1076

Bell Laboratories

Cover Sheet for Technical Memorandum

Title- UNIX Redocumentation and Test Report

Date- February 3, 1976

TM- 76-9131-3

Other Keywords -

Author(s)	Location and Room	Extension
J. T. Walker	PY 3D-111	7080
R. A. Zinke	PY 3D-111	7079

Charging Case - 39382

Filing Case - 39382-38

ABSTRACT

This memorandum represents a concise summary of the effort of Dept. 9131 in the UNIX redocumentation request. It describes the methodology employed in rewriting Setting Up UNIX and testing Procedures for Installing UNIX on the PDP 11 (the revised document); the debriefing of the test participants; and other matters.

The two documents referred to above are presented as appendices. Although this memorandum is primarily intended for UNIX, its methodology and content are applicable to the development and testing of other software system user documents.

Pages Text	13	Other	32	Total	45
No. Figures	-	No. Tables	-	No. Refs.	-

Address Label

SEE REVERSE SIDE FOR DISTRIBUTION LIST

Bell Laboratories

subject: UNIX Redocumentation and Test Report
Filing Case - 39382-38
Charging Case - 39382

date: February 3, 1976

from: J. T. Walker
Dept. 9131 - 7080
R. A. Zinke
Dept. 9131 - 7079
TM 76-9131-3

Memorandum for File

INTRODUCTION

Based on reports by the UNIX Support Group (Dept. 8234) and new users that existing UNIX documentation was not considered satisfactory, assistance was requested of the Personnel Subsystems Department (Dept. 9131) to improve the documents. Specifically, Dept. 9131 applied human factors principles and rewrote the high priority document Setting Up UNIX (See Appendix I). Following a technical review of the new document by Dept. 8234, revisions were made, and the revised document was evaluated by a representative user who followed the redocumented procedures.

REDOCUMENTATION METHODOLOGY

The following guidelines were applied in redocumenting Setting Up UNIX:

1. Determine the user's qualifications before writing, and continue to keep the user in mind until the document is finished. In this case the intended user is assumed to be familiar with the minicomputer hardware and its operation.

2. Strictly adhere to the main purpose of the document. Include only those topics, information items and procedures which are

necessary to the main purpose. The main purpose in this case is clear from the revised title: Procedures for Installing UNIX on the PDP 11 (See Appendix II). Procedural or descriptive information concerning routine maintenance, as opposed to installing UNIX, was considered to be relevant for inclusion in another document.

3. Organize the material into major activities.

4. Establish the logical sequence in which activities are performed.

5. Provide a brief, general introduction describing the purpose of the document and stating any general assumptions, rules, or warnings relative to the contents of the document or the intended user.

6. Give each major activity a sequence number, and a title expressed in clear action terms, set off in caps, underlined, or in bold face print for easy identification.

7. Immediately after each major activity title provide a brief general description of the procedure and its purpose. Also, state any particular requirements, rules, conditions, etc.

8. State all specific steps comprising a major activity in concise action terms. Place all specific steps in proper sequence with a sequence number. Place each step on a separate line for ease in following.

9. Where differences in procedural steps are required because of variations in equipment, etc., provide complete sets of alternative steps with the condition(s) clearly stated.

10. Follow the same rule for substeps as stated in item 8 but identify them by letters.

11. Indent step level activities in from the major activity title. Indent substeps in from the step level activities.

12. When documenting procedures requiring variable entries on the console terminal, always provide a prototype for the entry, and give an example immediately after the prototype entry.

13. Express the variables in the prototypes by using a consistent convention such as brackets and upper case letters, e.g.: login [LOGIN NAME].

14. Provide NOTES as necessary to explain the derivation of variable entries. These notes follow the stated procedures and are to be inserted <u>before</u> the examples. The expression "(SEE NOTE 2)", for example, will be provided on the same line of the procedure or entry to which the NOTE applies.

15. Include needed tables for the derivation of standard names and codes within the document.

16. When preparing tables, always use the left-most column for the item description or the familiar term that any user would identify. Then provide the associated standard code, abbreviation, or name to be used in column(s) to the right of the description.

17. Keep the language simple, i.e., at about an eight grade reading level. Avoid cute expressions.

18. Don't assume the user to possess any more knowledge or skills than actually can be justified.

19. Avoid uncommon terms or abbreviations unless the user can truly be assumed to understand them. If use of a new term cannot be avoided, introduce it with a definition.

20. Avoid use of the passive voice.

21. Provide feedback statements where necessary. Enclose these in parentheses. For example: (Wait for the tape to rewind - about 3 minutes).

22. Avoid the user of words like "should", "might", "ought to", etc. Such words imply a degree of uncertainty and indecisiveness. Instead, use the active voice.

23. Minimize the requirement for the user to make decisions. When decisions cannot be eliminated, provide sufficient information to assist the user in making the correct decision(s).

24. Generally, use positive rather than negative statements unless a negative statement (e.g., a warning or caution) would be more effective.

The above guidelines are applicable to the preparation of all procedural documents as well as most other types of user documents.

Test Participants

On January 12, 1976, J. Walker and R. Zinke (9131), with the technical assistance and cooperation of Ms. I. Winheim (8234), tested the revised document now entitled Procedures for Installing UNIX on the PDP 11. The PDP 11/45 maintained by Dept. 8234 at Murray Hill was used. Ms. E. Yamin (8624) served as the test subject. Prior to the test Ms. Yamin was interviewed and was found to meet the qualifications assumed of a typical user. She was a programmer who was relatively inexperienced with UNIX and even though she had 11 years of experience with minicomputers, she was not familiar with the PDP 11 series.

TEST STRATEGY

The test subject was briefed concerning the purpose of the test and her role. She was given a copy of the revised document and a supplemental document containing specific test case information and hardware definitions. (The supplemental hardware information would not be given to real users since it is assumed that they would be familiar with their particular system configuration, etc.). The subject was closely observed as she followed the document and performed the procedures. All problems and difficulties encountered by the subject were noted. The performance time for each procedure was also recorded.

OBSERVATIONS/REVISIONS

This section describes each critical event observed during the test. Events not applicable to or of no consequence to the test per se are not included here. After the test J. Walker and R. Zinke analyzed the test results and made appropriate revisions to the document.

Observations	Revisions
Procedure 1: Copy System from Magtape to Disk.	
1. Subject needed to make typing corrections on console terminal.	Provided procedures for correcting errors at the point where the user first uses the terminal.
2. At the end of step 12 subject went to the console to perform next step while machine was still processing.	Added statement immediately after step 12 advising the user to wait until the tape stops moving.
Procedure 2: Make a Special File for the Root Device and Place the Correct Boot on Your Disk	
1. Discovered document errors in the examples for RK and HP devices.	Changed the major device numbers from 5 to 0 for RK and 0 to 5 for HP.
Procedure 3: Make Special Files for Block and Character Devices	
1. Because of the wording in the introductory statement the subject thought she was supposed to turn back to previous procedure instead of going to step 1 of procedure 3.	Eliminated the reference to previous procedure and provided clear direction to the correct step.

