PERKIN-ELMER

EDITION VII WORKBENCH (VERSION 2.4)

BINARY SYSTEMS SOFTWARE PACKAGE

REFERENCE DOCUMENT

Consists of Packaging Information For:

Magnetic Tape (800 BPI) Package	04-164M31R04
Magnetic Tape (1600 BPI) Package	04-164M71R04
Disk (16Mb) Package	04-164MD1R04
Disk (16Mb with IDC Format) Package	04-164MG1R04
Disk (25Mb) Package	04-164MJ1R04

04-164M95R04

The information in this document is subject to change without notice. The Perkin-Elmer Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license, and is a product of The Perkin-Elmer Corporation. It can be used or copied only in a manner permitted by that license. Any copy of the described software must include all copyright notices. Title to and ownership of the described software and any copies thereof shall remain in The Perkin-Elmer Corporation.

The Perkin-Elmer Corporation assumes no responsibility for the use or reliability of the software.

The Perkin-Elmer Corporation
Data Systems Group
2 Crescent Place
Oceanport, NJ 07757

Printed in the United States of America

Table of Contents

1. PRODUCT IDENTIFICATION

Product Title:	Edition VII Workbench (Version 2.4) Binary Systems Software Package
Part Number:	04-164
Sales Numbers:	S50-316, S51-302, S51-303, S52-302, S52-303, S52-304, S52-305, S53-302, S53-303, S53-304, S53-305

2. AVAILABLE PACKAGES

This section discusses the available binary packages. Each sales number has a suffix which indicates the distribution media. Available media are:

Media	Part Number Suffix	Installation Type
800 BPI tape	B	tape
1600 BPI tape	C	tape
16MB MSM disc	P	msm cl3r
16MB IDC disc	Q	idc cl3r
25MB IDC disc	U	dl9r

The installation type is used in determining the installation procedure to follow. These media are shown as an underbar '_' in the tables below.

The following is a list of the available binary system packages for Edition VII Workbench (Version 2.4). Each of these packages allows for the specified number of ports, contains complete documentation, and a single 800 BPI tape, 1600 BPI tape, 13MB MSM format cartridge disc, 13MB IDC format cartridge disc, or 19MB Lark Cartridge disc as specified by the suffix.

Three types of systems are listed down the left side: initial CPU, additional CPU, and educational. The software for all of the entries in this table is identical; the only difference is in the type of license required. All entries in this table have the part number 04-164.

At the head of the table, Groups I, II, and III refer to processor types. Group I includes the 3205 processor, Group II includes the 7/32, 3210, 3220, and 3230, while Group III includes the 8/32, 3240, 3250, and 3250XP. Edition VII does not support the 3200MPS.

BINARY SYSTEMS

Ports	Group I	Group II	Group III
16 ports initial CPU	S51-302-AB_	S52-302-AB_	S53-302-AB_
16 ports additional CPU	S51-302-XB_	S52-302-XB_	S53-302-XB_
32 ports initial CPU	S51-303-AB_	S52-303-AB_	S53-303-AB_
32 ports additional CPU	S51-303-XB_	S52-303-XB_	S53-303-XB_
64 ports initial CPU	See note 1	S52-304-AB_	S53-304-AB_
64 ports additional CPU	See note 1	S52-304-XB_	S53-304-XB_
128 ports initial CPU	See note 1	S52-305-AB_	S53-305-AB_
128 ports additional CPU	See note 1	S52-305-XB_	S53-305-XB_
16 ports educational	S50-316-AB_	S50-316-AB_	S50-316-AB_

3. DOCUMENTATION

The Edition VII Workbench (Version 2.4) Binary Systems Software Documentation Package (Number 04-164F00M99R04) contains the following manuals:

Note 1: This system available by special quote only. Contact the UNIX support group at Pine Street (201)632-3333.

DOCUMENTS

Publication Number	Title
48-092 F01 R02	Edition VII Workbench Programmer's Reference Manual (Volume 1)
48-093 F01 R02	Edition VII Workbench Programmer's Reference Manual (Volume 2)
51-042	"The C Programming Language" by Brian Kernighan and Dennis Ritchie
51-043	"A UNIX Primer" by Ann Nichols Lomuto and Nico Lomuto

4. INTRODUCTION TO INSTALLATION

Edition VII is derived from UNIX* Version 7 software. The remainder of this document is also reprinted in Volume 2 of the Edition VII Workbench Programmer's Manual under the title Installing Edition VII.

The distribution media for Edition VII Workbench is one of the following:

* 9-track, 800 BPI magnetic tape
* 9-track, 1600 BPI magnetic tape
* CDD 16MB removable disc+, IDC format
* CDD 16MB removable disc+, MSM format
* Lark 25MB IDC disc+, IDC format

Binary installations receive a single tape or disc while source installations receive two tapes or discs. For source installations, the first tape or disc is a binary distribution, while the second tape or disc is a complete copy of the source.

*UNIX is a Trademark of Bell Laboratories.
+ The CDD 16MB disc, also known as a C13R, contains 16MB unformatted or 13MB formatted capacity. The 25MB Lark disc contains 25MB unformatted or 19MB formatted capacity.

The system as distributed contains binary images of the UNIX kernel and all the user level programs, along with source and manual sections for them -- about 2300 files altogether. The binary images, along with other things needed to flesh out the file system enough so Edition VII will run, are to be put on one file system called the 'root file system'. These files occupy some 3600 1k blocks, but are normally placed in a file system of 4800 1k blocks to allow for expansion. The second file system has libraries, the online manual, and the libraries necessary to regenerate the Edition VII kernel. These files occupy some 6750 1K blocks, but are normally placed in a file system of 25600 blocks to allow for inclusion of source and further expansion.

5. GENERAL GUIDELINES

The next three sections describe how to install Edition VII from tape and disc. The first two procedures use as an example installation on a non-standard address 67MB disc, while the third procedure deals with a non-standard address 19MB fixed disc. The actual disc you should use depends on the configuration supplied.

All disc packs used by Edition VII must be certified error-free and flag-free (i.e., no alternate tracks or sectors are allowed).

