Tishk International University
Science Faculty
IT Department

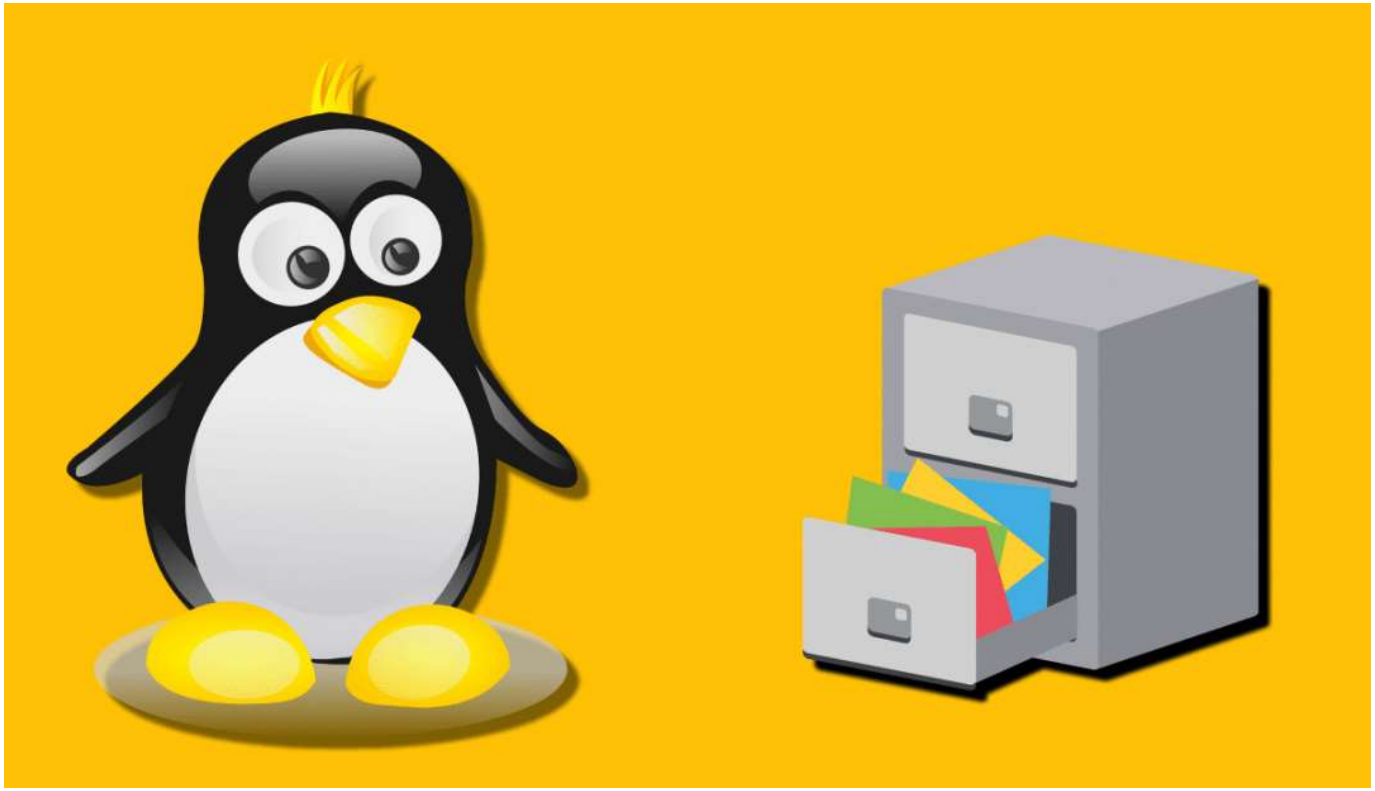
Open Source OS (Linux)

Lecture 6: File Systems

4th Grade - Fall Semester 2021-2022

Instructor: Alaa Ghazi

Lecture 6
File Systems



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 device.
- 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) .