Observations	Revisions
Procedure 3 (cont'd.)	
2. Although a note is provided in the document for the variable "x" in ttyx, subject used "x" literally from the device table.	Changed the x in the table and in the note to [X] thus making it more obvious that the last character is a variable.
3. Although subject was not affected, Ms. Winheim noticed a lack of definition in the document for the last character of the major device name.	Clarified by additional statement that the last character is the same as the minor device number.
Procedure 4: Install New Users	No problem, no revisions.
Procedure 5: Identify User Terminals	
1. Initially this procedure could not be completed because of insufficient space on the RK disk. Ms. Winheim made "source" available.	Provided a precautionary statement in the "Introduction" section advising against the use of an RK disk, because of insufficient storage space.
2. Although not a direct result of the test, it was observed that the definition of the third character of the terminal ID needed further clarification.	Added a statement and an example specifying the need to increment the third character for each additional terminal of a higher speed.

Observations	Revisions
Procedure 6: Time Conversion	
1. Subject expressed difficulty interpreting the introductory paragraph and the specific entries required. After some prompting she was able to perform.	Changed the wording to reduce ambiguity. Changed the prototype names to more specific ones. Provided additional wording to make the example clear. Changed the procedure title to Time Zone Conversion.
Procedure 7: Set the Current Date	
1. It was discovered that the definition for the month numbers was incorrect.	Changed definition from "00-11" to "01-12".
Procedure 8: Prepare UNIX to Accommodate Non-Standard TTY's	
1. Subject expressed that she did not fully understand this procedure.	Changed wording. Added a prototype with entry items clearly identified.
2. The entry format examples provided in the document assumed upper- and lower-case terminals but the terminal being used was upper-case only.	Provided explanation with example for making entries on an upper-case only terminal.

Observations	Revisions
Procedure 8 (cont'd.)	
3. The document was not very clear on how to determine terminal speed and mode codes. References to the <u>UNIX Programmer's Manual</u> were not helpful to subject.	Included speed and mode code tables in this section for easy look-up.
4. The document did not explain how to determine the getty.c program number which is a required entry in this procedure.	Added a NOTE explaining that this number is the same as that determined in procedure 5.
(Procedure is to be retested)	
Procedure 9: <u>Open System to Multiple Users</u>	
1. Although it seemed clear enough, subject misread the statement on console switch settings and entered the code for single user.	Changed the procedure from a negative to a positive statement, i.e., "set switches to 000002."
Procedure 10: <u>Load and Execute UNIX from Disk</u>	
1. Subject was doing well and almost completed this procedure when system appeared to go down. Considering the time of day,	No document problems. No revisions.

Observations | Revisions

Procedure 10 (cont'd.)

testing was concluded. (It was later determined that the console terminal simply ran out of paper)

Additional Observations

A total of 2 hours and 10 minutes was consumed for the documentation test. Part of this time was used for restarting procedure 1 because the tape was not initially set at load point. Portions of other procedures were delayed while recovering from typing errors. The manipulations required to make "source" available (due to lack of RK disk storage space) required several minutes. There was a faulty LOAD ADRS switch which did not return to place and had to be lifted after each use.

Post Test Debriefing

Following the test a meeting was held to give the test subject an opportunity to express her opinions concerning the document. The following is a summary of her comments:

1. Thought the document was well written and easy to follow, although there were some troublesome spots.

2. Suggested the document be looked at for inconsistencies, i.e., use of upper- vs. lower-case letters, examples v. literals, use of parentheses, etc. (Interestingly, while reviewing the document to identify such inconsistencies, subject suddenly realized that there was a format "code" which was used consistently.)

3. Felt a need at times to obtain information used or found in a previous procedure.

4. Felt the need for some prior editing knowledge for making corrections on the console terminal.

5. Much of the given device information provided in the supplemental information document was not entirely meaningful since she was not familiar with the various hardware options.

6. If actually responsible for setting up UNIX on her own system (instead of in a test situation) she would have taken more time and care. How much time? She suggested 4 hours.

7. When asked to describe the level of the person who could use the document, she said any experienced computer operator.

Items for Consideration

The following suggestions are presented here for future consideration:

1. If feasible provide a separate document for each model PDP 11, i.e., 11/40, 11/45, 11/70. This would reduce the irrelevant

material which must be contained in a single document as well as reduce the number of decisions imposed upon the user.

2. Since it was discovered in the test that a RK disk lacks sufficient storage space, it would seem appropriate to eliminate the procedures for a RK disk in the new document.

3. The following sections were omitted from the revised document:

   Floating Point Option
   File System Health
   Odds and Ends

   Floating Point Option was deemed unnecessary since the delivered system has already been recompiled. The other two sections were thought to be superfluous to installing UNIX. They were eliminated with the recommendation that their contents be rewritten and placed in a more suitable document(s) dealing with system maintenance procedures.

4. With each document produced for UNIX provide a cover page with title, etc., and a table of contents.

Further Testing

Additional testing is recommended at a date to be set in the near future. Procedure 11 still needs to be tested, and procedure 8 will need to be retested because of the extensive revisions which were made.

Test Conclusions

The test revealed certain weaknesses in the new document. These weaknesses have been corrected. Two procedures are rescheduled for testing. Although several documentation problems existed, the subject's performance was very good. Her performance is good evidence that the new document, for the most part, is effective in communicating the required procedures to the intended user. Considering the relatively low criticality of the weaknesses found, and in view of the revisions made, no retesting is recommended except for procedure 8.

J Walker
J. T. Walker

R. A. Zinke
R. A. Zinke

PY-9131-JTW-gl
RAZ

Attachments

Appendix I
Appendix II

SETTING UP UNIX - Issue Three

I. MANIFEST

Enclosed are:

1. The UNIX software on magtape or disk pack.
2. "Regenerating System Software"
3. Your configuration.
4. Sysgen shell procedure used.
5. Proto file
6. conf.c
7. low.s
8. name list of your system

II. SYSTEM ON MAGTAPE

If your system is on magtape, perform the following bootstrap procedure to obtain a disk pack.

1. Mount magtape on drive 0 at load point.
2. Mount formatted disk pack on drive 0.
3. Key in and execute at 100000
   for a TU10 magtape drive.

```
     012700
     172526
     010040
     012740
     060003
     000777
```

   The tape should move and the CPU loop.
   (This is <u>not</u> the DEC bulk ROM for tape.)

