Course Contents

Unit-03: Computer Software (6 Hrs.)

- Introduction; Types of Software; System Software; Application Software; Software Acquisition; Operating System (Introduction, Objectives of Operating System, Types of OS, Functions of OS, Process Management, Memory Management, File Management, Device Management, Protection and Security, User Interface, Examples of Operating Systems)

Course Contents

Unit-03: Computer Software

- **Introduction**

- **Types of Software: System Software Vs Application Software**

- **Software Acquisition**

- **Operating System**

  - **Introduction**

  - **Objectives of Operating System**

  - **Types of OS**

  - **Functions of OS (Process Management, Memory Management, File Management, Device Management, Protection and Security, User Interface)**

- **Examples of Operating Systems**

Computer Software

Chapter Objectives:

➢ The computer, as a machine, can do nothing without the software.

➢ Software is required for the functioning of computer.

➢ Software programs instruct computer about the actions to be performed, so as to get the desired output.

So purpose of this chapter is to introduce you to the different categories of software.

Computer Software

2.1. Introduction to software

- Computer is made with hardware and Software.

- Hardware is the physical equipments that are necessary for performing various operations i.e. reading and processing data, storing results and providing output to the users in a desired form.

- Hardware can't perform any task on its own and needs to be instructed about the task to be performed.

- So to enable the hardware to work, a set of computer programs are required and is called Computer Software.
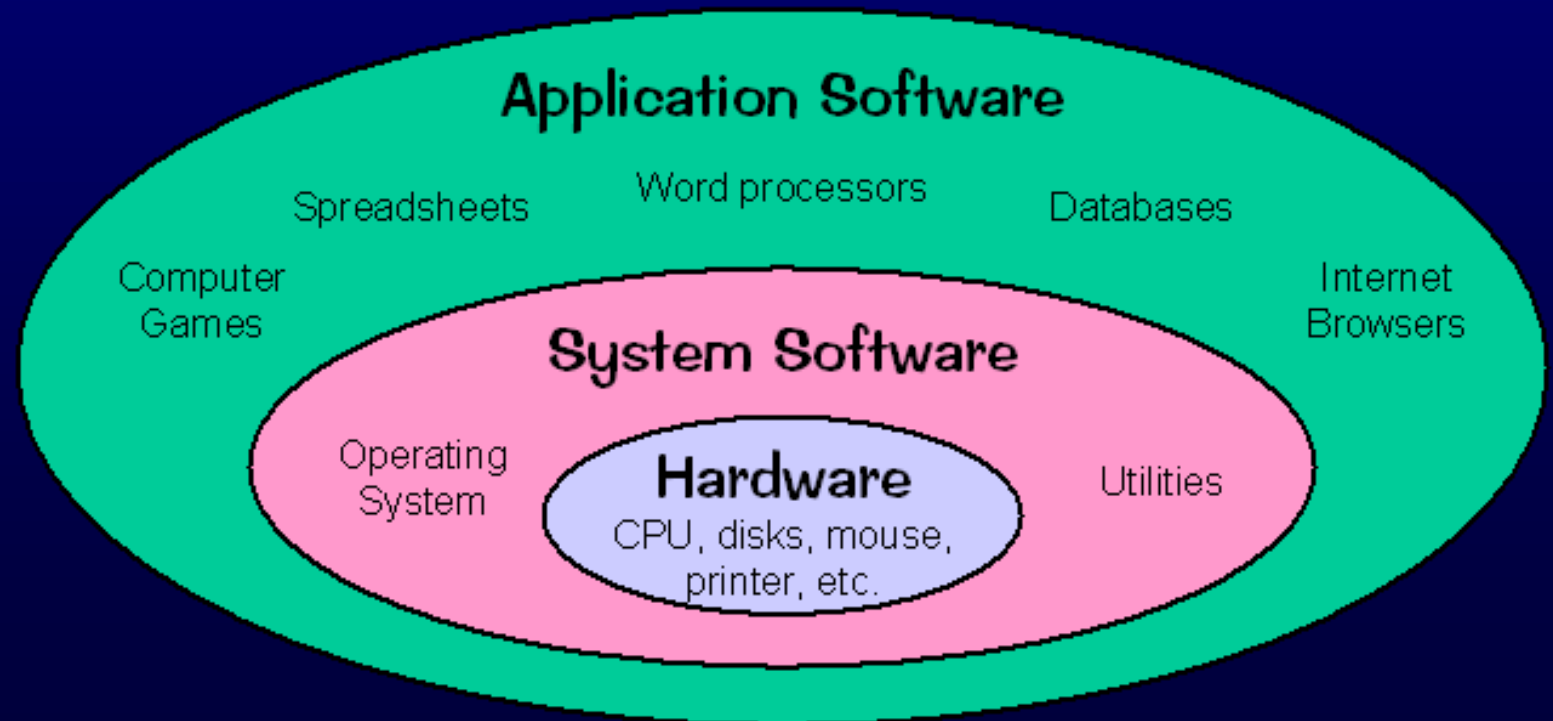
Computer Software

