

SECTION 4 **Configure Linux File System Partitions**

This section introduces some command you need to manage your Linux file system partitions.

Objectives

1. Finalize Partitioning
2. Configure LVM with Command Line Tools

Objective 1 Finalize Partitioning

The program **fdisk** is used for partitioning hard disks from the command line.

```
da10:~ # fdisk /dev/sda
```

```
The number of cylinders for this disk is set to 1111.  
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., old versions of LILO)  
2) booting and partitioning software from other OSs  
   (e.g., DOS FDISK, OS/2 FDISK)
```

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

To actually write your changes to the partition table on the disk, enter **w** (write).

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

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
WARNING: Re-reading the partition table failed with error 16: Device or  
resource busy.
```

```
The kernel still uses the old table.
```

```
The new table will be used at the next reboot.
```

```
Syncing disks.
```



When the new table is written, you are not asked for confirmation if you really want to do this.

As the output of **fdisk** says, you cannot directly use the new partition to create a file system on a new partition. You could now reboot as suggested, but you can also use the program **partprobe** to get the kernel to use the new partition table.

Objective 2 Configure LVM with Command Line Tools

Setting up LVM consists of several steps, with a dedicated tool for each:

- Tools to Administer Physical Volumes
- Tools to Administer Volume Groups
- Tools to Administer Logical Volumes

This is just a brief overview, not all available LVM tools are covered. To view the tools that come with LVM, enter **rpm -ql lvm2 | less** on a command line, and have a look at the corresponding manual pages for details on each of them.

Tools to Administer Physical Volumes

Partitions or entire disks can serve as physical volumes for LVM.

The ID of a partition used as part of LVM should be **Linux LVM, 0x8e**. However the ID **0x83, Linux**, works as well.

To use an entire disk as physical volume, it may not contain a partition table. Overwrite any existing partition table with **dd**:

```
da10:~ # dd if=/dev/zero of=/dev/hdd bs=512 count=1
```

The next step is to initialize the partition for LVM. The tool to use is **pvcreate**:

```
da10:~ # pvcreate /dev/hda9  
Physical volume "/dev/hda9" successfully created
```

pvscan shows the physical volumes and their use:

```
da10:~ # pvscan
PV /dev/hda9   lvm2 [242,95 MB]
Total: 1 [242,95 MB] / in use: 0 [0   ] / in no VG: 1 [242,95 MB]
```

The tool **pvmove** is used to move data from one physical volume to another (providing there is enough space), in order to remove a physical volume from LVM.

Tools to Administer Volume Groups

The tool **vgcreate** is used to create a new volume group. To create the volume group system, and add the physical volume /dev/hda9 to it, enter:

```
da10:~ # vgcreate system /dev/hda9
Volume group "system" successfully created
da10:~ # pvscan
PV /dev/hda9   VG system   lvm2 [240,00 MB / 240,00 MB free]
Total: 1 [240,00 MB] / in use: 1 [240,00 MB] / in no VG: 0 [0   ]
```

pvscan shows the new situation.

To add further physical volumes to the group, use **vgexpand**. Removing unused physical volumes is done with **vgreduce** after shifting data from the physical volume scheduled for removal to other physical volumes using **pvmove**. **vgremove** removes a volume group, providing there are no logical volumes in the group.

Tools to Administer Logical Volumes

To create a logical volume, use **lvcreate**, specifying the size, the name for the logical volume, and the volume group:

```
da10:~ # lvcreate -L 100M -n data system
Logical volume "data" created
```

The next step is to create a file system within the logical volume and mount it:

```
da10:~ # lvscan
ACTIVE                '/dev/system/data' [100,00 MB] inherit
da10:~ # mkreiserfs /dev/system/data
mkreiserfs 3.6.19 (2003 www.namesys.com)
...
ReiserFS is successfully created on /dev/system/data.
da10:~ # mount /dev/system/data /data
```

As shown above, **lvscan** is used to view the logical volumes. It shows the device to use for the formatting and mounting.

lvextend is used to increase the size of a logical volume. After that you can increase the size of the file system on that logical volume to make use of the additional space.

Before you use **lvreduce** to reduce the size of a logical volume, you have to reduce the size of the file system. Only then reduce the size of the logical volume. If you cut off parts of the file system by simply reducing the size of the logical volume without shrinking the file system first, you will lose data.

Summary

Objective	Summary
1. Finalize Partitioning	After changing the partitions using fdisk, you cannot directly create a file system on the new partition. You could now reboot, but you can also use the program partprobe .

Objective	Summary
2. Configure LVM with Command Line Tools	<p>To initialize a partition for LVM, use the tool pvcreate.</p> <p>pvscan shows the physical volumes and their use.</p> <p>The tool pvmove is used to move data from one physical volume to another.</p> <p>The tool vgcreate is used to create a new volume group.</p> <p>To add further physical volumes to the group, use vgexpand.</p> <p>Removing unused physical volumes is done with vgreduce after shifting data from the physical volume scheduled for removal to other physical volumes using pvmove.</p> <p>vgremove removes a volume group, providing there are no logical volumes in the group.</p> <p>To create a logical volume, use lvcreate.</p> <p>lvscan is used to view the logical volumes.</p> <p>lvextend is used to increase the size of a logical volume.</p> <p>Before you use lvreduce to reduce the size of a logical volume, you have to reduce the size of the file system.</p>
