z/OS

What you need to know about z/OS UNIX Shell Commands

NY Metro NaSPA June 17, 2008

Garth Godfrey

z/OS UNIX System Services Development

IBM Poughkeepsie, NY

ggodfrey@us.ibm.com

http://www.ibm.com/servers/eserver/zseries/zos/unix

http://www.ibm.com/servers/eserver/zseries/zos/unix/port_tools.html

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM*
IBM eServer
IBM e(logo)server*
IBM logo*
Language Environment*
MVS
On demand business logo
OS/390*
Parallel Sysplex*
RACF*
System z9
z/Architecture
z/OS*
zSeries*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Session Objectives

To improve your knowledge of:

- UNIX shell basics
- z/OS differences
- z/OS UNIX security facilities
- System set up and management

"Unix was not designed to stop people from doing stupid things, because that would also stop them from doing clever things." -- Doug Gwyn

z/OS UNIX

Topics
UNIX shell basics
Working with files & directories
Shell features (/bin/sh)
Security
System Setup & Management

Customizing your System through Profiles

IPL → /etc/rc

System-wide setup
Start daemons

```
_BPX_JOBNAME='CRON' /usr/sbin/cron
```

user login to sh — telnet, rlogin, OMVS, ssh

/etc/profile

***Common setup** for **every** sh user*
Set environment variables

```
export TZ=EST5EDT
export LANG=C
export LC_ALL=C
export PATH=/bin
```

$HOME/.profile

*Your **personal** setup*
Values override /etc/profile

```
export ENV=$HOME/.setup
export PS1='${PWD}> '
export PATH=$PATH:$HOME
```

$ENV

*Your **personal** setup*
*Run **every** shell*
Define shell options, aliases, functions

```
set -o vi
alias lsl="ls -l"
```

Shell Shortcuts: Wildcards and Variables

- **Wildcards** (filename expansion, globbing)
  - Matches filenames anywhere on the command line
  - * 0 or more characters (except a leading dot) `ls ab*`
  - ? any single character `ls ab?`
  - [ ] any characters within the brackets `ls *.[ch]`

    matches files that end with .c or .h
- **Shell Variables**
  - VAR=value *sets a shell variable*
  - export VAR *marks VAR for **export** to commands / scripts*
  - export VAR=value *marks VAR for **export** and sets the value*
    - *An exported variable is also called an **environment variable***
  - VAR=value *command* ***sets** a shell variable for the duration of command*
  - Variable name can contain only alphanumeric and _
    - Case-sensitive
  - value can contain any characters

Displaying Shell Variables

- **$VAR** *expands to value*
  - echo $HOME — *displays value of HOME*
  - print $HOME — *displays value of HOME*
  - PATH=$PATH:/u/godfrey/bin — *adds onto end of PATH*
  - PS1='<AQ>$PWD ==> ' — *sets prompt to* <AQ>$PWD ==> $PWD *is expanded before each command prompt*
  - Example:

```
<AQ>/u/godfrey/tst ==> echo $HOME
/u/godfrey
```

- **set** *without arguments – displays **all** variables in the shell environment*
- **env** *without arguments – displays **environment** variables*

Quoting in the shell

- **Single quotes**
  - No shell expansions

```
==> echo '$PWD'
$PWD
```

- **Double quotes**
  - Expand variables
  - Perform command substitution
  - Perform arithmetic substitutions
  - NO filename expansion (wildcard), tilde substitution, ...

```
==> echo $PATH
/bin
==> PATH="$PATH:$HOME/bin"
==> echo $PATH
/bin:/u/godfrey/bin
```

- **Backslash**
  - Escape next character from special meaning

```
==> echo \$PATH
$PATH
```

Shell Commands: Getting dangerous

- **rm** *remove files or directories*
  - rm junk — *removes file* junk
  - **Beware!** rm * — *removes* ***all files*** *in current directory*
  - **Beware! Beware!** rm -r * — *removes* ***entire directory tree***

- **cp** *copy files or directories*
  - cp *source_file target_file* — *creates or overlays target_file*
  - cp *.[ch] $OLDPWD — *copies all files ending in .c or .h to previous working directory*
  - cp myfile.c "//'ctware.c(myfile)' " — *copies file myfile.c to PDS*

Shell Commands: Searching files

- **find** *find files in **directory tree** matching criteria*
  - find . -name "*.c" *starting with cwd, find filenames ending with .c*
  - find /tmp/ -type d -user ctware *starting with /tmp, find directories owned by user 'ctware'*
- **grep** *search file contents for strings or regular expressions*
  - grep "Hello world" *.c *search all .c files for the string "Hello world"*
- **man** *display command **man**ual pages*
  - man find *displays manual for 'find' shell command*

Getting system info

- **`uname -aI`** *Displays information about the operating system*

```
z/OS      AQFT      08.00      01              2084
```

Op Sys *System* *Release* *Version* *Machine type*

Note: on z/OS, **`uname -a`** *will still report OS/390 by default*
`uname -aI` *will report z/OS values*