OR

If your machine is an 11/45 or an 11/70
this can be shortened. (On the 11/70 there
are more console switches and the addresses
below are for the 11/45 machines. To convert
the addresses to 11/70 simply set the unspecified
high order switches to the up position.)

```
     For TJU16
Make sure the following addresses are zeroed
             772442
             772444
             772446
Halt and load address 772472
Set switches to 001300 and deposit
Load address 772440
```

```
   Set switches to 000071
   Enable and deposit (in that order)

        For TU10
   Halt and load address 772522
   Set switches to 060003
   Enable and deposit (in that order)


4. Halt and restart the CPU at 0.
   (Load address 000000 -
   set switches to 773030 to come up single user
   Enable and start)
   The tape should rewind.
   The console should type '='.
5. Copy the magtape to disk by the following:
        =                                                  (machine)
        list<                                              (you)(<=CR/LF)
        (This will list all the files on the
         front of the tape. The last line
         will look like:
            tape size = NN (decimal)
         where NN is a number in decimal.)
        =                                                  (mch)
        copy<                                              (you)
        p for RP; f for RF; k for RK; 4 for RP04           (mch)
        p                (you) (or k,f,Or 4 as the case may be)
        m for TM11; u for TU16; c for TC11                 (mch)
        m                (you) (or u or c as the case may be)
        disk offset<                                       (mch)
        0<                                                 (you)
        tape offset<                                       (mch)
        NN<              (you) (where NN is gotten from list)
                (The tape should move)
        count<                                             (mch)
        4000<            (you) (the tape moves more)
        =                                                  (mch)
6. At this point you have a UNIX system on disk.
   It is advisable to come up on the disk from
   the tape at this point. This is done in order
   to copy the correct boot program onto the front
   of the pack being used as the root. (Due to the
   size of the new boot programs a utility package
   containing the necessary boots would not fit
   on block zero.) Make sure the switches are set
   to 773030 and respond to the "=" with:

   hpboot<      OR      rkboot<         OR      rpboot<
   4                    k                       p
                (note CR echoed by mch)
   /unix<               /unix<                  /unix<

   There will be no prompts after the boot program
```

is typed. The only way to tell that you are on the right track is if the machine echoes a carriage return. If no CR is echoed the wrong character was typed and this boot will not work. If this is the case repeat the above booting procedure from the tape (steps 3, 4, and 6). UNIX is now running on the disk pack. You know this by the console message "UNIX Release 3". A special file must be made in directory /dev for the root device. Section VI (Special Files) describes this procedure. NOTE all the special files may be entered at this time or they may be entered at a later time. With this completed, copy the correct boot (resides in the directory /usr/mdec/tu) to the device with the cp-I command. For example use one of the following:

```
cp /usr/mdec/tu/hpboot /dev/hp0
cp /usr/mdec/tu/rkboot /dev/rk0
cp /usr/mdec/tu/rpboot /dev/rp0
```

for theRP04, RK04/05, and the RP03 respectively. The correct boot program is now on block zero of the root device and any future boots can come up directly on this device.

III. SYSTEM ON DISK

You now have a UNIX system on disk and have moved the proper boot program to the front of the root pack (see above). You may want to extract and execute UNIX from this pack, if so key in and execute the following program at location 100000.

```
  RP03          RK          RP04
 012700       012700
 176726       177414        (see
 005040       005040        ROM
 005040       005040        below)
 005040       010040
 010040       012740
 012740       000005
 000005       105710
 105710       002376
 002376       005007
 005007
```

NOTE: These programs correspond to the DEC bulk ROM for RP and RK respectively.

OR

On an 11/45 or an 11/70 these programs can be shortened. Note that the 11/70 addresses must be converted as they were with the tape.

```
For an RK
        Halt and load address 777404
        Set switches to 000005
        Enable and deposit (in that order)
        Halt and load address 000000
        Set switches to 773030 (single user)
        Enable and start
For an RP03
        Halt and load address 776716
        Deposit a non-zero number
        Halt and load address 776714
        Set switches to 000205
        Enable and deposit (in that order)
        Halt and load address 000000
        Set switches to 773030 (single user)
        Enable and start
For an RP04
        Halt and load address 765000
        Set switches to 000070
        Enable and start
For an RS04
        Halt and load address 765000
        Set switches to 000100
        Enable and start
```

The CPU should loop. If the device being booted is the RP04 or the RS04 and you wish to come up single user set the console switches to 773030 at this point. At this point type p for RP or k for RK or 4 for RP04 or s for RS04. A CR will be echoed. Now type /unix followed by a CR. After a delay, UNIX will come up by announcing

```
(SINGLE USER)                     (MULTI USER)

UNIX Release x                    UNIX Release x
mem = xxx                         mem = xxx
                                  :login:
```

The mem message gives user available memory in .1K units. The number should be 120 (for 12K) or larger for UNIX to fully support all of its software.

UNIX is now running. The "UNIX Programmer's Manual" now applies. (Below references to section X entitled Y of the programmer's manual are given (Y-X).) The only valid user names are root and bin. The root is the super-user and bin is the owner of nearly every file in the file system.

Before UNIX is turned up completely, a few configuration dependent exercises must be performed. First login (login-I) as root. (This is not necessary if the system was brought up in single user mode.)

IV. FLOATING POINT

If your machine does not have the 11/45 floating point option, (this includes all 11/40 systems) the following command was executed before your system was delivered.

```
        sh /usr/sys/conf/flrc
```

This sequence recompiled the C compiler and the debugger not to use floating point. It also installed the floating point interpreter in the standard library and reloaded basic, and fortran to have their floating point interpreted.

V. TIME CONVERSION

If your machine is not in the eastern time zone, you must edit (ed-I) the subroutine /usr/source/s4/ctime.c to reflect your local time. The variable timezone should be changed to reflect the time difference between local time and GMT. For EST, this is 5*60*60; for PST it would be 8*60*60. For example to convert to pacific time:

```
        ed /usr/source/s4/ctime.c
        /5\*/s//8*/
```

Three lines farther, is the name of the standard time zone. It should be changed to reflect local standard time. Thus continuing the above example

```
        .3s/\e/\p/
```

The next line is the daylight time zone name, if any.

```
        .1s//\p/
        w
        q
```

NOTE that these two names are in upper case and will have to be edited using escapes (ed-IV) as shown above. Two lines farther is the daylight flag. It has the value 1 which causes the time to shift to daylight savings automatically between the last Sundays in April and October. Normally this will not have to be reset. After ctime.c has been edited (make sure it is rewritten), execute the following command.

```
sh /usr/sys/conf/tmrc
```

This will recompile all the programs that use the local time conversion. This execution takes about 7 minutes, and requires that the floating point corrections have been installed if needed. If the editing of ctime.c was botched, there will be diagnostics from its compilation.

