Understanding Storage Media

MODULE 5

Contents

Understanding Storage Media

5.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Know about Hard Disk Drive(HDD)
- Explain the working of HDD
- Identify various types of interfaces
- Recognise internal structure of HDD
- List different type of formatting
- Explain booting process
- Discover the boot sequence of Windows, Mac and Linux OS

5.2 HARD DISK DRIVE

A hard disk drive (HDD)[1], hard disk, hard drive or fixed disk is a data storage device used for storing and retrieving digital information using one or more rigid ("hard") rapidly rotating disks (platters) coated with magnetic material. The platters are paired with magnetic heads arranged on a moving actuator arm, which read and write data to the platter surfaces. Data is accessed in a random-access manner, meaning that individual blocks of data can be stored or retrieved in any order rather than sequentially. HDDs retain stored data even when powered off.

The primary characteristics of an HDD are its capacity and performance. Capacity is specified in unit prefixes corresponding to powers of 1000: a 1-terabyte (TB) drive has a capacity of 1,000 gigabytes (GB; where 1 gigabyte = 1 billion bytes). Typically, some of an HDD's capacity is unavailable to the user because it is used by the file system and the computer operating system, and possibly inbuilt redundancy for error correction and recovery. Performance is specified by the time required to move the heads to a track or cylinder (average access time) plus the time it takes for the desired sector to move under the head (average latency, which is a function of the physical rotational speed in revolutions per minute), and finally the speed at which the data is transmitted (data rate).

5.2.1 Working of HDD

An HDD records data by magnetizing a thin film of ferromagnetic material on a disk. Sequential changes in the direction of magnetization represent binary data bits. The data is read from the disk by detecting the transitions in magnetization. User data is encoded using an encoding scheme, such as run-length limited encoding, which determines how the data is represented by the magnetic transitions.
A typical HDD design consists of a spindle that holds flat circular disks, also called platters, which hold the recorded data. The platters are made from a non-magnetic material, usually

[1] https://en.wikipedia.org/wiki/Hard_disk_drive

aluminum alloy, glass, or ceramic, and are coated with a shallow layer of magnetic material typically 10–20 nm in depth, with an outer layer of carbon for protection.

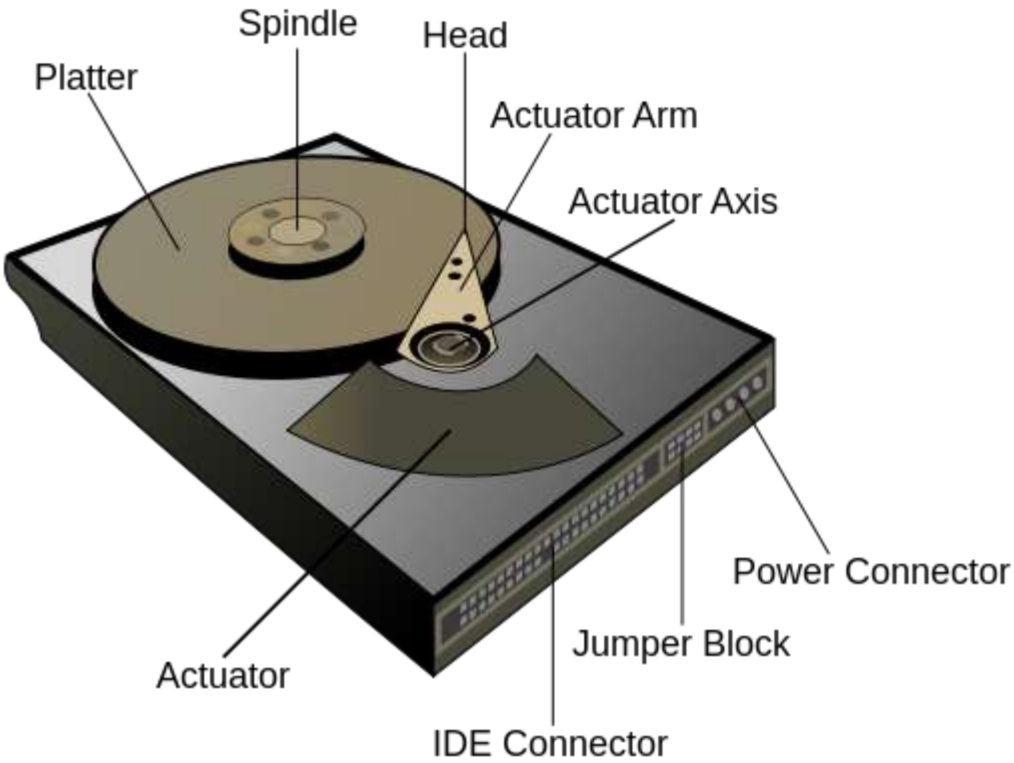


Figure 1: Hard disk drive of a computer[2]

The platters in contemporary HDDs are spun at speeds varying from 4,200 rpm in energy-efficient portable devices, to 15,000 rpm for high-performance servers. Information is written to and read from a platter as it rotates past devices called read-and-write heads that are positioned to operate very close to the magnetic surface, with their flying height often in the range of tens of nanometers. The read-and-write head is used to detect and modify the magnetization of the material passing immediately under it.

In modern drives there is one head for each magnetic platter surface on the spindle, mounted on a common arm. An actuator arm (or access arm) moves the heads on an arc (roughly radially) across the platters as they spin, allowing each head to access almost the entire surface of the platter as it spins. The arm is moved using a voice coil actuator or in some older designs a stepper motor. Early hard disk drives wrote data at some constant bits per second, resulting in all tracks having the same amount of data per track but modern drives (since the 1990s) use zone bit recording—increasing the write speed from inner to outer zone and thereby storing more data per track in the outer zones.

The two most common form factors for modern HDDs are 3.5-inch, for desktop computers, and 2.5-inch, primarily for laptops. HDDs are connected to systems by standard interface cables

[2] https://en.wikipedia.org/wiki/Hard_disk_drive#/media/File:Hard_drive-en.svg

such as PATA (Parallel ATA), SATA (Serial ATA), USB or SAS (Serial attached SCSI) cables. The details of various types of HDD interfaces are discussed in the next section.

