

NAME

fdisk - PC slice table maintenance utility

SYNOPSIS

fdisk [-**B**l*a*ipqstu] [-**b** *bootcode*] [-**1234**] [*disk*]

fdisk -**f** *configfile* [-**itv**] [*disk*]

PROLOGUE

In order for the BIOS to boot the kernel, certain conventions must be adhered to. Sector 0 of the disk must contain boot code, a slice table, and a magic number. BIOS slices can be used to break the disk up into several pieces. The BIOS brings in sector 0 and verifies the magic number. The sector 0 boot code then searches the slice table to determine which slice is marked "active". This boot code then brings in the bootstrap from the active slice and, if marked bootable, runs it. Under DOS, you can have one or more slices with one active. The DOS **fdisk** utility can be used to divide space on the disk into slices and set one active.

DESCRIPTION

This command is obsolete. Users are advised to use gpart(8) instead.

The FreeBSD utility, **fdisk**, serves a similar purpose to the DOS utility. The first form is used to display slice information or to interactively edit the slice table. The second is used to write a slice table using a *configfile*, and is designed to be used by other scripts/programs.

Options are:

-a Change the active slice only. Ignored if **-f** is given.

-b *bootcode*

Get the boot code from the file *bootcode*. Default is */boot/mbr*.

-B Reinitialize the boot code contained in sector 0 of the disk. Ignored if **-f** is given.

-f *configfile*

Set slice values using the file *configfile*. The *configfile* only modifies explicitly specified slices, unless **-i** is also given, in which case all existing slices are deleted (marked as "unused") before the *configfile* is read. The *configfile* can be '-', in which case standard input is read. See *CONFIGURATION FILE*, below, for file syntax.

WARNING: when **-f** is used, you are not asked if you really want to write the slices table (as you are in the interactive mode). Use with caution!

- i** Initialize sector 0 of the disk. Existing slice entries will be cleared (marked as unused) before editing. (Compare with **-u**.)
- I** Initialize sector 0 slice table for one FreeBSD slice covering the entire disk.
- p** Print a slice table in **fdisk** configuration file format and exit; see *CONFIGURATION FILE*, below.
- q** Be quiet. Benign warnings (such as "GEOM not found") are suppressed.
- s** Print summary information and exit.
- t** Test mode; do not write slice values. Generally used with the **-f** option to see what would be written to the slice table. Implies **-v**.
- u** Update (edit) the disk's sector 0 slice table. Ignored if **-f** is given.
- v** Be verbose. When **-f** is used, **fdisk** prints out the slice table that is written to the disk.
- 1234** Operate on a single slice table entry only. Ignored if **-f** is given.

The final disk name can be provided as a "bare" disk name only, e.g. *da0*, or as a full pathname. If omitted, **fdisk** tries to figure out the default disk device name from the mounted root device.

When called with no arguments, it prints the sector 0 slice table. An example follows:

```
***** Working on device /dev/ada0 *****
parameters extracted from in-core disklabel are:
cylinders=769 heads=15 sectors/track=33 (495 blks/cyl)

parameters to be used for BIOS calculations are:
cylinders=769 heads=15 sectors/track=33 (495 blks/cyl)

Warning: BIOS sector numbering starts with sector 1
Information from DOS bootblock is:
The data for partition 1 is:
sysid 165,(FreeBSD/NetBSD/386BSD)
start 495, size 380160 (185 Meg), flag 0
      beg: cyl 1/ sector 1/ head 0;
      end: cyl 768/ sector 33/ head 14
```

The data for partition 2 is:
 sysid 164,(unknown)
 start 378180, size 2475 (1 Meg), flag 0
 beg: cyl 764/ sector 1/ head 0;
 end: cyl 768/ sector 33/ head 14

The data for partition 3 is:
 <UNUSED>

The data for partition 4 is:
 sysid 99,(ISC UNIX, other System V/386, GNU HURD or Mach)
 start 380656, size 224234 (109 Meg), flag 80
 beg: cyl 769/ sector 2/ head 0;
 end: cyl 197/ sector 33/ head 14

The disk is divided into three slices that happen to fill the disk. The second slice overlaps the end of the first. (Used for debugging purposes.)

sysid is used to label the slice. FreeBSD reserves the magic number 165 decimal (A5 in hex).

start and *size* fields provide the start address and size of a slice in sectors.

flag 80 specifies that this is the active slice.

cyl, sector and *head* fields are used to specify the beginning and end addresses of the slice.

Note: these numbers are calculated using BIOS's understanding of the disk geometry and saved in the bootblock.

The **-i** and **-u** flags are used to indicate that the slice data is to be updated. Unless the **-f** option is also given, **fdisk** will enter a conversational mode. In this mode, no changes will be written to disk unless you explicitly tell **fdisk** to.

The **fdisk** utility will display each slice and ask whether you want to edit it. If you say yes, **fdisk** will step through each field, show you the old value, and ask you for a new one. When you are done with the slice, **fdisk** will display it and ask you whether it is correct. It will then proceed to the next entry.

Getting the *cyl*, *sector*, and *head* fields correct is tricky, so by default, they will be calculated for you; you can specify them if you choose to though.

