Lecture 4

Learning outcomes:

- *Software basic concepts*
- *Tasks of the operating system*

Software basic concepts

- Computer Software refers to the set of programs and instructions that perform several specific operations.

- ***For example***:

✓The antivirus that we use to protect our computer system is a type of Software.

✓The media players that we use to play multimedia files such as movies, music etc. are Software.

✓The Microsoft Office we use to edit the documents is a Software.

Why need software ?

❖***Software is needed for a variety of reasons, including:***

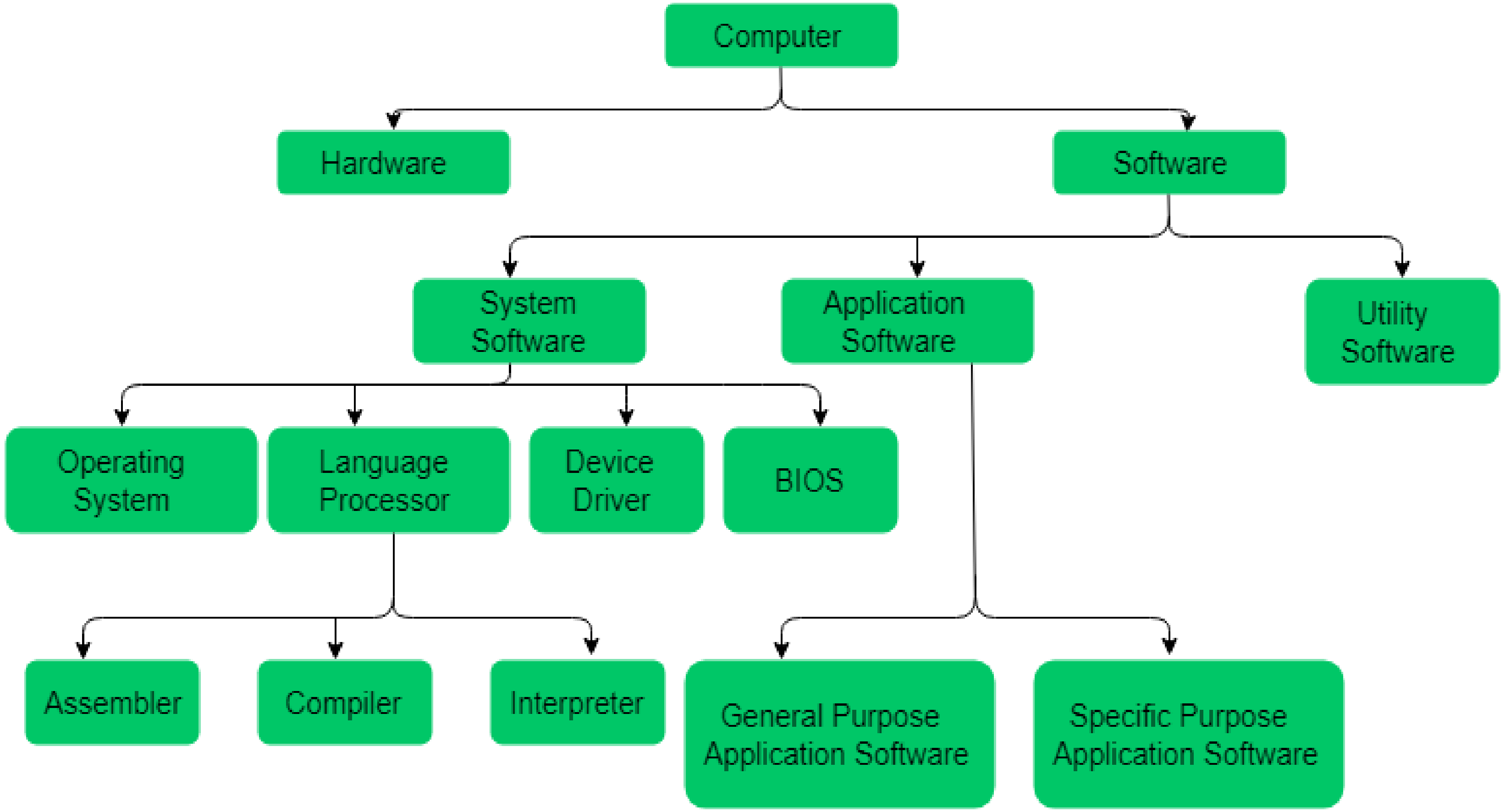
1. Automation: Software can automate repetitive and mundane tasks, allowing users to focus on more complex and strategic work.
2. Efficiency: Software can increase efficiency by streamlining workflows, reducing errors, and improving accuracy.
3. Scalability: Software can be designed to handle large volumes of data and users, making it easier to scale as a business grows.
4. Innovation: Software can enable new products and services that were previously impossible or difficult to achieve.
5. Communication: Software can facilitate communication and collaboration among team members and with customers, partners, and suppliers.
6. Decision-making: Software can provide data insights and analytics to support informed decision-making.

