิร์ฟเวอร์ X
 - ไฟล์สตาร์ทอัพ 346
- เซิร์ฟเวอร์ย่อย
 - การปิด การติดตาม 208
 - การเปิดการติดตาม 208
 - การเริ่มต้น 206
 - การแสดงผลสถานะ 207
 - การหยุด 206
 - คำอธิบายของ 204

เอ

- ฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเอง 578
- ฐานข้อมูลคอนฟิกูเรชันอุปกรณ์
 - การซิงโครไนซ์กับ Logical Volume Manager 419

ด

ดิสก์

- การเพิ่ม 391
- การเอาออก 434
- ดิสก์(ฮาร์ดไดรฟ์) 398
 - การกำหนดคอนฟิก 398
- ดิสก์ไดรฟ์
 - การกู้คืนข้อมูล
 - โดยไม่ต้องจัดรูปแบบใหม่ 410
 - การแก้ไขปัญหา 408
 - การจำกัดสิทธิ์เข้าถึงไดเรกทอรีบน 409
 - การติดตั้งพื้นที่ว่างจากดิสก์อื่น 410
 - การถอนติดตั้งระบบไฟล์บนดิสก์ 477
 - การทำให้พื้นที่ว่างบน 409
 - การลบไฟล์ที่เลิกใช้แล้วออกจาก 409
 - การวินิจฉัย 408
 - ให้ดูเพิ่มเติมที่ฟิลิคัลลวลุ่ม 401
- ดิสก์ไดรฟ์(ฮาร์ดไดรฟ์)
 - การแสดงผลการระบบไฟล์ 477
 - ความล้มเหลวของ
 - ตัวอย่างของการกู้คืนจาก 415
 - ต่อพ่วงโดยตรง 583
 - ที่ลิงก์แบบอนุกรม
 - โคดที่ติดตั้ง 583
- ดิสก์ไดรฟ์คงที่(ฮาร์ดไดรฟ์) 489
 - ให้ดูเพิ่มเติมที่ดิสก์ไดรฟ์ 409
- ดิสก์ไดรฟ์ที่ล้มเหลว
 - ตัวอย่างของการกู้คืนจาก 415
- ดิสก์โอเวอร์ไฟล์, การแก้ไข 489

ดิสเก็ต

- การคัดลอกไปยังหรือจาก 51
- การจัดการ 26
- การจัดรูปแบบ 50

ดิสเก็ต (ต่อ)

- การใช้เป็นสื่อสำเนาสำรอง 26
- ดิสเก็ตไดรฟ์
 - โคดที่ติดตั้ง 583
- ดีฟอลต์
 - การเปลี่ยนแปลง 347
- ดีฟอลต์เซลล์ 230
- ดีวีดี
 - ระบบไฟล์ 479
- ไดเรกทอรี
 - ไดเรกทอรี
 - ผลกระทบของแฟรกเมนต์ที่ใส่ต่อขนาดของ 513
 - ไดเรกทอรี 516
 - การคัดลอก 521
 - การจัดระเบียบ 517
 - การติดตั้ง 494
 - การทำงาน 517
 - การเปรียบเทียบเนื้อหา 524
 - การเปลี่ยนความเป็นเจ้าของ 323
 - การเปลี่ยนชื่อ 519
 - การเปลี่ยนแปลง 520
 - การเปลี่ยนสิทธิ์อนุญาต 327
 - การย้าย 519
 - การระบุด้วยคำย่อ 518
 - การลบ 523
 - การลิงก์ 224
 - การสร้าง 519
 - การแสดงผล 520
 - การแสดงผลเนื้อหา 521
 - การแสดงผลรายการไฟล์ 521
 - การแสดงผลรายการไฟล์ DOS 228
 - คำย่อ 518
 - โครงสร้าง 517
 - ชนิด 517
 - ชื่อพาร 518
 - ไดเรกทอรีย่อย 517
 - พารেন্ট 517
 - ภาพรวม 516
 - ระเบียบการตั้งชื่อ 518
 - ราก 516
 - สิทธิ์อนุญาต 324
 - โหมดการเข้าถึง 324
 - โสม 517
 - ไดเรกทอรี/export 472
 - ไดเรกทอรี/usr/share 470
 - ไดเรกทอรี\$HOME 518
 - ไดเรกทอรี~(โสม) 518
 - ไดเรกทอรี Shared Product Object Tree (SPOT) 472
 - ไดเรกทอรี SPOT 472
 - ไดเรกทอรีการทำงาน 517
 - ไดเรกทอรีพารেন্ট 517
 - ไดเรกทอรีโสม 517

ด

ตัวกรอง 379
ตัวกรอง `enscript` 666
ตัวดำเนินการ
 C เซลล์ 315
ตัวดำเนินการการเปลี่ยนทิศทางต่อท้าย 376
ตัวถูกดำเนินการ
 ในคำสั่ง 143
ตัวแปร
 CDPATH 236
 COLUMNS 236
 EDITOR 236
 ENV 236
 ERRNO 236
 FCEDIT 236
 FPATH 236
 HISTFILE 236
 HISTSIZ 236, 281
 HOME 236
 IFS 236
 Korn หรือ POSIX เซลล์ 236
 LANG 236
 LC_ALL 236
 LC_COLLATE 236
 LC_CTYPE 236
 LC_MESSAGES 236
 LINENO 236
 LINES 236
 MAIL 236
 MAILCHECK 236
 MAILPATH 236
 NLSPATH 236
 OLDPWD 236
 OPTARG 236
 OPTIND 236
 PATH 236
 PPID 236
 PS1 236
 PS2 236
 PS3 236
 PS4 236
 PWD 236
 RANDOM 236
 REPLY 236
 SECONDS 236
 SHELL 236
 TMOUT 236
 VISUAL 236
 การเอ็กซ์พอร์ต 350
 ขีดเส้นใต้ 236
 เซลล์เป้าหมาย 297, 298
 เซลล์เป้าหมายที่ผู้ใช้กำหนด 295