- **`who -a`** *list IPL time, release, all users*

```
Name       ST Line                Time          Idle     PID Hostname / Exit
   .          system boot          Jan 27 02:35
   .          run-level 03.18.00   Jan 27 02:35
BARRYL      + ttyp0000             Jan 27 20:52    .   16842884 (wecm-9-67-122-45.wecm.ibm.com)
ERINF       + ttyp0001             Jan 28 17:44    .   16842973 (wecm-9-67-92-242.wecm.ibm.com)
...
```

Getting process info

- **ps** *process status*
  - ps -ef *display all processes you are authorized to see*
  - *Only superusers can see all processes*

```
<AQ>/u/godfrey ==> ps -ef
     UID        PID        PPID  C    STIME TTY        TIME CMD
 GODFREY   16843107       66262  - 22:12:28 ?         0:00 /usr/sbin/sshd -
   f /etc/ssh/sshd_config
 GODFREY      66212    16843107  - 22:12:28 ttyp0016  0:00 sh -L
 GODFREY   33620720       66212  - 22:32:20 ttyp0016  0:00 ps -ef
```

  - *Many more options for* ps

z/OS UNIX

Topics
UNIX shell basics
Working with files & directories
Shell features (/bin/sh)
Security
System Setup & Management

File permissions (mode)

```
==> ls -l
-rw-r--r--   1 GODFREY  SHUT          127 Feb 15 12:03 c_template.c
drwxr-xr-x   2 GODFREY  SHUT         8192 Feb 15 12:03 data
-rw-r--r--   1 GODFREY  SHUT          218 Feb 15 12:03 getUTCtime.c
```

File type	User owner of file	Group owner of file	Other everyone else
d	rwx read/write/execute	r-x read/execute	r-x read/execute
	7	5	5

-	regular file
d	directory
l	symbolic link
e	external link
c	character special
	... and more

- Permissions are stored in the file system
- Access checking is done by the Security Product
  - **If effective UID = owner, check User permission**
  - **If effective GID or supplemental = group owner, check Group permission**
  - **Else check Other permission**

File Security

What permission bits mean for files and directories

Files		Directories
Read contents of file	Read	Read directory
Change contents of file	Write	Change, add, delete, entries in directory
Execute file	Execute	**Search** a directory

File Security – sticky bit

Files		Directories
Read contents of file	Read	Read directory
Change contents of file	Write	Change, add, delete, entries in directory
Execute file	Execute	Search a directory
Search for the program in **MVS** search order	**Sticky**	Files deleted / renamed only by owner (or superuser)

```
chmod +t /tmp/          set sticky bit on /tmp/ dir
ls -ld /tmp/            display /tmp/ dir permissions
drwxrwxrwt 23 SUFID  DEFLT1 204800 Jul 17 12:55 /tmp/
```

Listing extended attributes

- ls -E /bin/su *list* ***extended attributes*** *of* /bin/su

```
-rwxr-xr-x  -p--  2 CLASGEN  TASKS  81920 Dec 27 1999 /bin/su
```

program controlled

not allowed to run in a shared address space

Extended attributes

a	APF authorized
p	Program controlled
s	Shared AS allowed
l	System-shared library

Lots more options on ls

Setting file attributes

- **chmod** — *change mode (permission bits)*
  - chmod 777 *file* — *sets all permission bits on (anyone can read/write/execute)*
  - chmod u+x *file* — *sets user (owner) executable permission bit on*
  - chmod g+rw *file* — *sets group read / write permission bits*
  - chmod o-w *file1 file2* — *sets other write permission off on 2 files*
  - chmod +t *myprog* — *sets sticky bit on myprog*
- **extattr** — *change extended attributes*
  - extattr +p /usr/bin/trustme — *sets "**program control**" on*
    - like RDEFINE to PROGRAM class
    - requires BPX.FILEATTR.PROGCTL
  - extattr -s /bin/passwd — *sets "**share AS**" off*
    - spawn program in new address space

Editing ASCII files

- **Tag the file:** `chtag -tc ISO8859-1 ascfile`
  - Text file, containing chars encoded in ISO8859-1 (ASCII Latin 1)
  - Use chtag -p or ls -T to display file tags
- **Enable autoconversion**
  - shell (telnet, rlogin, ssh) environment

    ```
    export _BPXK_AUTOCVT=ON
    vi ascfile
    ```

    user controlled env var
  - 3270 environment

    ```
    BPXPRMxx parmlib
                  AUTOCVT(ON)
    oedit ascfile
    ```

    system-wide setting

    or ISPF edit in z/OS 1.9
- **The file will be stored as ASCII**

Editing ASCII files in ISPF *z/OS 1.9*

- **ISPF edit / browse support of z/OS UNIX files**
  - Entry panel

```
Other Partitioned, Sequential or VSAM Data Set, or z/OS UNIX file:
   Name . . . . . /u/godfrey/tst/ascii/helloA
   Volume Serial             (If not cataloged)
```

