Create Partitions and Filesystems

Information

These notes were originally written in the year 2000 as part of a set of LPI Exam 101 training materials. The LPI training course at Bromley College was subsequently discontinued and some of the sections of the notes modified and incorporated into our one-day System Administration Courses. The remainder of the notes have now been made publicly available on the linuxtraining.org.uk website.

If you are a beginner please do not be put off of training courses by these notes, as they are rather technical. On the other hand if you are a more experienced Linux user we hope you find the coverage of this topic refreshingly clear.

For full details of our current Linux training please visit the site:
http://ce.bromley.ac.uk/linux

If you have reached this page from a search engine and wish to see the full contents list for the published notes please visit the site:
http://www.linuxtraining.org.uk

We hope you find these notes useful, but please remember that they apply to the 2.2 kernel. I will update them when I have the time.

Clive Gould - 21st December 2004

Create Partitions and Filesystems

Objective 1

Create Partitions and Filesystem: Create disk partitions using fdisk, create hard drive and other media filesystems using mkfs.

Partitions

When you install a new hard drive it is first necessary to partition it and then format the partitions before installing any software.

There are two possible types of partition on a hard drive, primary and extended. Because of the nature of the partition table a particular drive is limited to a maximum of four primary partitions or three primary and one extended partition. You can get round the four partition limitation by using the extended partition which can contain a potentially unlimited number logical partitions. In practice it is not recommended to use more than twelve logical partitions on one disk. A computer can only be booted from a primary partition.

The syntax Linux uses to represent drives and partitions is illustrated in the following table:

`DRIVE_TYPE` **`d`** ***`DRIVE_LETTER PARTITION_NUMBER`***

`DRIVE_TYPE`	Where *`DRIVE_TYPE`* can be
	f = floppy
	h = ide
	s = scsi

`DRIVE_LETTER`	Where *`DRIVE_LETTER`* can be 0 or 1 for floppy a (1st hd), b (second hd), c (third hd) etc...

`PARTITION_NUMBER`
`Used only with hard drives`
	Where *`PARTITION_NUMBER`* can be
	0 - 4 reserved for primary or extended partitions
	5, 6, 7 etc. for logical partitions

Create Partitions and Filesystems

This is best illustrated by an example. The diagram below shows the partitioning for a typical Linux installation as displayed by fdisk:

```
Disk /dev/sda: 255 heads, 63 sectors, 1109 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1   *         1       255   2048256   83  Linux
/dev/sda2           256      1109   6859755    5  Extended
/dev/sda5           256       264     72261   82  Linux swap
/dev/sda6           265      1109   6787431   83  Linux
```

The corresponding **/etc/fstab** file is illustrated below:

```
[root@redhat /root]# cat /etc/fstab
/dev/sda1      /             ext2    defaults        1 1
/dev/sda6      /home         ext2    defaults        1 2
/dev/sda5      swap          swap    defaults        0 0
/dev/fd0       /mnt/floppy   ext2    noauto          0 0
/dev/cdrom     /mnt/cdrom    iso9660 noauto,ro       0 0
none           /proc         proc    defaults        0 0
none           /dev/pts      devpts  mode=0622       0 0
```

You can see the PC has one scsi hard drive (sda), with just one bootable primary partition (sda1) mounted on / and an extended partition (sda2), containing two logical partitions (sda5 and sda6), one a swap partition and the other mounted on /home.

Normally Linux distributions either create partitions automatically during the installation procedure, or you can use a custom install and create your own partitions using a utility like Disc Druid or fdisk. In these notes we will look at the deletion and creation of partitions using fdisk.

Create Partitions and Filesystems

Using Multiple Partitions

The minimum number of partitions you need are two, but many installations use multiple partitions. This is illustrated in the table below:

Minimum Partition Structure
/ (filesystem root)
swap

Typical Multiple Partition Structure
/ (filesystem root)
/var
/usr
/tmp
/home
swap

The advantages of using multiple partitions are:

- Any damage is limited to a single partition, which can be reformatted without losing the data on other partitions.
- Files like mail and log files, which enlarge dynamically, can be prevented from using up too much disk space.
- It is easier to upgrade system files without overwriting other data.
- Backups are easier.
- The time required to check the filesystem at boot is reduced.

One disadvantage of multiple partitions is:

- If a partition gets too full you cannot resize it, but have to delete it and recreate it.

Create Partitions and Filesystems

Partition Table Manipulator for Linux - fdisk

The utility **fdisk** can be used by the root to add or remove partitions on a hard drive. It cannot be used to resize an existing partition on a hard drive. If you wish to resize a partition to make room for a Linux installation you can use a DOS utility called **fips** (FAT 16 only) or a third party tool such as Partition Magic or Partition Commander.

Using the command fdisk, you can very easily erase everything on your hard disc, or make it inaccessible. It is vitally important to back up critical data before making any changes to an existing partition structure, and then be very careful about any changes you do make.

As a example, imagine you decide you want to delete an unused FAT16 logical partition on hda5. In the process of doing this fdisk automatically renumbers the existing logical partitions, making it impossible for LILO to find the / partition, and therefore impossible for you to boot the operating system.

Each operating system has its own version of fdisk. If you are creating partitions you want Linux to be able to access, you must use the Linux version. (If you wish to create partitions accessible from DOS you must use the DOS version of fdisk - check with the fdisk man page as the DOS fdisk has a bug in it!)

The fdisk utility can either be run from the hard drive on an existing Linux installation as root, or from boot floppy, depending on what you are trying to do.

Be sure to keep comprehensive notes when using fdisk. For each partition you should note the device, the start and end cylinders for the partition, the partition size in blocks and any error messages. (Each block is 1K byte).