ตัวแปร (ต่อ)

ตัวแปร
 SHELL PROMPT 236
ตัวแปร SHELL PROMPT 236
ที่กำหนดไว้ล่วงหน้า 236
ที่ผู้ใช้กำหนด 236
สภาวะแวดล้อม C เซลล์ 305
ตัวแปร CDPATH 236
ตัวแปร COLUMNS 236
ตัวแปร EDITOR 236
ตัวแปร ENV 236
ตัวแปร ERRNO 236
ตัวแปร FCEDIT 236
ตัวแปร FPATH 236
ตัวแปร HISTSIZE 236, 281
ตัวแปร HOME 236
ตัวแปร IFS 236
ตัวแปร LANG 236
ตัวแปร LC_ALL 236
ตัวแปร LC_COLLATE 236
ตัวแปร LC_CTYPE 236
ตัวแปร LC_MESSAGES 236
ตัวแปร LINENO 236
ตัวแปร LINES 236
ตัวแปร MAIL 236
ตัวแปร MAILCHECK 236
ตัวแปร MAILPATH 236
ตัวแปร NLSPATH 236
ตัวแปร OLDPWD 236
ตัวแปร OPTARG 236
ตัวแปร OPTIND 236
ตัวแปร PATH 236
ตัวแปร PPID 236
ตัวแปร PS1 236
ตัวแปร PS2 236
ตัวแปร PS3 236
ตัวแปร PS4 236
ตัวแปร PWD 236
ตัวแปร RANDOM 236
ตัวแปร REPLY 236
ตัวแปร SECONDS 236
ตัวแปร SHELL 236
ตัวแปร TMOUT 236
ตัวแปร VISUAL 236
ตัวแปรขีดเส้นใต้ 236
ตัวแปรเซลล์
 การเอ็กซ์พอร์ต 350
 นิยาม 230
 โลคัล 350
ตัวแปรที่กำหนดไว้ล่วงหน้า
 Korn หรือ POSIX เซลล์ 236
 เซลล์เป้าหมาย 298
ตัวแปรผู้ใช้กำหนด 236

ตัวแปรผู้ใช้กำหนด (ต่อ)
เซลล์เป้าหมาย 295
ตัวแปรสถานะแวดล้อม
C เซลล์ 305
การตั้งค่า 342
ตัวระบุ
นิยาม 230

ท

ทำซ้ำการรวบรวมเหตุการณ์ 687
เทป
การคัดลอกไปยังหรือจาก 53
การใช้เป็นสื่อสำเนาสำรอง 26
การตรวจสอบคุณภาพ 53
เทปไตรัฟ
การจัดการ 618
ไฟล์พิเศษสำหรับ 630
แอ็ดทริบิวต์
เปลี่ยนได้ 618, 620, 621, 622, 623, 624, 625, 626, 627,
628, 629
เทอร์มินัล
การลือก 336
การแสดงชื่อ 339
การแสดงการกำหนดคีย์ควบคุม 341
การตั้งค่าติดตั้ง 342
ที่ไม่ได้ใส่ใจ 322
สำหรับ BSD System Managers 374
เทอร์มินัล ASCII
การเพิ่ม 650
เทอร์มินัล X 650
เทอร์มินัล, ที่ลือกอ็อป 172
เทอร์มินัลแสดงอักขระ
การเพิ่ม 650

ท

นโยบายการกำหนดตารางเวลาการเขียน 443
นโยบายการจัดสรร intra-disk 448
นโยบายการจัดสรรของดิสก์อินเตอร์ 445
นโยบายตรวจสอบการเขียน 450
นาฬิกา
การรีเซ็ต 68
นาฬิกากระบบ
การทดสอบแบตเตอรี่ 68
การรีเซ็ต 68
นิพจน์
C เซลล์ 315
การค้นหาไฟล์ที่ตรงกัน 215
เงื่อนไข 242
นิพจน์ปกติ 212
นิยามเมนู 348

เน็ตเวิร์กเชิงลำดับชั้น 20
เน็ตเวิร์กแบบราบ 20

บ

บรรทัด
การนับจำนวนของ 220
บรรทัดของข้อความ
การผนวกเข้ากับไฟล์ 381
บล็อก
ต้นทุนประสิทธิภาพของ 506
บล็อกควบคุมโลจิคัลวอลุ่ม
ไม่ได้ป้องกันจากการเข้าถึงโลจิคัลวอลุ่มแบบดิบ 433
บัญชีการใช้เครื่องพิมพ์ 197
บัญชีการใช้ดิสก์ 196
บัญชีระบบ
การใช้ CPU
การแสดงผล 191
การตั้งค่า 186
การรายงานข้อมูล
ภาพรวม 174
กิจกรรมระบบ
ข้อมูล 179
ข้อผิดพลาด tacct
การแก้ไข 197
ข้อผิดพลาด wtmp
การแก้ไข 198
ข้อมูลการใช้เครื่องพิมพ์ 193
การรวบรวม 197
การรายงาน 178
ข้อมูลการใช้ดิสก์ 178, 192
การรวบรวม 196
ข้อมูลกิจกรรมระบบ
การแสดงผล 189
การแสดงผลในขณะรันคำสั่ง 189
ข้อมูลโปรเซส
การรวบรวม 195
การรายงาน 196
ข้อมูลเวลาเชื่อมต่อ 178, 192, 195
ความล้มเหลว
การกู้คืนจาก 183
ค่าธรรมเนียม
การรายงาน 179
การเรียกเก็บ 197
คำสั่ง
การรันจากคีย์บอร์ด 181
การรันโดยอัตโนมัติ 180
คำสั่ง runacct
การรีสตาร์ท 183
การเริ่มต้น 183
ปัญหา
การแก้ไขข้อผิดพลาด runacct 200