5.2.2 Interface

HDDs are accessed over one of a number of bus types, parallel ATA (PATA, also called IDE or EIDE; described before the introduction of SATA as ATA), Serial ATA (SATA), SCSI, Serial Attached SCSI (SAS), and Fibre Channel. Bridge circuitry is sometimes used to connect HDDs to buses with which they cannot communicate natively, such as IEEE 1394, USB and SCSI.

Modern interfaces connect an HDD to a host bus interface adapter with one data/control cable. Each drive also has an additional power cable, usually direct to the power supply unit. Now let us discuss various types of HDD interfaces in detail.

- **Small Computer System Interface (SCSI)**: originally named SASI for Shugart Associates System Interface, was standard on servers, workstations, Commodore Amiga, Atari ST and Apple Macintosh computers through the mid-1990s, by which time most models had been transitioned to IDE (and later, SATA) family disks. The range limitations of the data cable allows for external SCSI devices.

Figure 2: SCSI Interface[3]

- **Integrated Drive Electronics (IDE)**: later standardized under the name AT Attachment (ATA, with the alias P-ATA or PATA (Parallel ATA) retroactively added upon introduction of SATA) moved the HDD controller from the interface card to the disk drive. This helped to standardize the host/controller interface, reduce the programming complexity in the host device driver, and reduced system cost and complexity. The 40-pin IDE/ATA connection transfers 16 bits of data at a time on the data cable. The data cable was originally 40-conductor, but later higher speed requirements for data transfer to and from the HDD led to an "ultra DMA" mode, known as UDMA. Progressively swifter versions of this standard ultimately added the requirement for an 80-conductor variant of

[3] http://www.saypoint.net/2012/05/different-types-of-pc-hard-disk.html

the same cable, where half of the conductors provide grounding necessary for enhanced high-speed signal quality by reducing cross talk.

Figure 3: Parallel ATA[4]

- **EIDE:** was an unofficial update (by Western Digital) to the original IDE standard, with the key improvement being the use of direct memory access (DMA) to transfer data between the disk and the computer without the involvement of the CPU, an improvement later adopted by the official ATA standards. By directly transferring data between memory and disk, DMA eliminates the need for the CPU to copy byte per byte, therefore allowing it to process other tasks while the data transfer occurs.
- **Fibre Channel (FC):** is a successor to parallel SCSI interface on enterprise market. It is a serial protocol. In disk drives usually the Fibre Channel Arbitrated Loop(FC-AL) connection topology is used. FC has much broader usage than mere disk interfaces, and it is the cornerstone of storage area networks (SANs). Recently other protocols for this field, like iSCSI and ATA over Ethernet have been developed as well. Confusingly, drives usually use *copper* twisted-pair cables for Fibre Channel, not fibre optics. The latter are traditionally reserved for larger devices, such as servers or disk array controllers.
- **Serial Attached SCSI (SAS)**: The SAS is a new generation serial communication protocol for devices designed to allow for much higher speed data transfers and is compatible with SATA. SAS uses a mechanically identical data and power connector to standard 3.5-inch SATA1/SATA2 HDDs, and many server-oriented SAS RAID controllers are also capable of addressing SATA HDDs. SAS uses serial communication instead of the parallel method found in traditional SCSI devices but still uses SCSI commands.
- **Serial ATA (SATA)**: The SATA data cable has one data pair for differential transmission of data to the device, and one pair for differential receiving from the device, just like EIA-422. It requires that data be transmitted serially. A similar differential signalling system is used in RS485, LocalTalk, USB, FireWire, and differential SCSI.

[4] http://www.saypoint.net/2012/05/different-types-of-pc-hard-disk.html

Figure 4: SATA[5]

5.3 DETAILS OF INTERNAL STRUCTURE OF HDD

Before[6] trying to understand formatting, you first need to understand how a hard drive works. Let's start with the detailed description of physical formatting (or low-level formatting) and logical formatting (or high-level formatting). Even though hard drives can be very small, they still contain millions of bits and therefore need to be organized so that information can be located. This is the purpose of the file system.

5.3.1 Low-Level Formatting

The purpose of low-level formatting is to divide the disk surface into basic elements:

- tracks
- sectors
- cylinders

Remember that a hard drive consists of several circular platters rotating around an axis and covered on either side by a magnetic oxide which, since it is polarised, can be used to store data.

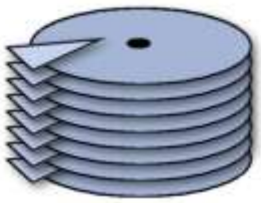


Figure 5: Platters in a HDD

The tracks are the concentric areas written on both sides of a platter.

[5] http://www.saypoint.net/2012/05/different-types-of-pc-hard-disk.html
[6] http://ccm.net/contents/626-formatting-formatting-a-hard-drive

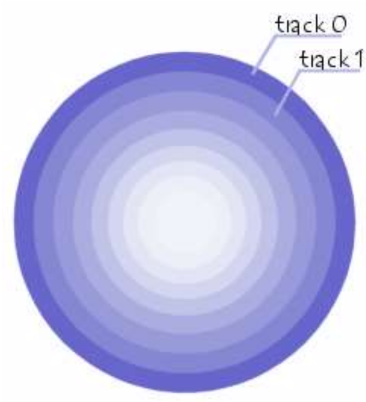


Figure 6: Tracks in a platter of HDD

Finally, these tracks are divided into pieces called sectors. There are millions of tracks and each has around 60 to 120 sectors.

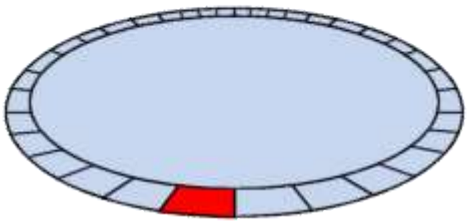


Figure 7: Tracks and sectors in a HDD

A **cylinder** refers to all the data located on the same track of different platters (i.e. vertically on top of each other) as this forms a "cylinder" of data in space.