2.1. Introduction to software

- Software refers to a set of computer programs that are required to enable the hardware to work and perform operations i.e. reading and processing data, storing results and providing output effectively.

- A computer software(program) is basically a set of logical instructions written in a computer programming language that tells the computer how to accomplish a task.

- Different sets of software can be loaded on the same hardware to perform different kinds of tasks.

Computer Software

Software broadly classified in 2 categories:

1. **System software**

2. **Application Software**

Computer Software

System Software:

System sw is a group of programs that direct the internal operations of computer system such as controlling I/O devices, managing the storage area within the computer etc.
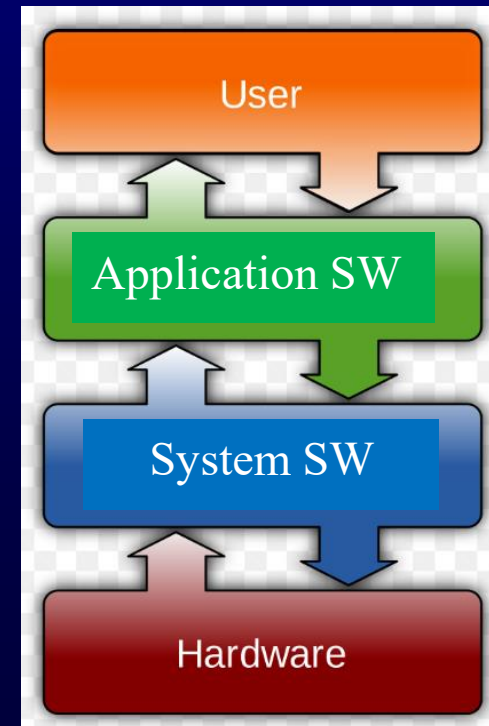
Functions of system software are:

- To provide basic functionality to computer,
- To control computer hardware, and
- To act as an interface between user, application sw and computer hardware.

System sw also provides the services(tools) for the development and execution of application **software.**

Computer Software

System software Vs Application Software:

➢ **System sw** provides the basic function of the computer while **application sw** is used by the users to perform specific tasks.

➢ **The system sw** interacts with hardware at one end and with **application sw** at the other end.