บัญชีระบบ (ต่อ)

ปัญหา (ต่อ)

- การแก้ไขเวลาที่ไม่ดี 200
- การแก้ไขสิทธิ์อนุญาตไฟล์ที่ไม่ถูกต้อง 199
- การแก้ไขไฟล์วันหยุดที่ล้าสมัย 193

ไฟล์

- ไฟล์ข้อมูล 182
- ไฟล์คำสั่ง runnact 184
- ภาพรวม 182
- รายงานและไฟล์ข้อมูลสรุป 182
- รูปแบบ 185

ไฟล์วันหยุด

- การอัปเดต 193

ภาพรวม 174

รายงาน

- งบประมาณ 179
- รายเดือน 177, 178
- รายวัน 175

เรีกคอร์ดสรุป 177

บันทึก JFS (ระบบไฟล์ที่เจอร์นัล)

ขนาดของ 506

แบดชโปรเซส 159

แบดเตอร์นาฟิกา 68

แบดเตอร์ระบบ 68

แบบแถบของดิสก์ 450

ไบต์

- การนับจำนวนของ 220

ป

ปฏิทิน

- การแสดง 151

ประวัติ

- การแก้ไข 148
- การทดแทนใน C เซลล์ 317
- รายการใน C เซลล์ 318

ประวัติคำสั่ง

- Korn หรือ POSIX เซลล์ 281
- การแก้ไข 148
- การทดแทน 281

ประสิทธิภาพ

- BSD System Managers 367
- การปรับปรุง
 - การกำหนดโลจิคัลวอลุ่มแบบดิส 433

ปัญหาเทอร์มินัล

- การหยุดโปรเซสที่เก่า 172

โปรแกรม

- การตัดลอกเอาต์พุตเข้าในไฟล์ 380

โปรแกรมแก้ไข

- ed 213
- emacs 269, 270
- gmacs 269, 270

โปรแกรมแก้ไข (ต่อ)

- vi 213, 269

- การแก้ไขแบบแทรก 269

โปรแกรมแก้ไข ed 213

โปรแกรมแก้ไข emacs

- การแก้ไขแบบแทรก 269, 270

โปรแกรมแก้ไข gmacs

- การแก้ไขแบบแทรก 269, 270

โปรแกรมแก้ไข vi 213

- การแก้ไขแบบแทรก 269, 272, 273, 274, 275

- การย้ายเคอร์เซอร์ 273

- คำสั่ง input edit 272

- คำสั่ง motion edit 273

- คำสั่ง search edit 273

- คำสั่ง text-modification edit 274

- คำสั่งแก้ไข เบ็ดเตล็ด 275

- คำสั่งแก้ไขที่ใช้กันทั่วไป 275

- โหมด อินพุต 272

- โหมดควบคุม 272

โปรเซส 140

- daemon 154

- zombie 154

- การจัดการของ 166, 167

- การจัดตารางเวลาสำหรับการดำเนินงานในภายหลัง 159

- การตรวจสอบสถานะ 155

- การตั้งลำดับความสำคัญแรกเริ่ม 157

- การเปลี่ยนลำดับความสำคัญ 157

- การเปลี่ยนลำดับความสำคัญของ 170

- การผูกเข้ากับตัวประมวลผล 171

- การมอนิเตอร์ของ 167

- การยกเลิกกระบวนการพื้นหน้า 158

- การยุติของ 170

- การรวบรวมข้อมูลการบัญชีบน 195

- การรีสตาร์ทที่หยุด 158

- การเริ่มต้น 155

- การลบออกจากตารางเวลา 161

- การสร้างรายงานการบัญชี 196

- การแสดงผลการใช้ CPU 191

- การแสดงให้เห็นทั้งหมด 155

- การแสดงผลรายการที่จัดตารางเวลา 160

- การหยุดกระบวนการพื้นหน้า 158

- คำอธิบาย 154

- แบดช์ 159

- พื้นหน้า 154

- พื้นหลัง 154

โปรเซส daemon 154

โปรเซสแปรผัน 384

โปรเซสแปรผัน

- การยกเลิกความล้มเหลวของ 419

โปรไฟล์

- ไฟล์ 57

- ภาพรวม 57

ผ

ผลิตภัณฑ์ซอฟต์แวร์

- การแสดงผลข้อมูลเกี่ยวกับ 340
- ผู้ใช้
 - กลุ่ม 323
 - การเปลี่ยนเป็นผู้ใช้อื่น 643
 - การแสดงผล ID ระบบ 644
 - การแสดงผลบุคคลที่ล็อกอิน 646
 - การแสดงผลข้อมูลกลุ่ม 325
 - คลาส 324
- ผู้บริโภคนเหตุการณ์ 675
- ผู้สร้างเหตุการณ์
 - นิยาม 675
- แผนผังไดรเร็กทอรี 517

พ

พร้อมต์

- การเปลี่ยน 352
- พรินเตอร์ PostScript
 - การพิมพ์ไฟล์ ASCII 666
- พอร์ตหลายโปรโตคอล
 - โค้ดที่ตั้ง 584