Set the current date (date-I).

VI. SPECIAL FILES

At this point, it would be wise to read all of the manuals and to augment this reading with hand to hand combat. It might be instructive to examine the shell command files for floating point and time conversions. The rest of the conversion assumes that sections I, IV, V, VII, and VIII of the programmer's manual have been read.

Next you must put all of the special files in the directory /dev using mknod - VIII. Print the configuration file:

```
cat /usr/sys/conf/conf.c
```

This is the major device switch of each device class (block and character). The block and character device configuration tables have been standardized by the UNIX Support Group. An entry for all common devices in both device classes has been made. If a device was not present in the system the entry in the bdevsw (block device switch) or cdevsw (character device switch) table would read:

```
&nodev,    &nodev,    &nodev,    &nodev,    &nodev,
```

The MAJOR device number for this entry would therefore not be associated with any specific file. The MAJOR device number is selected by counting the line number (from zero) of the device's entry in the device switch table, or, since conf.c has been standardized, by taking the number from the following table.

Block Devices

conf.c	TYPICAL NAME	MAJOR	device
rk	rk0	0	RK moving head disk
rp	rp0	1	RP moving head disk
rf	rf0	2	RS/RF fixed head disk
tm	mt0	3	TM/TU magtape
tc	tap0	4	TC/TU DECtape

Character Devices

conf.c	NAME	MAJOR	device	
kl	tty8	0	console tty	
pc	ppt	1	paper tape	
lp	lp	2	high speed line printer	
dc	ttyx	3	async serial line interface	
dh	ttyx	4	async serial line multiplexer	
dp		5	synchronous interface	
dj	ttyz	6	async serial line multiplexer	
dn		7	alternate console	
mem	mem	8	memory	
rk	rrk0	9	RK moving head disk	
rf	rrf0	10	RS/RF fixed head disk	for
rp	rrp0	11	RP moving head disk	physical
tm	rmt0	12	TM/TU magtape	I/O

Three devices have been added to the configuration table. These are RP04(hp), RS04(hs), and TU16(ht) where the parenthesized names are the ones that appear in CONF.C. If present in the configuration they would start at Block major device number 5 (five) and Character major device number 13 (thirteen). The major device numbers can not be gotten from the above table so they must be selected by counting the line number (from zero) in the device switch table.

The block special devices are entered into directory /dev first by executing the command:

```
                    /etc/mknod /dev/NAME b MAJOR MINOR
```

where NAME and MAJOR can be gotten from the preceeding table. The only exception being the TU16. If this device is being entered into /dev it should be called 'mt' not 'ht' as it is in the configuration tables. This is because the command TP(I) uses the pathname '/dev/mt' to execute. The MINOR device is the drive number, unit number, or partition number as described under each device in section IV of the programmer's manual. The last digit of the name (all given as 0 in the table above) should reflect the minor device number. For tapes where the unit is dial selectable, a special file may be made for each possible selection. If an RK, RP, TM, and TC were configured in the system the following special files might be created.

```
/etc/mknod /dev/rk0 b 0 0
/etc/mknod /dev/rk1 b 0 1
/etc/mknod /dev/rp0 b 1 0
/etc/mknod /dev/rp1 b 1 1
/etc/mknod /dev/mt0 b 3 0
          . . .
/etc/mknod /dev/mt7 b 3 7
/etc/mknod /dev/tap0 b 4 0
          . . .
/etc/mknod /dev/tap7 b 4 7
```

This sets up file rp0 and rp1 each as a half of one RP03 drive. Files rk0 and rk1 correspond to two RK03/RK05 drives. The tape files will handle up to eight handlers dialed to any unit.

The same goes for the character devices. Here the names are arbitrary except that devices meant to be used for teletype access should be named /dev/ttyX, where X is any character. The console is already in as:

```
/etc/mknod /dev/tty8 c 0 0
```

All 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. The open and close routines are the same as the standard ones for the device but there are special read and write routines. For example, to install raw magtape, the line

```
&tmopen,  &tmclose,  &tmread,  &tmwrite,  &nodev,
```

was added to the character device switch to make the appropriate special file for it (our name is "/dev/rmt0").

Another useful special file is "/dev/mem," which is configured in the system as distributed. The appropriate line to the character device switch is

```
&nulldev,  &nulldev,  &mmread,  &mmwrite,  &nodev,
```

and the name of the special file should be "/dev/mem." This file must exist for the ps command (I) to work.

VII. MOUNTED FILE SYSTEMS

If there are to be more file systems mounted than just the root, use mkfs-VIII to create the new file system and put its mounting in the file /etc/rc (see init-VII and mount-VIII). For example, to set up a second RK as a mounted file system.

```
/etc/mkfs /dev/rk1 4872
/etc/mount /dev/rk1 /mnt
ed /etc/rc
$a
/etc/mount /dev/rk1 /mnt
.
w
q
```

VIII. NEW USERS

Install new users by editing the password file /etc/passwd (passwd-V). You´ll have to make current directories for the new users in either /usr or /mnt and change their owners to the newly installed name. Login as each user to make sure the password file is correctly edited. For example:

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

This will make a new login entry for joe. His default current directory is /mnt/joe which has been created.

IX. MULTIPLE USERS

If UNIX is to support simultaneous access from more than just the console teletype, the file /etc/ttys has to be edited. Each line in this file describes a typewriter channel potentially available for logging-in. Each line has 3 characters: the first is a 0-or-1 flag which, if 1, causes the typewriter to have a login process for it and if 0 to be ignored. The second character is the last character of the typewriter name (E.g. "x" for "/dev/ttyx".) The third character is a digit which selects which of several programs is to be executed when a data connection is established on the line. The list is built into the init program

(VII) and currently the only program is /etc/getty and the only digit is 0. The /etc/ttys file as distributed contains lines for typewriters 0-9 and a-d but all except tty8 are turned off. For some historical reason tty8 is the name of the console typewriter. To add new typewriters 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 20 typewriters. Also note that if the special file is inaccessible when init tries to create a process for it, the system will thrash madly trying and retrying to open it.

Another problem occurs on dc interfaces that are wired to run at any of four speeds. Within /usr/sys/dmr/dc.c there is a table (dcrstab) containing speeds and two bits for hardware. There are four speeds flaged as 0,1,2,or 3 which should correspond to the hardware. The dc driver recieves a speed for a tty from a program called getty. The speed is a displacement into the dcrstab table. If the displacement falls on a zero slot an error condition is noted, otherwise the speed of the entry is used.

Getty cycles through speeds and login messages as specified in the itab table. The user depresses the interupt or the break key to tell getty to try the next speed.

