BOOT PROCEDURES(VIII) BOOT PROCEDURES(VIII)

NAME

boot procedures – MERT startup

DESCRIPTION

The advent of the 11/70 and its associated peripherals has changed the boot procedures. The following procedures apply only to C-language systems.

How to start MERT. MERT is started by placing it in core starting at location zero and transferring to zero. There are various ways to do this.

The *tp*(I) command places a bootstrap program on the otherwise unused block zero of the tape. The DECtape version of this boot program is called *tboot;* the magnetic tape version is called *mboot.* If *tboot* or *mboot* is read into location zero and executed there, it will type '=' on the console, read a *tp*(I) entry name, load that entry into core, and transfer to zero. Thus one way to start MERT is to maintain the UNIX/MERT code on a tape using *tp*(I). (Caution: the file /usr/mdec/tboot (DECtape) or /usr/mdec/mboot (magtape) must be present when the tape is made or updated.) Booting is then accomplished by executing a program which reads in and jumps to the first block of the tape. 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. In response to the '=' prompt, type the entry name of the system on the tape ('unix' is suggested) It is strongly recommended that a current version of the system be maintained in this way, even if another method of booting the system is usually used.

Another method of booting the system involves the otherwise unused block zero of each UNIX/MERT file system. One of four separate boot programs can be used in this method. The single-block program *uboot* reads a single character (either **p** or **k** for RP03 or RK04/05, drive 0) to specify which device is to be searched. The other boot programs are called *rkboot, rpboot,* and *hpboot.* These programs are also one block long, and they also read one character from the console, but they are designed to search one device type (drive zero in all cases), not one of two. *Rkboot* accepts the character **k** and searches the RK04/05 disk. *Rpboot* accepts the character **p** and searches the RP03 disk. *Hpboot* accepts the character **4** and searches the RP04 disk. All four bootstrap programs will then read a MERT pathname from the console, find the corresponding file on the device, load that file into core location zero, and transfer to it. *Uboot, rkboot, rpboot,* and *hpboot* operate under very severe space constraints. They supply no prompts, except a carriage return and line feed that are echoed after the **p, k,** or **4.** 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. These four bootstrap programs can reside in block zero of the device(s) they are capable of searching, or they may be loaded from a *tp* tape as described above. The correct bootstrap program can be placed on block zero at system generation by the *mkfs* (VIII) command or it can be placed there any time after that with the *cp* (I) command.

The standard DEC disk ROMs will load and execute *uboot, rkboot, rpboot,* and *hpboot* from block zero of the device.

The switches. The console switches play an important role in the use and especially the booting of MERT. 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 data display register only if the data display select knob is set to DISPLAY REGISTER. 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. They should contain 173030 if the system is to come up in single-user mode. Anything else will result in multiuser mode; conventionally, 173020 is set into the switches.

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 ratė faster than the bus error timeout (approx 10μs), then a permanent disk non-existent-memory error will occur.

ROM programs. Below are given 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 physical device into core location zero. The octal words constituting the program are listed on the left.

DECtape (drive 0) without search:

```
    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 (TU10) 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
```

Magtape (TU16) from load point:

```
The numbers in parentheses are for an 11/70
Zero the following addresses
          772442 (1772442)
          772444 (1772444)
          772446 (1772446)
Halt and load address 772472 (1772472)
Set switches to 001300 (0001300)
Load address 772440 (1772440)
Set switches to 000071 (0000071)
Enable and deposit
```

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
```

RP03 (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
```

RP04 (drive 0):

For an 11/70:

```
Halt and load address 1765000
Set switches to 0000070
Enable and start
```

For an 11/45 with optional ROM:

Halt and load address 773350
Enable and start

FILES

/usr/mdec/mboot – *tp* magtape bootstrap
/usr/mdec/tboot – *tp* DECtape bootstrap
/usr/mdec/uboot – file system bootstrap
/usr/mdec/tu/rkboot – RK04/05 file system bootstrap
/usr/mdec/tu/rpboot – RP03 file system bootstrap
/usr/mdec/tu/hpboot – RP04 file system bootstrap

SEE ALSO

tp (I), init (VIII), mkfs (VIII)

BUGS

If the console driver gets overridden accidentally in single-user mode, there is a provision to refresh it by setting the console switches to 173050. This feature works only, if switches are set at boot time.