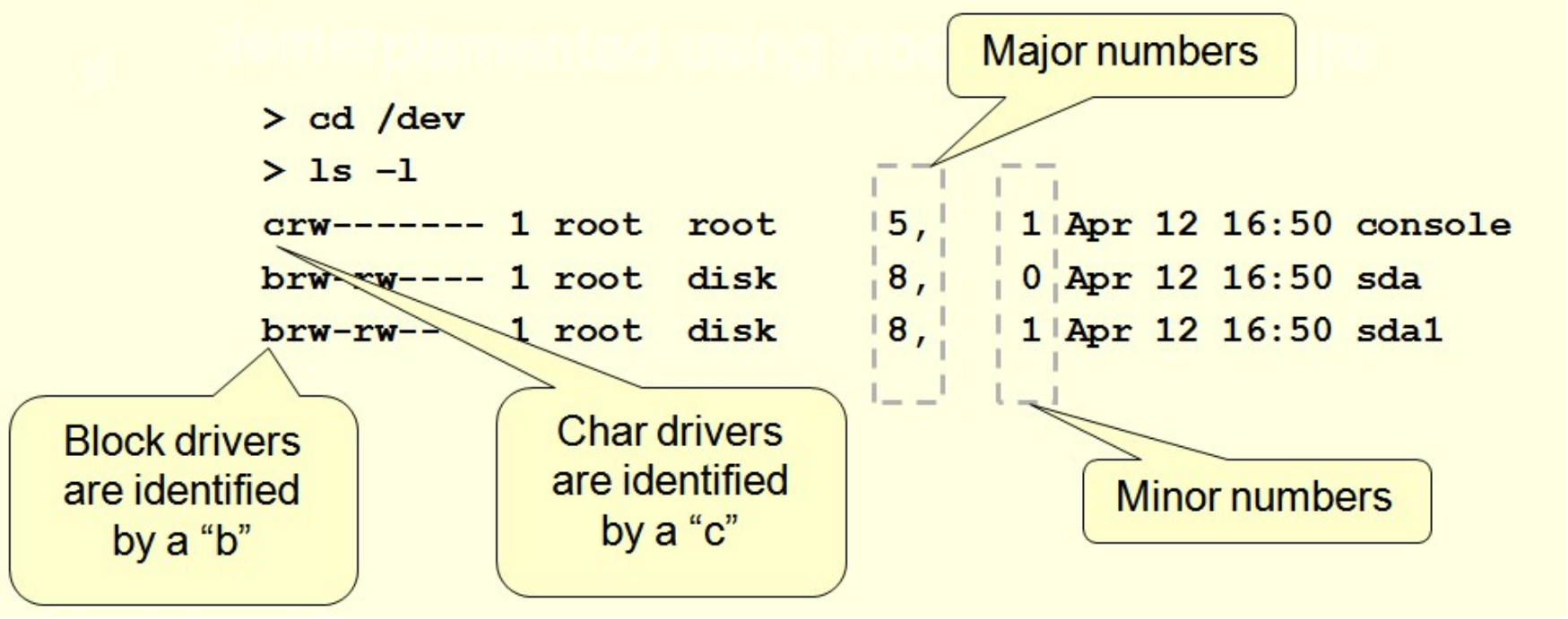
2) Block Devices: Devices that receive data in block transfers by using memory to buffer the transfers and can host a file system, such as a hard disk.

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.



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
sdb7	Seventh partition of the second hard disk

sr0	First CD-ROM
sr1	Second CD-ROM

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

null	Anything written to 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 the 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 the 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

Limitations

- The master boot record must be installed in the first 512 bytes of the hard disk.
- Only four primary partitions can be created on a storage device.
- The default block size of 512 bytes limits partitions to a maximum size of 2 TB = 2 x 2^40 B
- You cannot format an extended partition. However, you can create logical partitions inside an extended partition and format them.

