NAME

boot − from power on to the login prompt

DESCRIPTION

At power on the machine reads the first sector of the boot device into memory and executes it. This bootstrap code loads **/boot**, the Minix Boot Monitor. The monitor loads the kernel binaries from **/minix**, or the newest file in **/minix** if it is a directory.

The Minix system is now running, the different tasks initialize themselves and control is transferred to the last one, **init**.

Init is the grandparent of all Minix processes, it is responsible for starting login processes on each terminal, but first it runs **/etc/rc**.

/etc/rc checks the state of the system and starts daemons. First it sets the keyboard translation to the mapping in **/etc/keymap** if present, then it reads the time zone from **/etc/timeinfo** followed by a call to **readclock**(8) to set Minix time from the hardware clock. Next the file systems are checked if necessary and the **/usr** file system is mounted.

The system is now ready for multiuser startup, **/etc/rc** starts the **update**(8) and **cron**(8) daemons, and initializes the network services. **/etc/rc** finally recovers crashed editor buffers and cleans out the **tmp** directories.

Init reads **/etc/ttytab** and starts a **getty**(8) for each enabled terminal line to allow a user to log in.

BOOT ENVIRONMENT

Many features of the drivers inside the kernel are controlled by settings in the boot environment. The values of these variables are usually colon or comma separated numbers configuring the driver. **DPETH0 = 300:10** tells the ethernet driver to use I/O address 0x300, interrupt request 10, and the default memory address (0xD0000, values may be omitted) for the first ethernet board. (Note that IRQ 2 is redirected to IRQ 9 on AT's and PS/2's, so use 9 if a device is jumpered for 2.)

Variables that are special to both the monitor and the kernel are described in **monitor**(8). This section lists extra variables or variable settings:

hd = **at** | **bios** | **esdi** | **xt**

Choose the driver that is to be used for the hard disk, in order: IBM/AT (classic AT or newer IDE), BIOS (generic driver), ESDI (some PS/2's), or IBM/XT. By default the first of these drivers that is enabled is used. Most drivers are present in the kernel as distributed, but may be taken out by modifying **/usr/include/minix/config.h**. (An XT should always use the BIOS driver, not the XT driver, because BIOS calls are cheap on an XT. The XT driver can be used on AT machines with an old XT controller.) See **hd**(4).

dosd = **file** | **fat**

Choose the driver that is to be used as the "DOS virtual disk", either the "file" or the "FAT" driver. See **dosd**(4).

DPETH*n* = **on** | **off**

Turn an ethernet board on or off. The driver is by default in "sink" mode for all boards. The sink mode allows one to use the driver without an ethernet board installed. The driver will play /dev/null for that device, i.e. nothing comes in, and anything send out is dropped on the floor. If the board is turned on then the driver will use it to send out packets, if it is turned off then the driver will fail for that board.

DPETH*n* = *I/O-addr*:*irq*:*mem_addr*:*mem_size* (WD80x3)
DPETH*n* = *I/O-addr*:*irq*:**0** (NE2000)
DPETH*n* = *I/O-addr*:*irq*:*flags* (3c503)

Set the I/O address (hex), IRQ (decimal), memory address (hex), memory size (hex), or flags (hex) of the *n*-th ethernet board and turn it on. By default they are configured as 280:3:D0000 and 300:5:CC000 with the memory size set to 2000, 4000, or 8000 depending on the type of board found. For the Western Digital cards the IRQ must be what the board expects, but the memory address is programmed into the board by the driver. The SMC EtherEZ board, a WD8013 successor, has only 8K memory. This confuses the driver, so you need to explicitly specify the board size as being 2000. The memory address and size have no meaning for the Novell ethernet boards, but the address may be explicitly set to zero to indicate that the board **is** a Novell ethernet board. For the 3Com 3c503 the third parameter are flags, with the low bit indicates that the on-board tranceiver must be used if 0 (thin ethernet), or that an external tranceiver is used on the AUI port if set to 1. The IRQ is software settable, and must be specified as 2 (XT), 3, 4, 5, or 9 (AT). The memory address is set on the board by jumpers. The driver does not support I/O mode for the 3c503. (Note the little differences between board types. For the 8003/8013 and NE1000/NE2000 the IRQ is fixed and the memory address variable, for the 3c503 the IRQ is variable and the memory address is fixed, but need not be specified. Messy.)