As a safety measure, it is wise to make full backups of your system in case an error is made in installation. Also, all discs (except the one to be used as a system disc) should be set to write protect.

The console terminal must be at address X'10', using the highest baud rate and even parity.

In the installation procedures the machine's output is shown underlined, and explanatory comments are shown in brackets [].

Throughout the installation procedure there are references to the type of disc or tape being used. These references are in the form x(a,b) where x is the disc type, a is the drive number, and b is the offset from the start of the pack.

Disc types recognized are:

dsk	[10MB disc, fixed or removable platter]
d40	[40MB disc]
d67	[67MB disc]
d256	[256MB disc]
c13r	[cdd 13MB removable cartridge]
c13f	[cdd 13MB fixed]
c40f	[cdd 40MB fixed]
c67f	[cdd 67MB fixed]
d300	[300MB Capricorn]
d19r	[19MB Lark removable]
d19f	[19MB Lark fixed]
mt	[mag tape]

The drive number is 0 if the standard address is used (normally FC/FB/F0 for device, controller, and selch respectively for discs, and C5/F0 for device and selch for tapes), and "*" otherwise. If a "*" is used, you will be individually prompted for those three values, which should be two digit hexadecimal numbers in upper case. The controller is irrelevant for mag tape units, so any value will do. In this document, most examples assume non-standard addresses, since the majority of system devices are configured at non-standard addresses. However, prompts and responses for device/controller/selch are not shown.

The offset from the beginning of the pack is given in 512 byte blocks. The most commonly used values are 0 (/tmp file system), 19200 (root file system), and 28800 (/usr file system).

For example, the notation d67(0,19200) refers to the root file system of a 67MB disc at standard addresses.

Each line typed should be terminated by a carriage return. If a mistake is made while typing, the character '#' or backspace erases the last character typed on the current line, and the character '@' erases the entire line typed.

Some consoles cannot print lower case letters; adjust the instructions accordingly.

When creating file systems, the correct number of blocks per track must be known. The appropriate values are:

80 for d67 and c67f
304 for d256
16 for c13r and c13f
48 for c40f
256 for d300
31 for d19r and d19f

6. INSTALLATION FROM TAPE

Distribution tapes are either 9-track 800 BPI or 9-track 1600 BPI and contain the following files, separated by tapemarks:

File 0: (80-byte records)
Edition VII *boot*(8) program,
loadable by "50 Sequence"
File 1: (256-byte records)
Edition VII *boot*(8) program,
loadable by LSU (3200 series)
File 2: (1k-byte records)
Standalone *mkfs*(1) program
File 3: (1k-byte records)
Standalone *restor*(1) program
File 4: (1k-byte records)
Standalone *icheck*(1) program
File 5: (1k-byte records)
Standalone *adb*(1) program
File 6: (20480-byte records)
Root filesystem in *dump*(5) format
File 7: (10240-byte records)
'/usr' filesystem in *tar*(1) format
File 8: (10240-byte records)
'/' filesystem in *tar*(1) format

6.1. Starting the 1K Bootloader

Mount the distribution tape at load point *with the 'write' ring removed*.

6.1.1. Load the Appropriate 'boot' Program.

Boot may be loaded in one of several ways depending on hardware configuration. Tables 1 and 2 below describe device codes and "standard" hardware addresses; local installation device addresses may be different.

6.1.1.1. Loading 'boot' from Tape via "50 Sequence" (7/32 and 8/32)

A. Use the Display Panel to initialize low-memory device definitions as follows.

Address	Value	Meaning	
50	D500	al x'CF'	auto-load
52	00CF		
54	4300	b x'80'	branch to
56	0080		loaded program
78	xxyy		xx = address of input device (see Table 1 below) yy = device READ command

B. Clear the PSW status to zero [DTA-0-FN-SEL].

C. Start execution at address x'50' [DTA-50-ADR-RUN].

Table 1: Standard "50 Sequence" Devices

Device Address	READ Command	Device
02	94	Teletype reader
13	99	Paper tape reader/punch
45	A1	Cassette tape
85	A1	Magnetic tape at address 85
C5	A1	Magnetic tape at address C5

6.1.1.2. Loading 'boot' from Disc or Tape via LSU (3200 series)

A. Turn the 'IPL' switch to ENABLE and press the 'INIT' switch. A list of device mnemonics should appear on the console, followed by the prompt

```
DEVICE=
```

B. If the boot device is configured with "standard" device, controller and selch addresses, enter the corresponding device mnemonic from Table 2 below; otherwise enter 'OTHR'.

C. If 'OTHR' was specified above, respond to further prompts as applicable.

DEV#= device address 85
CODE= device type code 40
CTLR= controller address [illegible]
SLCH= selector channel address F1

D. Respond to the prompt

FILEMARKS=

with the decimal "1" (this is the number of files to skip on the standard distribution tape).

Table 2: Standard LSU Devices

See Loader Storage Unit (LSU) Programming Manual 29-450R01. Your LSU default addresses may be different. The following is the defaults used by the bootstrap.

LSU Mnemonic	Device Code	Device Address	Controller Address	Selch Address	Device
DS5F	31	C6	B6	F0	2.5MB disc
DS5F	32	C6	B6	F0	10MB disc (fixed)
DS5R	33	C6	B6	F0	10MB disc (removable)
	34	FC	FB	F0	40MB disc
DS67	35	FC	FB	F0	67MB MSM disc
D256	36	FC	FB	F0	256MB MSM disc
C13R	3B	FC	FB	F0	13MB CDD Cartridge
C13F	3C	FC	FB	F0	13MB CDD Fixed
C40F	3D	FC	FB	F0	40MB CDD Fixed
C67F	3E	FC	FB	F0	67MB CDD Fixed
D19R	2A	FC	FB	F0	19MB Lark Removable
D19F	2B	FC	FB	F0	19MB Lark Fixed
D300	2C	FC	FB	F0	300MB Capricorn
MG85	40	85	--	F0	800bpi magtape
MGC5	41	C5	--	F0	1600bpi magtape

Note: on the 3200 series, to load from 2.5MB or 40MB disc, use the device mnemonic / code for 10MB and 67MB disc respectively.