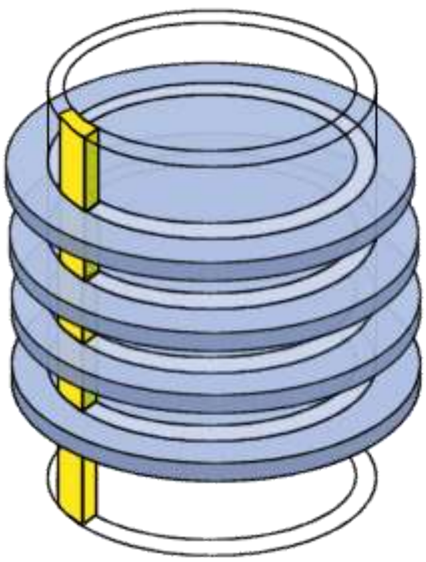


Figure 8: Cylinder in a HDD

Physical formatting therefore consists in organizing the surface of each platter into entities called trackers and sectors, by polarising the disk areas using the write heads. Tracks are numbered starting from 0, then the heads polarise concentrically the surface of the platters. When the head goes from one track to the next, it leaves a gap. Each track is itself organized into sectors (numbered starting from 1) and separated by *gaps*. Each of these sectors starts with an area reserved for system information called a *prefix* and ends with an area called a *suffix*. The purpose of low-level formatting is therefore to prepare the disk surface to receive data and to mark "defective sectors" using tests performed by the manufacturer. When you buy a hard drive, it has already undergone low-level formatting. SO YOU DO NOT NEED TO PERFORM LOW-LEVEL FORMATTING!

During the formatting, check tests (algorithms allowing the validity of sectors to be tested using checksums) are performed and each time a sector is considered defective, the (invalid) checksum is written in the prefix. It can no longer be used thereafter and is said to be "marked defective". When the disk reads the data, it sends a value that depends on the content of the sent packet, and which is initially stored with the data. The system calculates this value based on the data received, and then it compares it with the one that is stored with the data. If these two values are different, the data are no longer valid and there is probably a problem with the disk surface. The cyclic redundancy check (CRC), is based on the same principle to check the integrity of a file. Analysis utilities such as *scandisk* or *chkdsk* operate differently: they write data on sectors considered to be valid, and then read them and compare them. If they are the same, the utility goes on to the next sector, otherwise it marks the sector as defective.

5.3.2 High-level formatting

Logical formatting occurs after the low-level formatting. It creates a file system on the disks that will allow an operating system (DOS, Windows 95, Linux, OS/2, Windows NT, etc.) to use the disk space to store and access files. Operating systems use different file systems, so the type of logical formatting will depend on the operating system you install. So, if you format your disk with a single file system, this naturally limits the number and type of operating systems that you can install (in fact, you can only install operating systems that use the same file system). Fortunately, there is a solution to this problem which is to create partitions. Each of the partitions can effectively have its own file system, and you can therefore install different types of operating systems.

What is a partition?

Partitioning is the process of writing the sectors that will make up the partition table (which contains information on the partition: size in sectors, position with respect to the primary partition, types of partitions present, operating systems installed, etc.). When a partition is created, it is given a *volume name* which allows it to be easily identified.

The partitioning of a hard drive occurs after the drive has been physically formatted but before it is logically formatted. It involves creating areas on the disk where data will not be mixed. It can be used, for example, to install different operating systems that do not use the same file system. There will therefore be at least as many partitions as there are operating systems using different file systems. If you are using just one operating system, a single partition the full size of the disk is sufficient, unless you want create several partitions so as to have, for example, several drives on which data are kept separate.

There are three types of partitions: **primary partitions**, **extended partitions** and **logical drives**. A disk may contain up to four primary partitions (only one of which can be active), or three primary partitions and one extended partition. In the extended partition, the user can create logical drives (i.e. "simulate" several smaller-sized hard drives).

Let's look at an example where the disk contains one primary partition and one extended partition made up of three logical drives (later we will look at multiple primary partitions):

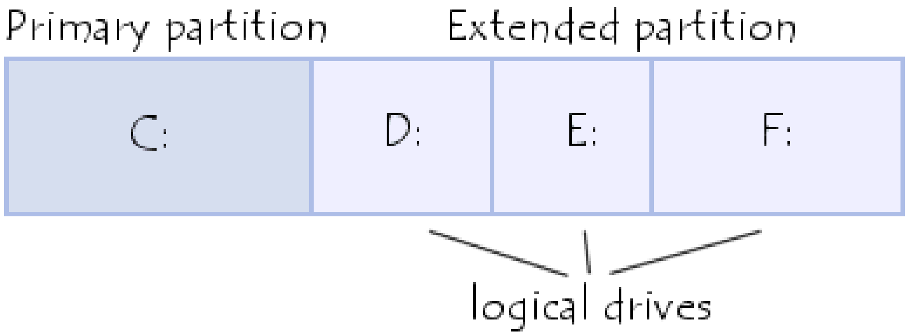


Figure 9: Partitining of HDD

For DOS systems (DOS, Windows 9x), only the primary partition is bootable, and is therefore the only one on which the operating system can be started.

5.3.3 Glossary of some important terms

5.3.3.1 Slack space

The unused space at the end of a file in a file system that uses fixed size clusters (so if the file is smaller than the fixed block size then the unused space is simply left). Often contains deleted information from previous uses of the block[7].

5.3.3.2 Lost Cluster

A lost cluster is a series of clusters on the hard disk drive that are not associated with a particular file. Data is there but it's unknown as to what file that data belongs to. Error checking tool looks for those and tries to repair them. Usually it can repair them, but sometimes it can't. If we're in a situation where we can't repair lost clusters that usually means we've got some serious disk errors and we're going to be losing some data[8].

[7] https://en.wikipedia.org/wiki/Glossary_of_digital_forensics_terms
[8] http://www.utilizewindows.com/pc-fundamentals/storage/333-hardware-and-software-disk-optimization

5.3.3.3 Bad Sector

A bad sector is a sector on a computer's disk drive or flash memory that is either inaccessible or unwriteable due to permanent damage, such as physical damage to the disk surface or failed flash memory transistors[9].

5.3.3.4 Master Boot Record

The **boot sector[10]** (called the **Master Boot Record** or **MBR**) is the first sector of a hard drive (cylinder 0, head 0, sector 1), it contains the main partition table and the code, called the **boot loader**, which, when loaded into memory, will allow the system to boot up. After it is loaded into memory, this programme will determine from which system partition to boot, and will start the programme (called the *bootstrap*) which will start up the operating system present on that partition. This disk sector also contains all information concerning the hard drive (manufacturer, serial number, number of bytes per sector, number of sectors per cluster, number of sectors, etc.). This sector is therefore the most important one on the hard drive and is also used by the BIOS setup to recognize the hard drive. In other words, without it your hard drive is useless, which makes it a favourite target for viruses.