DPETH*n***_EA** = *e0*:*e1*:*e2*:*e3*:*e4*:*e5*
: Set the ethernet address of the *n*-th ethernet board. The address is normally obtained from the ethernet board, so only in exceptional circumstances is this setting ever needed. (Use the address of the main server if you want a career change.)

AHA0 = *I/O-addr*:*bus-on*:*bus-off*:*tr-speed*
: Configure the Adaptec 154xA SCSI host adapter to use the given I/O address (hex), Bus-on time (decimal), Bus-off time (decimal) and transfer speed (hex). The default is 330:15:1:00. The default transfer speed is always 5.0 Mb/s (code 00) ignoring the jumper settings.

sd*n* = *target*,*lun*
: Program SCSI disk **sd***n* to have the given target and logical unit number. The target and lun of a tape or other SCSI device may be changed by setting the **sd***n* variable that would be used had it been a disk. So tape device st7 can be set to target 4, lun 1 with sd35=4,1.

MCD = *I/O-addr*:*irq*
: I/O address (hex) and IRQ (decimal) of the Mitsumi CD-ROM driver, by default 300:10.

dosd*n* = *file*
dosd*n* = *partition:file*
: Tells the DOS virtual disk driver for device **dosd***n* (*n* is 0 or 5) to use a given MS-DOS file as a disk. Depending on whether the "file" or the "FAT" driver is being used **dosd***n* is either set to a normal DOS file name, or to a Minix hard disk partition name followed by a simple DOS file name. See **dosd**(4).

TCP/IP CONFIGURATION

To use TCP/IP you have to compile a kernel with networking enabled, and unless you are running standalone you have to enable the ethernet driver. See the **DPETH***n* boot variable above. The driver supports these ethernet cards: Western Digital 8003, Western Digital 8013, SMC Elite Ultra 16, Novell NE1000 and NE2000, 3Com Etherlink II (3c503). Many newer variants of the WD8013, now under the SMC brand, are also supported.

You are likely to use TCP/IP in one of three situations:

Standalone with no connection to a network.

In a small network with no support from a "big" host.

Connected to a large network with address and name servers.

In each situation you need a different set of configuration files.

Standalone

If **rarpd** notices that there is no connection to the outside world then it will configure the default network to address **10.0.0.1**. You need to create one file, **/etc/hosts**, that should look like this (using the name "flotsam" as the example name of the machine):

```
127.0.0.1    localhost
10.0.0.1     flotsam
```

Small Network

In a network where the Minix machine can't obtain its IP address and name from a different host you need specify the ethernet address to host name translation in the **/etc/ethers** file for use by **rarpd**.

Suppose your machine is to be named "flotsam", and another machine in the network is named "jetsam". The file **/etc/ethers** should at least contain an entry for flotsam:

```
0:0:c0:a:77:23     flotsam
```

Use **hostaddr −e** to find out what the six octet ethernet address of a host is. Use the address as printed: lowercase hex digits, no leading zeros. The **/etc/hosts** file shows all IP addresses of the machines flotsam needs to know about, and the netmask of the 10.*.*.* network:

```
127.0.0.1    localhost
10.0.0.1     flotsam
10.0.0.2     jetsam
255.0.0.0    0.0.0.10.in-addr.arpa
```

Warning! Do not add ethernet addresses of diskless workstations to your ethers file. A Sun for instance has the stupid habit of booting from the first RARP server that answers, probably your Minix machine...

Large Network

In a network with a central network administration your machine's IP address and name are given by the RARP and name services of the special servers on the network. For a new machine you need to apply for an IP address and host name with your network administrator supplying the ethernet address of your machine. You don't need any configuration files now, the **irdpd** and **nonamed** daemons automatically find a router and a name server.

Note that no knowledge of the IP address or hostname of the Minix machine itself is necessary, it all comes from the RARP and name servers. A series of Minix machines can therefore set up identically. Even if you have no RARP or name servers you can still set them up identically if you list all the Minix hosts in the hosts and ethers files.