❖Depending on its use and area of implementation, Software can be divided into 3 major types:

1.System Software

2.Application Software

3.Utility Software



System Software

- ❖These are the software that directly allows the user to interact with the hardware components of a computer system. As the humans and machines follow different languages, there has to be an interface that will allow the users to interact with the core system, this interface is provided by the software. The system software can be called the main or the alpha software of a computer system as it handles the major portion of running a hardware. This System Software can be further divided into four major types:
- ❑**The Operating System** – It is the main program that governs and maintains the inter-cooperation of the components of a computer system. For eg., Microsoft Windows, Linux, Mac OS etc.
- ❑**The Language Processor** – The hardware components present in the computer system does not understand human language.

There are three types of languages involved in the world of human-machine interaction:

1. Machine-Level Language:

➢The Machine-Level Language only understand the digital signals or the binary codes or the binary language which consist of strings of 0’s and 1’s. These are totally machine dependent language.

➢Machine-Level Language is the final form of any program, regardless of what language it was originally written in. This is what the computer ultimately executes.

2. High-Level Language:

➢These are the simple English statements, that humans use to program and code as it is easy to read and understand to the human world. For eg., Java, C, C++, Python etc.

➢High-Level Language is translated into Assembly-Level Language or directly into machine code(Machine-Level Language) by a compiler or interpreter.

3. Assembly-Level Language:

- These are the Low-Level Language(LLL), that forms a correspondence between machine level instruction and general assembly level statements. Assembly language uses a mnemonics to represent each low-level machine instruction or operation-code also called the op-codes. For eg., ADD instruction is used to add two entities, the HALT instruction is used to stop a process etc. It is a machine dependent language and varies from processor to processor.
- Assembly-Level Language serves as an intermediate step between human-readable code and machine code (Machine-Level Language). It offers more control than high-level languages but requires knowledge of the hardware.
- Programmers use assembly language to optimize critical sections of code for maximum performance, especially in real-time systems, embedded systems, or performance-critical software like video games or operating system kernels.
- Assembly code is translated into machine code by a program called an assembler. The assembler converts assembly instructions into the corresponding binary code for the processor to execute.

- **Assembler:** This language processor is used to convert the assembly language into machine level language.

- **Compiler:** This language processor is used to convert High-Level Language into machine level language in one go, thus execution time is fast. The error detection is difficult in a compiler. Programming Languages like C, C++ and Scala use compiler.

- **Interpreter:** This language processor is also used to convert High-Level Language into machine level language line-by-line, thus execution time is slow. Error-detection is easier in an interpreter as it reports as soon as a bug is caught and restarts the process. This consumes unnecessary memory. Programming Languages like Python, Ruby and Java uses an interpreter.

- ❑**The Device Drivers** – The device drivers and the device programs or the system software that acts as an interface between the various Input-Output device and the users or the operating system. For eg., the Printers, Web cameras come with a driver disk that is needed to be installed into the system to make the device run in the system.

- ❑**The BIOS** – It stands for Basic Input Output System and is a small firmware, that controls the peripheral or the input-output devices attached to the system. This software is also responsible for starting the OS or initiating the booting process.

Application Software

❖These are the basic software used to run to accomplish a particular action and task. These are the dedicated software, dedicated to performing simple and single tasks. For eg., a single software cannot serve to both the reservation system and banking system. These are divided into two types:

➢The General Purpose of Application Software: These are the types of application software that comes in-built and ready to use, manufactured by some company or someone. For eg.

✓Microsoft Excel – Used to prepare excel sheets.

✓VLC Media Player – Used to play audio/video files.

✓Adobe Photoshop – Used for designing and animation and many more.

- ➢The Specific Purpose of Application Software: These are the type of software that is customizable and mostly used in real-time or business environment. For eg.

- ✓Ticket Reservation System
- ✓Healthcare Management System
- ✓Hotel Management System
- ✓Payroll Management System

Utility Software

❖ These are the most basic type of software which provides high utility to the user and the system. These perform the basic but daily need tasks. For eg.