5.4 THE BOOTING PROCESS

The boot process[11] of a modern system involves multiple phases. The following components are involved in the boot process. They are each executed in this order:

- Power Supply Unit
- BIOS and CMOS
- POST tests
- Reading the Partition Table
- The Bootloader
- The Kernel
- OS Kernel

1. **Power Supply Unit:** When the power button is pressed, an electric circuit is closed which causes the power supply unit to perform a self test. In order for the boot process to continue, this self test has to complete successfully. If the power supply cannot confirm the self test, there will usually be no output at all. Most modern x86 computers, especially those using the ATX standard, will have two main connectors to the motherboard: a 4-pin connector to power the CPU, and a 24-pin connector to power other motherboard components. If the self test passes successfully, the PSU will send a signal to the CPU on the 4-pin connector to indicate that it should power on.

[9] https://en.wikipedia.org/wiki/Bad_sector
[10] http://ccm.net/contents/626-formatting-formatting-a-hard-drive
[11] http://ops-school.readthedocs.org/en/latest/boot_process_101.html

2. **BIOS and CMOS:** At its core, the Basic Input/Output System (*BIOS*) is an integrated circuit located on the computer's motherboard that can be programmed with firmware. This firmware facilitates the boot process so that an operating system can load.
   Let's examine each of these in more detail:
   a. *Firmware* is the software that is programmed into Electrically Erasable Programmable Read-Only Memory (EEPROM). In this case, the firmware facilitates booting an operating system and configuring basic hardware settings.
   b. An *integrated circuit* (IC) is what you would likely think of as a stereotypical "computer chip" - a thin wafer that is packaged and has metal traces sticking out from it that can be mounted onto a printed circuit board.

   Your BIOS is the lowest level interface you'll get to the hardware in your computer. The BIOS also performs the Power-On Self Test, or POST. Once the CPU has powered up, the first call made is to the BIOS. The first step then taken by the BIOS is to ensure that the minimum required hardware exists:

   - CPU
   - Memory
   - Video card

   Once the existence of the hardware has been confirmed, it must be configured. The BIOS has its own memory storage known as the CMOS (Complimentary Metal Oxide Semiconductor). The CMOS contains all of the settings the BIOS needs to save, such as the memory speed, CPU frequency multiplier, and the location and configuration of the hard drives and other devices.

   The BIOS first takes the memory frequency and attempts to set that on the memory controller. Next the BIOS multiply the memory frequency by the CPU frequency multiplier. This is the speed at which the CPU is set to run. Sometimes it is possible to "overclock" a CPU, by telling it to run at a higher multiplier than it was designed to, effectively making it run faster. There can be benefits and risks to doing this, including the potential for damaging your CPU.

3. **POST tests:** Once the memory and CPU frequencies have been set, the BIOS begins the Power-On Self Test (POST). The POST will perform basic checks on many system components, including:
   - Check that the memory is working
   - Check that hard drives and other devices are all responding
   - Check that the keyboard and mouse are connected (this check can usually be disabled)
   - Initialise any additional BIOSes which may be installed (e.g. RAID cards)
4. **Reading the Partition Table:** The next major function of the BIOS is to determine which device to use to start an operating system. A typical BIOS can read boot information from the devices below, and will boot from the first device that provides a successful response. The order of devices to scan can be set in the BIOS:
   - Floppy disks

- CD-ROMs
- USB flash drives
- Hard drives
- A network

We'll cover the first four options here. For booting over the network, please refer to *http://networkboot.org/fundamentals/*.

There are two separate partition table formats: Master Boot Record (MBR) and the GUID Partition Table (GPT). We'll illustrate how both store data about what's on the drive, and how they're used to boot the operating system.

a. **Master Boot Record (the old way):** Once the BIOS have identified which drive it should attempt to boot from, it looks at the first sector on that drive. These sectors should contain the Master Boot Record.

   The MBR has two component parts:

   - The boot loader information block (448 bytes)
   - The partition table (64 bytes)

   The boot loader information block is where the first program the computer can run is stored. The partition table stores information about how the drive is logically laid out.

   The MBR has been heavily limited in its design, as it can only occupy the first 512 bytes of space on the drive (which is the size of one physical sector). This limits the tasks the boot loader program is able to do. The execution of the boot loader literally starts from the first byte. As the complexity of systems grew, it became necessary to add "chain boot loading". This allows the MBR to load an another program from elsewhere on the drive into memory. The new program is then executed and continues the boot process.

   If you're familiar with Windows, you may have seen drives labelled as "C:" and "D:" - these represent different logical "partitions" on the drive. These represent partitions defined in that 64-byte partition table.

b. **GPT - The GUID Partition Table (the new way):** The design of the IBM-Compatible BIOS is an old design and has limitations in today's world of hardware. To address this, the United Extensible Firmware Interface (UEFI) was created, along with GPT, a new partition format.

   There are a few advantages to the GPT format, specifically:

   - A Globally-Unique ID that references a partition, rather than a partition number. The MBR only has 64 bytes to store partition information - and each partition definition is 16 bytes. This design allows for unlimited partitions.

➢ The ability to boot from storage devices that are greater than 2 TBs, due to a larger address space to identify sectors on the disk. The MBR simply had no way to address disks greater than 2 TB.
➢ A backup copy of the table that can be used in the event that the primary copy is corrupted. This copy is stored at the ‘end’ of the disk.

There is some compatibility maintained to allow standard PCs that are using old BIOS to boot from a drive that has a GPT on it.

5. **The Bootloader:** The purpose of a bootloader is to load the initial kernel and supporting modules into memory.
6. **Kernel:** The kernel is the main component of any operating system. The kernel acts as the lowest-level intermediary between the hardware on your computer and the applications running on your computer. The kernel abstracts away such resource management tasks as memory and processor allocation. The kernel and other software can access peripherals such as disk drives by way of device drivers. Let us examine the booting process of some popular Operating Systems.

VIDEO LECTURE

5.4.1 Linux Boot Process

The following are the 6 high level stages of a typical Linux boot process[12].