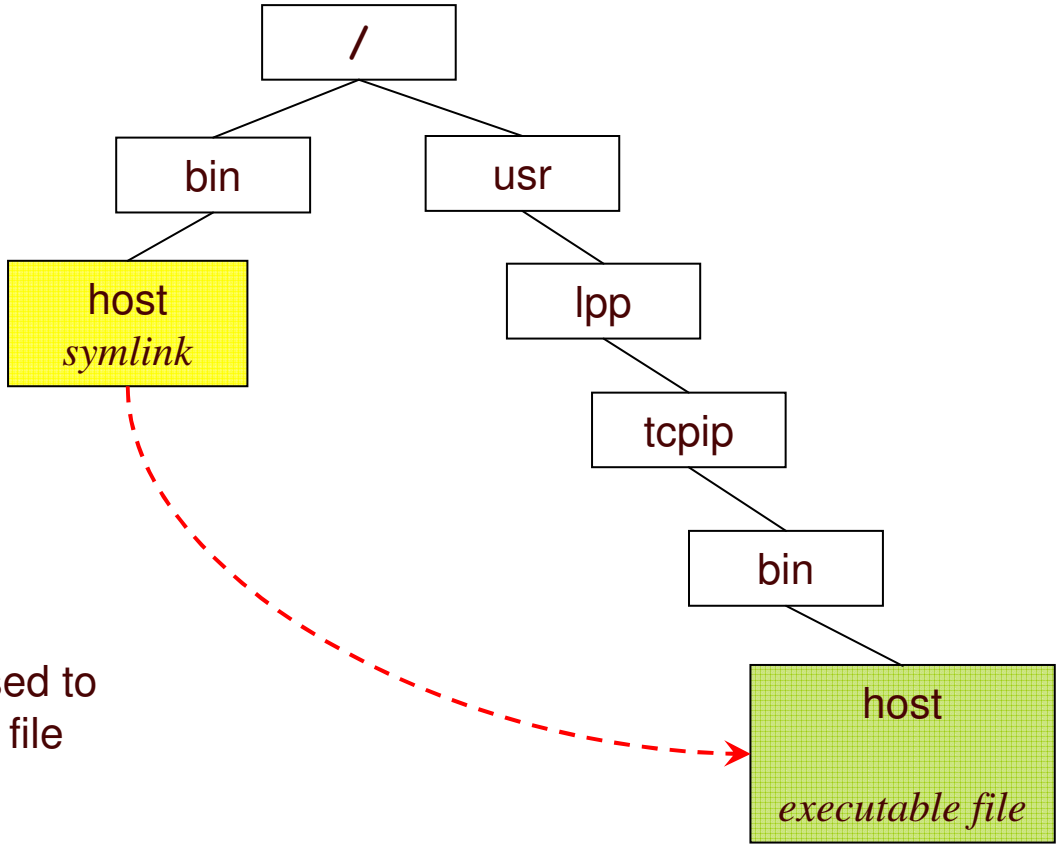
  - Specify z/OS UNIX pathname on
    COMPARE, COPY, CREATE, MOVE, REPLACE
  - **ASCII** file editting - APAR OA22284
    - EA command in 3.17
    - Automatically recognizes file tagged with ccsid=819
    - Allows character insertion
    - Replaces SOURCE ASCII and LF
  - **The file will be stored as ASCII**

Symbolic Link (symlink or soft link)

`ln -s` *old new*

- Establish an alternate path name for a **file**

`ln -s /usr/lpp/tcpip/bin/host /bin/host` *creates file symlink*



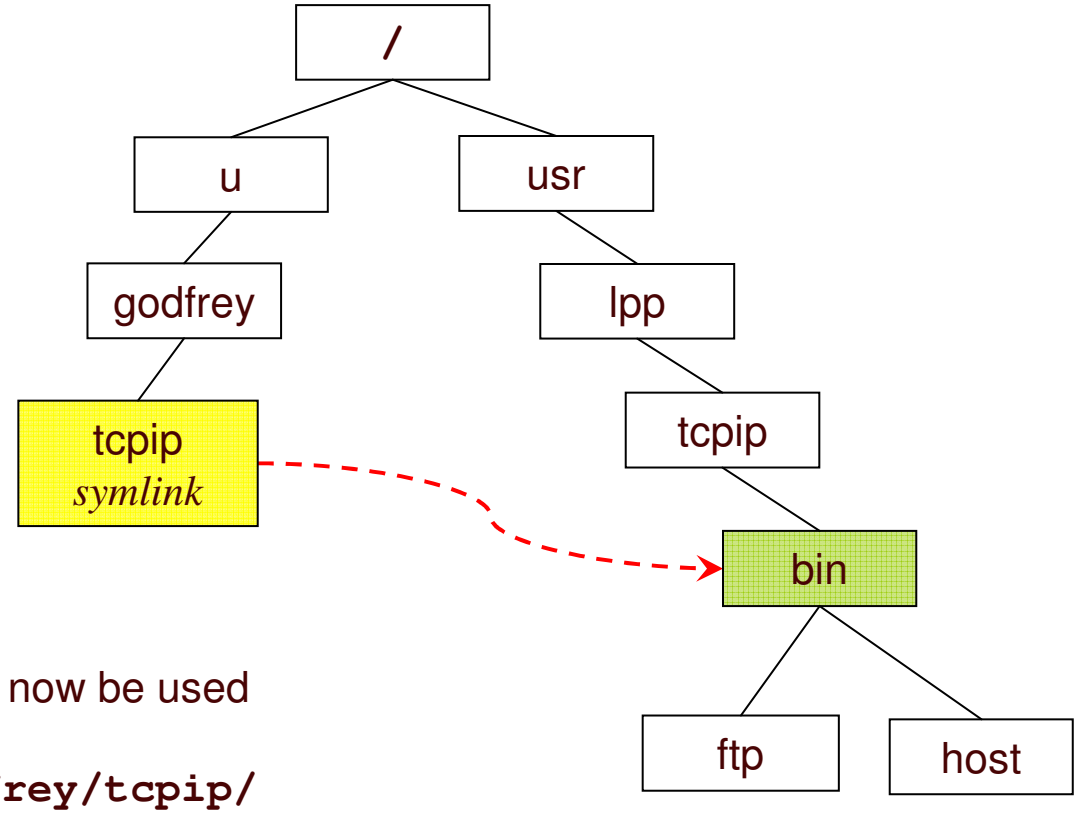
/bin/host can now be used to refer to the executable file