Workarounds

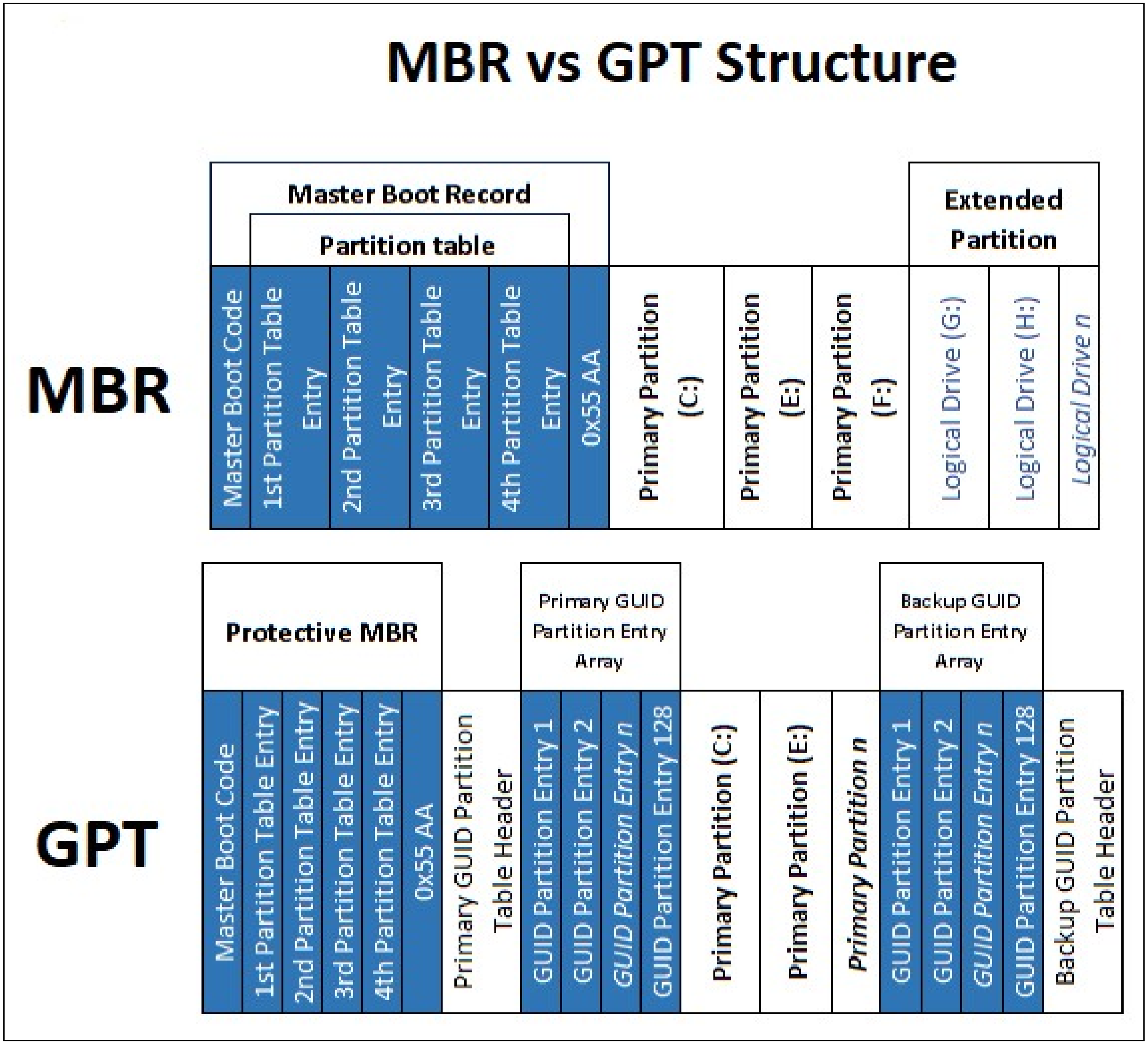
- Logical Block Addressing (LBA) allows the use of larger hard disks.
- Use of 4,096 byte sectors increases the maximum partition size on a disk.
- Extended partitions can contain many logical partitions

2.2 GUID Partition Table(GPT)

- **GUID Partition Table (GPT),** Uses only one type of partition. There are no primary, extended, or logical partitions.
- GPT supports extremely large storage devices and partitions.
- On a device formatted with GPT the maximum partition size is

8 Zi B = 2 x 2^70 B

- GPT allows up to 128 partitions on a storage device.
- Stores a copy of the partition table in the first and last sectors of the storage device. If one copy gets corrupted, then the redundant copy can be used instead.
- Verifies the integrity of the partition table using a *cyclic redundancy check* (CRC).
- Assigns unique IDs to each storage device and partition.


MBR vs GPT Structure
MBR
Master Boot Record
Partition table
Master Boot Code
1st Partition Table Entry
2nd Partition Table Entry
3rd Partition Table Entry
4th Partition Table Entry
0x55 AA
Primary Partition (C:)
Primary Partition (E:)
Primary Partition (F:)
Extended Partition
Logical Drive (G:)
Logical Drive (H:)
Logical Drive n
GPT
Protective MBR
Master Boot Code
1st Partition Table Entry
2nd Partition Table Entry
3rd Partition Table Entry
4th Partition Table Entry
0x55 AA
Primary GUID Partition Table Header
Primary GUID Partition Entry Array
GUID Partition Entry 1
GUID Partition Entry 2
GUID Partition Entry n
GUID Partition Entry 128
Primary Partition (C:)
Primary Partition (E:)
Primary Partition n
Backup GUID Partition Entry Array
GUID Partition Entry 1
GUID Partition Entry 2
GUID Partition Entry n
GUID Partition Entry 128
Backup GUID Partition Table Header