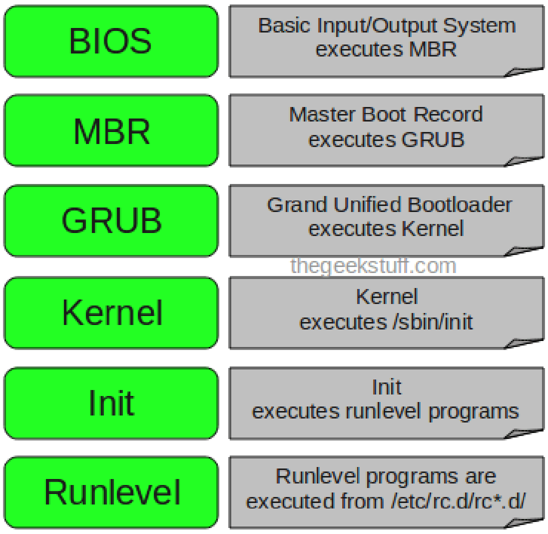


Figure 10: Linux boot process

Setp1: BIOS

- BIOS stands for Basic Input/Output System
- Performs some system integrity checks
- Searches, loads, and executes the boot loader program.
- It looks for boot loader in floppy, cd-rom, or hard drive. You can press a key (typically F12 of F2, but it depends on your system) during the BIOS startup to change the boot sequence.
- Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
- So, in simple terms BIOS loads and executes the MBR boot loader.

Step 2: MBR

- MBR stands for Master Boot Record.
- It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
- MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes.
- It contains information about GRUB (or LILO in old systems).
- So, in simple terms MBR loads and executes the GRUB boot loader.

Step3: GRUB

- GRUB stands for Grand Unified Bootloader.

[12] https://pacesettergraam.wordpress.com/2012/08/11/stages-of-linux-boot-process/

- If you have multiple kernel images installed on your system, you can choose which one to be executed.
- GRUB displays a splash screen, waits for few seconds, if you don't enter anything, it loads the default kernel image as specified in the grub configuration file.
- GRUB has the knowledge of the filesystem (the older Linux loader LILO didn't understand filesystem).
- Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this).
- GRUB just loads and executes Kernel and initrd images.

Step 4: Kernel

- Mounts the root file system as specified in the "root=" in grub.conf
- Kernel executes the /sbin/init program
- Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a 'ps -ef | grep init' and check the pid.
- initrd stands for Initial RAM Disk.
- initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

Step 5: Init

- Looks at the */etc/inittab* file to decide the Linux run level.
- Following are the available run levels
    - 0 – halt
    - 1 – Single user mode
    - 2 – Multiuser, without NFS
    - 3 – Full multiuser mode
    - 4 – unused
    - 5 – X11
    - 6 – reboot
- Init identifies the default initlevel from */etc/inittab* and uses that to load all appropriate program.
- Execute '*grep initdefault /etc/inittab'* on your system to identify the default run level
- If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
- Typically you would set the default run level to either 3 or 5.

Step 6: Runlevel programs

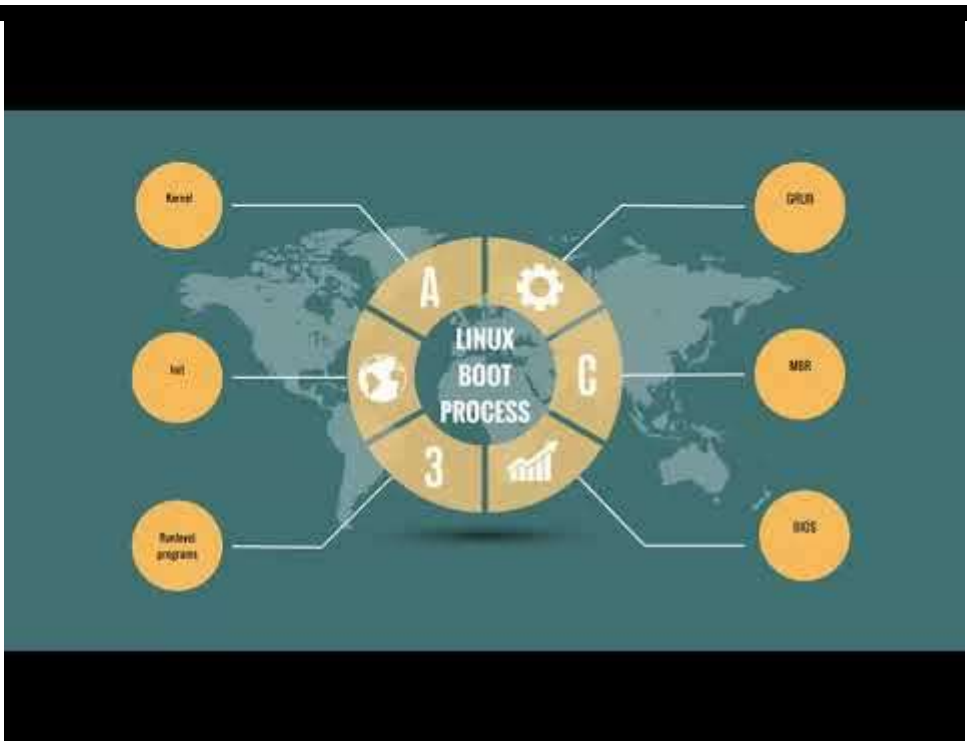
- When the Linux system is booting up, you might see various services getting started. For example, it might say "starting sendmail …. OK". Those are the runlevel programs, executed from the run level directory as defined by your run level.

- Depending on your default init level setting, the system will execute the programs from one of the following directories.
    - Run level 0 – /etc/rc.d/rc0.d/
    - Run level 1 – /etc/rc.d/rc1.d/

- ➢ Run level 2 – /etc/rc.d/rc2.d/
- ➢ Run level 3 – /etc/rc.d/rc3.d/
- ➢ Run level 4 – /etc/rc.d/rc4.d/
- ➢ Run level 5 – /etc/rc.d/rc5.d/
- ➢ Run level 6 – /etc/rc.d/rc6.d/

- Please note that there are also symbolic links available for these directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.
- Under the /etc/rc.d/rc*.d/ directories, you would see programs that start with S and K.
- Programs starts with S are used during startup. S for startup.
- Programs starts with K are used during shutdown. K for kill.
- There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.
- For example, S12syslog is to start the syslog deamon, which has the sequence number of 12. S80sendmail is to start the sendmail daemon, which has the sequence number of 80. So, syslog program will be started before sendmail.