The itab entry looks as follows:

```
     itab:
                itn300;tnmes;tn300
                itty37;ttymes;tty37
                0
```

This represents the initial speed and parity settings, the address of the login message, and the subsequent speed and parity settings respectively for 300 baud and tty model 37 terminals.

A tty entry is as follows:

```
     itty37:
                .byte   5,5                     /speed 5
                0,340                           /any parity, raw
```

The first two bytes represent the displacement into the dcrstab and dctstab tables in the dc driver. A 5, for example, is 150 baud speed 1. The second entry (0,340) is the bit setting for the parameters such as parity, raw mode, echo, as described in STTY(II) of the Reference Manual.

The entries in the dc driver are for selectable speeds 134.5, 150, 300, 1200 baud and correspond to a DC11-AG. Any other DC11 requires that the entries in the dcrstab and dctstab tables be changed.

Another common modification is to build a new getty for use with a "funny" terminal (for example a CRT). This is done by adding entries to the itab table as described above. For example to add a 1200 baud entry the following procedure could be used:

```
        chdir /usr/source/s1
        cp getty.s ngetty.s
        ed ngetty.s
        /itab/
        a
        it1200; tnmes; tn1200
        .
        /ttymes/
        i
        it1200:
                .byte 9,9
                0;340           /any parity, raw
        tn1200:
                .byte 9,9
                0;10310         /any parity, echo, no tab, del
        .
        w
        q
        as ngetty.s
        ld a.out
        mv a.out /etc/ngetty
```

There are now two getty files and init.c must be edited to show this. Within /usr/source/s1/init there is a list of getty file names and the number of getty files. To do this change ncom, and the table com:

```
define ncom equal to the number of getty files
        *com[ncom]
                add the new getty file name
Then compile init.c as
        cc -n -s init.c
        mv a.out /etc/init
```

The final file to edit is /etc/ttys to make sure all the ttys are enabled. Remember the third digit of each entry in the ttys file is the displacement into the com table of init.

It is possible to add or drop typewriters without rebooting the system by sending a "hangup" signal to the initialization process; init catches this signal and rereads the /etc/ttys file. It then terminates processes on those lines that are to be dropped and creates them for those that are to be added. It is careful not to disturb processes on lines that are not being changed. When this signal is sent init examines the console switches and obeys them as if the system was being rebooted; that is, if the switches indicate a single-user system, everyone is logged out, and conversely other switch settings cause a multiple-user system to be continued or started. The hangup signal is sent by executing the command

```
        kill -1 1
```

by the super-user. The "-1" indicates the hangup signal; the "1" is the process number of init. [Note: this whole feature is new and is not documented elsewhere. Init used to have the names of the typewriters built into it and had to be recompiled for each change in typewriter configuration.]

X. FILE SYSTEM HEALTH

Periodically (say every day or so) and always after a crash, you should check all the file systems for consistency (check-VIII). It is quite important to execute sync (VIII) before rebooting or taking the machine down. This is done automatically every 30 seconds by the update program (VIII) 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. It should be pointed out that our RP controller has failed three times, each time in such a way that all information on the disk was wiped out without any error status from the controller. Complete and incremental dumps are easily done with the dump command (VIII) but restoration of individual files is painful. Dumping of files by name is best done by tp (I) but the number of files is limited. Finally if there are enough drives entire disks can be copied using cp-I, or preferably with a program provided by you which uses raw I/O to copy a track or so at time.

XI. ODDS AND ENDS

The programs dump, check and df (source check.c, dump.c, and df.c all in /usr/source/s1) have to be changed to reflect your default mounted file system devices. Print the first few lines of these programs and the changes will be obvious.

The source that is provided on the file system is quite space consuming. You will want to probably dump it offline (with dump-VIII, tp-I or cp-I as above) and remove it. Care should be exercised in deleting any files in the following directories: /bin, /etc, /lib, /usr/bin, /usr/lib, /usr/games. Also the following files should not be deleted: /usr/fort/fc1, /usr/sys/unix, /dev/tty8.

The source of the system proper is kept in /usr/sys. Most of the subsystem source is kept in /usr/source. For a complete list of the file system, use check-VIII or du-I (with -l flag).

If there are any programs that are not part of your system and you wish to retain a tape of them, this can be arranged .

APPENDIX II

PROCEDURES FOR INSTALLING UNIX ON THE PDP11

Introduction

Use the following set of procedures when initially installing the UNIX software on your PDP11. Follow the procedures appropriate to your machine model and type of tape drive or disk as indicated below. These procedures assume that your machine has been powered up and has passed DEC diagnostic tests during installation.

First, copy the software from magnetic tape to a disk pack. It is suggested that you avoid using an RK disk pack because it is not large enough to store all of the UNIX software.

1. Copy System From Magtape to Disk

1. Mount the magtape on drive 0 at load point.
2. Mount a formatted disk pack on drive 0.

(Unformatted disk packs can be formatted using the DEC supplied software diagnostics.)

3. Set console control knob to CONS PHY position.
4. Set console control knob to DATA PATHS position.
5. Depress HALT switch.

6A. For a TU10 magtape drive on a PDP11/40, do the following:

a. Set console switches to starting address 100000.
b. Depress the LOAD ADRS switch.
c. Set console switches to:

```
012700 and raise DEP switch
172526        "
010040        "
012740        "
060003        "
000777 and raise DEP switch
```

d. Set switches again to starting address 100000.
e. Depress LOAD ADRS switch.
f. Raise ENABLE switch.
g. Depress START switch.

(The tape will move and the CPU will loop. This is **not** the DEC bulk ROM for tape.)

or

6B. For a TU10 magtape drive on a PDP11/45, do the following:

a. Set console switches to 772522.
b. Depress LOAD ADRS switch.
c. Set switches to 060003.
d. Raise ENABLE switch.
e. Raise DEP switch.

(The tape will move.)

or

6C. For a TU16 magtape drive and either machine model 11/45 or 11/70, do the following:

a. Set console switches to 772442.
b. Depress LOAD ADRS switch.
c. Set switches to 000000.
d. Raise DEP switch three times.
e. Depress HALT switch.
f. Set switches to 772472.
g. Depress LOAD ADRS switch.
h. Set switches to 001300.
i. Raise DEP switch.
j. Set switches to 772440.
k. Depress LOAD ADRS switch.
l. Set switches to 000071.
m. Raise ENABLE switch.
n. Raise DEP switch.

(The tape will move.)

7. Depress HALT switch.
8. Set switches to 000000.
9. Depress LOAD ADRS switch.
10. Raise ENABLE switch.
11. Depress START switch.

(The tape will rewind, and the character "=" will print on the console terminal.)

12. Now go to the console terminal. If you detect a typing error on the console terminal while entering information and before the carriage return is typed, you can make corrections as follows:

a. For an incorrect character discovered immediately, type the sharp-character "#" immediately after the character in error to erase it, retype the character correctly and continue making the entry.

b. For an incorrect character discovered within the line, type the at-character, "@" to erase all characters typed thus far and then retype the line.