พาธ

- ไดรเร็กทอรี 518

พาร์ติชันฟิสิกัล

- ขนาด 428
- นิยาม 428

พาร์ติชันโลจิคัล

- นโยบายการจัดสรรของดิสก์อินเตอร์ 446
- นิยาม 429

พารามิเตอร์

- Korn หรือ POSIX เซลล์ 251, 253
- ตำแหน่ง 251
- ที่กำหนดไว้ล่วงหน้า 253
- ที่ระบุ 251
- ในคำสั่ง 143

- พิเศษ 251, 253

พารามิเตอร์ตำแหน่ง 251

- เซลล์เป้าหมาย 286

พารามิเตอร์ที่ระบุ 251

พารามิเตอร์พิเศษ 251

พื้นที่การเลื่อนหน้า

- การจัดสรร 454
- การเปลี่ยนขนาดของ hd6 459
- การเปลี่ยนลักษณะของ 458
- การย้าย hd6 459
- การลบ 458
- คำสั่งสำหรับการจัดการ 456
- ภาพรวม 453, 454
- ลักษณะสำหรับการสร้าง 456

พื้นที่การเลื่อนหน้า (ต่อ)

- โหมดการจัดสรร early 454

- โหมดการจัดสรร late 454

พื้นที่การเลื่อนหน้า

- AIX for BSD System Managers 362

พื้นที่ว่าง

- การแสดงผลที่มีอยู่ 475

พื้นที่ว่างเปล่า

- การตีความของ 299
- นิยาม 230

พื้นที่สลับ

- ให้ดูที่พื้นที่การเลื่อนหน้า 453

โปรซีเดเจอร์การกู้คืน

- การเข้าถึงระบบที่จะไม่บูต 22

- การรีบูตระบบโดยใช้ กราฟิก planar 8

โปรซีเดเจอร์การกู้คืนจากดิสก์ไดรฟ์ที่ล้มเหลว

- ตัวอย่างของ 415

ไพพ์ 379

ไพพ์ไลน์

- นิยาม 230, 379

พ

พอนต์

- การเปลี่ยน 351

- การแสดงผลรายการที่พร้อมใช้งาน 339

ฟิลิคัลวอลุ่ม

- การกำหนดคอนฟิกดิสก์ 398

- การเปลี่ยนในกลุ่มวอลุ่มที่ทำสำเนา 401

- การย้ายเนื้อหา 396

- การสร้างจากดิสก์ไดรฟ์ที่มีอยู่ 400

- คัดลอก JFS ไปยังที่อื่น 514

- คัดลอกโลจิคัลวอลุ่มไปยังที่อื่น 392

- นิยาม 426

แฟร็กเมนต์

- ขนาดของ

- การระบุ 504

- ข้อจำกัดสำหรับไดรเวอร์อุปกรณ์ 513

- ต้นทุนประสิทธิภาพของ 506

- ผลกระทบต่อการใช้ดิสก์ 501

- ผลกระทบต่อการทำสำเนาสำรอง/เรียกคืน 513

- และหมายเลขตัวแปรของโอ-โหนด 501

แฟล็ก 141

- ในคำสั่ง 142

- สำหรับคำสั่ง pr 665

- สำหรับคำสั่ง qprt 656, 666

แฟล็กคำสั่ง 141

ไฟล์ของการมอนิเตอร์เหตุการณ์ 677

ไฟล์ 516

- /dev/null 377

- /etc/environment 344

- /etc/passwd 230

ไฟล์ (ต่อ)

- /etc/profile 344
- /etc/security/passwd 322
- .hushlogin 643
- .mwmrc 348
- .profile 345
- .Xdefaults 347
- .xinitrc 347
- ASCII 209
- HISTFILE 281
- การกำหนดหมายเลขบรรทัด 223
- การเก็บถาวร 53
- การขยาย 43
- การเขียนไปยังเอาต์พุต 221
- การค้นหาสตริง 218
- การคลายบีบอัด 43
- การคลายแพ็คเกจ 43
- การตัดลอก 214
- การตัดลอกจากDOS 226
- การตัดลอกจากเทปหรือดิสก์ 52
- การตัดลอกจากหน้าจอ 381
- การตัดลอกไปยังDOS 227
- การจัดการ 213
- การจัดรูปแบบสำหรับการแสดง 217
- การจับคูนีพจน์ 215
- การดึงข้อมูลจากหน่วยเก็บ 53
- การต่อ 377
- การตัดฟิลต์ที่เลือกจาก 221
- การติดตั้ง 494
- การทำสำเนาสำรอง 54
- การบีบอัด 41
- การเปรียบเทียบ 220, 524
- การเปลี่ยนความเป็นเจ้าของ 323
- การเปลี่ยนชื่อ 213
- การเปลี่ยนสิทธิ์อนุญาต 327
- การผนวกข้อความหนึ่งบรรทัด 381
- การผสมบรรทัดของหลาย 222
- การย้าย 213
- การรวม 377
- การระบุชนิด 216
- การระบุตำแหน่งส่วน 144
- การเรียกคืน 30, 34, 55
- การเรียงลำดับข้อความ 218
- การลบ 213
- การลบDOS 227
- การลบคอลัมน์ 223
- การลบที่ลิงก์ 226
- การลิงก์ 224, 225
- การวางข้อความ 222
- การสร้างด้วยการเปลี่ยนทิศทางจากคีย์บอร์ด 377
- การแสดงชนิด 216
- การแสดงผลเนื้อหา 217
- การแสดงผลบรรทัดแรก 220

ไฟล์ (ต่อ)