Symbolic Link (symlink or soft link)

`ln -s` *old new*

- Establish an alternate path name for a **directory**

`ln -s /usr/lpp/tcpip/bin/ /u/godfrey/tcpip` *creates directory symlink*



/u/godfrey/tcpip/ can now be used

e.g. `ls -l /u/godfrey/tcpip/`

Packaging directory trees

- **pax** *package files or directories into portable archives*
  - pax -wzf /tmp/my_dir.pax **.** — ***create*** *a compressed archive containing working directory tree*
  - pax -wzf "//'posix.godfrey.tar(mydir)'" **.** — *write archive to PDS member*
  - pax -vf import.pax — ***list*** *attributes of files in archive*
  - pax -rf from_AIX.pax -o from=ISO8859-1, to=IBM-1047 "*.c" — ***extract*** *the .c files from an archive translating from ASCII to EBCDIC*
  - pax -rwpe **.** newdir — ***copy*** *working dir tree to newdir, preserving file attributes*
- Hints:
  - cd to directory and use relative names
  - do **not** write archive into the same directory
  - 1.8 and later: Use **-x pax** to save all attributes

Getting File System info

- **df** *display filesystem status for **all mounted filesystems***
- **df -Pvk .** *display filesystem status for filesystem containing cwd*

```
Filesystem          1024-blocks        Used  Available  Capacity Mounted on
OMVS.HFS.GODFREY         720000      626960      91916       88% /u/godfrey
HFS, Read/Write, Device:16544, ACLS=Y
File System Owner : AQFT        Automove=Y      Client=N
Filetag : T=off   codeset=0
```

- **mount** *__mount__ a file system or __list__ mountpoints*
  - **/usr/sbin/mount -q /u** *list mountpoints under /u*
- **fuser** *display PID's with open files*
  - **fuser -cu /usr/lpp/dfs** *list processes and users with open files in filesystem*

skulker

- **`skulker [-irw] [-l`** ***logfile*****`]`** ***directory days_old***

- removes files in the directory older than the specified number of days

- shell script in /**samples**
  - copy to /bin/skulker or /usr/sbin/skulker
  - may be modified by installation
  - **Protect it from hackers!** (make it non-writable)
    as it is usually run with superuser authority
  - APAR **OA16107** for z/OS 1.6 - 1.8

- *e.g.* skulker /tmp/ 100
  - deletes files in /tmp older than 100 days
  - trailing slash follows a /tmp symlink to another directory

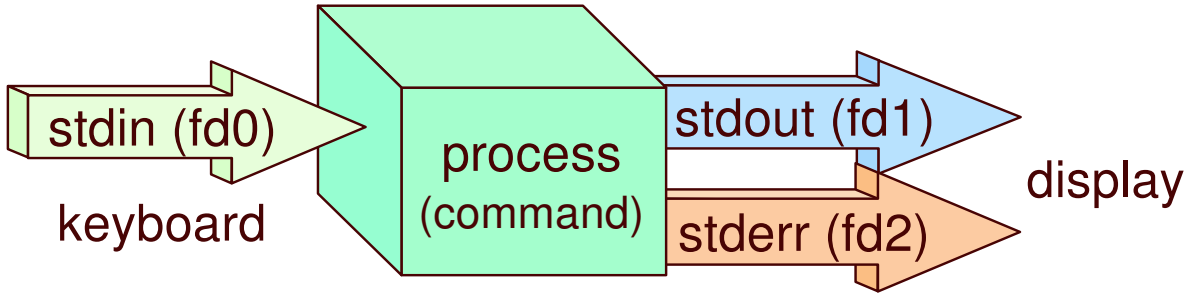
- use **cron** to schedule it to run regularly

z/OS UNIX

Topics
UNIX shell basics
Working with files & directories
Shell features (/bin/sh)
Security
System Setup & Management

stdin / stdout / stderr

- **Process** is the environment of a running program including:
  - program image
  - open files
  - userid
  - data
  - current directory



- **Command** runs in a process
- **Redirect** stdin/stdout/stderr
  - files
  - pipelined commands