6.1.2. Verifying the 1K Bootloader

After the bootstrap has been successfully loaded the console will type

```
Boot-1k
:
```

If instead of the Boot-1k message the console displays a ">" sign, microcode, or the wait light remains continually dim, follow the instructions given in Table 3 to initialize memory.

Table 3: Machine Code for Initializing Memory*

To initialize memory press the halt switch on the system and type in the following machine code:

```
@100   F840
@102   0004
@104   0000
@106   2454
@108   F860
@10A   00XX    [XXFFFC = top of core]
@10C   FFFC    [XX is in 64k byte units, values are:]
@10E   5044    [03 = 256k]
@110   0000    [07 = 512k]
@112   C140    [0F = 1mb]
@114   FFF8    [1F = 2mb]
@116   8800    [3F = 4mb]
P
=0
<
```

Then press the enable and init switches on the system, and reboot.

If the system refuses to leave microcode, press the halt switch and type in the following:

```
@0x84   5000
@0x86   6000
P
=0
<
```

Now press the enable and init switches on the system, and reboot.

6.2. Restoring the Root File System

*Perkin-Elmer Field Information Bulletin 209

6.2.1. Making the Root File System

Make an empty file system by running the mkfs program as follows:

```
[to bring in the program mkfs]
: mt(*,2)
Mkfs
file system size:(1k-byte blocks) 4800
file system: d67(*,19200)
isize = XX
interleave (m): 8
blocks per cylinder (n): 80
m/n = XX
[after a while]
exit called
Boot-1k
:
```

The correct value for blocks per cylinder can be found in section 2.

6.2.2. Copying in the Root File System

The next thing to do is to restore the data onto the new empty file system. To do this you respond to the ':' printed in the last step with

```
[to bring in the program restor]
: mt(*,3)
Restor
tape? mt(*,6)
disk? d67(*,19200)
Hit RETURN key when ready to restore to disk.
[the tape moves, perhaps 5-10 minutes pass]
end of tape
Boot-1k
:
```

You now have an Edition VII root file system on msm0c.

6.3. Booting Edition VII

You should have the bootstrap running, left over from the last step above; if not, repeat the boot process [section 3.1] again. Then enter

```
: d67(*,19200)edition7
```

The machine will type the following:

```
Edition VII Software for use on CPU Serial Number xxxxx
   under license number xxxxx with The Perkin-Elmer Corporation
Copyright The Perkin-Elmer Corporation 1983
All rights reserved
Not to be construed as a published work
May not be copied without permission of Perkin-Elmer
Memory = xxx.xx K
#
```

The *memory* message gives the memory available to user programs in kilobytes.

Edition VII is now running, and the 'Edition VII Programmer's Manual' applies; references below of the form X(Y) mean the subsection named X in section Y of the manual. The '#' is the prompt from the Shell, and indicates you are the super-user. The user name of the super-user is 'root', with no password required, *HOWEVER*, *be sure to set a password* (*see passwd*(*1*)) *as soon as possible or your system may be compromised*!

6.4. Setting up /tmp, /u, and Restoring /usr

6.4.1. Creating the /tmp File System

Next make and mount a file system for /tmp. This can be done by typing the following commands

```
/etc/mkfs /dev/rmsm0a 4800 8 80
/etc/mount /dev/msm0a /tmp
```

The correct number of blocks per cylinder can be found in section 2.

6.4.2. Putting the Bootstrap on Disc

Before anything further is done the bootstrap program should be copied to the disc. This is done by using the command

```
/etc/mkboot -9600 /boot /dev/msm0
```

Now the Perkin-Elmer LSU bootstrap is usable. See *boot*(8) for further information.

6.4.3. Creating the /u File System

It is wise to create a file system separate from "root" and "usr" for the users' files to reside. Use the following commands to make the file system then mount it.

```
        /etc/mkfs /dev/rmsm0e 25600 8 80
        /etc/mount /dev/msm0e /u
```

6.4.4. Creating the /usr File System

The next thing to do is to restore the '/usr' file system from the tape.

```
        /etc/mkfs /dev/rmsm0d 25600 8 80
```

At this point mount the '/usr' file system (mount(8)), change to the /usr directory (cd(1)), skip 7 files on the tape, and use tar to read the tape (tar(1)). To do this type the following:

```
        /etc/mount /dev/msm0d /usr
        cd /usr
        tpc ff 7         [skip 7 files on the tape]
        tar xvp          [extract the 'usr' file system]
```

Now unload the tape and store it in a safe place.

6.5. Loading System Source

This section applies only to those installations with system source.

To load system source, load the second tape with the 'write' ring removed, and restore the files as follows

```
        mkdir /usr/src
        cd /usr/src
        tar xvpf /dev/rmmt0
        mkdir /usr/doc
        cd /usr/doc
        tar xvpf /dev/rmmt0
```

The tape will advance for several minutes. When it completes, remove the tape from the drive and store it in a safe place. The layout of source is described in "Regenerating System Software" and the hier(7) manual page. The /usr/doc directory contains most documents in volume 2 of the Edition VII Programmer's Manual. It may be removed from the system to save space (about 1MB).

6.6. Example of loading from tape & booting from an D67 disc

```
DEVICE=OTHR
DEV#=85
CODE=40
SLCH=F1
FILEMARKS=1
Boot-1k
: mt(*,2)
Mkfs
file system size:(1k-byte blocks) 4800
file system: d67(*,19200)
isize = XX
interleave (m): 8
blocks per cylinder (n): 80
m/n = XX
exit called
Boot-1k
: mt(*,3)
Restor
tape? mt(*,6)
disk? d67(*,19200)
Hit RETURN key when ready to restore to disk.
end of tape
Boot-1k
: d67(*,19200)edition7
Edition VII Software for use on CPU Serial Number xxxxx
      under license number xxxxx with The Perkin-Elmer Corporation
Copyright The Perkin-Elmer Corporation 1983
All rights reserved
Not to be construed as a published work
May not be copied without permission of Perkin-Elmer
Memory = xxx.xx K
# /etc/mkfs /dev/rmsm0a 4800 8 80
# /etc/mount /dev/msm0a /tmp
# /etc/mkboot -9600 /boot /dev/msm0
# /etc/mkfs /dev/rmsm0e 25600 8 80
# /etc/mount /dev/msm0e /u
# /etc/mkfs /dev/rmsm0d 25600 8 80
# /etc/mount /dev/msm0d /usr
# cd /usr
# tpc ff 7
# tar xvp
[Following commands for source only]
# mkdir /usr/src
# cd /usr/src
# tar xvpf /dev/rmmt0
# mkdir /usr/doc
# cd /usr/doc
# tar xvpf /dev/rmmt0
```

7. INSTALLATION FROM CDD REMOVABLE DISC

This section describes the procedures to use when installing Edition VII from a CDD 16MB (unformatted capacity) removable disc. This disc is generally referred to as a C13R (which stands for Cdd 13mb (formatted) Removable).