Netmasks

An important parameter of a network is the netmask. A netmask is a bit mask that you apply to IP addresses to find out if two IP addresses are on the same network. So if a network interface has address 172.16.102.65 with netmask 255.255.255.224 then the 32 addresses 172.16.102.64 through 172.16.102.95 are all on the same network as 172.16.102.65. The **rarpd** daemon figures out the netmask by repeatedly looking up the netmask corresponding to an interfaces' network address: ip-address AND current-netmask. Suppose for example that the address for an interface is 172.16.102.65. The initial netmask for this class B address is 255.255.0.0. So we look up the netmask associated with 172.16.0.0 (172.16.102.65 & 255.255.0.0). This lookup yields 255.255.255.0, so now look up 172.16.102.0 to find 255.255.255.224. Lastly we look up 172.16.102.64 to find 255.255.255.224 again so we use it as the netmask. These netmasks can be specified in **/etc/hosts** as follows:

```
255.255.255.0      0.0.16.172.in-addr.arpa
255.255.255.224    0.102.16.172.in-addr.arpa
255.255.255.224    64.102.16.172.in-addr.arpa
```

This should not be necessary on a large network that has name servers if the Network Administrators have done their jobs properly. Check with **ifconfig** if the netmask is as it should be. If not then try to find the netmask manually with **host** commands on the appropriate *.*.*.in-addr.arpa names. The initial netmask you start with is 255.0.0.0 for a class A network (0.*.*.* to 127.*.*.*), 255.255.0.0 for a class B network (128.*.*.* to 191.*.*.*) and 255.255.255.0 for class C networks (192.*.*.* to 223.*.*.*). If you've found the mistake then you can either try to convince your Network Administrator to fix it, or you can add the appropriate entries to /etc/hosts to please **rarpd**.

Simpler configuration tools

The **rarpd**, **irdpd** and **nonamed** daemons are complex little programs that try to obtain information about their surroundings automatically to tell the machine what its place in the network is. It should come as no surprise that there are simpler utilities to configure a machine. On a memory starved machine it may even be wise to configure a machine statically to get rid of the daemons. The first daemon, **rarpd**, can be replaced by:

ifconfig −h *host-IP-address* **−n** *netmask*

to set the IP address of the machine. Note that this is only necessary if there is no external RARP service. The second daemon **irdpd** can be replaced by setting a static route:

add_route −g *router-IP-address*

(if there is a router.) The last daemon, **nonamed**, can be replaced by an entry in **/etc/resolv.conf** that specifies an external name daemon:

nameserver *nameserver-IP-address*

Edit **/etc/rc** just above the point where the network daemons are started to add the calls to **ifconfig** and **add_route**. Remove the names of the daemons that no longer needed. Note that these changes undo all the efforts to make Minix TCP/IP autoconfigurable. Make very sure that all the IP addresses are correct, and that the IP address of your machine is unique. (Mistakenly using the address of a main server will make all other machines look at your machine, and will make all the users of all other machines look at you.)

FILES

/boot	Minix Boot Monitor.
/minix	Kernel image, or directory containing them.
/etc/rc	First of the system initialization files.
/etc/hosts	Name to IP address mapping.
/etc/ethers	Name to ethernet address mapping.
/etc/resolv.conf	Name resolver configuration.

SEE ALSO

monitor(8), **init**(8), **inet**(8), **loadkeys**(8), **readclock**(8), **fsck**(1), **update**(8), **cron**(8), **ttytab**(5), **getty**(8), **hostaddr**(1), **ifconfig**(8), **irdpd**(8), **nonamed**(8), **rarpd**(8), **hosts**(5), **ethers**(5), **resolv.conf**(5), **netdefault**(8).

DIAGNOSTICS

Checking File Systems.

If the system has crashed then **fsck** is called for the root and /usr file systems. It is wise to reboot if the root file system must be fixed.

Finish the name of device to mount as /usr: /dev/

If the name of the /usr file system has not been set in /etc/fstab. You can type a device name, say **fd0**.

hostaddr: unable to fetch IP address

TCP/IP misconfiguration. The RARP may have failed because the ethernet address of the machine is not entered in either the remote or the local ethers file. Either talk to your Network Administrator, or make an ethers and a hosts file.

1.2.3.4 login:

If you see an IP address instead of a host name then the system failed to translate the IP address. Either talk to your Network Administrator to have the reverse address translation tables fixed, or make a hosts file.

NOTES

The 10.*.*.* network can be used for private networks. Next to this class A network there are 16 private class B networks, 172.16.*.* to 172.31.*.*, and 256 private class C networks, 192.168.0.* to 192.168.255.*. RFC-1597 will tell you why they are good, and RFC-1627 why they are bad.

BUGS

Indefinite hangs are possible if I/O addresses or IRQ's are wrong. A driver may babble about addresses and IRQ's, but that does not mean that what it says is true, it may just be configured that way. It is very difficult to find peripherals on a PC automatically, and Minix doesn't even try.

Rarpd should learn to also speak the BOOTP and DHCP protocols.

AUTHOR

Kees J. Bot (kjb@cs.vu.nl)