✓Antivirus Softwares: These provide protection to the computer system from unwanted malware and viruses. For eg., QuickHeal, McAfee etc.

✓Disk Defragmenter Tools: These help the users to analyse the bad sectors of the disk and rearrange the files in a proper order.

✓Text-editors: These help the users to take regular notes and create basic text files. For eg., Notepad, Gedit etc.

Advantages of software :

- There are several advantages of software, including:

- **Increased efficiency:** Software can automate repetitive tasks, reduce errors, and streamline workflows, leading to increased productivity and efficiency.
- **Improved accuracy:** Software can perform complex calculations and data processing with high accuracy and consistency, reducing the risk of human error.
- **Scalability:** Software can be designed to handle large volumes of data and users, making it easier to scale as a business grows.

- **Cost-effectiveness:** Software can reduce costs by eliminating the need for manual processes, reducing errors and waste, and improving resource allocation.
- **Innovation:** Software can enable the development of new products and services, driving innovation and competitiveness.
- **Better communication and collaboration:** Software can facilitate communication and collaboration among team members and with customers, partners, and suppliers, leading to better outcomes and stronger relationships.
- **Data analysis:** Software can provide data insights and analytics to support informed decision-making and drive business growth.

Dis-advantages of software :

❖There are also some potential disadvantages to consider, including:

- ➢**Cost:** Software development can be expensive, especially for complex or customized applications.
- ➢**Security vulnerabilities:** Software can be vulnerable to security threats such as hacking, viruses, and malware, which can compromise data and systems.
- ➢**Compatibility issues:** Software may not be compatible with all hardware or operating systems, which can limit its usefulness and require additional investments.
- ➢**Maintenance and updates:** Software requires ongoing maintenance and updates to ensure optimal performance and security, which can be time-consuming and costly.
- ➢**Learning curve:** New software may require significant training and time to learn, which can impact productivity and require additional resources.
- ➢**Dependence on technology:** Over-reliance on software can create a dependence on technology, which can limit flexibility and creativity and increase the risk of disruptions.

Difference Between System Software and Application Software

System Software	***Application Software***
It is designed to manage the resources of the computer system, like memory and process management, etc.	It is designed to fulfill the requirements of the user for performing specific tasks.
Written in a low-level language.	Written in a high-level language.
Less interactive for the users.	More interactive for the users.
System software plays vital role for the effective functioning of a system.	Application software is not so important for the functioning of the system, as it is task specific.
It is independent of the application software to run.	It needs system software to run.