Redirection

- Changes a command's standard input / output to a file

> **Redirect** standard **output** to a file
(creates a new file or overlays an existing file)

>> **Append** standard output to a file
(creates the file if it does not exist)

< **Redirect** standard **input** from a file

```
cat file1 file2 >file3
```
create (or overlay) file3

```
date >>logfile
```
appends time stamp to logfile

```
find /u/devan 2>/dev/null
```
discards error messages
redirects file descriptor 2 to the "bit bucket"

```
sort <list.txt >sorted_list.txt
```
sorts list.txt *and writes output to* sorted_list.txt

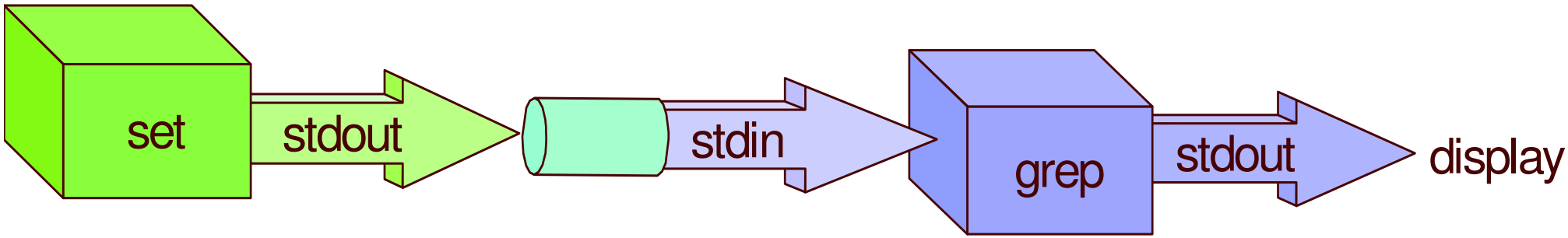
Pipelined commands

- Pipes direct the **stdout** of one command into the **stdin** of the next command

set | grep PATH

set *displays all variables,*

grep *displays lines containing the string* "PATH"




```
<AQ>/u/godfrey ==> set | grep PATH
LIBPATH="/lib:/usr/lib:.:/bin"
MANPATH="/usr/man/%L"
NLSPATH="/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/%L/%N.cat"
PATH="/u/godfrey/relL/maint/bin:/bin:/usr/lpp/java/J2/J1.3/bin"
```

Using Pipes

- `ls -ltr | tail`
  - *List the 10 most recently modified files*
- `man pax | head -n30`
  - *Show the first 30 lines of the pax manual page*
- `find . -type f -print | wc  -l`
  - *Displays the total number of files in the current working directory and all of its subdirectories.*

Command substitution

- Substitutes stdout from one command into another command
- Backquotes *old style*

```
==> echo There are `ls | wc -w` files in this directory.
There are 99 files in this directory.
```

- $( ) *new style*

```
==> echo User `whoami` owns files under cwd:  $(find . -user $(whoami))
User GODFREY owns files under cwd: .  ./date ./dirjunk ./dirjunk/junk
```

Diagnosing problems

- $? *Exit status of **last** command*

  0 = success

  >0 = failure

  126 = not executable

  127 = not found

  >127 = terminated by signal

  - **`echo $?`** *to display **exit status***

  - **`kill -l nnn`** *to report the **terminating signal** if >127*

```
==> kill -l 137
KILL
```

z/OS UNIX

Topics
UNIX shell basics
Working with files & directories
Shell features (/bin/sh)
Security
System Setup & Management

Changing identity

- **su** *switch userid, starting a child shell*
  - su user2 *Prompts for password*
    Changes UNIX and MVS identity
  - su *Switch to **UID 0** if permitted to the **BPX.SUPERUSER** FACILITY class profile*
    Changes UNIX identity
    Maintains invoking user's shell environment

garth

```
> su frank
Enter the password ...
> id
uid=922(GARTH) gid=...
```

same cwd
same env vars

```
> id
uid=42(FRANK) gid=...
```

- **id** *show current identity and groups*

```
uid=922(GARTH) gid=2821(SHUT) groups=0(TASKS),16(MHV)
```

  - id diane *show diane's uid/gid/groups*

Superuser Granularity

- To minimize the users with BPX.SUPERUSER . . . or UID 0
- **UNIXPRIV** class Resource Names Supported in RACF:
  - CHOWN.UNRESTRICTED
  - FILE.GROUPOWNER.SETGID
  - RESTRICTED.FILESYS.ACCESS
  - SHARED.IDS
  - SUPERUSER.FILESYS.ACLOVERRIDE
  - SUPERUSER.FILESYS
  - SUPERUSER.FILESYS.CHANGEPERMS
  - SUPERUSER.FILESYS.CHOWN
  - SUPERUSER.FILESYS.MOUNT
  - SUPERUSER.FILESYS.QUIESCE
  - SUPERUSER.FILESYS.PFSCTL
  - SUPERUSER.FILESYS.VREGISTER
  - SUPERUSER.IPC.RMID
  - SUPERUSER.PROCESS.GETPSENT
  - SUPERUSER.PROCESS.KILL
  - SUPERUSER.PROCESS.PTRACE
  - SUPERUSER.SETPRIORITY