After all the slices are processed, you are given the option to change the "active" slice. Finally, when all

the new data for sector 0 has been accumulated, you are asked to confirm whether you really want to rewrite it.

The difference between the **-u** and **-i** flags is that the **-u** flag edits (updates) the existing slice parameters while the **-i** flag is used to "initialize" them (old values will be ignored); if you edit the first slice, **-i** will also set it up to use the whole disk for FreeBSD and make it active.

NOTES

The automatic calculation of starting cylinder etc. uses a set of figures that represent what the BIOS thinks the geometry of the drive is. These figures are taken from the in-core disklabel by default, but **fdisk** initially gives you an opportunity to change them. This allows you to create a bootblock that can work with drives that use geometry translation under the BIOS.

If you hand craft your disk layout, please make sure that the FreeBSD slice starts on a cylinder boundary.

Editing an existing slice will most likely result in the loss of all data in that slice.

You should run **fdisk** interactively once or twice to see how it works. This is completely safe as long as you answer the last question in the negative. There are subtleties that **fdisk** detects that are not fully explained in this manual page.

CONFIGURATION FILE

When the **-f** option is given, a disk's slice table can be written using values from a *configfile*. The syntax of this file is very simple; each line is either a comment or a specification, as follows:

comment ...

Lines beginning with a **#** are comments and are ignored.

g *spec1 spec2 spec3*

Set the BIOS geometry used in slice calculations. There must be three values specified, with a letter preceding each number:

cnum Set the number of cylinders to *num*.

hnum Set the number of heads to *num*.

snum Set the number of sectors/track to *num*.

These specs can occur in any order, as the leading letter determines which value is which;

however, all three must be specified.

This line must occur before any lines that specify slice information.

It is an error if the following is not true:

```
1 <= number of cylinders
1 <= number of heads <= 256
1 <= number of sectors/track < 64
```

The number of cylinders should be less than or equal to 1024, but this is not enforced, although a warning will be printed. Note that bootable FreeBSD slices (the "/" file system) must lie completely within the first 1024 cylinders; if this is not true, booting may fail. Non-bootable slices do not have this restriction.

Example (all of these are equivalent), for a disk with 1019 cylinders, 39 heads, and 63 sectors:

```
g   c1019 h39  s63
g   h39   c1019 s63
g   s63   h39  c1019
```

p *slice type start length*

Set the slice given by *slice* (1-4) to type *type*, starting at sector *start* for *length* sectors. If the *start* or *length* is suffixed with a *K*, *M* or *G*, it is taken as a *Kilobyte*, *Megabyte* or *Gigabyte* measurement respectively. If the *start* is given as "*" it is set to the value of the previous partition end. If the *length* is given as "*" the partition end is set to the end of the disk.

Only those slices explicitly mentioned by these lines are modified; any slice not referenced by a **p** line will not be modified. However, if an invalid slice table is present, or the **-i** option is specified, all existing slice entries will be cleared (marked as unused), and these **p** lines will have to be used to explicitly set slice information. If multiple slices need to be set, multiple **p** lines must be specified; one for each slice.

These slice lines must occur after any geometry specification lines, if one is present.

The *type* is 165 for FreeBSD slices. Specifying a slice type of zero is the same as clearing the slice and marking it as unused; however, dummy values (such as "0") must still be specified for *start* and *length*.

Note: the start offset will be rounded upwards to a head boundary if necessary, and the end offset

will be rounded downwards to a cylinder boundary if necessary.

Example: to clear slice 4 and mark it as unused:

```
p 4 0 0 0
```

Example: to set slice 1 to a FreeBSD slice, starting at sector 1 for 2503871 sectors (note: these numbers will be rounded upwards and downwards to correspond to head and cylinder boundaries):

```
p 1 165 1 2503871
```

Example: to set slices 1, 2 and 4 to FreeBSD slices, the first being 2 Gigabytes, the second being 10 Gigabytes and the forth being the remainder of the disk (again, numbers will be rounded appropriately):

```
p 1 165 63 2G
p 2 165 * 10G
p 3 0 0 0
p 4 165 * *
```

a *slice*

Make *slice* the active slice. Can occur anywhere in the config file, but only one must be present.

Example: to make slice 1 the active slice:

```
a 1
```

FILES

/boot/mbr The default boot code.

SEE ALSO

boot0cfg(8), bsdlable(8), gpart(8), newfs(8)

HISTORY

A version of **fdisk** first appeared in the Mach Operating System. It was subsequently ported to 386BSD.

AUTHORS

fdisk for Mach Operating System was written by Robert Baron <rvb@cs.cmu.edu>. It was ported to 386BSD by Julian Elischer <julian@tfs.com>.

BUGS

The default boot code will not necessarily handle all slice types correctly, in particular those introduced since MS-DOS 6.x.

The entire utility should be made more user-friendly.

Most users new to FreeBSD do not understand the difference between "slice" and "partition", causing difficulty to adjust.

You cannot use this command to completely dedicate a disk to FreeBSD. The `bsdlabel(8)` command must be used for this.