Tasks of the operating system

Operating System

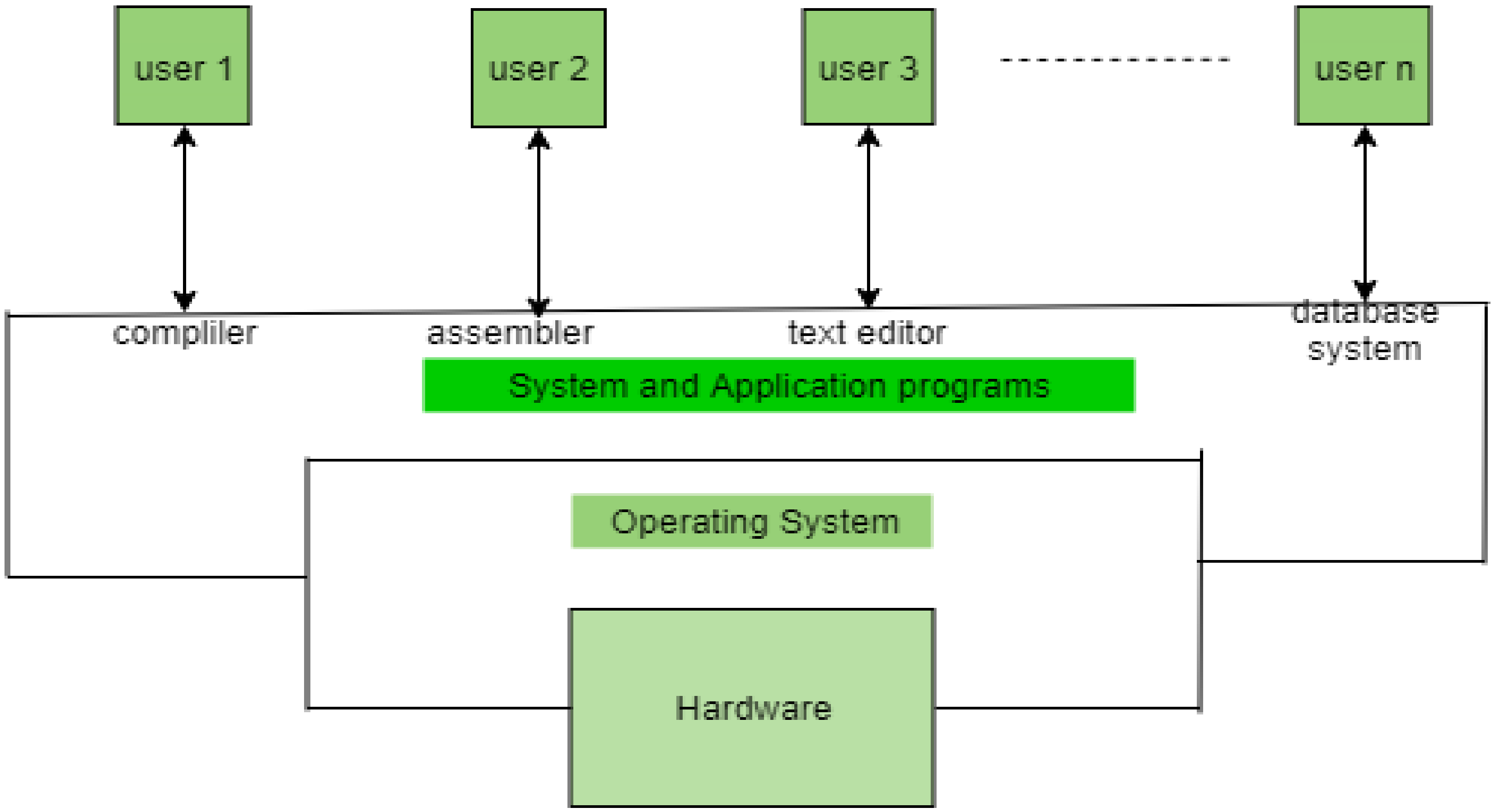
- An operating system (OS) is a program that acts as an interface between the system hardware and the user. It is software that manages and controls the entire set of resources and effectively utilizes every part of a computer.

- Moreover, it handles all the interactions between the software and the hardware. All the working of a computer system depends on the OS at the base level.

- Further, it performs all the functions like handling memory, processes, the interaction between hardware and software, etc.

- The fundamental goal of an Operating System is to execute user programs and to make tasks easier. Various application programs along with hardware systems are used to perform this work.

The figure shows how OS acts as a medium between hardware units and application programs.



Why are Operating Systems Used?

- Operating System is used as a communication channel between the Computer hardware and the user. It works as an intermediate between System Hardware and End-User. Operating System handles the following responsibilities:

- ➢It controls all the computer resources.
- ➢It provides valuable services to user programs.
- ➢It coordinates the execution of user programs.
- ➢It provides resources for user programs.
- ➢It provides an interface (virtual machine) to the user.
- ➢It hides the complexity of software.
- ➢It supports multiple execution modes.
- ➢It monitors the execution of user programs to prevent errors.

Primary Functions of Operating Systems

❖The primary functions of an operating system can be divided into five categories:

- Process Management
- Memory Management
- File Systems Management
- Device Management
- Security and Privacy

Process management

- ❖This involves managing every process that the system hardware executes. The OS decides the order in which processes are performed, prioritizing them based on variables like efficiency and necessity.
- ❖This deals with the management of the Central Processing Unit (CPU). The operating system takes care of the allotment of CPU time to different processes. When a process finishes its CPU processing after executing for the allotted time period, this is called scheduling. There is various type of scheduling techniques that are used by the operating systems:

- ➢**Shortest Job First(SJF):** The process which needs the shortest CPU time is scheduled first.
- ➢**Round Robin Scheduling:** Each process is assigned a fixed CPU execution time in a cyclic way.
- ➢**Priority-Based Scheduling (Non-Preemptive):** In this scheduling, processes are scheduled according to their priorities, i.e., the highest priority process is scheduled first. If the priorities of the two processes match, then schedule according to arrival time.

Priority-Based Scheduling (Non-Preemptive):

- When referring to a lower priority number in the context of Priority-Based Scheduling, it means that a process is considered to be more important and should be executed before other processes.

- In many scheduling systems, lower numbers indicate higher priority, so a process with a priority number of **1** will be considered more urgent or important than a process with a priority number of **3**, for example.