VIDEO LECTURE



5.4.2 Mac OS Boot Sequence

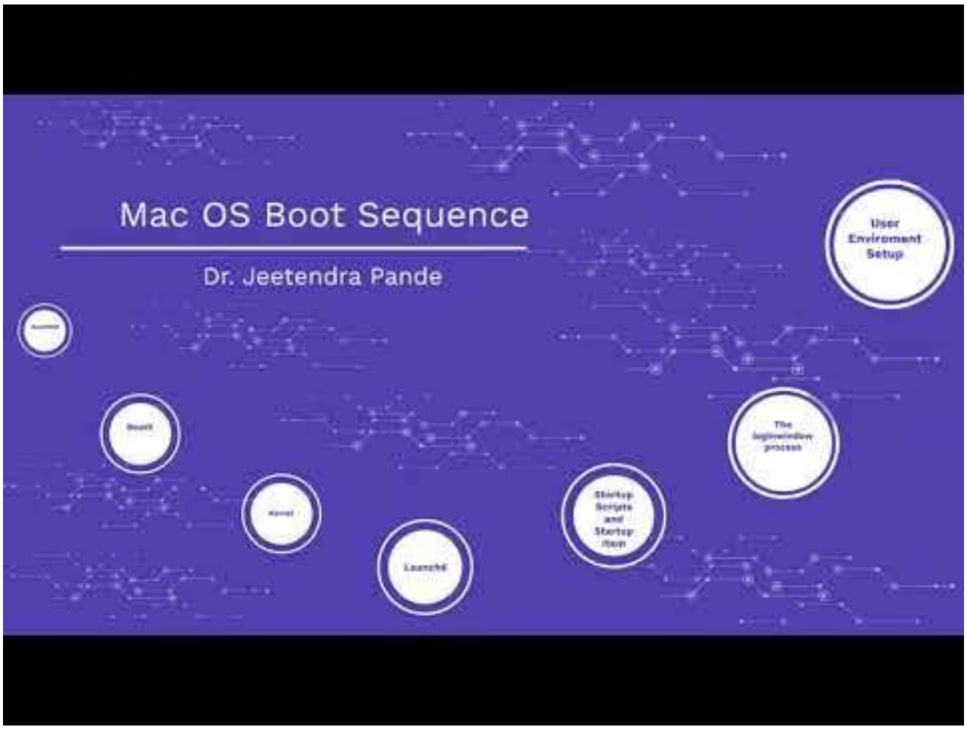
The boot process starts with the activation of BootROM, the basic Macintosh ROM, which performs a Power On Self Test to test hardware essential to startup[13]. On the passing of this

[13] https://en.wikipedia.org/wiki/BootX_(Apple)

test, the startup chime is played and control of the computer is passed to OpenFirmware. OpenFirmware initializes the Random Access Memory, Memory Management Unit and hardware necessary for the ROM's operation. The OpenFirmware then checks settings, stored in NVRAM, and builds a list of all devices on a device tree by gathering their stored FCode information.

On the completion of this task, BootX takes over the startup process configuring the keyboard and display, claiming and reserving memory for various purposes and checking to see if various key combinations are being pressed. After this process has been completed BootX displays the grey Apple logo, spins the spinning wait cursor, and proceeds to load the kernel and some kernel extensions and start the kernel.

VIDEO LECTURE



The detailed description of the above is as follows:

- **BootROM:** As the name suggest, BootROM is a ROM (Read only Memory) which contains boot programmes viz. POST and Open Firmware.
    - POST: Power-On Self Test is the initial process which check the functionality of the basic hardware attached to the computer including RAM.

- Open Firmware: The remaining hardware is initialized by Open Firmware. It also checks all the hardware associated with the systems and builds the initial device tree.

- **BootX:** The BootX initialize the kernel and the drivers required to boot the system from the cached set of device drivers. In case it is not present, it is loaded for /System/Library/Extensions for the same.

- **Kernel:** Once the Kernel is loaded, it initialises the I/O kit which is used to control Input/Output devices. After this, the kernel initiates the launched process.

- **Launchd:** it is process used for bootstrapping and is responsible for starting every system process. It also manages system initialization and starts the loginwindow process. During system initialization the system launchd process automatically starts /System/Library/LaunchDaemons, /Library/LaunchDaemons, /Library/StartupItems, and /etc/rc.local. The launch also manages daemons, a program who manages service request.

- **Startup Scripts and Startup Item:** As soon as /etc/rc.local is executed, it initialises the basic system by performing file-system consistency check and initiating SystemStarter process, a process used for launching startup items. For system configuration related information, it refers to /etc/hostconfig.

- **The loginwindow Process:** this process displays the login screen that allows the user to authenticate, and then sets up and manages the graphical interface user environment based on the account preferences.

- **User Enviroment Setup:** After the user's credentials are authenticated, the user environment setup is performed based on the user's preference.