The C13R disc is set up in a manner essentially similar to that of a distribution tape. It contains the following four disc sections:

msm0a:Standalone boot program, pointed to by volume header (60-512 byte blocks)

msm0b:File system with standalone boot, adb, icheck, mkfs, restor (340-512 byte blocks)

msm0c:Root file system in dump(5) format (about 4000-1K blocks)

msm0d:/usr file system in dump(5) format (about 8100-1K blocks)

7.1. Starting the 1K Bootloader

To bring the system up, insert the C13R cartridge into the drive (which is generally at address FC/FB/F0 for device address, controller address, and selch address, respectively) and bring it up to speed. Set the protection switch to ON for the upper (removable) portion. If you are restoring to the lower (fixed) portion, set the protection to OFF for that disc. When it is ready, bring in the 1k bootloader as discussed under tape loading using either the 50 Sequence or the LSU (see section 3.1). The 'normal' case (i.e., with an LSU and the disc at the standard address) requires depressing the ENABLE and INIT buttons, and responding to the DEVICE= prompt with 'C13R' and the FILE= prompt with 'BOOT'. The system should respond

```
Boot-1k
:
```

If you do not get this message, see the section 3.1.3 for how to initialize memory.

7.2. Bring in Root File System

7.2.1. Creating the Root File System

First, make an empty root file system with the following procedure:

```
[to bring in the program mkfs]
: cl3r(*,60)mkfs
Mkfs
file system size:(1k-byte blocks) 4800
file system: d67(*,19200)
isize = XX
interleave (m): 8
blocks per cylinder (n): 80
m/n = XX
[after a while]
exit called
Boot-1k
:
```

See section 2 for the correct value for blocks per cylinder. You now have an empty root file system.

7.2.2. Restoring the Root File System

Next restore the contents of the root file system using the standalone restor program:

```
[to bring in the program restor]
: cl3r(*,60)restor
Restor
tape? cl3r(*,400)
disk? d67(*,19200)
Hit RETURN key when ready to restore to disk.
[10-15 minutes pass]
end of tape
Boot-1k
:
```

You now have a root file system.

7.3. Bring in /usr File System

7.3.1. Create the /usr File System

First build the file system using the sequence:

```
[to bring in the program mkfs]
: cl3r(*,60)mkfs
Mkfs
file system size:(1k-byte blocks) 25600
file system: d67(*,28800)
isize = XX
interleave (m): 8
blocks per cylinder (n): 80
m/n = XX
[after a while]
exit called
Boot-1k
:
```

7.3.2. Restore the /usr File System

Now restore the contents of the /usr file system:

```
[to bring in the program restor]
: cl3r(*,60)restor
Restor
tape? cl3r(*,9400)
disk? d67(*,28800)
Hit RETURN key when ready to restore to disk.
[20-25 minutes pass]
end of tape
Boot-1k
:
```

You have now created the /usr file system.

7.4. Booting Edition VII

You are now ready to bring up Edition VII in single-user mode. Use the following command:

```
: d67(*,19200)edition7
```

The machine will type the following:

```
Edition VII Software for use on CPU Serial Number xxxxx
   under license number xxxxx with The Perkin-Elmer Corporation
Copyright The Perkin-Elmer Corporation 1983
All rights reserved
Not to be construed as a published work
May not be copied without permission of Perkin-Elmer
Memory = xxx.xx K
#
```

The memory message gives the memory available to user programs in kilobytes.

Edition VII is now running, and the 'Edition VII Programmer's Manual' applies; references below of the form X(Y) mean the subsection named X in section Y of the manual. The '#' is the prompt from the Shell, and indicates you are the super-user. The user name of the super-user is 'root', with no password required, *HOWEVER, be sure to set a password (see passwd(1)) as soon as possible or your system may be compromised*!

7.5. Setting up /tmp and /u, Creating a Boot Block

7.5.1. Setting up /tmp File System

The next step is to create and mount a file system for /tmp. This can be done as follows:

```
/etc/mkfs /dev/rmsm0a 4800 8 80
/etc/mount /dev/msm0a /tmp
```

7.5.2. Setting up /u File System

Now create the /u file system:

```
/etc/mkfs /dev/rmsm0e 25600 8 80
/etc/mount /dev/msm0e /u
```

7.5.3. Create a Boot Block

Now create a boot block. This is extremely important, as you will not be able to boot directly from your system disc until you have a boot block. Use the command

```
/etc/mkboot -9600 /boot /dev/msm0
```

The Perkin-Elmer LSU bootstrap is now usable. See boot(8) for further information.

Now execute a couple of sync(2) commands to flush out all data and then spin the disc down. Remove the c13r disc and store it in a safe place. If you do not have source, insert a new disc, turn the write protect to OFF, and spin the drive back up to speed.

7.6. Loading System Source

This section applies only to those installations with system source.

To load system source, load the second disc and spin it up to speed, and restore the files as follows

```
mkdir /mnt
/etc/mount /dev/msml /mnt -r
mkdir /usr/src
cd /usr/src
tar xvpf /mnt/source
mkdir /usr/doc
cd /usr/doc
tar xvpf /mnt/documents
/etc/umount /dev/msml
```

Depending on your configuration, the removable disc may have a name other then msml. You can verify the correct name by looking in your configuration file for the cl3r disc. You can verify the correct disc by doing an fsck(8) on the disc, and if it does not have any errors mounting the disc. If it is the source disc, there will be a large file called 'source' and another called 'documents' on the disc.

