

Name

fdisk - Partition table manipulator for Linux

Synopsis

fdisk [-uc] [-b *sectorsize*] [-C *cylns*] [-H *heads*] [-S *sects*] *device*

fdisk -l [-u] [*device...*]

fdisk -s *partition...*

fdisk -v

fdisk -h

Description

Hard disks can be divided into one or more logical disks called *partitions*. This division is described in the *partition table* found in sector 0 of the disk.

In the BSD world one talks about 'disk slices' and a 'disklabel'.

Linux needs at least one partition, namely for its root file system. It can use swap files and/or swap partitions, but the latter are more efficient. So, usually one will want a second Linux partition dedicated as swap partition. On Intel compatible hardware, the BIOS that boots the system can often only access the first 1024 cylinders of the disk. For this reason people with large disks often create a third partition,

just a few MB large, typically mounted on */boot*, to store the kernel image and a few auxiliary files needed at boot time, so as to make sure that this stuff is accessible to the BIOS. There may be reasons of security, ease of administration and backup, or testing, to use more than the minimum number of partitions.

fdisk (in the first form of invocation) is a menu driven program for creation and manipulation of partition tables. It understands DOS type partition tables and BSD or SUN type disklabels.

fdisk doesn't understand GUID Partition Table (GPT) and it is not designed for large partitions. In particular case use more advanced GNU ***parted***(8).

The *device* is usually */dev/sda*, */dev/sdb* or so. A device name refers to the entire disk. The old systems without libata (a library used inside the Linux kernel to support ATA host controllers and devices) make a difference between IDE and SCSI disks. In such a case the device name will be */dev/hd** (IDE) or */dev/sd** (SCSI).

The *partition* is a *device* name followed by a partition number. For example, ***/dev/sda1*** is the first partition on the first hard disk in the system. See also Linux kernel documentation (the *Documentation/devices.txt* file).

A BSD/SUN type disklabel can describe 8 partitions, the third of which should be a 'whole disk' partition. Do not start a partition that actually uses its first sector (like a swap partition) at cylinder 0, since that will destroy the disklabel.

An IRIX/SGI type disklabel can describe 16 partitions, the eleventh of which should be an entire 'volume' partition, while the ninth should be labeled 'volume header'. The volume header will also cover the

partition table, i.e., it starts at block zero and extends by default over five cylinders. The remaining space in the volume header may be used by header directory entries. No partitions may overlap with the volume header. Also do not change its type and make some file system on it, since you will lose the partition table. Use this type of label only when working with Linux on IRIX/SGI machines or IRIX/SGI disks under Linux.

A DOS type partition table can describe an unlimited number of partitions. In sector 0 there is room for the description of 4 partitions (called 'primary'). One of these may be an extended partition; this is a box holding logical partitions, with descriptors found in a linked list of sectors, each preceding the corresponding logical partitions. The four primary partitions, present or not, get numbers 1-4. Logical partitions start numbering from 5.

In a DOS type partition table the starting offset and the size of each partition is stored in two ways: as an absolute number of sectors (given in 32 bits) and as a Cylinders/Heads/Sectors triple (given in 10+8+6 bits). The former is OK - with 512-byte sectors this will work up to 2 TB. The latter has two different problems. First of all, these C/H/S fields can be filled only when the number of heads and the number of sectors per track are known. Secondly, even if we know what these numbers should be, the 24 bits that are available do not suffice. DOS uses C/H/S only, Windows uses both, Linux never uses C/H/S.

If possible, **fdisk** will obtain the disk geometry automatically. This is not necessarily the physical disk geometry (indeed, modern disks do not really have anything like a physical geometry, certainly not something that can be described in simplistic Cylinders/Heads/Sectors form), but is the disk geometry that MS-DOS uses for the partition table.

Usually all goes well by default, and there are no problems if Linux is the only system on the disk. However, if the disk has to be shared with other operating systems, it is often a good idea to let an fdisk from another operating system make at least one partition. When Linux boots it looks at the partition table, and tries to deduce what (fake) geometry is required for good cooperation with other systems.

Whenever a partition table is printed out, a consistency check is performed on the partition table entries. This check verifies that the physical and logical start and end points are identical, and that the partition starts and ends on a cylinder boundary (except for the first partition).

Some versions of MS-DOS create a first partition which does not begin on a cylinder boundary, but on sector 2 of the first cylinder. Partitions beginning in cylinder 1 cannot begin on a cylinder boundary, but this is unlikely to cause difficulty unless you have OS/2 on your machine.

