Unit – 1 "Software Fundamentals"

Software:

- Software is a set of instructions, data or programs used to operate computer system and execute specific tasks.
- Software is a collection of instructions that enable the user to interact with a computer, its hardware, or perform any task.
- It is the opposite of hardware, which describes the physical aspects of a computer.
- Software is a generic term used to refer to applications, scripts and programs that run on a device.

Hardware:

- Computer hardware includes the physical parts of a computer, such as the central processing unit, monitor, mouse, keyboard, computer data storage, graphics card, sound card, speakers and motherboard.

Relationship between Hardware & Software:

- Hardware and software are mutually dependent on each other. Both of them must work together to make a computer produce a useful output.
- Software cannot be utilized without supporting hardware. Hardware without set of programs to operate upon cannot be utilized and is useless.
- To get a particular job done on the computer, relevant software should be loaded into the hardware.
- Hardware is a one-time expense. Software development is very expensive and is a continuing expense.
- Different software applications can be loaded on a hardware to run different jobs.
- A software acts as an interface between the user and the hardware.
- If hardware is the 'heart' of a computer system, then software is its 'soul'. Both are complimentary to each other.

System Software:

- ➢ System Software is a set of programs that control and manage the operations of computer hardware.
- ➢ It also helps application programs to execute correctly.
- ➢ System Software are designed to control the operation and extend the processing functionalities of a computer system.
- ➢ System software makes the operation of a computer faster, effective, and secure.
- ➢ Example: Operating systems like Android, Linux, iOS, Windows, etc.
- ➢ Programs included in a system software package are called system program and the programmers who prepare them are called system programmers.
- ➢ The **BIOS (basic input/output system)** gets the computer system started after you turn it on and manages the data flow between the operating system and attached devices such as the hard disk, video adapter, keyboard, mouse, and printer.
- ➢ The **boot program** loads the operating system into the computer's main memory or random access memory (RAM).
- ➢ An **assembler** takes basic computer instructions and converts them into a pattern of bits that the computer's processor can use to perform its basic operations.
- ➢ Other examples of system software are operating system, programming language, translator, utility programs and communication software.

Application Software:

- ➢ Application Software is a program that does real work for the user. It is mostly created to perform a specific task for a user.
- ➢ Application Software acts as a mediator between the end-user and System Software. It is also known as an application package.

Unit – 1 "Software Fundamentals"

- ➤ This type of software is written using a high-level language like C, Java, VB. Net, etc.
- ➤ It is user-specific and is designed to meet the requirements of the user.
- ➤ You can also install multiple Application Software on a single System Software.
- ➤ You can store this kind of software on CDs, DVDs, flash derive, or any portable storage devices.
- ➤ Example: Word-processing, Spreadsheet, Database, etc.
- ➤ Application Software is a program that does real work for the user.
- ➤ It is mostly created to perform a specific task for a user.
- ➤ Application Software acts as a mediator between the end-user and System Software. It is also known as an application package.

Point of difference	System Software	Application Software
1. Definition	System Software maintain the system resources and give the path for application software to run.	Application software is built for specific tasks.
2. Language	Low level languages are used to write the system software.	High level languages are used to write the application software.
3. Purpose	Its a general purpose software.	While its a specific purpose software.
4. System running	Without system software, system can't run.	While without application software system always runs.
5. Working	System software runs when system is turned on and stop when system is turned off.	While application software runs as per the user's request.
6. Programming	System Software programming is complex than application software.	Application software programming is simpler as comparison to system software.
7. Examples	Operating systems	Notepad, Photoshop. Microsoft office, etc.

Different Types of Application Software:

1. Application Suite: Has multiple applications bundled together. Related functions, features and user interfaces interact with each other.

2. **Enterprise Software:** Addresses an organization's needs and data flow in a huge distributed environment.
3. **Enterprise Infrastructure Software:** Provides capabilities required to support enterprise software systems.
4. **Information Worker Software:** Addresses individual needs required to manage and create information for individual projects within departments.
5. **Content Access Software:** Used to access content and addresses a desire for published digital content and entertainment.
6. **Educational Software:** Provides content intended for use by students.
7. **Media Development Software:** Addresses individual needs to generate and print electronic media for others to consume

Various Examples Of Application Software Are:

➢ Word processing software
➢ Database programs
➢ Entertainment software
➢ Business software
➢ Educational software
➢ Computer-aided design(CAD) software
➢ Spreadsheet software etc.