Proceed as follows on the console terminal; be sure to press carriage return after each entry except where indicated by "<NO CR>".

```
[you]      list

[mach]     m (for TM11); c (for TC11); u (for TJU16)

       (Type of Tape Controller)

[you]      m    or     c    or      u              <NO CR>

[mach]     (The machine will list all the files on the front
       of the tape.  The last line will look like:
                tape size = NN [decimal].)
       =

[you]      copy

[mach]     p (for RP); f (for RF); k (for RK); 4 (for RJP04)

       (Type of Disk Controller)

[you]      p    or        f     or   k     or     4         <NO CR>

[mach]     m (for TM11); c (for TC11); u (for TJU16)

       (Type of Tape Controller)

[you]      m     or    c     or     u                      <NO CR>

[mach]     disk offset

[you]      0

[mach]     tape offset

[you]      (Type in the tape size number, NN which the
       machine gave you earlier.)

       (The tape will move.)

[mach]     count

[you]      4000

       (Wait until the tape stops moving - about three minutes.)
```

```
[mach]      =
```

(You now have a copy of the UNIX system on disk, however the correct boot program for your root device [disk] must be moved to the front of the disk pack.)

13. Set console switches to 773030.
14. Return to the console terminal and continue. Press carriage return after each entry except where indicated by "<NO CR>".

```
[you]       rpboot(for RP03) or rkboot(for RK05) or hpboot(for RP04 or RP03)

               (Type of Disk Drive)

[you]       p <NO CR> or k <NO CR> or 4 <NO CR>

[mach]      (Carriage returned)

[you]       /unix

[mach]      UNIX Release 3
          mem=NNN
```

(The above message indicates that UNIX is running on the disk pack. The number, NNN, indicates the amount of space available for user programs in units of memory blocks [64 bytes].)

2. Make A Special File For The Root Device and Place the Correct Boot on Your Disk

Now make a special file in directory "/dev" for the root device. Perform one of the following procedures according to the type of disk you have. Be sure to press the carriage return after each line you type.

For rp (RP03 disk on an RP11 controller)

```
/etc/mknod /dev/rp0  b  1  0
cp /usr/mdec/tu/rpboot /dev/rp0
```

or

For rk (RK05 disk on an RK11 controller)

```
/etc/mknod /dev/rk0  b  0  0
cp /usr/mdec/tu/rkboot /dev/rk0
```

or

For hp (RP04 or RP05 disk on an RJP04 controller)

```
/etc/mknod /dev/hp0  b  5  0
cp /usr/mdec/tu/hpboot /dev/hp0
```

(The correct boot program is now on block zero of the root device and any future boots can come up directly on this device. UNIX is now running on your machine. All UNIX documents and commands now apply.)

3. Make Special Files For Other Block And Character Devices

Make a special file for each device on your PDP11. Put all of the special files in the directory "/dev", using the procedures which follow the device table.

The following block and character device configuration tables have been standardized for UNIX:

BLOCK DEVICES	NAME	MAJOR DEVICE NUMBER
RK moving head disk (RK11/RK03 or RK05)	rk0	0
RP moving head disk (RP11/RP03)	rp[0]	1
RF/RS fixed head disk (RF11/RS11)	rf[0]	2
TM/TU magtape (TM11/TU10)	mt[0]	3
TC/TU DECtape (TC11/TU56)	tap[0]	4

CHARACTER DEVICES	NAME	MAJOR DEVICE NUMBER
Console Teletype (TTY)	tty3	0
Paper Tape (PC11)	ppt	1
High Speed Line Printer (LP11)	lp	2
Async Serial Line Interface (DL11)	tty[X]	3
Async Serial Line Multiplexer (DH11)	tty[X]	4
Synchronous Interface (DP11)	dp[0]	5
Async Serial Line Multiplexer (DJ11)	tty[X]	6
Alternate Console	tty[X]	7
Memory	mem	8
RK moving head disk (RK11/RK03 or RK05)	rrk[0]	9
RF/RS fixed head disk (RF11/RS03 or RS04)	rrf[0]	10
RP moving head (RP11/RP03)	rrp[0]	11
TM/TU magtape (TM11/TU10)	rmt[0]	12

For devices with the NAME ending in "[0]", "[0]" will be the same as the MINOR DEVICE NUMBER. For devices with the name "tty[X]", "[X]" may be any alpha or numeric character.

Although the following devices are not included in the standard table, use the names provided here.

DEVICE	NAME
RP04/RP05	hp0
RS03/RS04	hs0
TU16	mt0

1. For each block device in the table which matches your system configuration, make a special file using the following format:

```
/etc/mknod /dev/[NAME] b [MAJOR DEVICE NUMBER]  [MINOR DEVICE NUMBER]
```

   a. Use the NAME given in the table (See Note 1.).
   b. Use b to indicate <u>block</u> device.
   c. Use the MAJOR DEVICE NUMBER from the table (See Note 1.).
   d. Use 0 for the MINOR DEVICE NUMBER, unless there is more than one drive, unit, or subdevice (See Note 2).

   For example, if the block device is a RP03 moving head disk, enter the following:

```
/etc/mknod /dev/rp0 b 0 0
```

<u>Note 1</u>:

If your system configuration includes a device which is not listed in the table, use the appropriate NAME and the next number in sequence for the MAJOR DEVICE NUMBER. For example, if you have a block device which is other than the five listed in the table, the MAJOR DEVICE NUMBER would be 5, then 6, etc. The MAJOR DEVICE NUMBER for the next character device not listed in the table would be 13, then 14, etc.

<u>Note 2</u>:

The MINOR DEVICE NUMBER is used to designate the drive number, unit number, or subdevice number. If you use a single device as if it were two or more devices, then two or more separate special files must be created. The difference would be specified in the MINOR DEVICE NUMBER designation where the first one is specified as 0 and the second one is specified as 1, etc. The same thing goes for each tape which is dial selectable. A separate special file must be created for each possible dial selection, by using the same MAJOR DEVICE NUMBER designation but providing a different MINOR DEVICE NUMBER.

For example, if the block device is a second RP03 moving head disk, enter the following:

```
/etc/mknod /dev/rp1 b 0 1
```

2. For each character device in the list which matches your system configuration, prepare a similar entry as you did for the block devices. Use "c" to indicate character device. Notes 1 and 2 above also apply to the character devices.

   For example, if the character device is a paper tape, enter the following:

```
/etc/mknod /dev/ppt c 1 0
```

(See UNIX Programmer's Manual (IV) for more information on the special files.)