5.4.3 Boot Sequence in Windows 7

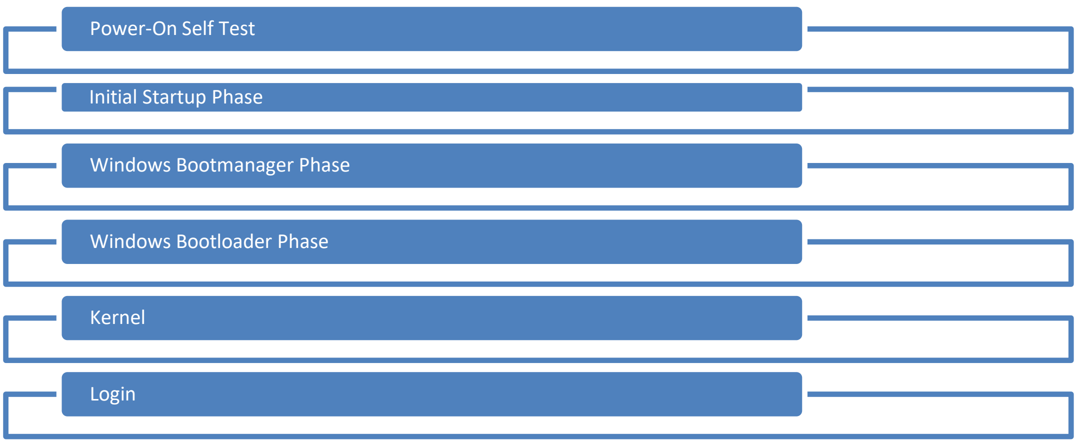


Figure 11: Window 7 boot sequence

1. In the first step, when the system is powered-on the *Basic Input Output System* (BIOS) and RAM is loaded and BIOS performs the hardware diagnostics based by initiating Power-On Self Test(POST).
2. In the second step, the BIOS locates MBR(Master Boot Record), which is located at the first sector of the first hard drive, to find the active drive, bootable partition and reads boot sector.
3. The boot sector loads *bootmgr*, which looks for the active partition on the drive and load *Boot Configuration Data(BCD)* data store. The information stored in BCD to find and load the selected operating system.
4. When the Windows 7 OS is selected, *bootmgr* executes a program called *winload.exe*, which takes the charge of the further process of loading windows. The following screen will appear in the monitor:

Figure 12: Windows login screen

5. The winload.exe starts ntoskrnl.exe which initiates all the necessary files needed to load and run Windows 7 OS.
6. The OS is loaded and *winlogon* is executed to provide the login interface. After authentication, the system settings are loaded based on the users credential and preferences.

VIDEO LECTURE

5.5 SUMMARY

1. A hard disk drive (HDD), hard disk, hard drive or fixed disk is a data storage device used for storing and retrieving digital information using one or more rigid rapidly rotating disks (platters) coated with magnetic material.
2. Data is accessed in a random-access manner, meaning that individual blocks of data can be stored or retrieved in any order rather than sequentially. HDDs retain stored data even when powered off.
3. The primary characteristics of an HDD are its capacity and performance.
4. An HDD records data by magnetizing a thin film of ferromagnetic material on a disk.
5. A typical HDD design consists of a spindle that holds flat circular disks, also called platters, which hold the recorded data.
6. The two most common form factors for modern HDDs are 3.5-inch, for desktop computers, and 2.5-inch, primarily for laptops.
7. HDDs are connected to systems by standard interface cables such as PATA (Parallel ATA), SATA (Serial ATA), USB or SAS (Serial attached SCSI) cables.
8. The purpose of low-level formatting is to divide the disk surface into basic elements viz. tracks, sectors and cylinders.
9. Operating systems use different file systems, so the type of logical formatting will depend on the operating system you install.
10. The partitioning of a hard drive occurs after the drive has been physically formatted but before it is logically formatted.

11. There are three types of partitions: primary partitions, extended partitions and logical drives.
12. Partitioning is the process of writing the sectors that will make up the partition table.
13. The unused space at the end of a file in a file system that uses fixed size.
14. A lost cluster is a series of clusters on the hard disk drive that are not associated with a particular file.
15. A bad sector is a sector on a computer's disk drive or flash memory that is either inaccessible or unwriteable due to permanent damage, such as physical damage to the disk surface or failed flash memory transistors.
16. The boot sector is the first sector of a hard drive (cylinder 0, head 0, sector 1), it contains the main partition table and the code, called the boot loader, which, when loaded into memory, will allow the system to boot up.
17. Firmware is the software that is programmed into Electrically Erasable Programmable Read-Only Memory (EEPROM).
18. The purpose of a bootloader is to load the initial kernel and supporting modules into memory.
19. The kernel is the main component of any operating system. The kernel acts as the lowest-level intermediary between the hardware on your computer and the applications running on your computer.

5.6 CHECK YOUR PROGRESS

1. Fill in the blanks

i. The primary characteristics of an HDD are its ____________ and ____________.
ii. An HDD records data by magnetizing a thin film of ____________ material on a disk.
iii. SCSI stands for ________________________.
iv. A ____________ refers to all the data located on the same track of different platters.
v. ____________ is the process of writing the sectors that will make up the partition table.
vi. BIOS stands for ________________________.
vii. ____________is the software that is programmed into Electrically Erasable Programmable Read-Only Memory.
viii. CMOS stands for ________________________.
ix. The purpose of a ____________is to load the initial kernel and supporting modules into memory.
x. ____________is a development from Silicon Graphics.
xi. IBM has developed a file system for tape called the ________________________.
xii. A ________________________is a file system that acts as a client for a remote file access protocol, providing access to files on a server.

2. State True or False

i. The data is read from the disk by detecting the transitions in magnetization. True
ii. The platters are made from a non-magnetic material, usually aluminum alloy, glass, or ceramic.
iii. The 40-pin IDE/ATA connection transfers 32 bits of data at a time on the data cable.
iv. EIDE was an unofficial update (by Western Digital) to the original IDE standard.
v. EIDE have DMA transfer functionality.
vi. Physical level formatting is also known as high level formatting.
vii. The tracks are the concentric areas written on both sides of a platter.
viii. The partitioning of a hard drive occurs after the drive has been physically formatted but before it is logically formatted.
ix. The partition table stores information about how the drive is logically laid out.
x. The file system is based on management of clusters.
xi. The choice of file system does not depends on the operating system that you are using.
xii. NTFS system provides higher security and better performance than the FAT system.
xiii. Several operating systems coexist on the same machine.
xiv. The BIOS does not have its own memory storage.

5.7 ANSWERS TO CHECK YOUR PROGRESS

1. Fill in the blanks

i. capacity , performance.
ii. ferromagnetic
iii. Small Computer System Interface.
iv. cylinder
v. Partitioning
vi. Basic Input/Output System.
vii. Firmware .
viii. Complimentary Metal Oxide Semiconductor.
ix. bootloader
x. XFS
xi. Linear Tape File System.
xii. network file system

2. State True or False

i. True
ii. True
iii. False
iv. True
v. True

vi. False
vii. True
viii. True
ix. True
x. True
xi. False
xii. True
xiii. True
xiv. False

5.8 FURTHUR READING

Computer Storage. (n.d.). Retrieved Sep. 21, 2015, from http://www.utilizewindows.com/: http://www.utilizewindows.com/pc-fundamentals/storage/333-hardware-and-software-disk-optimization

Carrier, B. File System Forensic Analysis.