- การแสดงบรรทัดสุดท้าย 221
- การอัดแน่น 41
- ความเป็นเจ้าของ 323
- ชื่อพาร 210, 518
- ตัวอธิบาย 375, 378
- นิพจน์ปกติ 212
- ไบนารี 209
- ปฏิบัติการ 209
- แผนผัง 464
- ไฟล์.env 345
- เมตาอักขระ 212
- ระเบียบการตั้งชื่อ 210
- สภาวะแวดล้อม 344
- สำหรับBSD System Managers 357, 362
- สิทธิ์อนุญาต 209, 324
- โหมดการเข้าถึง 324
- ไฟล์/dev/null 377
- ไฟล์/etc/environment 344
- ไฟล์/etc/inittab
 - การเปลี่ยน 9
- ไฟล์/etc/profile 57, 344
- ไฟล์/etc/security/passwd 322
- ไฟล์.env 345
- ไฟล์.hushlogin 643
- ไฟล์.mwmrc 348
- ไฟล์.profile 57, 345
- ไฟล์.Xdefaults 347
- ไฟล์.xinitrc 347
- ไฟล์DOS
 - การตัดลอก 226
 - การแปลง 226
 - การลบ 227
 - การแสดงรายการเนื้อหา 228
- ไฟล์inittab 9
 - srcmstr daemon ใน 205
- ไฟล์motd 70
- ไฟล์PostScript
 - การแปลงจากASCII 666, 667
- ไฟล์ล็อกอิน
 - ไฟล์/etc/profile 57
- ไฟล์ข้อความ
 - การกำหนดหมายเลขบรรทัด 223
 - การค้นหาสตริง 218
 - การต่อ 377
 - การตัดส่วน 221
 - การเรียงลำดับ 218
 - การลบคอลัมน์ 223
 - การวางส่วน 222
 - การสร้างจากอินพุตคีย์บอร์ด 377
- ไฟล์ที่ลิงก์
 - การลบ 226
- ไฟล์โปรไฟล์ 343

ไฟล์แม่พิมพ์ 449
ไฟล์ร็อดส์
 การแก้ไข 347, 348
ไฟล์ลือกอิน
 /etc/environment 344
 /etc/profile 344
 .profile 345
 ไฟล์.env 345
 ไฟล์.profile 57
ไฟล์สตาร์ทอัพ
 AIXwindows 346
 C เซลล์ 299
 เซิร์ฟเวอร์ X 346
 ระบบ 343

ก

ภาษา
 สองทาง 342
ภาษาสองทาง 342

ม

มุมมองระดับสูงของ AIX Event Infrastructure
 นิยาม 676
เมตาอักขระ 212
 การใส่ไว้ในเครื่องหมายอัฒภาคใน Korn หรือ POSIX เซลล์ 243
 นิยาม 230
แมนเพจ
 การค้นหาด้วยการค้นหาคีย์เวิร์ด 153
แม่พิมพ์
 คีย์บอร์ด 340

ร

รหัสผ่าน
 การตั้งค่าเป็น null 648
 การเปลี่ยนหรือการตั้งค่า 647
 คำแนะนำ 647
 คำอธิบาย 641
ระดับการรัน
 การระบุ 16
 การแสดงผลประวัติ 16
ระดับการรันของระบบ 16
ระบบ
 การกำหนดสถานะแวดล้อมเอง 350, 351, 352
 การจัดการ 464
 การบัญชี 641
 การเปลี่ยนพร้อมท์ 352
 การเปิดเครื่อง 642
 การรักษาความปลอดภัย 321

ระบบ (ต่อ)
 การเริ่มต้น 4
 การแสดงชื่อ 646
 ตัวแปรดีฟอลต์ 344
 ไฟล์สตาร์ทอัพ 343
 สถานะแวดล้อม 336
ระบบปฏิบัติการ
 การลือกเอาต์ 643
ระบบไฟล์
 คำอธิบาย 641
ระบบการจัดรูปแบบข้อความ 666
ระบบการบัญชี
 BSD System Managers 360
 การใช้ CPU
 การแสดงผล 191
 การตั้งค่า 186
 การรายงานข้อมูล
 ภาพรวม 174
 ข้อผิดพลาด tacct
 การแก้ไข 197
 ข้อผิดพลาด wtmp
 การแก้ไข 198
 ข้อมูลการใช้เครื่องพิมพ์ 178, 197
 การแสดงผล 193
 ข้อมูลการใช้ดีสก์ 196
 การรายงาน 178
 การแสดงผล 192
 ข้อมูลกิจกรรมระบบ
 การรายงาน 179
 การแสดงผล 189
 การแสดงผลในขณะรันคำสั่ง 189
 ข้อมูลโปรเซส
 การรวบรวม 195
 การรายงาน 196
 ข้อมูลเวลาเชื่อมต่อ
 การรวบรวม 195
 การรายงาน 178
 การแสดงผล 192
 ความล้มเหลว
 การกู้คืนจาก 183
 ค่าธรรมเนียม
 การรายงาน 179
 การเรียกเก็บ 197
 คำสั่ง
 การรันจากคีย์บอร์ด 181
 การรันโดยอัตโนมัติ 180
 ภาพรวม 180
 คำสั่ง runacct
 การรีสตาร์ท 183
 การเริ่มต้น 183
ปัญหา
 การแก้ไขข้อผิดพลาด runacct 200
 การแก้ไขไฟล์วันหยุดที่ล้าสมัย 193

ระบบการบัญชี (ต่อ)

ปัญหา (ต่อ)

