

Roadmap

1. Device Management
2. Disk Partitioning
3. File System Management
4. Mounting File Systems
5. Hard and Symbolic Links



1. Device Management

- Linux is device independent, which improves its portability from one system to another.
- A **device driver** is a computer program which defines how to perform communication between kernel and hardware devices.
- A driver provides a software interface to hardware devices, enabling operating systems and other computer programs to access hardware functions without a need to know precise details of the hardware being used.
- Linux treats devices as if they are files, and you can access devices the same way you access files in Linux.
- Special device files can be found under the directory **/dev**.

Classes of Device Drivers:

1. **Character Devices:** Devices that send data transfers character by character (like a keyboard).
2. **Block Devices:** Devices that receive data in block transfers by using memory to buffer the transfers and can host a file system, such as a hard disk.
3. **Network Devices:** Their function is to send and receive packets of information as directed by the network subsystem of the kernel.
 - Linux identifies each device by **two numbers**
 - Major number identifies the **device driver**.
 - Minor number specifies the particular **device**



Devices Listing

- Exploring the devices can be done through **ls -l** command as shown in the example below
- The first letter in file permissions indicate the device class.

```

> cd /dev
> ls -l
crw----- 1 root  root
brw-rw---- 1 root  disk
brw-rw---- 1 root  disk
  
```

The terminal output shows three lines of device information. Callouts explain the following parts:

- Block drivers are identified by a "b"**: Points to the 'b' in the permissions of the two disk entries.
- Char drivers are identified by a "c"**: Points to the 'c' in the permissions of the console entry.
- Major numbers**: Points to the first number in the device numbers (5, 8, 8).
- Minor numbers**: Points to the second number in the device numbers (1, 0, 1).

Linux Device Naming Examples:

fd0	First Floppy Drive
fd1	Second Floppy Drive

sda	First hard disk
sdb	Second hard disk
sda1	First partition of the first hard disk
sda7	Seventh partition of the first hard disk

sr0	First CD-ROM
Sr1	Second CD-ROM

cdrom	Symbolic link to the CD-ROM device file
mouse	Symbolic link to the mouse device file

null	Anything written on this device will disappear
zero	One can endlessly read zeros out of this device

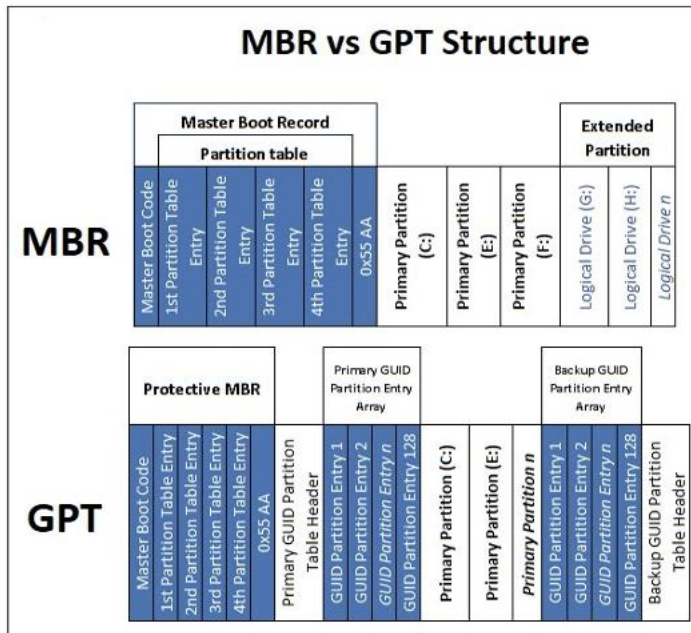


2. Disk Partitioning

- **Disk Partitioning:** is dividing a single hard drive into many logical drives using partitioning tools.
- **Partition:** Logical storage unit which allows treating a single physical device as multiple ones, allowing a different File system on each partition.
- **Primary Partition:** It is a partition which can hold operating system boot files and cannot be further subdivided into logical drives and must be formatted with a file system.
- **Extended Partition:** It is a partition which can be further subdivided into a number of logical partitions and cannot be directly formatted with a file system. However, logical partitions within an extended partition can be formatted with a file system.

2.1 Legacy MBR Partition Table

2.2 GUID Partition Table (GPT)



Disk Management Commands

fdisk

- A command line utility used to manage partitions on a hard disk. It works on MS-DOS, Windows and Linux.

gdisk

- is a text-mode menu-driven program for creation and manipulation of partition tables.



- Create and delete and display information about partition GPT partitions.
- Convert an MBR partition table to a GPT partition table.

Gparted

- A GUI tool to create, delete and modify MBR and GPT partitions.

3. File System Management

- **File System** is a method for storing and organizing files on Linux.
- **A journaling filesystem** keeps a record of the changes that are being made to the filesystem.
- An **inode** is a data structure that stores everything about a file apart from its name and actual content and it specifies where a file's data physically exists on a disk.
- **Superblock** is a component in every file system which contains information about the file system, such as: Type, Size, and Status.
- Linux maintains multiple redundant copies of the superblock in every file system.

Common File System's Types

Type	Description
ext2	The Second Extended File System (ext2) is one of the oldest Linux file systems still available.
ext3	The Third Extended File System (ext3) is an updated version of ext2 that supports journaling.
ext4	The Fourth Extended File System (ext4) includes all of the features found with ext2 and ext3.
swap	It is used as virtual memory (the portion of the hard disk used to temporarily store portions of main memory) by the operating system.
NTFS	Microsoft operating systems use NTFS (New Technology File System). Linux provides limited support for NTFS
VFAT	VFAT is a FAT32 file system for Linux. VFAT includes long name support.
XFS	It is proficient at handling large files, offers smooth data transfers, and provides journaling. It also can reside on a regular disk partition or on a logical volume

4. Mounting File Systems

- **Mounting** is the attaching of an additional file system to the currently accessible file system of a computer (making use of device file).
- At least one partition is mounted during booting process. Normally is the first partition in the first hard disk (/dev/sda1)
- Mounting can be done for any storage device (USB, CDROM, ...).



- A storage device can be mounted in a directory in the file system.
- You should mount storage devices in empty directories, since mounting a volume to a directory that contains data makes the data inaccessible.

Mounting Commands

Command	Description
mount	Mount a volume or device Syntax: mount -- [file--dev] [mnt--point]: -r: mounting in read--only mode. -t: kind of file system mounted. -o: specify owner Example: mount /dev/sdb1 mydisk-o uid=student
df	View which file systems are mounted to specific mount points.
umount	disconnects the device from the rest of the system. Doing this requires that no process is making use of the file system to umount. syntax: umount [mnt--point] umount[device]

5. Hard and Symbolic Links

- In Linux, “everything is a file” and a file is fundamentally a link to an inode ().
- A **hard link** is a file that points to the same underlying inode, as another file.
- In case you delete one file, it removes one link to the underlying inode. Example to create a hard link to file1

ln myfile.txt my-hard-link

- A **symbolic link** (soft link) is a link to another filename in the filesystem. Example to create a hard link to file1

ln-s myfile.txt my-soft-link

- Another important difference between the two types of links is that hard links can only work within the same filesystem while symbolic links can go across different filesystems