Disk Management Commands

fdisk

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

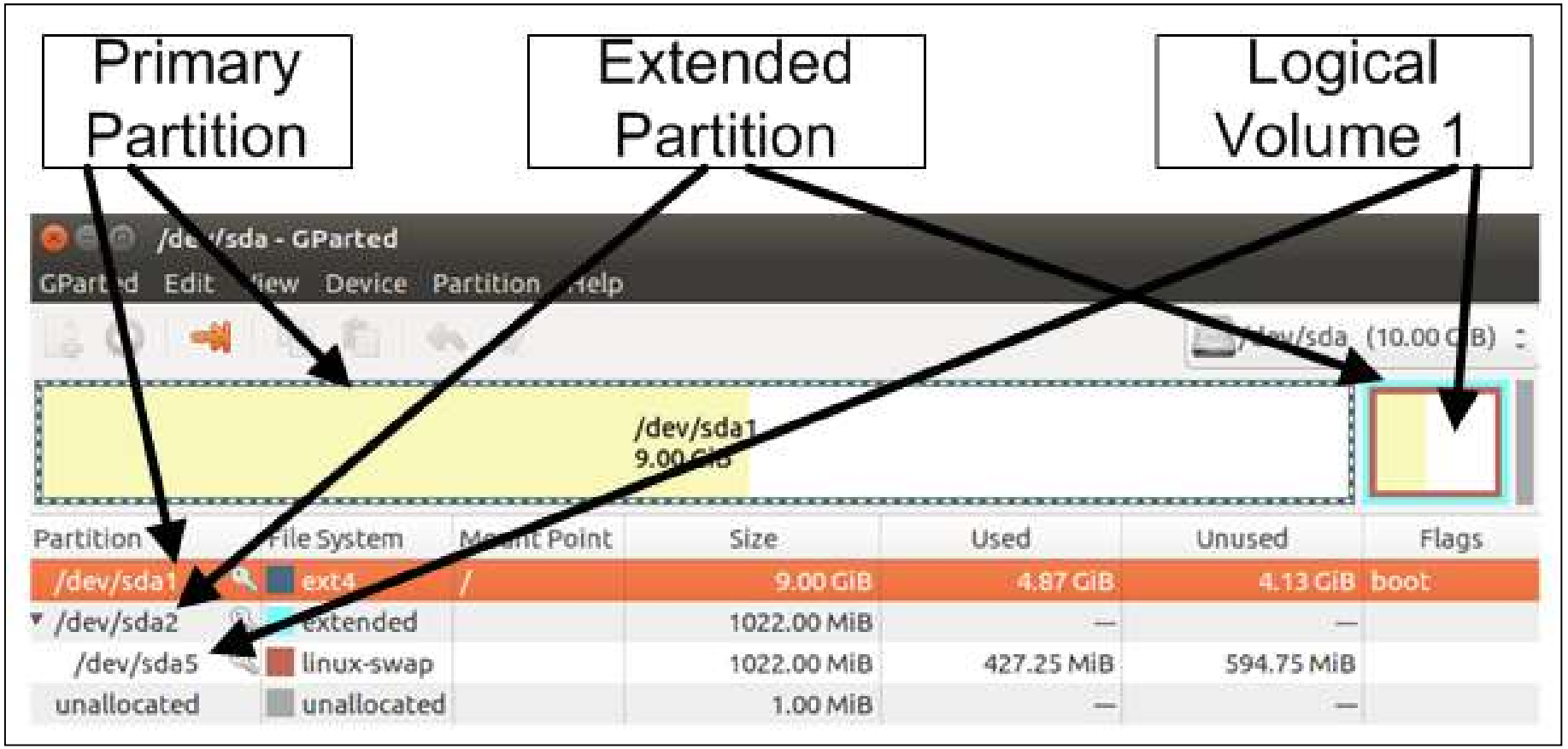
gdisk

- is a text-mode menu-driven program for creation and manipulation of partition tables.
- Create and delete and display information about a 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.