➢ **The application sw** interacts with the **system sw** and the users of the computer.
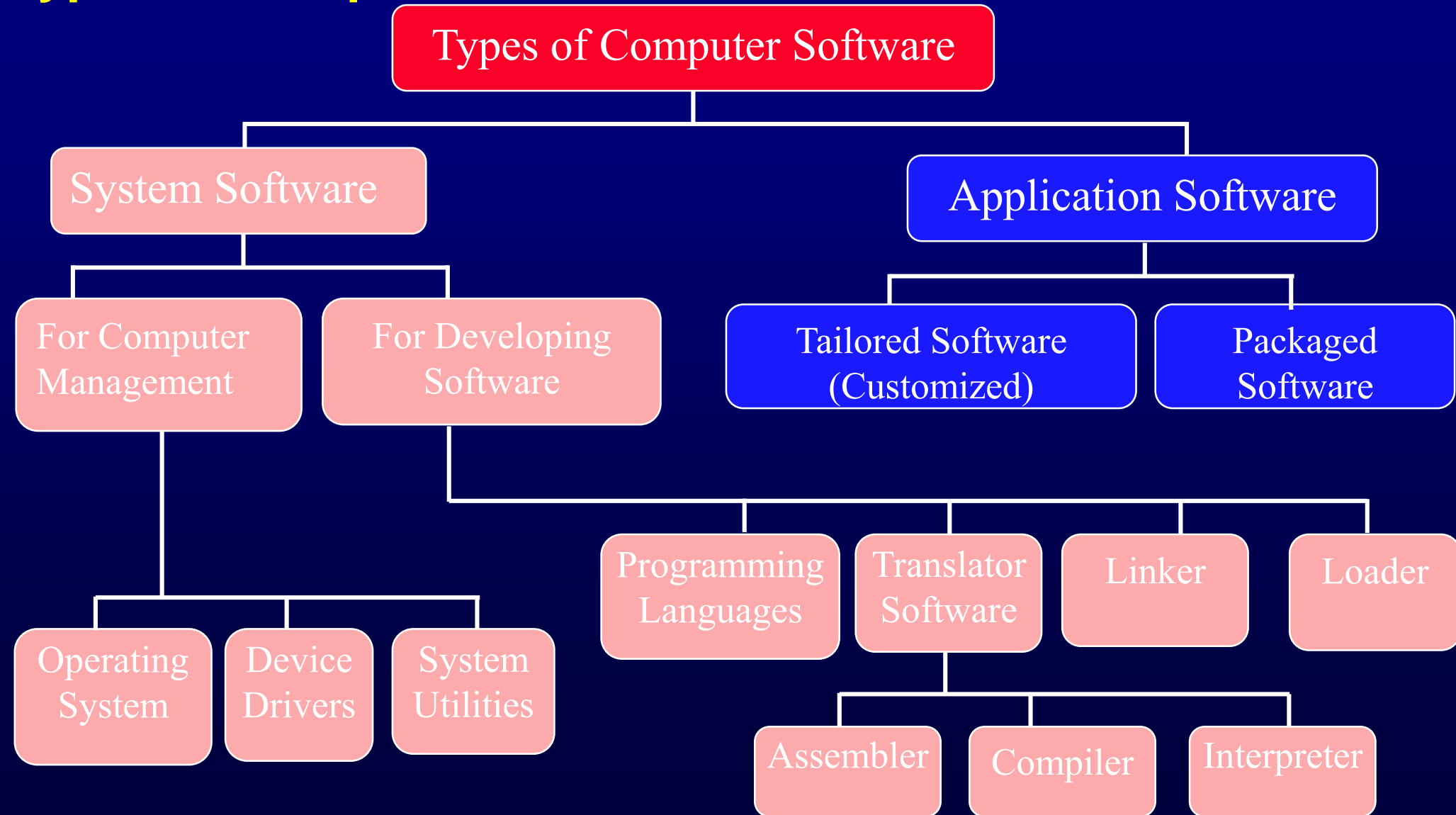
Computer Software

System software Vs Application Software:

System Software (SS)	Application Software (AS)
SS is used for operating computer hardware.	AS is used by user to perform specific task.
SS are installed on the computer when operating system is installed.	AS are installed according to user's requirements.
In general, the user does not interact with SS because it works in the background.	In general, the user interacts with AS.
SS can run independently. It provides platform for running application softwares.	AS can't run independently. They can't run without the presence of system software.
Some examples of SS are OS, compiler, assembler, debugger, driver, etc.	Some examples of AS are word processor, web browser, media player, etc.

