

**NAME**

**boot** - system bootstrapping procedures

**DESCRIPTION**

**Power fail and crash recovery.** Normally, the system will reboot itself at power-up or after crashes. An automatic consistency check of the file systems will be performed, and unless this fails, the system will resume multi-user operations.

**Cold starts.** Most i386 PCs attempt to boot first from floppy disk drive 0 (sometimes known as drive A:) and, failing that, from hard disk drive 0 (sometimes known as drive C:, or as drive 0x80 to the BIOS). Some BIOSes allow you to change this default sequence, and may also include a CD-ROM drive as a boot device.

Some newer PCs boot using UEFI firmware, not BIOS. That process is described in `uefi(8)`.

A three-stage bootstrap is employed. Control is passed from the boot blocks (bootstrap stages one and two) to a third-stage bootstrap program, `loader(8)`. This third stage provides more sophisticated control over the booting process than it is possible to achieve in the boot blocks, which are constrained by occupying limited fixed space on a given disk or slice.

The remainder of this subsection deals only with the boot blocks. The `loader(8)` program is documented separately.

After the boot blocks have been loaded, you should see a prompt similar to the following:

```
>> FreeBSD/x86 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

The automatic boot will attempt to load `/boot/loader` from partition 'a' of either the floppy or the hard disk. This boot may be aborted by typing any character on the keyboard at the 'boot:' prompt. At this time, the following input will be accepted:

- ? Give a short listing of the files in the root directory of the default boot device, as a hint about available boot files. (A ? may also be specified as the last segment of a path, in which case the listing will be of the relevant subdirectory.)

*bios\_drive:interface(unit,[slice,]part)filename [-aCcDdghmnPprsv] [-Sspeed]*  
Specify boot file and flags.

*bios\_drive*

The drive number as recognized by the BIOS. 0 for the first drive, 1 for the second drive, etc.

*interface*

The type of controller to boot from. Note that the controller is required to have BIOS support since the BIOS services are used to load the boot file image.

The supported interfaces are:

ad	ST506, IDE, ESDI, RLL disks on a WD100[2367] or lookalike controller
fd	5 1/4" or 3 1/2" High density floppies
da	SCSI disk on any supported SCSI controller

*unit* The unit number of the drive on the interface being used. 0 for the first drive, 1 for the second drive, etc.

*[slice,]part*

The partition letter inside the BSD portion of the disk. See `bsdlabel(8)`. By convention, only partition 'a' contains a bootable image. If sliced disks are used ("fdisk partitions"), any *slice* (1 for the first slice, 2 for the second slice, etc.) can be booted from, with the default (if not specified) being the active slice or, otherwise, the first FreeBSD slice. If *slice* is specified as 0, the first FreeBSD slice (also known as "compatibility" slice) is booted from.

*filename*

The pathname of the file to boot (relative to the root directory on the specified partition). Defaults to `/boot/kernel/kernel`. Symbolic links are not supported (hard links are).

**[-aCcDdghmnPpqrsv] [-Sspeed]**

Boot flags:

<b>-a</b>	during kernel initialization, ask for the device to mount as the root file system.
<b>-C</b>	try to mount root file system from a CD-ROM.
<b>-c</b>	this flag is currently a no-op.
<b>-D</b>	boot with the dual console configuration. In the single configuration, the console will be either the internal display or the serial port, depending on the state of the <b>-h</b> option below. In the dual console configuration, both the internal display and the serial port will become the console at the same time, regardless of the state of the <b>-h</b> option.

- d** enter the DDB kernel debugger (see `ddb(4)`) as early as possible in kernel initialization.
- g** use the GDB remote debugging protocol.
- h** force the serial console. For instance, if you boot from the internal console, you can use the **-h** option to force the kernel to use the serial port as its console device.
- m** mute the console to suppress all kernel console input and output during the boot.
- n** ignore key press to interrupt boot before `loader(8)` is invoked.
- P** probe the keyboard. If no keyboard is found, the **-D** and **-h** options are automatically set.
- p** pause after each attached device during the device probing phase.
- q** be quiet, do not write anything to the console unless automatic boot fails or is disabled. This option only affects second-stage bootstrap, to prevent next stages from writing to the console use in combination with the **-m** option.
- r** use the statically configured default for the device containing the root file system (see `config(8)`). Normally, the root file system is on the device that the kernel was loaded from.
- s** boot into single-user mode; if the console is marked as "insecure" (see `ttys(5)`), the root password must be entered.
- Sspeed**  
set the speed of the serial console to *speed*. The default is 115200 unless it has been overridden by setting `BOOT_COMCONSOLE_SPEED` in `make.conf(5)` and recompiling and reinstalling the boot blocks.
- v** be verbose during device probing (and later).

Use the `/boot.config` file to set the default configuration options for the boot block code. See `boot.config(5)` for more information about the `/boot.config` file.

## FILES

`/boot.config`  
parameters for the boot blocks (optional)

`/boot/boot1` first stage bootstrap file

`/boot/boot2` second stage bootstrap file

`/boot/loader` third stage bootstrap

`/boot/kernel/kernel`  
default kernel

`/boot/kernel.old/kernel`  
typical non-default kernel (optional)

## DIAGNOSTICS

When disk-related errors occur, these are reported by the second-stage bootstrap using the same error

codes returned by the BIOS, for example "Disk error 0x1 (lba=0x12345678)". Here is a partial list of these error codes:

- 0x1 Invalid argument
- 0x2 Address mark not found
- 0x4 Sector not found
- 0x8 DMA overrun
- 0x9 DMA attempt across 64K boundary
- 0xc Invalid media
- 0x10 Uncorrectable CRC/ECC error
- 0x20 Controller failure
- 0x40 Seek failed
- 0x80 Timeout

**NOTE:** On older machines, or otherwise where EDD support (disk packet interface support) is not available, all boot-related files and structures (including the kernel) that need to be accessed during the boot phase must reside on the disk at or below cylinder 1023 (as the BIOS understands the geometry). When a "Disk error 0x1" is reported by the second-stage bootstrap, it generally means that this requirement has not been adhered to.

## SEE ALSO

ddb(4), boot.config(5), make.conf(5), mount.conf(5), ttys(5), boot0cfg(8), btxld(8), config(8), efibootmgr(8), efivar(8), gpart(8), gptboot(8), gptzfsboot(8), halt(8), loader(8), nextboot(8), reboot(8), shutdown(8), uefi(8), zfsbootcfg(8)

## BUGS

The bsdlable format used by this version of BSD is quite different from that of other architectures.

Due to space constraints, the keyboard probe initiated by the **-P** option is simply a test that the BIOS has detected an "extended" keyboard. If an "XT/AT" keyboard (with no F11 and F12 keys, etc.) is attached, the probe will fail.