Using gparted to discover 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 Systems' 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

3. Mounting File Systems

- **Mounting** is the attaching of an additional filesystem to the currently accessible filesystem 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 --<options> [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]**

4. Hard and Symbolic Links

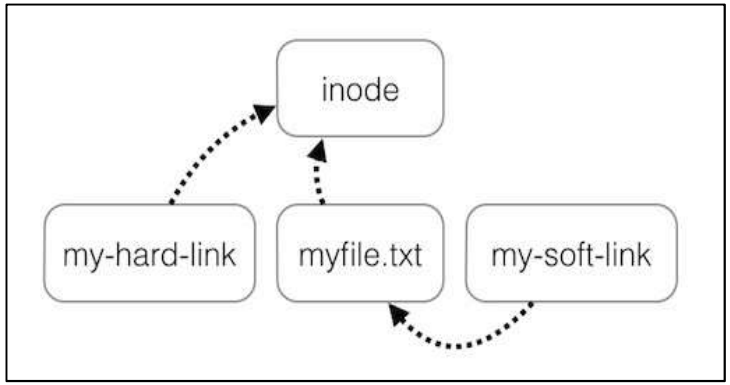
- In Linux, "everything is a file" and a file is fundamentally a link to an inode ().
- A **hard link** is a file that points to the same underlying inode, as another file.
- In case you delete one file, it removes one link to the underlying inode. Example to create a hard link to file1

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.



LAB 6
File Systems

LAB 6 TEST1: MBR Partitioning

1) **In the setting of your virtual machine in VirtualBOX, add a new disk to the SATA controller with 2GB capacity**
2) Install and run **gparted** application using apt-get
3) The disk will be seen inside gparted as **/dev/sdb**
4) Using device menu: create a **msdos** partition table
5) Using partition menu: create 4 primary partitions each of size 256 MB with ext3 file system and **Apply**
6) Try to add fifth primary partition of size 256 MB, record the error
7) Delete one primary partition
8) Add Extended partition of 1 GB
9) Inside it create two logical partitions with 256 MB with **ntfs** file system and Apply
10) Delete all partitions

LAB 6 TEST2: GPT Partitioning

1) Using device menu: create a **gpt** partition table
2) Using partition menu: create 4 primary partitions each of size 256 MB with **ntfs** file system and **Apply**
3) Try to add fifth primary partition of size 256 MB with **ntfs** file system and **Apply**

LAB 6 TEST2: Partition Mounting

1) Provide screen shoots and comments for all steps below
2) Open a new Terminal window
3) Use the command df to list mounting points
4) **Create folder mydisk**
5) Mount sdb1 as mydisk

sudo mount /dev/sdb1 mydisk –o uid=student

6) Use the command df to list mounting points
7) Go inside mydisk and create a new file using cat
8) List the files inside mydisk
9) Go outside mydisk
10) Unmount my disk using the command umount

sudo umount mydisk

11) Use the command df to list mounting points
12) Go inside mydisk and List the files inside mydisk
13) Using gparted delete all partitions in /dev/sdb

LAB 6 TEST4: Hard and Symbolic Links

1) Under **/home/student** create a folder **test1**
2) Inside **test1** create a file called **file1** using touch command
3) Create hard link to **file1**
4) List the directory contents using the command

ls –li

Provide screen shoot and comments

1) Under **/home/student** change to folder **test1**
2) Inside test1 make sure there is a file called **file1**
3) Create soft link to **file1**
4) List the directory contents using the command

ls –li

Provide screen shoot and comments for all steps above

Root Password Reset in Single-User Mode (not required in exam or report)

- From GRUB menu stop over the first option, press 'e' to edit your boot entry (the Ubuntu entry)
- Find the kernel line (starts with linux /boot/) and add init="/bin/bash" at the end of the line.
- Press Ctrl + X to reboot with these settings and enter single user mode, and system will show the root prompt.
- Enter the commands:
- Provide screen shoot and comments for all steps above

```
# mount -rw -o remount  /
# passwd
# reboot -f
```


```
root@(none):/# mount -rw -o remount /
root@(none):/# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@(none):/# reboot -f
```

File System Check in Single-User Mode (not required in exam or report)

- From GRUB menu stop over the first option, press 'e' to edit your boot entry (the Ubuntu entry)
- Find the kernel line (starts with linux /boot/) and add init="/bin/bash" at the end of the line.
- Press Ctrl + X to reboot with these settings and enter single user mode, and system will show the root prompt.
- Enter the commands:

```
# fsck –y /dev/sda1
# reboot -f
```

- Provide screen shoot and comments for all steps above