A sync() and a BLKRRPART ioctl() (reread partition table from disk) are performed before exiting when the partition table has been updated. Long ago it used to be necessary to reboot after the use of fdisk. I do not think this is the case anymore - indeed, rebooting too quickly might cause loss of not-yet-written data. Note that both the kernel and the disk hardware may buffer data.

DOS 6.x WARNING

The DOS 6.x FORMAT command looks for some information in the first sector of the data area of the partition, and treats this information as more reliable than the information in the partition table. DOS FORMAT expects DOS FDISK to clear the first 512 bytes of the data area of a partition whenever a size change occurs. DOS

FORMAT will look at this extra information even if the /U flag is given -- we consider this a bug in DOS FORMAT and DOS FDISK.

The bottom line is that if you use cfdisk or fdisk to change the size of a DOS partition table entry, then you must also use **dd** to zero the first 512 bytes of that partition before using DOS FORMAT to format the partition. For example, if you were using cfdisk to make a DOS partition table entry for /dev/sda1, then (after exiting fdisk or cfdisk and rebooting Linux so that the partition table information is valid) you would use the command "dd if=/dev/zero of=/dev/sda1 bs=512 count=1" to zero the first 512 bytes of the partition.

BE EXTREMELY CAREFUL if you use the **dd** command, since a small typo can make all of the data on your disk useless.

For best results, you should always use an OS-specific partition table program. For example, you should make DOS partitions with the DOS FDISK program and Linux partitions with the Linux fdisk or Linux cfdisk program.

Options

-b *sectorsize*

Specify the sector size of the disk. Valid values are 512, 1024, 2048 or 4096. (Recent kernels know the sector size. Use this only on old kernels or to override the kernel's ideas.) Since util-linux-ng 2.17 fdisk differentiates between logical and physical sector size. This option changes both sector sizes to *sectorsize*.

-h

Print help and then exit.

-c

Switch off DOS-compatible mode. (Recommended)

-C *cyls*

Specify the number of cylinders of the disk. I have no idea why anybody would want to do so.

-H *heads*

Specify the number of heads of the disk. (Not the physical number, of course, but the number used for partition tables.) Reasonable values are 255 and 16.

-S *sects*

Specify the number of sectors per track of the disk. (Not the physical number, of course, but the number used for partition tables.) A reasonable value is 63.

-l

List the partition tables for the specified devices and then exit. If no devices are given, those mentioned in */proc/partitions* (if that exists) are used.

-u

When listing partition tables, give sizes in sectors instead of cylinders.

-s *partition*

The *size* of the partition (in blocks) is printed on the standard output.

-v

Print version number of **fdisk** program and exit.

Bugs

There are several *fdisk programs around. Each has its problems and strengths. Try them in the order **cfdisk**, **fdisk**, **sfdisk**. (Indeed, **cfdisk** is a beautiful program that has strict requirements on the partition tables it accepts, and produces high quality partition tables.

Use it if you can. **fdisk** is a buggy program that does fuzzy things - usually it happens to produce reasonable results. Its single advantage is that it has some support for BSD disk labels and other non-DOS partition tables. Avoid it if you can. **sfdisk** is for hackers only - the user interface is terrible, but it is more correct than fdisk and more powerful than both fdisk and cfdisk. Moreover, it can be used noninteractively.)

These days there also is **parted**. The cfdisk interface is nicer, but parted does much more: it not only resizes partitions, but also the filesystems that live in them.

The IRIX/SGI type disklabel is currently not supported by the kernel. Moreover, IRIX/SGI header directories are not fully supported yet.

The option 'dump partition table to file' is missing.

See Also

[***cfdisk***](#)(8), [***sfdisk***](#)(8), [***mkfs***](#)(8), [***parted***](#)(8), [***partprobe***](#)(8), [***kpartx***](#)(8)

Availability

The fdisk command is part of the util-linux-ng package and is available from <ftp://ftp.kernel.org/pub/linux/utils/util-linux-ng/>.

Referenced By

[***addpart***](#)(8), [***ddpt***](#)(8), [***delpart***](#)(8), [***ext2online***](#)(8), [***gparted***](#)(8), [***guestfs***](#)(3), [***lilo***](#)(8), [***mkswap***](#)(8), [***ntfsresize***](#)(8), [***partx***](#)(8), [***photorec***](#)(8), [***pvcreate***](#)(8), [***pvresize***](#)(8), [***resize2fs***](#)(8), [***sg_dd***](#)(8), [***sgm_dd***](#)(8), [***sgp_dd***](#)(8), [***syslinux***](#)(1), [***testdisk***](#)(8)