4. Install New Users

Every user (i.e. each distinct login name) requires an entry in the password file, "/etc/passwd". However, it is not necessary for each user to have a distinct initial working directory. The password file serves as a list of all possible UNIX system users. Be sure to include super users, i.e., those who are responsible for maintaining the system and who are privileged to manipulate system directories and files. (See UNIX Programmer's Manual SU(VIII).)

1. To edit the password file, use the following format for each new user:

```
ed /etc/passwd
$
```

(The last line of the password file will be printed. You can determine the new NUMERICAL USER ID by increasing the present NUMERICAL USER ID by one.)

```
a
[LOGIN NAME]::[NUMERICAL USER ID]::[INITIAL WORKING DIRECTORY]:
.
w
q
```

For example, if the login name is joe and he is the tenth user and the initial working directory /usr/joe is chosen, enter the following:

```
ed /etc/passwd
$a
joe::10::/usr/joe
.
w
q
```

2. Once the name of the initial working directory has been created in the password file, create the directory. To make the initial working directory, use the following format for each new user:

```
mkdir [INITIAL WORKING DIRECTORY]
chown [LOGIN NAME] [INITIAL WORKING DIRECTORY]
```

For example, if the login name is joe and the initial working directory /usr/joe is chosen, enter the following:

```
mkdir /usr/joe
chown joe /usr/joe
```

3. Check to be sure each new user is correctly established by logging in each user as follows:

```
login [LOGIN NAME]
ls -la
```

(The contents of the initial working directory, the symbols "." and "..", will be listed.)

5. Identify User Terminals

Edit the file "/etc/ttys" to include all terminals which are to have access to the UNIX system. Each line added to this file will describe a user terminal line potentially available for logging-in when the system is brought up multi-user. Be sure that a special file has already been created for each terminal device before editing the "/etc/ttys" file and use the following format:

```
ed /etc/ttys
$a
[THREE CHARACTERS]
.
w
q
```

Each line in the file has three characters:

First character:

Enter "1" to activate the terminal line. Enter "0" to leave the terminal line inactive.

Second character:

Enter the last character of the terminal special file NAME. This character must be the same as determined in section 3 when you created special files for devices (e.g. "5" for /dev/tty5).

Third character:

Enter the character which corresponds to the speed of the terminal -

0 for 300, 150, 110 baud
1 for 150 baud
2 for 1200 baud

(For each additional nonstandard terminal, increase the third character by one, e.g., for a 1800 baud terminal the third character would be a "3".)

This character indicates which program in "getty.c" is to be executed when a data connection is made on the line. Other entries can be added to "getty.c" to support other terminal types and speeds.

For example, to activate a 300 bps terminal on line 5, enter the following:

```
ed /etc/ttys
$a
150
.
w
q
```

Note:

If there are to be more than twenty (20) terminals, the "tabsize" in "init.c" will have to be increased and "init.c" will have to be recompiled. See UNIX Programmer's Manual CC(I) on the UNIX C compiler.

6. Time Zone Conversion

UNIX is already set for the eastern time zone. Skip this procedure if you are in the eastern time. If your machine is not located in the eastern time zone, make the location known to the system by editing the subroutine "/usr/source/s4/ctime.c" and recompiling all the programs that use the local time. To do so, change the time zone, the standard zone name and the daylight zone name; and execute the shell. The time zone is the difference between the local time and Greenwich Mean Time.

1. To make the conversion, use the following format:

```
ed/usr/source/s4/ctime.c
/5\*/s//[TIME ZONE]
.3s/E/[STANDARD ZONE NAME]/
.1s//[DAYLIGHT ZONE NAME]/
w
q
```

For example, to convert from Eastern Standard Time (EST) which is 5*60*60 (the difference from Greenwich Mean Time)to Pacific Standard Time (PST) which is an additional three hours or 8*60*60, enter the following:

```
ed /usr/source/s4/ctime.c
/5\*/s//8*/
.3s/E/P/
.1s//P/
w
q
```

(The time conversion has now been made.)

2. To recompile all the programs that use the local time conversion, enter the following command:

```
sh /usr/sys/conf/tmrc
```

(The execution takes about seven minutes.)

7. Set The Current Date

After making the time conversion, set the current date. To set the current date, use the following format:

```
date [MMDDHHMM[YY]],
```

where MM is the month number (01-12); DD is the day number (01-31); HH is the hour number (00-23); MM is the minute number (00-59); and YY is the last two digits of the year number. YY is optional and the current year is the default if no year is given. If no argument is given, the current date is printed to the second.

For example, to set the date for December 2, 1975,12:45, enter the following command:

```
date 12021245
```

You may verify that the date has been set by calling "date" as follows:

```
date
```

(The console terminal should return the date specified by you.)

8. Prepare UNIX To Support Non-Standard TTY Terminals

UNIX supports ASCII terminals such as the GE Terminet 300, the Memorex 1240, Execuport, TI, TTY37, and most display terminals. The existing software is set to support terminals with only the following characteristics:

speeds of 110, 150, 300, 1200 baud
full duplex
upper and lower case characters

If your system is to support terminals with other speeds or mode characteristics you must define the speed for each terminal and add the appropriate mode characteristics to the "speeds" and "itab" tables in "/usr/source/s1/getty.c" using the procedures which follow the mode and speed tables below.

The following mode and speed tables have been standardized for UNIX:

MODE CHARACTERISTICS	MODE CODE
	HUPCL
Echo and print tabs as spaces	XTABS
Map upper case to lower on input	LCASE
Echo (full duplex)	ECHO
Map carriage return into line feed	CRMOD
Raw mode	RAW
Odd parity allowed on input	ODDP
Even parity allowed on input	EVENP
Odd and Even parity allowed on input	ANYP
	CR1
	CR2
Select one of four algorithms for carriage return delays	CR3
	NL1
	NL2
Select one of four algorithms for new line delays	NL3
	TAB1
	TAB2
Select one of four algorithms for tab delays	TAB3
Select one of two algorithms for form feed and vertical tab delays	FF1

STANDARD TERMINAL SPEEDS	SPEED CODES
Hang Up Dataphone	0
50 baud	1
75 baud	2
110 baud	3
134.5 baud	4
150 baud	5
200 baud	6
300 baud	7
600 baud	8
1200 baud	9
1800 baud	10
2400 baud	11
4800 baud	12
9600 baud	13
External A	14
External B	15

1. To support a non-standard terminal use the following format:

```
chdir /usr/source/s1
ed getty.c
/speeds/
```

(For console terminals with only upper-case characters, enter a backslash character immediately before each special and upper-case character in this procedure; e.g., ANYP would be entered as ANYP, etc.)

```
/B1200/
a
define  B[SPEED]  [SPEED CODE]
.
/};/
i
/* table ´[TABLE NUMBER - SEE NOTE 1]´ -- [SPEED] */
´[TABLE NUMBER]´, ´[TABLE NUMBER]´,
ANYP+RAW+NL1+CR1, [MODE CODES - SEE NOTE 2],
B[INPUT SPEED], B[OUTPUT SPEED],
"\n\r\033;login: ",
.
w
q
mv /etc/getty /etc/ogetty
cc -s -o -n getty.c
mv a.out /etc/getty
```

<u>Note</u> <u>1</u>:

This number corresponds to the getty.c program to be executed when a data connection is made on the line. The character is the same as that determined as the third character in Procedures 5.