การแก้ไขเวลาที่ไม่มีดี 200

การแก้ไขสิทธิอนุญาตไฟล์ที่ไม่ถูกต้อง 199

ไฟล์

ไฟล์ข้อมูล 182

ไฟล์คำสั่ง runacct 184

ภาพรวม 182

รายงานและไฟล์ข้อมูลสรุป 182

รูปแบบ 185

ไฟล์วันหยุด

การอัปเดต 193

ภาพรวม 174

รายงาน

งบประมาณ 179

รายเดือน 177, 178

รายวัน 175

เรีกคอร์ดสรุป 177

ระบบที่ใช้งานไม่ได้

การรีสตาร์ท ระบบ 13

ระบบที่ไม่ได้ใช้งาน

การตรวจสอบโปรเซส 11

การตรวจสอบฮาร์ดแวร์ 10

ระบบที่ไม่ได้ใช้งานอยู่

การตรวจสอบโปรเซส 11

การตรวจสอบฮาร์ดแวร์ 10

ระบบที่ไม่ทำงาน

การรีสตาร์ท ระบบ 13

ระบบปฏิบัติการ

การปิด 144

การล็อกอิน 642

การแสดงชื่อ 646

การโหลด 13

ระบบผู้ใช้เดี่ยว

การเปลี่ยนระดับการรันบน 17

ระบบไฟล์

/opt 466

/proc 466

CDRFS 479

Journaled File System (JFS) 464

UDFS 479

กลุ่ม

การติดตั้ง 477

การถอนติดตั้ง 477

การแก้ไขที่เสียหาย 487

การตรวจสอบความสอดคล้องกัน 51

การตรวจสอบคุณภาพของ 481

การติดตั้ง 477, 494

การถอนติดตั้ง 477

การทำารซ่อมแซมแบบโต้ตอบ 51

การทำสำเนาสำรองโดยใช้สคริปต์ 48

การทำสำเนาสำรองระบบไฟล์ผู้ใช้ 29

การบายพาส 433

ระบบไฟล์ (ต่อ)

การบีบอัดข้อมูล 509

การลดขนาดในกลุ่มวอลุ่มราก 482

คำสั่งสำหรับการจัดการ 474, 475

โครงสร้าง 466

ชนิด

ซีดีรอม 499

ดีวีดีรอม 499

ระบบไฟล์ที่เจอร์นัล (JFS) 499

ระบบไฟล์ที่เจอร์นัลที่เพิ่มประสิทธิภาพ (JFS2) 499

ระบบไฟล์เน็ตเวิร์ก (NFS) 499

ดิสก์โอเวอร์โพลว 489

ตัวอย่าง 210

เทคนิคการเจอร์นัล 464

แผนผังไฟล์

ไดเรกทอรี/export 472

ไดเรกทอรี/usr/share 470

ภาพรวม 465

ระบบไฟล์/(ราก) 467

ระบบไฟล์/usr 469

ระบบไฟล์/var 471

ระบบไฟล์ราก(/) 467

พื้นที่ว่างที่มีอยู่ 475

แฟรกเมนต์ 501

ไฟล์ขนาดใหญ่ 508

ไฟล์ที่กระจัดกระจาย 507

ภาพรวม 464

ภารกิจการจัดการ 474

ระบบไฟล์ซีดีรอม (CDRFS) 499

ระบบไฟล์ที่เจอร์นัล (JFS) 499

ระบบไฟล์ที่เจอร์นัลที่เพิ่มประสิทธิภาพ (JFS2) 499

ระบบไฟล์เน็ตเวิร์ก (NFS) 499

ราก 466

รูปภาพ 513

สำหรับ BSD System Managers 373

สื่อบันทึกออปติคัลการอ่าน/เขียน 479

ไอ-โหนด 501

โสม 466

ระบบไฟล์/(ราก) 467

ระบบไฟล์/opt 466

ระบบไฟล์/proc 466

ระบบไฟล์/var 471

ระบบไฟล์ CDRFS 479

ระบบไฟล์ซีดีรอม (CDRFS) 499

ระบบไฟล์ที่เจอร์นัล (JFS) 499

ระบบไฟล์ที่เจอร์นัลที่เพิ่มประสิทธิภาพ (JFS2) 499

ระบบไฟล์ที่เปิดใช้งาน

พื้นที่ว่าง 508

เรขาคณิตไฟล์ขนาดใหญ่ 508

สร้าง 508

ระบบไฟล์เน็ตเวิร์ก (NFS) 499

ระบบไฟล์ราก 466

ระบบไฟล์ราก(/) 467

- ระบบไฟล์โฮม 466
- ระบบย่อย
 - การปิดการติดตาม 208
 - การเปิดการติดตาม 208
 - การรีเฟรช 207
 - การเริ่มต้น 206
 - การแสดงผลสถานะ 207
 - การหยุด 206
 - คุณสมบัติของ 204
- ระบบหลายผู้ใช้
 - การเปลี่ยนระดับการรันบน 17
- ระเบียบการตั้งชื่อ
 - ไดเรกทอรี 518
 - ไฟล์ 210
- รายการ
 - นิยาม 230
- รายการกำหนดพารามิเตอร์
 - นิยาม 230
- รายการคำสั่ง
 - สำหรับ Korn หรือ POSIX เชลล์ 241
- รายการควบคุมการเข้าถึง 328
- รายการควบคุมการเข้าถึง
 - การคงไว้ 328
 - ตัวอย่าง 334
 - ตัวอย่างของ AIX ACL 331
 - สำหรับอ็อบเจกต์ระบบไฟล์ 328
- รายการคำสั่ง
 - apropos 153
 - cal 151
 - factor 153
 - ของ C เชลล์ 307
 - ของเชลล์เป้าหมาย 287
 - สำหรับ Korn หรือ POSIX เชลล์ 242
- รีโมต
 - เชลล์ 230
 - ล็อกอิน 641
- รีสตาร์ทระบบ 13
- รูปแบบข้อผิดพลาด 688
- เรียกคืน
 - ผลกระทบของแฟรกเมนต์บน 513