Software Hierarchy
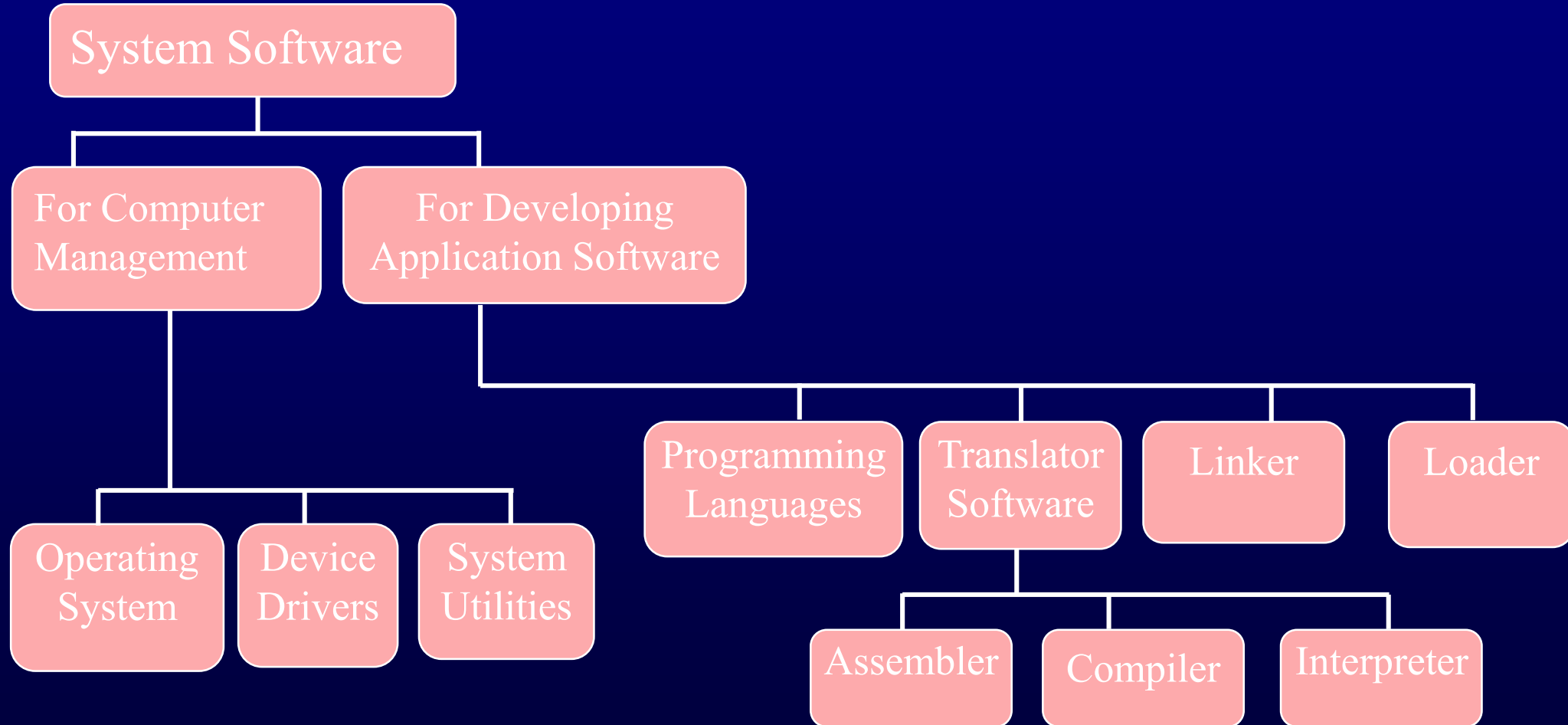
Types of computer Software:

System Software

System software for Computer Management:

➤ System SW includes Operating system, device drivers, and system utilities.

➤ System software is required for managing the operations performed by the components of computer (CPU, Memory, file storage) and Input-output devices attached to the computer.

➤ It provides services required for the development and execution of application software.

➤ The programming language software, translator software, loader, and linker are also categorized as system software, and are required for the application software development.

Computer Software

System Software:

Computer Software

SS for computer management:

1. Operating System
2. Device Driver
3. System Utilities

SS for Developing Application Software:

1. Programming Languages
   - High-level Language
   - Assembly Language
   - Machine Language
2. Translator Software
   - Assembler
   - Compiler
   - Interpreter
3. Linker
4. Loader

System Software

Operating System:

➢ Different kinds of application software use specific hardware resources of a computer like CPU, I/O devices and memory.