Access Control Lists

- UNIX files are protected with POSIX permission bits

User			Group			Other		
read	**w**rite	e**x**ecute	**r**ead	**w**rite	e**x**ecute	**r**ead	**w**rite	e**x**ecute

- Can only specify permissions for file owner (user), group owner, and everybody else

- **Access Control Lists** permit/restrict access to specific users and groups

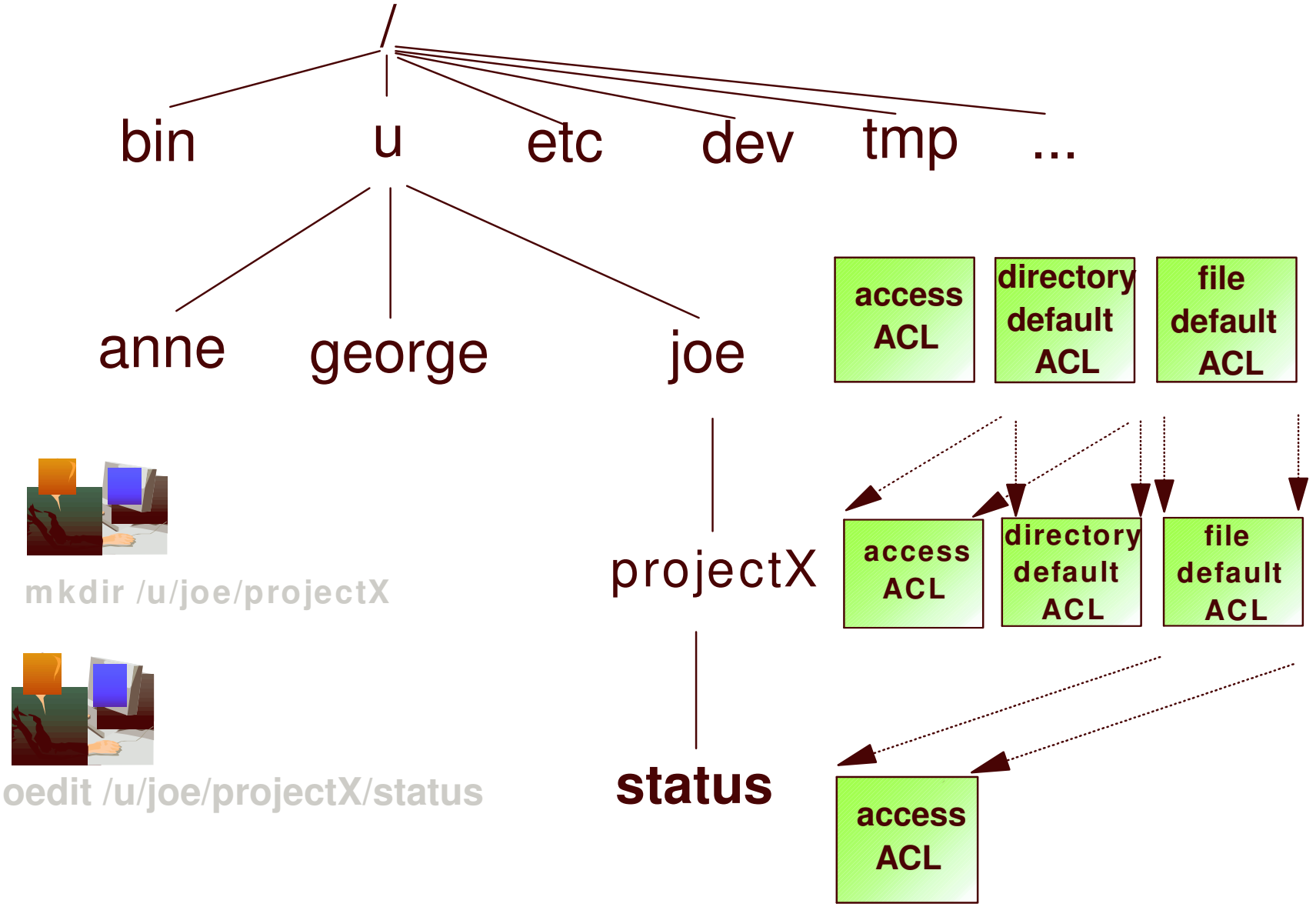
Access Control Lists (ACLs) Overview

- Traditional UNIX approach
- Contained within the file system
  - File security is portable
  - Deleted automatically if the file is removed
- Not protected by RACF profiles
- Managed using UNIX shell commands, or ISHELL
- Supports inheritance for new files and subdirectories

ACL Inheritance

- Can establish default (or 'model') ACLs on a directory
- They will get automatically applied to new files/directories created within the directory
- Separate default ACL used for files and (sub)directories
- Can reduce administrative overhead

ACL Inheritance example



New terms

- **base ACL entries** = permission bits
  - user::*rwx*
  - group::*rwx*
  - other::*rwx*

- **extended ACL entries**
  - user:***uid***:*rwx*
  - group:***gid***:*rwx*

  uid = userid or UID

  gid = group or GID

  - default:user:***uid***:*rwx*
  - default:group:***gid***:*rwx*

  - fdefault:user:***uid***:*rwx*
  - fdefault:group:***gid***:*rwx*

setfacl command

- setfacl -s ***entries*** [***path*** ...]
  - set (replace) entire ACL
  - must include base ACL entries (permission bits)
- setfacl -S ***file*** [***path*** ...]
  - set (replace) entire ACL from file
  - must include base ACL entries (permission bits)
- setfacl -D ***type*** ... [***path*** ...]
  - delete extended ACL entries of matching type
- setfacl -m|M|x|X ***EntryOrFile*** [***path*** ...]
  - modify or delete extended ACL entries

setfacl command . . .