The source will be copied in for several minutes. When it completes, execute a couple of sync(2) commands to flush the buffers and spin the disc down. Then remove the disc from the drive and store it in a safe place. Insert a new disc in the drive, turn the write protect to OFF, and spin the drive back up to speed.

The layout of source is described in "Regenerating System Software" and the hier(7) manual page.

7.7. Example of Loading from Disc & Booting from an D67 disc

```
DEVICE=C13R
VOL=ED7, FILE=BOOT
Boot-1k
: c13r(*,60)mkfs
Mkfs
file system size:(1k-byte blocks) 4800
file system: d67(*,19200)
isize = XX
interleave (m): 8
blocks per cylinder (n): 80
m/n = XX
exit called
Boot-1k
: c13r(*,60)restor
Restor
tape? c13r(*,400)
disk? d67(*,19200)
Hit RETURN key when ready to restore to disk.
end of tape
Boot-1k
: c13r(*,60)mkfs
Mkfs
file system size:(1k-byte blocks) 25600
file system: d67(*,28800)
isize = XX
interleave (m): 8
blocks per cylinder (n): 80
m/n = XX
exit called
Boot-1k
: c13r(*,60)restor
Restor
tape? c13r(*,9400)
disk? d67(*,28800)
Hit RETURN key when ready to restore to disk.
: d67(*,19200)edition7
Edition VII Software for use on CPU Serial Number xxxxx
      under license number xxxxx with The Perkin-Elmer Corporation
Copyright The Perkin-Elmer Corporation 1983
All rights reserved
Not to be construed as a published work
May not be copied without permission of Perkin-Elmer
Memory = xxx.xx K
# /etc/mkfs /dev/rmsm0a 4800 8 80
# /etc/mount /dev/msm0a /tmp
# /etc/mkfs /dev/rmsm0e 25600 8 80
# /etc/mount /dev/msm0e /u
# /etc/mkboot -9600 /boot /dev/msm0
[Following commands for source only]
# mkdir /mnt
# /etc/mount /dev/msm1 /mnt -r
# mkdir /usr/src
# cd /usr/src
```

```
        # tar xvpf /mnt/source
        # mkdir /usr/doc
        # cd /usr/doc
        # tar xvpf /mnt/documents
        # /etc/umount /dev/msml
```

8. INSTALLATION FROM LARK REMOVABLE DISC

This section describes the procedures to use when installing Edition VII from a Lark 25MB (unformatted capacity) removable disc. This disc is generally referred to as a D19R (as the disc is a 19MB (formatted) removable pack). This installation procedure is typically used on 3205s where the D19F (the fixed portion of the disc) is used as the system disc.

The D19R disc is set up in a manner essentially similar to that of a distribution tape. It contains the following three disc sections:

msm0a:Standalone boot program, pointed to by volume header (60-512 byte blocks), plus a file system with standalone boot, adb, icheck, mkfs, restor (340-512 byte blocks)

msm0b:Root file system in dump(5) format (about 4000-1K blocks)

msm0c:/usr file system in tar(1) format (about 8100-1K blocks)

8.1. Starting the 1K Bootloader

To bring the system up, insert the D19R cartridge into the drive (which is generally at address FC/FB/F0 for device address, controller address, and selch address, respectively) and bring it up to speed. Set the protection switch to ON for the upper (removable) portion. If you are restoring to the lower (fixed) portion, set the protection to OFF for that disc. When it is ready, bring in the 1k bootloader as discussed under tape loading using either the 50 Sequence or the LSU (see section 3.1). The 'normal' case (i.e., with an LSU and the disc at the standard address) requires depressing the ENABLE and INIT buttons, and responding to the DEVICE= prompt with 'D19R' and the FILE= prompt with 'BOOT'. The system should respond

```
Boot-1k
:
```

If you do not get this message, see the section 3.1.3 for how to initialize memory.

8.2. Bring in Root File System

8.2.1. Creating the Root File System

First, make an empty root file system with the following procedure:

```
[to bring in the program mkfs]
: d19r(*,0)mkfs
Mkfs
file system size:(1k-byte blocks) 4800
file system: d19f(*,0)
isize = XX
interleave (m): 8
blocks per cylinder (n): 31
m/n = XX
[after a while]
exit called
Boot-1k
:
```

See section 2 for the correct value for blocks per cylinder if you have other than a D19F as a system disc. You now have an empty root file system.

8.2.2. Restoring the Root File System

Next restore the contents of the root file system using the standalone restor program:

```
[to bring in the program restor]
: d19r(*,0)restor
Restor
tape? d19r(*,9600)
disk? d19f(*,0)
Hit RETURN key when ready to restore to disk.
[10-15 minutes pass]
end of tape
Boot-1k
:
```

You now have a root file system.

8.3. Booting Edition VII

You are now ready to bring up Edition VII in single-user mode. Use the following command:

```
: d19f(*,0)edition7
```

The machine will type the following:

```
Edition VII Software for use on CPU Serial Number xxxxx
   under license number xxxxx with The Perkin-Elmer Corporation
Copyright The Perkin-Elmer Corporation 1983
All rights reserved
Not to be construed as a published work
May not be copied without permission of Perkin-Elmer
Memory = xxx.xx K
#
```

The *memory* message gives the memory available to user programs in kilobytes.

Edition VII is now running, and the 'Edition VII Programmer's Manual' applies; references below of the form X(Y) mean the subsection named X in section Y of the manual. The '#' is the prompt from the Shell, and indicates you are the super-user. The user name of the super-user is 'root', with no password required, *HOWEVER, be sure to set a password* (*see passwd*(*1*)) *as soon as possible or your system may be compromised*!

8.4. Bring in /usr File System

First build the file system using the sequence:

```
/etc/mkfs /dev/rmsm0c 9744 8 31
```

Now mount(8) the /usr file system and restore the contents from the removable disc using tar(1):

```
/etc/mount /dev/msm0c /usr
cd /usr
tar xvpf /dev/rmsm1c
```

Depending on your configuration, the removable disc may have a name other then msm1. You can verify the correct name by looking in your configuration file for the dl9r disc.