Cooper, M. (2014, March). *General overview of the Linux file system*. Retrieved Oct. 22, 2015, from Linux Documentation Project : http://www.tldp.org/LDP/intro-linux/html/sect_03_01.html

Different Types Of PC Hard Disk Drives (HDD). (n.d.). Retrieved Oct. 21, 2015, from SAYPOINT: http://www.saypoint.net/2012/05/different-types-of-pc-hard-disk.html

File Systems: FAT, NTFS, and HFS+. (n.d.). Retrieved Oct. 22, 2015, from www.study.com: http://study.com/academy/lesson/files-systems-fat-ntfs-hfs-and-ffs.html

Hagen, W. V. (2002). *Linux Filesystems.* SAMS.

How Things Work/Hard Drive. (2012, April 2012). Retrieved Oct. 22, 2015, from Wikibooks: https://en.wikibooks.org/wiki/Wikijunior:How_Things_Work/Hard_Drive

Silberschatz, Galvin, & Gagne. (2006). *Operating System Principles.* Wiley.

Singh, A. (2006). *Mac OS X Internals: A Systems Approach.*

Smidsrød, R. (n.d.). *The fundamentals of network booting*. Retrieved Oct. 22, 2015, from Networkboot: http://networkboot.org/fundamentals/.

Stages of linux boot process. (2012, Aug. 11). Retrieved Oct. 21, 2015, from https://pacesettergraam.wordpress.com: https://pacesettergraam.wordpress.com/2012/08/11/stages-of-linux-boot-process/

Thakur, D. (n.d.). *What is Booting? Type of Booting*. Retrieved Oct. 22, 2015, from Computer Notes: http://ecomputernotes.com/fundamental/disk-operating-system/what-is-booting-type-of-booting

Windows Disk File Systems. (2010). General Books LLC.

Witjes, C., & Stremlau, A. (n.d.). *The Windows 7 Boot Process (sbsl)*. Retrieved Oct. 22, 2015, from http://social.technet.microsoft.com/: http://social.technet.microsoft.com/wiki/contents/articles/11341.the-windows-7-boot-process-sbsl.aspx

5.9 MODEL QUESTIONS

1. What is Hard Disk Drive? What are its main characteristics?
2. Explain the working of HDD.
3. Explain various interfaces of HDD in detail.
4. Differentiate between high level formatting and low level formatting.
5. What is cyclic redundancy check (CRC)?
6. Define the terms:
    a. Slack space
    b. Lost cluster
    c. Bad sector
7. What is master boot record?
8. What is booting? Explain the booting process of Window 7 in detail.
9. What is a file system? Why it is used?
10. Give the details of file systems that different Operating System supports.
11. What is journaling?
12. What are different types of File Systems? Explain in detail.
13. What is flat file system?

REFERENCES, ARTICLE SOURCE & CONTRIBUTORS

Bad Sector. (2015, Sep. 26). Retrieved Oct. 21, 2015, from Wikipedia: https://en.wikipedia.org/wiki/Bad_sector

BootX (Apple). (2015, May 25). Retrieved Oct. 21, 2015, from Wikipedia: https://en.wikipedia.org/wiki/BootX_(Apple)

Computer Storage. (n.d.). Retrieved Sep. 21, 2015, from http://www.utilizewindows.com/: http://www.utilizewindows.com/pc-fundamentals/storage/333-hardware-and-software-disk-optimization

Different Types Of PC Hard Disk Drives (HDD). (n.d.). Retrieved Oct. 21, 2015, from SAYPOINT: http://www.saypoint.net/2012/05/different-types-of-pc-hard-disk.html

File system. (2015, Oct. 18). Retrieved Oct. 21, 2015, from Wikipedia: https://en.wikipedia.org/wiki/File_system

File systems. (2014, Jan.). Retrieved Oct. 21, 2015, from https://yakking.branchable.com/: https://yakking.branchable.com/posts/filesystems/

Formatting - Formatting a hard drive. (2015, Oct. 01). Retrieved Oct. 21, 2015, from ccm.net: http://ccm.net/contents/626-formatting-formatting-a-hard-drive

Formatting - Formatting a hard drive. (2015, Sep.). Retrieved Oct. 21, 2015, from ccm.net: http://ccm.net/contents/626-formatting-formatting-a-hard-drive

Glossary of digital forensics terms. (2015, Sep. 09). Retrieved Oct. 21, 2015, from Wikipedia: https://en.wikipedia.org/wiki/Glossary_of_digital_forensics_terms

Hard disk drive. (2015, Oct. 16). Retrieved Oct. 21, 2015, from Wikipedia: https://en.wikipedia.org/wiki/Hard_disk_drive

Stages of linux boot process. (2012, Aug. 11). Retrieved Oct. 21, 2015, from https://pacesettergraam.wordpress.com: https://pacesettergraam.wordpress.com/2012/08/11/stages-of-linux-boot-process/

The file system. (2015). Retrieved Oct. 21, 2015, from ccm.net: http://ccm.net/contents/624-the-file-system

Vig, A. (2013). *The boot process*. Retrieved Oct. 21, 2015, from http://ops-school.readthedocs.org: http://ops-school.readthedocs.org/en/latest/boot_process_101.html

EXPERT PANEL

Dr. Jeetendra Pande, Associate Professor- Computer Science, School of Computer Science & IT, Uttarakhand Open University, Haldwani

Dr. Ajay Prasad, Sr. Associate Professor, University of Petroleum and Energy Studies, Dehradun

Dr. Akashdeep Bharadwaj, Professor, University of Petroleum and Energy Studies, Dehradun

Mr. Sridhar Chandramohan Iyer, Assistant Professor- Universal College of Engineering, Kaman, Vasai, University of Mumbai

Mr. Rishikesh Ojha, Digital Forensics and eDiscovery Expert

Ms. Priyanka Tewari, IT Consultant

Mr. Ketan Joglekar, Assistant Professor, GJ College, Maharastra

Dr. Ashutosh Kumar Bhatt, Associate Professor, Uttarakhand Open University, Haldwani

Dr. Sangram Panigrahi, Assistant Professor, Siksha 'O' Anusandhan, Bhubaneswar

This MOOC has been prepared with the support of