- An ACL can be set from contents of a file
  - **`setfacl -S ~/acls/ateam  rel4dir`**

    where ~/acls/ateam contains an entire ACL (e.g.):

    u::rwx
    g::r-x
    o::---
    g:shut:rwx
    g:testers:r-x

- Allows use of "named ACLs"

- An ACL can be set from stdin, and thus piped in from a getfacl command
  - **`getfacl YourFile | setfacl -S - MyFile`**

getfacl Display ACL contents

- getfacl MyFile
  - Displays file name, user owner, and group owner
  - Displays base POSIX permissions in "ACL format"
  - Displays access ACL entries

```
#file:  MyFile
#owner: BRUCE
#group: RACFDEV
user::rwx
group::r--
other::r--
user:GARTH:rwx
group:RACFDEV:r-x
```

find Command substitution

- Useful in command substitution
  - Permit group ALPHA to search every directory under /u/godfrey/tools

```
setfacl -m g:ALPHA:r-x $(find /u/godfrey/tools -type d)
```

  - Remove user TED from all ACL entries

```
setfacl -qx u:TED,d:u:TED,f:u:TED $(find / -acl_user TED)
```

  - Add the group ALPHA to every access list in /u/shr/ which contains an entry for UNIXGRP:

```
setfacl -m  g:ALPHA:rwx $(find /u/shr -acl_entry UNIXGRP)
```

RACF Access Checking with ACLs

- Takes into account base POSIX permissions and access ACLs

- ACLs only used if the **FSSEC** class is active
  - **SETROPTS CLASSACT(FSSEC)**
    will activate use of ACLs in Unix file authority checks

  - Make sure that FSSEC is **not active** until you are ready to use ACLs
    - The class need not be active to create ACLs

- **setfacl** can be used to create ACLs at any time

z/OS UNIX

Topics
UNIX shell basics
Working with files & directories
Shell features (/bin/sh)
Security
System Setup & Management

cron daemon

- Clock daemon
- Runs user commands at specified dates / times
- Runs cron jobs with user's authority
- Regularly scheduled: **crontab**

  Run once: **at**
  batch

cron daemon set up

If you mount the root file system R/O

- Set up an **/var/spool** directory (or **/etc/spool**)
  - Consider a separate mounted file system
  - Permissions 755
  - Copy existing **/usr/spool** to **/var/spool**
  - Create symbolic link **/usr/spool** pointing to **/var/spool**
- Set up an **/etc/cron** directory
  - Permissions 755
  - Copy existing **/usr/lib/cron** to **/etc/cron**
  - Create symbolic link **/usr/lib/cron** pointing to **/etc/cron**
- *The cron program creates and writes to filenames in* **/usr/spool** *and* **/usr/lib/cron**, *which are redirected to a* **R/W** *file system*

cron set up: queuedefs

/usr/lib/cron/queuedefs (or /etc/cron/queuedefs)

- Defines the cron queues
- cp /samples/queuedefs /usr/lib/cron/queuedefs
- Default queues **a, b, c** for "at" jobs, "batch" jobs, "crontab" jobs

```
c.5j2n15w
```

- **c** **queue name**
- **5** **jobs running simultaneously**
  (*increase this for realistic workloads*)
- **2** **nice value - *mapped to* BPXPRMxx PRIORITYGOAL *or* PRIORITYPG**
- **15** **if 5 jobs already running, wait 15 seconds before retry**

cron set up: allow / deny

- Who may use the **at** command?
  - /usr/lib/cron/**at.allow** *list of users allowed*
  - /usr/lib/cron/**at.deny** *list of users who are **not allowed***
  - To allow **all users**
    - create an **empty at.deny** file, and no at.allow

- Who may use the **crontab** command?
  - /usr/lib/cron/**cron.allow** *list of users allowed*
  - /usr/lib/cron/**cron.deny** *list of users who are **not allowed***
  - To allow **all users**
    - create an **empty cron.deny** file, and no cron.allow

Starting the cron daemon

- Start by user with authority to setuid
  - UID 0
  - READ authority to FACILITY class profile **BPX.DAEMON**
- Typically called from /etc/rc or /etc/inittab (z/OS 1.8)
  - or by a started procedure
- Only one cron daemon can be running on a system
- **TZ** environment variable should be set
  - **cron** uses this **time zone** when matching **crontab** entries
  - **at** jobs use the TZ of the user