You have now created the /usr file system.

8.5. Creating a Boot Block

Lark based systems do not have /tmp file systems, so this step which is present in the tape and cartridge installation procedures is intentionally omitted. In addition, the /u file system is typically on the removable disc, so that procedure is omitted also.

Now create a boot block. This is extremely important, as you will not be able to boot directly from your system disc until you have a boot block. Use the command

```
/etc/mkboot -9600 /boot /dev/msm0
```

The Perkin-Elmer LSU bootstrap is now usable. See boot(8) for further information.

Now execute a couple of sync(2) commands to flush out all data and then spin the disc down. Remove the d19r disc and store it in a safe place. If you do not have source, insert a new disc and spin the drive back up to speed. You can then create a /u file system on the removable disc.

8.6. Loading System Source

This section applies only to those installations with system source.

To load system source, load the second disc and spin it up to speed, and restore the files as follows

```
mkdir /mnt
/etc/mount /dev/msm1 /mnt -r
mkdir /usr/src
cd /usr/src
tar xvpf /mnt/source
mkdir /usr/doc
cd /usr/doc
tar xvpf /mnt/documents
/etc/umount /dev/msm1
```

As mentioned in section 8.4, the removable disk may be other than msm1. The correct value can be found in the configuration or you can verify the correct disc by doing an fsck(8) on the disc, and if it does not have any errors mounting the disc. If it is the source disc, there will be a large file called 'source' and another called 'documents' on the disc.

The source will be copied in for several minutes. When it completes, execute a couple of sync(2) commands to flush the buffers and spin the disc down. Then remove the disc from the drive and store it in a safe place. Insert a new disc in the drive and spin the drive back up to speed.

The layout of source is described in "Regenerating System Software" and the hier(7) manual page.

8.7. Example of Loading from Lark Removable & Booting from a Lark Fixed disc

```
DEVICE=D19R
VOL=ED7, FILE=BOOT
Boot-1k
: d19r(*,0)mkfs
Mkfs
file system size:(1k-byte blocks) 4800
file system: d19f(*,0)
isize = XX
interleave (m): 8
blocks per cylinder (n): 31
m/n = XX
exit called
Boot-1k
: d19r(*,0)restor
Restor
tape? d19r(*,9600)
disk? d19f(*,0)
Hit RETURN key when ready to restore to disk.
end of tape
Boot-1k
: d19f(*,0)edition7
Edition VII Software for use on CPU Serial Number xxxxx
      under license number xxxxx with The Perkin-Elmer Corporation
Copyright The Perkin-Elmer Corporation 1983
All rights reserved
Not to be construed as a published work
May not be copied without permission of Perkin-Elmer
Memory = xxx.xx K
# /etc/mkfs /dev/rmsm0c 9744 8 31
# /etc/mount /dev/msm0c /usr
# cd /usr
# tar xvfp /dev/rmsm1c
# /etc/mkboot -9600 /boot /dev/msm0
[Following commands for source only]
# mkdir /mnt
# /etc/mount /dev/msm1 /mnt -r
# mkdir /usr/src
# cd /usr/src
# tar xvpf /mnt/source
# mkdir /usr/doc
# cd /usr/doc
# tar xvpf /mnt/documents
# /etc/umount /dev/msm1
```

9. CUSTOMIZATION

9.1. Reconfiguration

The EDITION VII system running contains the system described in your configurator. This should include your first six discs, up to three tapes, and up to six communications multiplexors. The line speeds may be wrong. You may need to alter the configuration file to reflect the true state of your machine.

It is wise at this point to know how to configure the system. Print (cat(1)) the file /usr/sys/conf/makefile. This file is input to the program *make*(1) which if invoked with 'make all' will recompile all of the system source and install it in the correct libraries.

The program mkconf(1) prepares files that describe a given configuration (See *mkconf*(1m)). In the /usr/sys/conf directory, the file 'conf.xxxxx' was input to mkconf to produce the version of the system in /edition7. You will need to edit it to describe your own configuration. Then run mkconf; it will generate the files param.s (assembly parameters) and c.c (configuration tables). Take a careful look at these files to make sure that all the devices that you have are included at the right addresses.

After all the corrections have been made, use make(1) to recompile the system (or recompile individually if you wish: use the makefile as a guide). If you compiled individually, say 'make edition7' in the directory /usr/sys/conf. The final object file (edition7) should be moved to the root, and then booted to try it out. It is best to name it something like /nedition7 so as not to destroy the working system until you are sure it works. Or boot it from the current directory with :

```
d67(*,28800)sys/conf/edition7
```

See *boot*(8) for a discussion of booting. Note: before taking the system down, always (!!) perform a sync(1) to force delayed output to the disc.

9.2. Special Files

Next you must put in special files for the new devices in the directory /dev using mknod(1). The file /usr/sys/conf/mkdev.sh (created by mkconf) will give you an idea of how to create special files. Do not use this file without prior examination, as it is not complete. You can also print the configuration file c.c created above, although this method is somewhat more difficult. C.c is the major device switch of each device class (block and character). There is one line for each device configured in your system and a null line for place holding for those devices not configured. The essential block special files were installed above; for any new devices, the major device number is selected by counting the line number (from zero) of the device's entry in the block configuration table. Thus the first entry in the table bdevsw would be major device zero. This number is also printed in the table along the right margin.

The minor device is the drive number, unit number or partition as described under each device in section 4 of the manual. You can also add entries for other disc drives.

In reality, device names are arbitrary. It is usually convenient to have a system for deriving names, but it doesn't have to be the one presented above.

Some further notes on minor device numbers. The 10MB disc driver uses the 010 bit of the minor device number to indicate whether or not to interleave a file system across more than one physical device. See dsk(4) for more detail. The magtape driver uses the 0200 bit to indicate whether or not to rewind the tape when it is closed. By convention, tape special files with the 0200 bit on have a 'm' prepended to their name, as in /dev/mmt0 or /dev/rmmt1. Again, see mt(4).