<u>Note</u> <u>2</u>:

Enter the mode codes that describe your terminal joining each with a plus sign, "+". Use the standard "defines" listed in the mode characteristics table above.

For example, if the non-standard terminal is a 9600 baud device (e.g. a CRT), enter the following on your console terminal:

```
chdir /usr/source/s1
ed getty.c
/speeds/
/B1200/
a
#define  B9600   13
.
/};/
i
/* table '3' -- 9600 */
'3', '3',
ANYP+RAW+NL1+CR1, [MODE CODES],
B9600, B9600,
"\n\r\033;login: ",
.
w
q
mv /etc/getty /etc/ogetty
cc -s -o -n getty.c
mv a.out /etc/getty
```

2. Edit the file "/etc/ttys" to include the nonstandard terminal using the following format:

```
ed /etc/ttys
a
[THREE CHARACTERS]
.
w
q
```

where the first character may activate the terminal line, the second character is the same as the last character of the special file NAME, and the third character is the TABLE NUMBER which indicates the program within "getty.c" to be executed.

For example to include a 9600 baud terminal, enter the following:

```
ed /etc/ttys
a
1q3
.
w
q
```

9. Open System To Multiple Users

1. Set console switches to 000002.
2. Return to the console terminal and press the "CONTROL" key and the letter "D" at the same time.

(The standard "login:" message will appear after several seconds indicating that the UNIX system is ready for general use.)

10. Load and Execute UNIX from Disk

Use the following procedures only when the UNIX file system needs to be replaced in core from a disk already containing the proper file system, boot program, special files, etc. These procedures assume that your machine has been powered up using the DEC procedures for your particular PDP11.

1. Turn console control knob to CONS PHY position.
2. Turn console control knob to DATA PATHS position.
3. Depress HALT switch.

4A. For an RP03 disk on a PDP11/40, do the following:

- a. Set console switches to starting address 100000.
- b. Depress HALT switch.
- c. Set console switches to:

```
012700 and raise DEP switch.
176726     "
005040     "
005040     "
005040     "
010040     "
012740     "
000005     "
105710     "
002376     "
005007 and raise DEP switch.
```

(This program corresponds to the DEC ROM.)

- d. Set switches again to the starting address 100000.
- e. Depress LOAD ADRS switch.
- f. Raise ENABLE switch.
- g. Depress START switch.

4B. For an RK disk on a PDP11/40, do the following:

a. Set console switches to starting address 100000.
b. Depress LOAD ADRS switch.
c. Set console switches to :

```
012700 and raise DEP switch.
177414      "
005040      "
005040      "
010040      "
012740      "
000005      "
105710      "
002376      "
005007 and raise DEP switch.
```

(This program corresponds to the DEC bulk ROM.)

d. Set switches again to starting address 100000.
e. Depress LOAD ADRS switch.
f. Raise ENABLE switch.
g. Depress START switch.

4C. For an RK disk on a PDP11/45 , do the following:

a. Set console switches to address 777404.
b. Depress LOAD ADRS switch.
c. Set switches to 000005.
d. Raise ENABLE switch.
e. Raise DEP switch.
f. Depress HALT switch.
g. Set switches to address 000000.
h. Depress LOAD ADRS switch.
i. Raise ENABLE switch.
j. Depress START switch.

4D. For an RP03 disk on a PDP11/45 , do the following:

a. Set console switches to address 776716.
b. Depress LOAD ADRS switch.
c. Set switches to 000001.
d. Raise DEP switch.
e. Depress HALT switch.
f. Set switches to 776714.
g. Depress LOAD ADRS switch.
h. Set switches to 000205.
i. Raise ENABLE switch.
j. Raise DEP switch.
k. Depress HALT switch.
l. Set switches to address 000000.
m. Depress LOAD ADRS switch.
n. Raise ENABLE switch.
o. Depress START switch.

4E. For an RP04 disk on a PDP11/70, do the following:

a. Set console switches to address 765000.
b. Depress LOAD ADRS switch.

c. Set switches to 000070.
d. Raise ENABLE switch.
e. Depress START switch.

4F. For an RS04 disk on a PDP11/70, do the following:

a. Set console switches to 765000.
b. Depress LOAD ADRS switch.
c. Set switches to 000100.
d. Raise the ENABLE switch.
e. Depress the START switch.

(The CPU will loop.)

5. Set console switches to 773030.
6. Return to console terminal and enter the following; be sure to press carriage return after each entry:

[you] p (for RP); or k (for RK); or 4 (for RP04); or s (for RS04)

(Type of Disk Drive)

[mach] (Carriage returned by machine)

[you] /unix

[mach] UNIX Release x
mem = xxx

(The above message indicates that the loading of the UNIX system from disk has been completed.)

7. At this time , check to see that the block device file systems are intact. This is particularly important if you are reloading the system after software or hardware failures. To perform the check use the "check" command and the following format for each file:

check /dev/[NAME]

For example , if the block device is a RP03 moving head disk, enter the following:

check /dev/rp0

(A report of each file system checked will be printed out, and will contain appropriate information about the file system as well as diagnostics if something is wrong.) (See UNIX Programmer's Manual CHECK (VIII).)

8. When you are satisfied with the file system checks, do the following:

   a. Set console switches to 000002.

   b. Return to the console terminal and press the "CONTROL" key and the letter "D" at the same time.

(The standard ";login:" message will appear after several seconds indicating that the UNIX file system is ready for general use.)

11. Create and Mount Additional File Systems

File systems in addition to the "root" file system may be created using the "/etc/mkfs" program and mounted using "/etc/mount". To create a new file system and mount it on the file "/etc/rc", use the following format:

```
/etc/mkfs /dev/[NAME] [NUMBER OF BLOCKS ON THE DEVICE]
/etc/mount /dev/[NAME] /usr
ed /etc/rc
$a
/etc/mount /dev/[NAME] /usr
.
w
q
```

For example, if a second RP moving head disk is set up as a mounted file system, enter the following:

```
/etc/mkfs /dev/rp1 4872
/etc/mount /dev/rp1 /usr
ed /etc/rc
$a
/etc/mount /dev/rp1 /usr
.
w
q
```

(An additional file system has now been created and mounted.)