ล

- ล็อก JFS 393
- ล็อก JFS2 393
- ล็อกของระบบไฟล์ 393
- ล็อกอิน
 - ID ผู้ใช้ 322
 - การยืนยันข้อความ 643
 - การแสดงผลชื่อ 645
 - ชื่อ 641
 - เชลล์ 230

- ล็อกอิน (ต่อ)
 - ไดเรกทอรี 642
 - ภาพรวม 642
- ล็อกเอาต์
 - ภาพรวม 642
- ลำดับ Ctrl-C 11
- ลำดับความสำคัญของโปรเซส 170
- ลิงก์
 - การลบ 226
 - การสร้าง 225
 - ชนิด 224
 - ภาพรวม 224
 - สัญลักษณ์ 224
 - ฮาร์ด 224
- โลจิคัลพาร์ติชัน
 - การกำหนดขนาดของ 482
- โลจิคัลวอลุ่ม
 - hotspots 451
 - กลยุทธ์สำหรับ 441
 - การเปลี่ยนชื่อ 391
 - การเปลี่ยนดิสก์ 416
 - การเพิ่มระบบไฟล์บนรายการใหม่ 473
 - การย้ายเนื้อหาไปยังระบบอื่น 396
 - ขนาด
 - การตรวจสอบ 473
 - การเพิ่ม 473
 - การลด 473
 - ซีดจำกัด 418
 - คัดลอกไปยังฟิสิคัลวอลุ่มอื่น 392
 - ติบ
 - กำหนด 433
 - นโยบายตรวจสอบการเขียน 450
 - นโยบายกลุ่มวอลุ่ม 453
 - นิยาม 428
 - แบบแถบ 450
 - ไฟล์แม็พ 449
- โลจิคัลวอลุ่มแบบติบ
 - กำหนด 433

ว

- เวิร์กสเตชันแบบไม่มีดิสก์
 - ความปลอดภัยในการติดตั้ง 496

ส

- สตริง
 - การค้นหาในไฟล์ข้อความ 218
- สตาร์ทอัพ
 - การควบคุมหน้าต่างและแอ็พพลิเคชัน 347
- สถานะการออก
 - Korn หรือ POSIX เชลล์ 250

สนับสนุนไดรฟ์ Blu-ray แบบ USB 632

สภาวะแวดล้อม

การแสดงผลปัจจุบัน 341

ค่าติดตั้ง 344

ไฟล์ 344

ระบบ 336

สภาวะแวดล้อมเซลล์

การกำหนดเอง 57

สภาวะแวดล้อมผู้ใช้

การกำหนดเอง 57

สภาวะแวดล้อมของระบบ 59

การยกเลิกการจัดสรรตัวประมวลผลแบบไดนามิก 59, 61

ข้อความของวัน 70

เซอวิวิการจัดการข้อมูลเวลา 58

โปรไฟล์ 57

โหมด 64-บิต 58

สมนาม

r 147

การลบ 281

การสร้าง 281

การแสดงผลรายการ 281

การเอ็กซ์พอร์ต 281

ที่ติดตาม 282

ไม่สนับสนุน 281

สมนาม r 147

สมนามที่ติดตาม 282

ส่วนขยายเคอร์เนล AIX Event Infrastructure 674

สัญญาณ

SIGINT 269

SIGQUIT 269

สัญญาณ SIGINT 269

สัญญาณ SIGQUIT 269

สายเคเบิล

การตรวจสอบการเชื่อมต่อ 614

สำเนาสำรอง 44, 45

กลยุทธ์สำหรับการจัดการ

การวางแผน 27

แนวทางสำหรับ 24

กลุ่มวอลุ่มที่ผู้ใช้กำหนด 45

การใช้คำสั่ง smit 55

การทำซ้ำระบบ (การลอกแบบ) 28

การทำตามที่จัดตารางเวลาอย่างสม่ำเสมอ 48

การนำไปใช้โดยใช้สคริปต์ 48

การบีบอัดไฟล์ 41

การบีบอัดไฟล์ก่อน 41

การเรียกคืนไฟล์ 34

การเรียกคืนข้อมูล 30

คำสั่ง, รายการของ 23

นโยบาย 24

ผลกระทบของแฟร็กเมนต์บน 513

ผู้จัดการระบบ BSD 360

โพรซีเจอร์สำหรับข้อมูลระบบและผู้ใช้ 27

โพรซีเจอร์สำหรับระบบไฟล์ผู้ใช้ 29

สำเนาสำรอง (ต่อ)

