

## Linux Partition HOWTO

---

# 5. Partitioning with fdisk

This section shows you how to actually partition your hard drive with the **fdisk** utility. Linux allows only 4 primary partitions. You can have a much larger number of logical partitions by sub-dividing one of the primary partitions. Only one of the primary partitions can be sub-divided.

*Examples:*

1. Four primary partitions
2. Mixed primary and logical partitions

## 5.1. fdisk usage

**fdisk** is started by typing (as root) **fdisk device** at the command prompt. *device* might be something like `/dev/hda` or `/dev/sda` **fdisk** commands you need are:

**p** print the partition table

**n** create a new partition

**d** delete a partition

**q** quit without saving changes

**w** write the new partition table and exit

Changes you make to the partition table do not take effect until you issue the write (**w**) command. Here is a sample partition table:

Disk `/dev/hdb`: 64 heads, 63 sectors, 621 cylinders  
Units = cylinders of 4032 \* 512 bytes

Device	Boot	Start	End	Blocks	Id	System
<code>/dev/hdb1</code>	*	1	184	370912+	83	Linux
<code>/dev/hdb2</code>		185	368	370944	83	Linux
<code>/dev/hdb3</code>		369	552	370944	83	Linux
<code>/dev/hdb4</code>		553	621	139104	82	Linux swap

The first line shows the geometry of your hard drive. It may not be physically accurate, but you can accept it as though it were. The hard drive in this example is made of 32 double-sided platters with one head on each side

(probably not true). Each platter has 621 concentric tracks. A 3-dimensional track (the same track on all disks) is called a cylinder. Each track is divided into 63 sectors. Each sector contains 512 bytes of data. Therefore the block size in the partition table is  $64 \text{ heads} * 63 \text{ sectors} * 512 \text{ bytes}$  er...divided by 1024. for discussion on problems with this calculation.) The start and end values are cylinders.

## 5.2. Four primary partitions

*The overview:*

Decide on the size of your swap space and where it ought to go. Divide up the remaining space for the three other partitions.

Example:

I start fdisk from the shell prompt:

```
# fdisk /dev/hdb
```

which indicates that I am using the second drive on my IDE controller. When I print the (empty) partition table, I just get configuration information.

```
Command (m for help): p
```

```
Disk /dev/hdb: 64 heads, 63 sectors, 621 cylinders
Units = cylinders of 4032 * 512 bytes
```

I knew that I had a 1.2Gb drive, but now I really know:  $64 * 63 * 512 * 621 = 1281982464$  bytes. I decide to reserve 128Mb of that space for swap, leaving 1153982464. If I use one of my primary partitions for swap, that means I have three left for ext2 partitions. Divided equally, that makes for 384Mb per partition. Now I get to work.

```
Command (m for help): n
```

```
Command action
```

```
  e   extended
```

```
  p   primary partition (1-4)
```

```
p
```

```
Partition number (1-4): 1
```

```
First cylinder (1-621, default 1): <RETURN>
```

```
Using default value 1
```

```
Last cylinder or +size or +sizeM or +sizeK (1-621, default 621): +384M
```

Next, I set up the partition I want to use for swap:

```
Command (m for help): n
```

```
Command action
```

```
  e   extended
```

```
  p   primary partition (1-4)
```

```
p
Partition number (1-4): 2
First cylinder (197-621, default 197): <RETURN>
Using default value 197
Last cylinder or +size or +sizeM or +sizeK (197-621, default 621): +128M
```

Now the partition table looks like this:

Device	Boot	Start	End	Blocks	Id	System
/dev/hdb1		1	196	395104	83	Linux
/dev/hdb2		197	262	133056	83	Linux

I set up the remaining two partitions the same way I did the first. Finally, I make the first partition bootable:

```
Command (m for help): a
Partition number (1-4): 1
```

And I make the second partition of type swap:

```
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 82
Changed system type of partition 2 to 82 (Linux swap)
Command (m for help): p
```

The end result:

```
Disk /dev/hdb: 64 heads, 63 sectors, 621 cylinders
Units = cylinders of 4032 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hdb1	*	1	196	395104+	83	Linux
/dev/hdb2		197	262	133056	82	Linux swap
/dev/hdb3		263	458	395136	83	Linux
/dev/hdb4		459	621	328608	83	Linux

Finally, I issue the write command (w) to write the table on the disk.

Side topics:

- [Section 10.2](#)
- [Section 10.1](#)
- [Section 10.3](#)

## 5.3. Mixed primary and logical partitions

*The overview:* create one use one of the primary partitions to house all the extra partitions. Then create logical partitions within it. Create the other primary partitions before or after creating the logical partitions.

### Example:

I start fdisk from the shell prompt:

```
# fdisk /dev/sda
```

which indicates that I am using the first drive on my SCSI chain.

First I figure out how many partitions I want. I know my drive has a 183Gb capacity and I want 26Gb partitions (because I happen to have back-up tapes that are about that size).

$183\text{Gb} / 26\text{Gb} = \sim 7$

so I will need 7 partitions. Even though fdisk accepts partition sizes expressed in Mb and Kb, I decide to calculate the number of cylinders that will end up in each partition because fdisk reports start and stop points in cylinders. I see when I enter fdisk that I have 22800 cylinders.

```
> The number of cylinders for this disk is set to 22800.  There is  
> nothing wrong with that, but this is larger than 1024, and could in  
> certain setups cause problems with: 1) software that runs at boot  
> time (e.g., LILO) 2) booting and partitioning software from other  
> OSs (e.g., DOS FDISK, OS/2 FDISK)
```

So, 22800 total cylinders divided by seven partitions is 3258 cylinders. Each partition will be about 3258 cylinders long. I ignore the warning msg because this is not my boot drive

Since I have 4 primary partitions, 3 of them can be 3258 long. The extended partition will have to be  $(4 * 3258)$ , or 13032, cylinders long in order to contain the 4 logical partitions.

I enter the following commands to set up the first of the 3 primary partitions (stuff I type is bold ):

```
Command (m for help): n  
Command action  
  e   extended  
  p   primary partition (1-4)  
p  
Partition number (1-4): 1  
First cylinder (1-22800, default 1): <RETURN>  
Using default value 1  
Last cylinder or +size or +sizeM or +sizeK (1-22800, default 22800): 3258
```

The last partition is the extended partition:

```
Partition number (1-4): 4  
First cylinder (9775-22800, default 9775): <RETURN>  
Using default value 9775  
Last cylinder or +size or +sizeM or +sizeK (9775-22800, default 22800): <RETURN>
```

Using default value 22800

The result, when I issue the print table command is:

```
/dev/sda1      1      3258  26169853+  83  Linux
/dev/sda2     3259     6516  26169885   83  Linux
/dev/sda3     6517     9774  26169885   83  Linux
/dev/sda4     9775    22800 104631345   5   Extended
```

Next I segment the extended partition into 4 logical partitions, starting with the first logical partition, into 3258-cylinder segments. The logical partitions automatically start from /dev/sda5.

```
Command (m for help): n
First cylinder (9775-22800, default 9775): <RETURN>
Using default value 9775
Last cylinder or +size or +sizeM or +sizeK (9775-22800, default 22800): 13032
```

The end result is:

```
Device Boot      Start         End      Blocks   Id  System
/dev/sda1        1         3258   26169853+   83  Linux
/dev/sda2       3259         6516   26169885   83  Linux
/dev/sda3       6517         9774   26169885   83  Linux
/dev/sda4       9775        22800 104631345    5   Extended
/dev/sda5       9775        13032   26169853+   83  Linux
/dev/sda6      13033        16290   26169853+   83  Linux
/dev/sda7      16291        19584   26459023+   83  Linux
/dev/sda8      19585        22800   25832488+   83  Linux
```

Finally, I issue the write command (w) to write the table on the disk. To make the partitions usable, I will have to format each partition and then mount it.

## 5.4. Submitted Examples

I'd like to submit my partition layout, because it works well with any distribution of Linux (even big RPM based ones). I have one hard drive that ... is 10 gigs, exactly. Windows can't see above 9.3 gigs of it, but Linux can see it all, and use it all. It also has much more than 1024 cylinders.

**Table 7. Partition layout example**

Partition	Mount point	Size
/dev/hda1	/boot	(15 megs)
/dev/hda2	windows 98 partition	(2 gigs)
/dev/hda3	extended	(N/A)
/dev/hda5	swap space	(64 megs)
/dev/hda6	/tmp	(50 megs)

Partition	Mount point	Size
/dev/hda7	/	(150 megs)
/dev/hda8	/usr	(1.5 gigs)
/dev/hda9	/home	(rest of drive)

I test new kernels for the USB mass storage, so that explains the large /boot partition. I install LILO into the MBR, and by default I boot windows (I'm not the only one to use this computer).

I also noticed that you don't have any REAL examples of partition tables, and for newbies I HIGHLY suggest putting quite a few up. I'm freshly out of the newbie stage, and partitioning was what messed me up the most.

---