How to Assign Priorities:

- Priorities can be assigned in a number of ways, depending on the type of system and the needs of the user or application. Here's how priorities might be assigned:

- ***System-Assigned Priorities:***

- Operating System or system scheduler can automatically assign priorities to processes based on their type.
- For example, kernel-level processes (critical system tasks) may automatically be given a higher priority (lower number) than user-level processes.
- Real-time tasks or tasks that are critical to system performance, like handling interrupts or managing input/output devices, are given higher priority than non-essential tasks.

➢***User-Assigned Priorities:***

- In some systems, a user or administrator can manually assign priority levels to processes.

- This is common in systems that support real-time applications, where certain tasks need to be completed within specific time constraints.

- For example, a user may assign a high priority to a video rendering task over a background file transfer.

➢***Application-Assigned Priorities:***

- Programs or applications can assign priorities to their own threads or processes based on the specific needs of the software.

- For example, in a web server, the thread handling incoming requests might be given a higher priority than threads performing background cleanup tasks.

Examples of Assigning Priorities:

- Process A (Handling system updates) → Assigned priority 1 (very high priority).

- Process B (Running a background music player) → Assigned priority 4 (low priority).

- Process C (Compiling code) → Assigned priority 2 (high priority).

Memory management

- ❖The operating system oversees your computer's memory, allocating space to processes when they run and deallocating them when they halt.
- ❖An Operating System performs the following activities for Memory Management:

- ✓It keeps track of primary memory, i.e., which bytes of memory are used by which user program. The memory addresses that have already been allocated and the memory addresses of the memory that has not yet been used.
- ✓In multiprogramming, the OS decides the order in which processes are granted memory access, and for how long.
- ✓It Allocates the memory to a process when the process requests it and deallocates the memory when the process has terminated or is performing an I/O operation.

File systems management

- The OS also manages your data files. It uses a systematic way to store, arrange, and retrieve files and data. Furthermore, it keeps track of all the data, ensuring that it doesn't get lost or overwritten accidentally.
- The operating system manages the files, folders, and directory systems on a computer. Any data on a computer is stored in the form of files and the operating system keeps the information about all of them using the File Allocation Table (FAT), or a data structure called an inode in Linux. The FAT stores general information about files like filename, type (text or binary), size, starting address, and access mode (sequential/indexed sequential/direct/relative). The file manager of the operating system helps to create, edit, copy, allocate memory to the files, and also updates the FAT. The operating system also takes care that files are opened with proper access rights to read or edit them.

Device management

- An operating system manages all input, output, and storage devices. The system ensures that your devices run as effectively as possible and that their essential data is stored safely.

- The Operating System communicates with the hardware and the attached devices and maintains a balance between them and the CPU. This is all the more important because the CPU processing speed is much higher than that of I/O devices.

Security and privacy

- Maintaining system security is crucial. The OS ensures the security of the system by limiting hardware access to trusted applications and programs.

- It uses methods like password protection and controlled user access to uphold privacy and data integrity.

Operating System Services

- The main purpose of the operating system is to provide an environment for the execution of programs. Thus, an operating system provides certain services to programs and the users of those programs.

1. Program Execution

- The operating system provides a convenient environment where users can run their programs.
- The operating system performs memory allocation to programs, and load them into appropriate location so that they can execute. The users do not have to worry about all these tasks.

2. I/O Operations

- In order to execute a program, it usually requires an I/O operation. For example, it may need to read a file and print the output.
- When all these I/O operations are performed users cannot control I/O devices.
- All I/O is performed under the control of the operating system.

3. Communication

- The various processes executing on a system may need to communicate in order to exchange data or information.
- The operating system provides this communication by using a facility for message passing. In message passing packets of information are moved between processes by the operating system.

Types of Operating Systems

❖There are various types of operating systems. These are what they are:

❑ Batch OS

➢The jobs and tasks are not forwarded to the CPU directly in this system by the OS. It functions by combining similar job types into a single category. We also refer to this group as a “batch.” Hence, batch operating system. The payroll system, a bank statement, etc. are some examples.

❑ Time-Shared OS

➢Time-shared OS refers to a system that runs multiple tasks simultaneously. Because the system can run various tasks concurrently as needed. As a result, they each share CPU time individually. We also refer to it as multitasking as a result. Quantum is the amount of time that each task receives.

❑ Distributed OS

➢There are multiple CPUs present in this system. All of the processors receive equal task distribution from the OS. There is no shared memory or clock time between the processors. Through various communication channels, OS manages all of its communication.

❑ Network OS

➢A server is connected to a variety of systems in these OS. Sharing resources like files, printers, applications, etc. is made possible by this system. Additionally, it provides the ability to manage these resources.