➢ OS controls and coordinates the use of hardware among the different application software and the users.

➢ OS provides an interface that is convenient for the user to use, and facilitates efficient operations of the computer system resources.
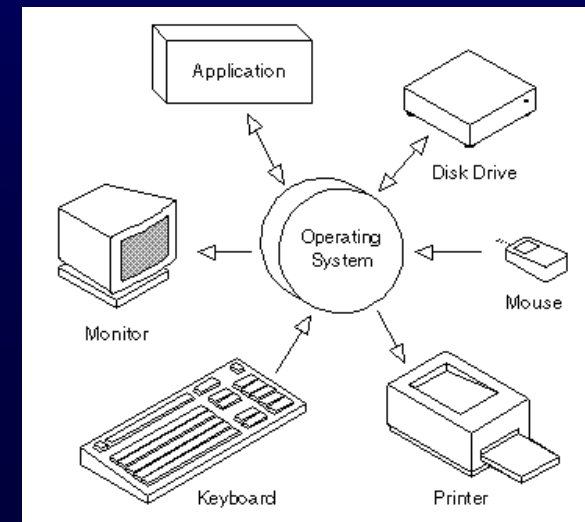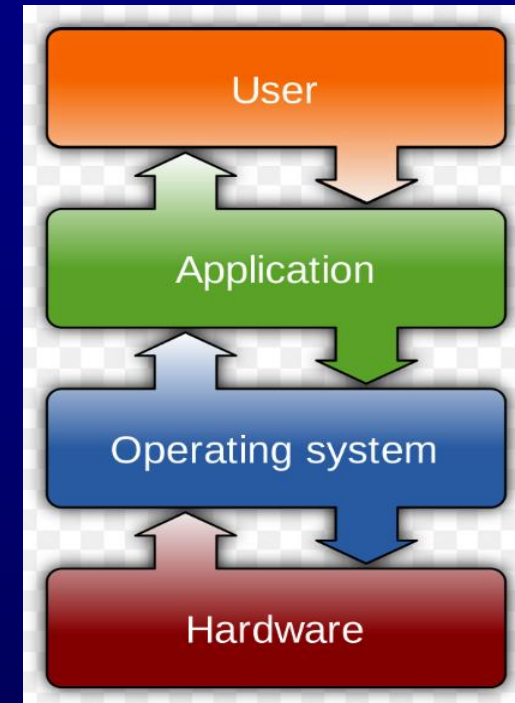
System Software

Operating System: The key functions of OS are:

➢ It provides an environment in which users and application software can do work.

➢ It manages different resources of the computer like the CPU time, memory space, file storage, I/O devices etc. During the use of computer by other programs or users, operating system manages various resources and allocates them whenever required, efficiently.

➢ It controls the execution of different programs to prevent occurrence of error.

➢ It provides a convenient interface to the user in the form of commands and graphical interface, which facilitates the use of computer.

Some available operating systems are Microsoft Disk Operating System (MS-DOS), Windows7, Windows XP, Linux, UNIX, and Mac OS X Snow Leopard.

System Software

Device Driver:

➢ A device driver acts as a translator between the hardware and the software that uses the devices.

➢ Devices(printer, speakers, microphone, webcam, scanner, digital camera etc.) that are connected to the computer must be installed its corresponding device driver for proper working of a device.

➢ For example, when we give a command to read data from the hard disk, the command is sent to the hard disk driver and is translated to a form that the hard disk can understand. The device driver software is typically supplied by the respective device manufacturers.

➢ Operating system comes preloaded with some commonly used device drivers i.e. for mouse, webcam, and keyboard and the operating system can automatically detect the device when it is connected to the computer. Such devices are called plug and play devices.

System Software

Device Driver:

➢ Each device has its own device driver.

➢ Whenever a new device is connected to a computer, its device driver has to be loaded in the computer's memory, to enable use of the device.

➢ When you buy a new printer, you get the device driver CD with it. You must install the device driver on your computer, to use the new printer. Each printer comes with its own device driver.