ไฟล์ 23

ไฟล์ผู้ใช้ 29

ภาพรวม 23

เมธอด 23

ระบบไฟล์ผู้ใช้ 29

สื่อบันทึก 26

สิ่งอำนวยความสะดวกโปรเซสรวม

Korn หรือ POSIX เซลล์ 257

สิทธิอนุญาต

ไดเรกทอรี 327

ไฟล์ 327

สื่อบันทึกหน่วยเก็บ 23

สื่อบันทึกออปติคัล

การใช้ระบบไฟล์ในการอ่าน/เขียน 479

ห

หน่วยเก็บ โลจิคัลวอลุ่ม

นโยบายการจัดสรร intra-disk 448

องค์ประกอบ 385

หน่วยเก็บ โลจิคัลวอลุ่ม

กลุ่มวอลุ่ม 427

กลุ่มวอลุ่มที่ไม่ใช่องค์ประกอบ 387

ขนาดสูงสุด 429

ดิสก์โอเวอร์ไฟล์ 489

นโยบายการกำหนดตารางเวลาการเขียน 443, 444

นโยบายการจัดสรรของดิสก์อินเตอร์ 445

นิยาม 426

พาร์ติชันโลจิคัล 429

ฟิลิคัลวอลุ่ม 426

ระบบไฟล์ 429

โลจิคัลวอลุ่ม 428

หน้าจอ

การคัดลอกจอแสดงผลไปยังไฟล์ 380

การคัดลอกไปยังไฟล์ 381

การล้าง 380

การแสดงความในตัวของอักขรขนาดใหญ่ 382

การแสดงความในหน้าจอในแต่ละครั้ง 217

หมายเลข PID 154

หมายเลขตัวแปรของโอ-โหนด 504

และแฟร็กเมนต์ 501

หมายเลขประจำตัวโปรเซส 154

หมายเลขอ้างอิงโหนดดัชนี 516

หมายเลขโอ-โหนด 209, 224, 516

เหตุการณ์คัลสเตอร์ 707

เหตุการณ์ที่เกิดขึ้นซึ่งไม่พร้อมใช้งาน

นิยาม 683

โหมด 64-บิต

ชุดไฟล์ 58

โหมดการเข้าถึง

ไฟล์ 324

โหมต อินพุต

นิยาม 272

โหมตการเข้าถึง

การควบคุม 324

การแสดงผลข้อมูลกลุ่ม 325

การแสดงผลแทนโดยใช้ตัวเลขของ 325

การแสดงผลแทนโดยใช้สัญลักษณ์ของ 324

คลาสผู้ใช้ 324

ไต่เรียกทอริ 324

โหมตควบคุม 272

โหมตผู้ใช้เดี่ยว 56

โหมตอินพุต

คำสั่ง input edit 272

อ

องค์ประกอบ

กลุ่มวอลุ่มที่ไม่ใช่องค์ประกอบ 387

การเปลี่ยนเป็นสถานะไม่ใช่องค์ประกอบ 387

นิยาม 385

อ็อปติคัลไดร์ฟ

การกำหนดคอนฟิก 422

อ็อพชัน

ในคำสั่ง 142

อักขระ

การใส่ไว้ในเครื่องหมายัญประกาศใน Korn หรือ POSIX เซลล์ 243

อักขระ wildcard 211

เครื่องหมายคำถาม 211

ดอกจัน 211

นิยาม 230

อาร์กิวเมนต์

ในคำสั่ง 143

อินเตอร์เฟซการเขียนโปรแกรมแอปพลิเคชัน

Workload Manager (API) 569

อินพุต

การเปลี่ยนทิศทาง 375

ตัวดำเนินการเปลี่ยนทิศทาง 377

อินพุตมาตรฐาน 375

อินพุตมาตรฐาน

การคัดลอกไปยังไฟล์ 380

การเปลี่ยนทิศทาง 377

อิมเมจเปิดเครื่องใหม่

การสร้าง 14

อุปกรณ์ 423

MPIO

การเดินทางเคเบิล 594

การกำหนดคอนฟิกจำนวนมาก 423

การกำหนดคอนฟิกอ็อปติคัลไดร์ฟการอ่าน/เขียน 422

การกำหนดใหม่ 612

การตรวจสอบการเชื่อมต่อ 614

การตรวจสอบซอฟต์แวร์ 612

การตรวจสอบสถานะของ 612

อุปกรณ์ (ต่อ)

การตรวจสอบสภาพความพร้อม 616

การตรวจสอบแอ็ดทริบิวต์ 612

การติดตั้ง 422

การเปลี่ยนแอ็ดทริบิวต์ 612

การรันการวิเคราะห์ 616

การแสดงผลข้อมูลเกี่ยวกับ 337

คลาส 579

โค้ดที่ตั้ง 580

ที่มีความสามารถ MPIO 593

สถานะ 580

สำหรับ BSD System Managers 372

โหนด 578

อุปกรณ์ IDE

การติดตั้ง 574

ฐานข้อมูลคอนฟิกูเรชันแบบกำหนดเอง 578

ตัวควบคุมของเทปไดร์ฟ 575

แอ็ดเตรสของเทปไดร์ฟ 575

แอ็ดทริบิวต์แบบกำหนดเอง 578

อุปกรณ์ SCSI

โค้ดที่ตั้ง 582

เอกสาร here 255, 378

เอกสารออนไลน์อินพุต 378

เอาต์พุต

การทิ้งด้วยไฟล์ /dev/null 377

การเปลี่ยนทิศทาง 375

การเปลี่ยนทิศทางไปยังไฟล์ 376

ตัวดำเนินการเปลี่ยนทิศทาง 376

เอาต์พุตการวินิจฉัย 375

เอาต์พุตข้อผิดพลาด 375

เอาต์พุตข้อผิดพลาดมาตรฐาน

การเปลี่ยนทิศทาง 378

เอาต์พุตมาตรฐาน 375

การเปลี่ยนทิศทาง 376

การผนวกเข้ากับไฟล์ 376

แอ็ดทริบิวต์

สนับสนุนโดย Korn หรือ POSIX เซลล์ 251

ไอ-โหนด 504

จำนวนไบนารีต่อ (NBPI)

การระบุ 504

และแฟร็กเมนต์ 501

หมายเลขตัวแปรของ 504

ไอ-โหนด, จำนวนของ 506

อี

ฮาร์ดดิสก์ 398



พิมพ์ในสหรัฐอเมริกา