The naming of character devices is similar to block devices. Here the names are even more arbitrary except that devices meant to be used for teletype access should (to avoid confusion, no other reason) be named /dev/ttyX, where X is some string (as in '00' or 'library'). The files console, mem, kmem, and null are already correctly configured.

The disc and magtape drivers provide a 'raw' interface to the device which provides direct transmission between the user's core and the device and allows reading or writing large records. The raw device counts as a character device, and should have the name of the corresponding standard block special file with 'r' prepended. Thus the raw magtape files would be called /dev/rmtX. These special files should be made.

When all the special files have been created, care should be taken to change the access modes (chmod(1)) on these files to appropriate values (probably 600 or 644).

9.3. Time Conversion

If the system reflects the incorrect timezone this can be modified by editing /usr/sys/conf/conf.xxxxx to reflect the proper time difference between local time and GMT in hours. Common time zone values include 0 for GMT, 5 for EST, 6 for CST, 7 for MST, and 8 for PST. If daylight savings time (DST) is used in your area, set the "daylight" value in the configuration file to 1. Note that DST works in the U.S.A., but may not work elsewhere (due to varied rules).

9.4. Disc Layout

If there are to be more file systems mounted than just the root and /usr, use mkfs(1) to create any new file system and put its mounting in the file /etc/rc (see init(8) and mount(8)). (You might look at /etc/rc anyway to see what has been provided for you.)

There are two considerations in deciding how to adjust the arrangement of things on your discs: the most important is making sure there is adequate space for what is required; secondarily, throughput should be maximized. Swap space is a critical parameter. The system as distributed has 9500 blocks for swap space. This should be large enough so running out of swap space never occurs. You may want to change these if local wisdom indicates otherwise.

Many common system programs (C, the editor, the assembler, etc.) create intermediate files in the /tmp directory, so the file system where this is stored also should be made large enough to accommodate most high-water marks. If you leave the root file system as distributed (except as discussed above) there should be no problem. All the programs that create files in /tmp take care to delete them, but most are not immune to events like being hung up upon, and can leave dregs. The directory should be examined every so often and the old files deleted.

Exhaustion of user-file space is certain to occur now and then; the only mechanisms for controlling this phenomenon are occasional use of du(1), df(1), quot(1), threatening messages of the day, and personal letters.

The efficiency with which Edition VII is able to use the CPU is largely dictated by the configuration of disc controllers. For general time-sharing applications, the best strategy is to try to split user files, the root directory (including the /tmp directory) and the swap area among three controllers.

Once you have decided how to make best use of your hardware, the question is how to initialize it. If you have the equipment, the best way to move a file system is to dump it (dump(8)) to magtape, use mkfs(1) to create the new file system, and restore (restor(8)) the tape. If for some reason you do not want to use magtape, dump accepts an argument telling where to put the dump; you might use another disc. Sometimes a file system has to be increased in logical size without copying. The super-block of the device has a word giving the highest address which can be allocated. For relatively small increases, this word can be patched using the debugger (adb(1)) and the free list reconstructed using icheck(1). The size should not be increased very greatly by this technique, however, since although the allocatable space will increase, the maximum number of files will not (that is, the i-list size cannot be changed). Read and understand the description given in file system(5) before playing around in this way. You may want to see section msm(4) for some suggestions on how to lay out the information on MSM discs.

If you have to merge a file system into another existing one, the best bet is to use tar(1). If you must shrink a file system, the best bet is to dump the original and restor it onto the new filesystem. However, this might not work if the i-list on the smaller filesystem is smaller than the maximum allocated inode on the larger. If this is the case, reconstruct the filesystem from scratch on another filesystem (perhaps using tar(1)) and then dump it. If you are playing with the root file system and only have one drive the procedure is more complicated. What you do is the following:

1. GET A SECOND PACK!!!!

2. Dump the current root filesystem (or the reconstructed one) using dump(8).

3. Bring the system down and mount the new pack.

4. Retrieve the Edition VII distribution tape and perform sections 3.1 through 3.2.1 (or 4.1 to 4.2.1) at the beginning of this document, substituting the desired file system size instead of 4800 when asked for 'file system size'.

5. Perform section 3.2.2 (or 4.2.2) up to the point where the 'tape' question is asked. At this point mount the tape you made just a few minutes ago. Continue with section 3.2.2 (4.2.2) above substituting 'mt(0,0)' for 'mt(0,6)'.

9.5. New Users

There are two methods for installing new users: using the automated procedure mkuser(8), or by manually editing the password file /etc/passwd (see passwd(5)) and group file /etc/group (see group(5)). One of these procedures should be used once multi-user mode is entered (see init(8)). The following discussion applies to manual creation of users; the mkuser utility does all of these functions for you.

First, you will have to make a current directory for each new user and change its owner to the newly installed name. A .profile should be placed in each users login directory to setup the desired system attributes (i.e. PATH). The default is PATH=/bin:/usr/bin:/usr/ucb: this will give you standard Edition 7 commands. Login as each user to make sure the password file is correctly edited. For example:

```
ed /etc/passwd
$a
joe::10:1::/u/joe:
w
q
mkdir /u/joe
chown joe /u/joe
login joe
ls -la
login root
```

This will make a new login entry for joe, who should be encouraged to use passwd(1) to give himself a password. His default current directory is /u/joe which has been created. The delivered password file has the user guest in it to be used as a prototype.

9.6. Multiple Users

If Edition VII is to support simultaneous access from more than just the console terminal, the file /etc/ttys (ttys(5)) has to be edited. To add a new terminal be sure the device is configured and the special file exists, then set the first character of the appropriate line of /etc/ttys to 1 (or add a new line). Note that init.c will have to be recompiled if there are to be more than 100 terminals. Also note that if the special file is inaccessible when init tries to create a process for it, the system will thrash trying and retrying to open it.

The /etc/ttys file is read each time the user enters multi-user mode. To enable a terminal which is listed in that file but has a first character of 0 (indicating that the line is disabled), use the attach(8) command. Once a line is attached, it will be enabled until it is disabled using the detach(8) command, even if the system is rebooted.