Managing the cron log

- /usr/spool/cron/log
  - Contains a history of cron jobs run
    - both successful and failing
  - Must be cleaned up periodically

```
LOG=/usr/spool/cron/log
cp $LOG /bkup/cronlog.$(date +%m%d%Y.%T) &&   >$LOG
```

  - copies the cron log to new file: cronlog.06012008.18:02:16
  - if successful, empties the cron log
  - Using command substitution, "AND" operator, redirection

Scheduling cron jobs

- **at** *run a command at specified time*
  - `at -f bigcopy.sh 23:00` *run bigcopy.sh script at 11:00 pm*
  - `at midnight Friday`
    `/u/admin/build >/u/admin/log 2>/u/admin/errlog`
    `^D` *end keyboard input of commands*
  - `at -l` *list user's at jobs*

- **crontab** *schedule regular jobs*
  - `crontab mycronjobs` *mycronjobs contains entries*

min hour day-of-month month day-of-week command-string

```
0 0 * * *   /u/admin/daily_bup  >>/etc/bup_log 2>>&1   #midnight daily
30 1 * * 6  /u/admin/weekly_bup >>/etc/bup_log 2>>&1   #1:30 every Sat
```

  - `crontab -l` *list user's crontab jobs*

cron hints

- **crontab** jobs run with
  - clean environment
  - user's HOME, LOGNAME (MVS identity, even if shared UID)
  - PATH=/bin *use full pathnames or set PATH*
  - SHELL=/bin/sh
- at jobs inherit (most of) user's environment
- Do **not** change or put other files in the spool directory
  - /usr/spool/cron/crontabs/ *username file created by crontab*
  - /usr/spool/cron/atjobs/ *nnnnnn.a file created by at*

BPXPRMxx Limit Management

- Monitor and manage Unix System Services parmlib values through operator messages and commands

- Console messages are issued
  - as the usage reaches 85%, 90%, 95% and 100% of the current limit
  - as the usage decreases and when it drops below 85%

Managing BPXPRMxx Parmlib Values

- Display command options
  - **D OMVS,L** to display the **system** wide parmlib limits
  - **D OMVS,L,PID=*nnnnnnnn*** to display the specific limits for a **process**
  - **D OMVS,PFS** to display the high water mark for each **sockets** PFS

- commands to set the limit values

SETOMVS / SET OMVS
- the parmlib values take effect immediately

SETOMVS PID=nnnnnnnn
- to change the limit for a specific process

BPXPRMxx parmlib limits monitored

SYSTEM level limits:

- MAXPROCSYS
- MAXUIDS
- MAXPTYS
- MAXMMAPAREA
- MAXSHAREPAGES
- IPCMSGNIDS
- IPCSEMNIDS
- IPCSHMNIDS
- IPCSHMSPAGES
- IPCMSGQBYTES
- IPCMSGQMNUM
- IPCSHMMPAGES
- SHRLIBRGNSIZE
- SHRLIBMAXPAGES

PROCESS level limits:

- MAXFILEPROC
- MAXFILESIZE
- MAXPROCUSER
- MAXQUEUEDSIGS
- MAXTHREADS
- MAXTHREADTASKS
- IPCSHMNSEGS
- MAXCORESIZE
- MAXMEMLIMIT

SOCKETS Address Family level limit:

- MAXSOCKETS

UNIX User Limits

- Stored in **OMVS** segment of user profile
  - CPUTIMEMAX
  - ASSIZEMAX
  - FILEPROCMAX
  - PROCUSERMAX
  - THREADSMAX
  - MMAPAREAMAX
  - MEMLIMIT

ADDUSER ... OMVS(CPU(100) ASSIZEMAX(200M) ...)

Monitoring Message controls

SETOMVS LIMMSG=NONE

SYSTEM

ALL

- **LIMMSG=NONE** (default)
  - **No** console messages issued for any of the limits.
- **LIMMSG=SYSTEM**
  - Console messages will be issued for
    - SYSTEM level limits
    - PROCESS level limits for a process if limit
      - *is defined in the user's OMVS segment*
      - *was changed via the SETOMVS PID= command*
- **LIMMSG=ALL**
  - Console messages issued for all SYSTEM and PROCESS level limits

Session Summary

- You should now know enough to be dangerous

  Remember:

  "Unix was not designed to stop people from doing stupid things, because that would also stop them from doing clever things."
  -- Doug Gwyn

- Experiment (on a test system) to learn more

- Ask questions

References

- z/OS V1R9.0 UNIX System Services **User's Guide** (SA22-7801)
  - Lots of introductory material for shell users
- z/OS V1R9.0 UNIX System Services **Command Reference** (SA22-7802)
  - The gory details of all shell commands
- z/OS V1R9.0 UNIX System Services **Planning** (SA22-7800)
  - System customization
- z/OS V1R9.0 UNIX System Services **Messages and Codes** (SA22-7807)
- z/OS UNIX System Services
  - http://www.ibm.com/servers/eserver/zseries/zos/unix
- UNIX System Services Online Bookshelf:
  - http://publibz.boulder.ibm.com/cgi-bin/bookmgr/Shelves/BPXZSH50