NAME

boot procedures – UNIX startup

DESCRIPTION

The advent of the new system has changed the boot procedures. *These procedures apply only to C-language systems.*

How to start UNIX. UNIX is started by placing it in core starting at location zero and transferring to zero. There are various ways to do this. If UNIX is still intact after it has been running, the most obvious method is simply to transfer to zero.

The *tp* command places a bootstrap program on the otherwise unused block zero of the tape. The DECtape version of this program is called *tboot,* the magtape version *mboot.* If *tboot* or *mboot* is read into location zero and executed there, it will type '=' on the console, read in a *tp* entry name, load that entry into core, and transfer to zero. Thus the next easiest way to run UNIX is to maintain the UNIX code on a tape using *tp.* Then when a boot is required, execute (somehow) a program which reads in and jumps to the first block of the tape. In response to the '=' prompt, type the entry name of the system on the tape (we use plain 'unix'). It is strongly recommended that a current version of the system be maintained in this way, even if the first or third methods of booting the system are usually used.

The standard DEC ROM which loads DECtape is sufficient to read in *tboot,* but the magtape ROM loads block one, not zero. If no suitable ROM is available, magtape and DECtape programs are presented below which may be manually placed in core and executed.

A third method of rebooting the system involves the otherwise unused block zero of each UNIX file system. The single-block program *uboot* will read a UNIX pathname from the console, find the corresponding file on a device, load that file into core location zero, and transfer to it. The current version of this boot program reads a single character (either **p** or **k** for RP or RK, both drive 0) to specify which device is to be searched. *Uboot* operates under very severe space constraints. It supplies no prompts, except that it echos a carriage return and line feed after the **p** or **k.** No diagnostic is provided if the indicated file cannot be found, nor is there any means of correcting typographical errors in the file name except to start the program over. *Uboot* can reside on any of the standard file systems or may be loaded from a *tp* tape as described above.

The standard DEC disk ROMs will load and execute *uboot* from block zero.

The switches. The console switches play an important role in the use and especially the booting of UNIX. During operation, the console switches are examined 60 times per second, and the contents of the address specified by the switches are displayed in the display register. (This is not true on the 11/40 since there is no display register on that machine.) If the switch address is even, the address is interpreted in kernel (system) space; if odd, the rounded-down address is interpreted in the current user space.

If any diagnostics are produced by the system, they are printed on the console only if the switches are non-zero. Thus it is wise to have a non-zero value in the switches at all times.

During the startup of the system, the *init* program (VIII) reads the switches and will come up single-user if the switches are set to 173030.

It is unwise to have a non-existent address in the switches. This causes a bus error in the system (displayed as 177777) at the rate of 60 times per second. If there is a transfer of more than 16ms duration on a device with a data rate faster than the bus error timeout (approx 10×s) then a permanent disk non-existent-memory error will occur.

ROM programs. Here are some programs which are suitable for installing in read-only memories, or for manual keying into core if no ROM is present. Each program is position-independent but should be placed well above location 0 so it will not be overwritten. Each reads a block from the beginning of a device into core location zero. The octal words constituting the program are listed on the left.

DECtape (drive 0) from endzone:

```
012700          mov     $tcba,r0
177346
010040          mov     r0,-(r0)        / use tc addr for wc
012710          mov     $3,(r0)         / read bn forward
000003
105710  1:      tstb    (r0)            / wait for ready
```

```
002376          bge     1b
112710          movb    $5,(r0)         / read (forward)
000005
000777          br      .               / loop; now halt and start at 0
```

DECtape (drive 0) with search:

```
012700  1:      mov     $tcba,r0
177346
010040          mov     r0,-(r0)        / use tc addr for wc
012740          mov     $4003,-(r0)     / read bn reverse
004003
005710  2:      tst     (r0)
002376          bge     2b              / wait for error
005760          tst     -2(r0)          / loop if not end zone
177776
002365          bge     1b
012710          mov     $3,(r0)         / read bn forward
000003
105710  2:      tstb    (r0)            / wait for ready
002376          bge     2b
112710          movb    $5,(r0)         / read (forward)
000005
105710  2:      tstb    (r0)            / wait for ready
002376          bge     2b
005007          clr     pc              / transfer to zero
```

Caution: both of these DECtape programs will (literally) blow a fuse if 2 drives are dialed to zero.

Magtape from load point:

```
012700          mov     $mtcma,r0
172526
010040          mov     r0,-(r0)        / usr mt addr for wc
012740          mov     $60003,-(r0)    / read 9-track
060003
000777          br      .               / loop; now halt and start at 0
```

RK (drive 0):

```
012700          mov     $rkmr,r0
177414
005040          clr     -(r0)
005040          clr     -(r0)
010040          mov     r0,-(r0)
012740          mov     $5,-(r0)
000005
105710  1:      tstb    (r0)
002376          bge     1b
005007          clr     pc
```

RP (drive 0)

```
012700          mov     $rpmr,r0
176726
005040          clr     -(r0)
005040          clr     -(r0)
005040          clr     -(r0)
010040          mov     r0,-(r0)
012740          mov     $5,-(r0)
000005
105710  1:      tstb    (r0)
002376          bge     1b
005007          clr     pc
```

FILES

/usr/sys/unix – UNIX code
/usr/mdec/mboot – *tp* magtape bootstrap
/usr/mdec/tboot – *tp* DECtape bootstrap
/usr/mdec/uboot – file system bootstrap

SEE ALSO

tp(I), init(VII)