9.7. File System Health

Periodically (say every day or so) and always after a crash, you should check all the file systems for consistency (icheck(1), dcheck(1), fsck(1)). It is quite important to execute sync(8) before rebooting or taking the machine down. This is done automatically every 30 seconds by the update(8) program when a multiple-user system is running, but you should do it anyway to make sure.

Dumping of the file system should be done regularly, since once the system is going it is very easy to become complacent. Complete and incremental dumps are easily done with dump(8). Dumping of files by name is best done by tar(1) but the number of files is somewhat limited. Finally if there are enough drives entire discs can be copied using cp(1), or preferably with dd(1) using the raw special files and an appropriate block size.

9.8. System Naming

The uucp and mail utilities are aware of the name of the system being run on. This name does not matter unless you are in a network of some sort, in which case you should edit /usr/include/whoami.h to have a seven character (or less) system name. Then connect to /usr/custom/mail and do a 'make cp' to install the new version of mail. Next connect to /usr/custom/uucp and do a 'make cp' to install the current versions of all uucp commands. Be careful not to delete the .o files in these two directories, as you will not be able to recreate these utilities without them.

Users with source licenses can delete the /usr/custom directory, as those utilities can be created from /usr/src/cmd/mail and /usr/src/cmd/uucp.

In addition, the special file /dev/cul0 should be linked to whatever port your dial-out line is plugged into. Special file /dev/cua0 is normally linked to /dev/null, as Perkin-Elmer hardware does not support CUA devices. Depending on the type of dial-out modem you use you may not even need a /dev/cul0 line; this requires experimentation. The relevant programs to try this with are cu(1c) and uucp(1c).

9.9. Saving Space

The Edition VII system as distributed contains a complete binary (and source, if applicable). For those systems which are tight on disc space, there are various ways to reduce the amount of space taken by the system. They fall in two general categories: those files which are relevant to other systems and can be deleted, and those files which grow and should be trimmed periodically.

Each Edition VII system has several copies of those programs which differ among different types of CPUs. Note that in each case the file names are foo.cputype (where cputype is 732, 832, 3205, 3210, 3220, 3230, 3240, and 3250). Do NOT delete foo, as it is needed, but foo.* may be deleted (since foo is simply a link to foo.cputype for the appropriate cputype). It is, of course, a good idea to make a tape backup of any files before deleting them from the system. The following files fall into this category:

```
/bin/ps.*
/bin/pstat.*
/bin/adb.*
/usr/ucb/w.*
/usr/include/sys/local.*
/usr/sys/sys/LIB1.*
/usr/sys/dev/LIB2.*
```

The following files and directories grow on a normal basis, and should be checked and trimmed regularly:

```
/usr/spool/uucp/LOGFILE
/usr/spool/uucp/SYSLOG
/usr/tmp/*
/tmp/*
/usr/lib/learn/log/*
/usr/lib/learn/play/*
/usr/games/lib/snake*
/usr/dist/spellhist
```

In addition, the /usr/adm directory contains a number of files which grow on a regular basis. These include:

messages: log of system messages generated by dmesg(8)
awkhist: log of awk(1) invocations
lpr.error*: log of printer errors generated by lpd(8)
sulog*: log of su(1) invocations
wtmp*: log of login(1) invocations

Those files marked with a * grow if they exist, but are not created by the corresponding programs if they do not exist. Other programs may also use /usr/adm as a logging directory.

The man(1) utility keeps formatted copies of manual pages in /usr/man/cat? directories. This makes subsequent invocations of man fast, since the page need not be reformatted. However, it can take quite a bit of space. Options for restricting the space used include removing the files /usr/man/cat?/* on a regular basis or removing the directories altogether (in which case man(1) will reformat the pages every time). Keeping all pages formatted will take over 1MB of disc space.

Finally, various files can be moved from one location to another depending where space is available. In particular, libraries can be moved between /lib and /usr/lib freely, as the loader looks in both places. Utility programs can be moved between /bin and /usr/bin, although this requires more care. Any utility which is useful in single user mode should be in /bin, and some utilities (e.g., login, sh, csh) are expected to be in /bin.

9.10. Known Problems, Odds and Ends

The files /etc/fstab and /etc/rc should be changed to reflect your normally mounted file systems if other than /dev/msm0a as /tmp, /dev/msm0c as root, /dev/msm0d as /usr, and /dev/msm0e as /u, (/dev/msm0a as root, /dev/msm0c as /usr for Lark-based systems) plus any other special things you want done on start-up. In addition, /usr/lib/crontab should be modified to execute those commands needed on a regular interval and /etc/motd should be set to your desired message-of-the-day.

There are a number of known problems with Edition VII which can be easily avoided. If an off-line disc is mounted, the system will hang until the disc is brought on-line. Thus, only spinning discs should be mounted. If a disc being accessed is taken off-line, the system will hang until the disc is brought back on-line.

For MAC machines only: MAC machines have an option of where the hardware MAC registers reside. This hardware selectable option is set depending on how many device addresses are in the system. The standard address of 0x300 corresponds to the default value for 'ndevs' in the configuration file which is 256 (see mkconf(8)). The other two MAC addresses are 0x500 for 512 devices or 0x900 for 1024 devices. Distributed systems contain an 'ndevs' value set appropriately for the highest hardware address genned into the system. If additional hardware addresses are added, then the MAC address must be changed, and the 'ndevs' value in the configuration file must be changed. Note that the MAC address and 'ndevs' value _must_ agree or the system will not boot.

For example, if a new comm mux is to be added at address 0x1FC on a machine which was previously set up with a MAC at address 0x300, then the system must _first_ be regenerated with the new device address and an 'ndevs' value of 512. _Next_, the new comm mux should be added to the hardware and the MAC restrapped for address 0x500. _Then_, the new system can be booted.

For MAT machines only: the 'ndevs' value should reflect the highest device address for efficiency reasons. Setting the 'ndevs' value too low (e.g., a value of 256 when there is an address above 256) may cause the system to crash when devices above address 256 cause interrupts. However, unlike on MAC machines, setting the 'ndevs' value too high (e.g., 1024 when the highest address is less than 256) will only waste space, but will not prevent the system from running.