On the following two pages are listed some of the main interactive commands used with fdisk together with examples of their use.

Create Partitions and Filesystems

fdisk interactive command	*Explanation and Example*

d	**Delete existing partition**
	`Command (m for help): d`
	`Partition number (1-8): 5`
	
	`Command (m for help):`

m	**Display help menu**
	`Command (m for help): m`
	`Command action`
	`   a   toggle a bootable flag`
	`   b   edit bsd disklabel`
	`   c   toggle the dos compatibility flag`
	`   d   delete a partition`
	`   l   list known partition types`
	`   m   print this menu`
	`   n   add a new partition`
	`   o   create a new empty DOS partition table`
	`   p   print the partition table`
	`   q   quit without saving changes`
	`   s   create a new empty Sun disklabel`
	`   t   change a partition's system id`
	`   u   change display/entry units`
	`   v   verify the partition table`
	`   w   write table to disk and exit`
	`   x   extra functionality (experts only)`
	
	`Command (m for help):`

Create Partitions and Filesystems

<table>
<tr><th><em>fdisk interactive command</em></th><th><em>Explanation and Example</em></th></tr>
<tr><td><strong>n</strong></td><td>

Create new partition

```
Command (m for help): n
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
l
First cylinder (952-1024, default 952): 952
Last cylinder or +size or +sizeM or +sizeK (952-
1015, default 1015): 1015

Command (m for help):
```

Note: you will need to specify the starting and either the ending cylinder, or the partition size e.g. +80M

</td></tr>
<tr><td><strong>p</strong></td><td>

Display Partition Table

```
Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 1027 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot Start   End    Blocks   Id  System
/dev/hda1   *      1   261   2096451    b  Win95 FAT32
/dev/hda2        262  1024   6128797+   5  Extended
/dev/hda5        952  1015    514048+   6  FAT16
/dev/hda6       1016  1024     72261   82  Linux swap
/dev/hda7        262   516   2048224+  83  Linux
/dev/hda8        517   951   3494106   83  Linux

Command (m for help):
```

</td></tr>
<tr><td><strong>q</strong></td><td>

Quit without saving changes to the Partition Table

```
Command (m for help): q

[root@ext7144 /root]#
```

</td></tr>
</table>

Create Partitions and Filesystems

fdisk interactive command	*Explanation and Example*

t	**Change Partition Type**
	`Command (m for help): `**`t`**
	`Partition number (1-8): `**`6`**
	`Hex code (type L to list codes): `**`82`**
	`Changed system type of partition 6 to 82 (Linux swap)`
	
	Note: By default fdisk creates Linux native partitions. You will need to use the **t** command to turn a native partition into a swap partition. A list of partition types can be obtained using the **l** interactive command

w	**Write changes to Partition Table - EXTREME CARE!**
	
	`Command (m for help): `**`w`**
	`The partition table has been altered!`
	
	`Calling ioctl() to re-read partition table.`
	`Syncing disks.`
	
	`[root@ext7144 /root]#`

Additional notes when using fdisk for partitioning:

- The swap partition is used to enhance the amount of virtual memory on your system. Your swap size should be twice the size of the physical RAM on your system, up to a maximum size of 128 Mb. However, you can have up to sixteen separate swap partitions. The minimum recommended size for a swap partition is 16 Mb.

- If you are using LILO to boot your system, the partition where LILO is located must be entirely within the first 504 Mb (1023) cylinders of your hard drive.

Create Partitions and Filesystems

Build a Linux File System - mkfs

Once you have partitioned your hard disc you will need to format the partitions. You can create a filesystem (called formatting a disc with MSDOS) for any supported filesystem, using the **mkfs** command. Linux supports a wide variety of filesystems including FAT16, FAT32, NTFS and ext2.

The syntax for mkfs is shown below:

```
mkfs option(s) device blocks
```

Common mkfs options are:

Option	*Explanation*
-c	Check the device for bad blocks when creating the file system. It is a good idea to always use this option.
-t *FS_TYPE*	Specifies the fype of file system to be built (default ext2) To find out the file system types your kernel supports look at the contents of the directory **/usr/src/linux/fs**
-v	Displays all the commands used to create the file system

A number of examples of using mkfs are shown overleaf.

To create an ext2 filesystem on **/dev/hda8** enter the following command as root:

```
[root@ext7144 /root]# mkfs -t ext2 -c /dev/hda8 3494106
```

Note: When creating a filesystem you need to be very careful to make sure the device and block arguments are correct. If you get either wrong you can seriously damage data on your hard drive. This is where the notes you made during partitioning come in useful.

Create Partitions and Filesystems

To create an ext2 filesystem on a 1.44 Mb floppy disc enter the following command as root:

```
[root@ext7144 /root]# mkfs -t ext2 -c /dev/fd0 1440
mke2fs 1.17, 26-Oct-1999 for EXT2 FS 0.5b, 95/08/09

Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
184 inodes, 1440 blocks
72 blocks (5.00%) reserved for the super user
First data block=1
1 block group
8192 blocks per group, 8192 fragments per group
184 inodes per group

Checking for bad blocks (read-only test): done
Writing inode tables: done
Writing superblocks and filesystem accounting information:
done
[root@ext7144 /root]#
```

If you wish instead to create an MSDOS filesystem on a 1.44 Mb floppy disc enter the following command as root:

```
[root@ext7144 /root]# mkfs -t msdos -c /dev/fd0 1440
mkfs.msdos 2.2 (06 Jul 1999)
[root@ext7144 /root]#
```

In the above examples the switch **-t** lets you specify the type of filesystem to be built and the switch **-c** causes mkfs to check the device for bad blocks before building the filesystem.

Note: you can also format a floppy drive using **`fdformat /dev/fd0`**