Stand Alone Applications Software:

➢ Stand-alone software is an application that does not come bundled with (or require another software package) in order to run.
➢ Essentially, it's software that can "stand on its own" without help from the Internet or another computer process.
➢ Stand-alone software is installed on your computer, which is very different than online software that runs via your internet browser.

> ➢ Quicken and Microsoft Money is two examples of stand-alone software packages that don't require anything more than the operating system on your computer (whether it's Windows or another operating system).

Types of Stand Alone Software:

> ➢ **Software that runs on its own without an internet connection:** An example is an anti-virus software which can be installed on your computer after inserting an installation disc into your computer or laptop disc drive or portable disc drive.
> ➢ **Software that isn't part of a bundle:** Many times, this is software that comes with computer hardware or an electronic device. A bundle can also mean several software packages sold together, but most often, you will receive bundled software when you purchase a new computer. A bundle can include antivirus, money management, photo-related and other software.
> ➢ **A program that runs separately from all other computer processes:** This is a program that doesn't rely on other software in order to function.
> ➢ **A portable application that doesn't need to be installed on your computer:** An example is a software program that runs on its own using a disc or flash drive. When not in use, you can easily eject the disc or flash drive. The program, therefore, is self-contained and, conveniently, doesn't take up space on your hard drive.
> ➢ **An expansion pack for gaming:** If you enjoy playing video games online, you know that gaming software offers expansion packs. In most cases, these packs are "add-ons" to the game. Add-ons might be new weapons or other new items that can be incorporated into an existing game.

Single-User Operating System:

➤ Single user operating system is designed especially for home computers. A single user can access the computer at a particular time.

➤ The single-user operating system allows permission to access your personal computer at a time by a single user, but sometimes it can support multiple profiles.

➤ It can also be used in official work and other environments as well.

➤ The computers based on this operating system have a single processor to execute only a single program at all times.

➤ This system provides all the resources such as CPU, and I/O devices, to a single user at a time.

➤ The operating system for those computers which support only one computer. In this operating system, another user cannot interact with another working user.

➤ **Single-User Single-Tasking:** Operating system allows a single user to execute one program at a particular time. Some functions such as printing a document, and downloading images and videos are performed in one given frame. Ex: MS-DOS

➤ **Single-User Multi-Tasking:** Operating system allows a single user to execute multiple programs at the same time, the single user can perform multiple tasks at a time. This type of operating system is found on personal desktops and laptops. The most popular single-user multi-tasking is Microsoft windows. This single-user multi-tasking can be pre-emptive or cooperative. Ex: Windows, Mac OS

Multi-User Operating System:

➤ A multi-user operating system is an operating system that permits several users to access a single system running to a single operating system.

- These systems are frequently quite complex, and they must manage the tasks that the various users connected to them require.
- Users will usually sit at terminals or computers connected to the system via a network and other system machines like printers.
- A multi-user operating system varies from a connected single-user operating system in that each user accesses the same operating system from different machines.
- The main goal of developing a multi-user operating system is to use it for time-sharing and batch processing on mainframe systems.
- This multi-user operating system is now often used in large organizations, the government sector, educational institutions like large universities, and on servers' side such as Ubuntu Server or Windows Server.
- These servers allow several users to access the operating system, and hardware at the same time.
- It is usually responsible for handling memory and processing for other running programs, identifying and using system hardware, and efficiently handling user interaction and data requests.
- It's especially important for an operating system, a multi-user operating system because several users rely on the system to function properly at the same time.

Multi-User Application Software:

- Multi-user software is software that allows access by multiple users of a computer.
- **Time-sharing systems** are multi-user systems.
  - o Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing.

- ➢ Most **batch processing systems** for mainframe computers may also be considered "multi-user", to avoid leaving the CPU idle while it waits for I/O operations to complete.
    - o The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.
- ➢ However, the term "multitasking" is more common in this context.

Client-Server Architecture:

- ➢ Client-Server architecture is a network architecture in which each computer or process over the network is either a client or a server.
- ➢ **Servers** are powerful computers or processes dedicated to managing disk drivers (file servers), print servers, or network traffic.
- ➢ **Client** are PCs or workstations on which the users run applications. Clients rely on servers for resources (i.e. files, devices and processing power).

Client — Request → Response — Server
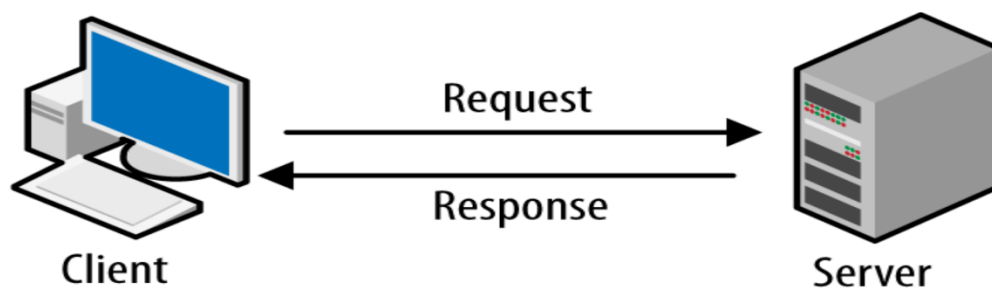
- ➢ Client/server architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client.

➤ This type of architecture has one or more client computers connected to a central server over a network or Internet connection. This system shares computing resources.

➤ Client/server architecture may also be referred to as a networking computing model because all the requests and services are delivered over a network.

➤ Client/server architecture is a producer-consumer computing architecture where the server acts as the producer and the client as a consumer.

➤ The server houses and provides high-end, computing-intensive services to the client on demand. These services can include applications access, storage, file sharing, printer access and/or direct access to the server's raw computing power.
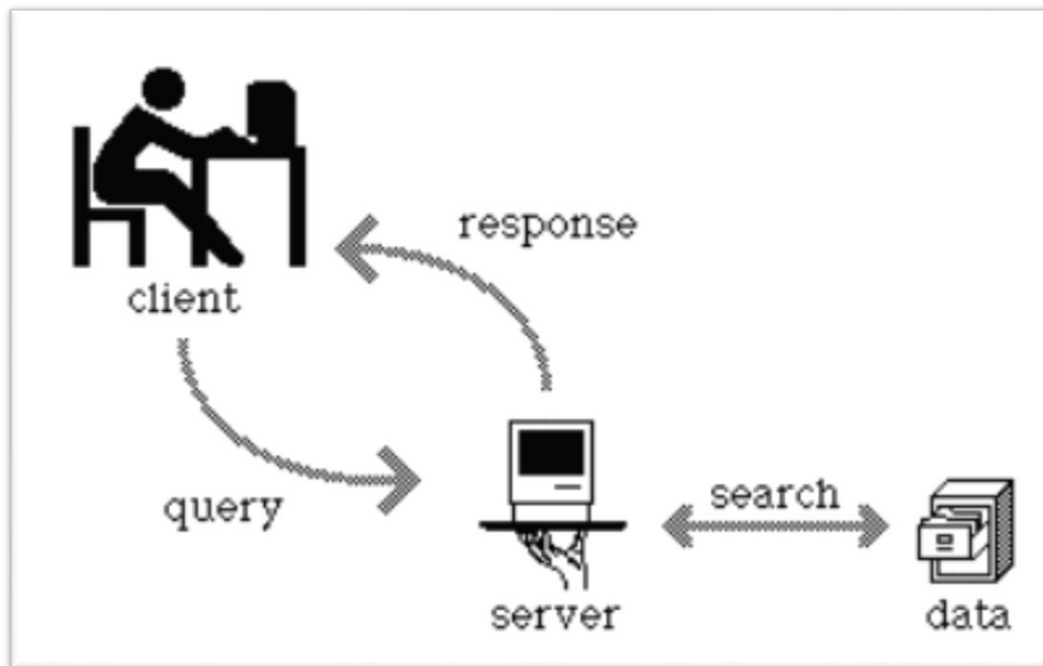
➤ Client/server architecture works when the client computer sends a resource or process request to the server over the network connection, which is then processed and delivered to the client.

- ➢ A server computer can manage several clients simultaneously, whereas one client can be connected to several servers at a time, each providing a different set of services.
- ➢ In its simplest form, the Internet is also based on client/server architecture where the Web server serves many simultaneous users with Web page and or website data.

Dedicated Servers:

- ➢ It is a computer set aside for a specific task, such as hosting a resource intensive application or website.
- ➢ They does the task of managing disk drives, print servers or network traffic.
- ➢ Ex: Novell (it provides wide networking services ranging).

Non-Dedicated Servers:

- ➢ It is a computer that allows to use itself as a workstation and provides facilities of sharing its resources to other computers in the network.
- ➢ They does a task of workstations/client as well as servers.
- ➢ Ex: Linux.