➢ Device drivers can be character or block device drivers. Character device drivers are for character based devices like keyboard, which transfer data character by character. Block device driver are for devices that transfer data as a block, like in hard disk.

System Software

System Utilities:

System utility software is required for the maintenance of computer. System utilities are used for supporting and enhancing the programs and the data in computer. Some system utilities may come embedded with OS and others may be added later on. Some examples of system utilities are:

- Anti-virus utility to scan computer for viruses (Figure 6.5).

- Data Compression utility to compress the files.

- Cryptographic utility to encrypt and decrypt files.

- Disk Compression utility to compress contents of a disk for increasing the capacity of a disk.

- Disk Partitioning to divide a single drive into multiple logical drives. Each drive is then treated as an individual drive and has its own file system.

- Disk Cleaners to find files that have not been used for a long time. It helps the user to decide what to delete when the hard disk is full.

System Software

System Utilities:

➢ Backup Utility to make a copy of all information stored on the disk. It also restores the backed up contents in case of disk failure.

➢ System Profiling Utility provides detailed information about the software installed on the computer and the hardware attached to it.

➢ Network Managers to check the computer network and to log events.
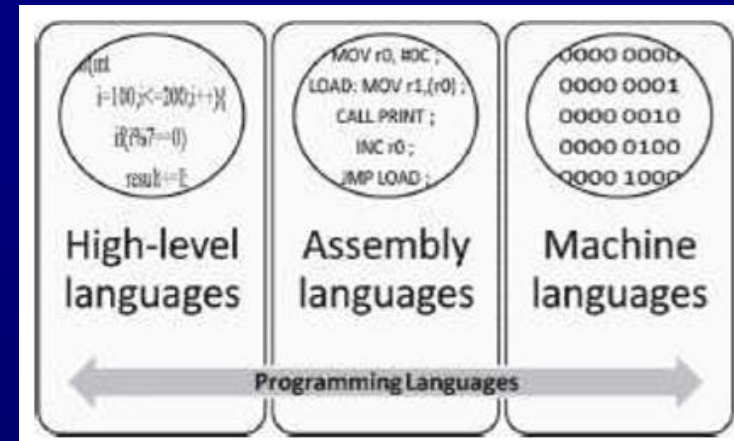
System Software

Programming Languages:

➢ A Programming Language consists of a set of vocabulary and grammatical rules, to express the computations and tasks that the computer has to perform.

➢ Programming languages are used to write a program, which controls the behavior of computer, codify the algorithms precisely, or enables the human-computer interface.

➢ Each language has a unique set of keywords (words that it understands) and a special syntax for organizing program instructions.

➢ The programming language should be understood, both by the programmer (who is writing the program) and the computer.

➢ A computer understands the language of 0·s and 1·s, while the programmer is more comfortable with English-like language.

➢ Programming Language usually refers to high-level languages like COBOL, BASIC, FORTRAN, C, C++, Java etc. Programming languages fall into three categories
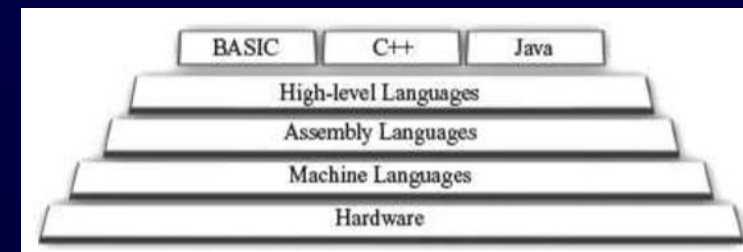
System Software

Programming Languages:

Programming languages fall into three categories:

➢ **Machine Language** is what the computer can understand but it is difficult for the programmer to understand. Machine languages consist of numbers only. Each kind of CPU has its own unique machine language.

➢ **Assembly Language** falls in between machine language and high-level language. They are similar to machine language, but easier to program in, because they allow the programmer to substitute names for numbers.

➢ **High-level Language** is easier to understand and use for the programmer but difficult for the computer.

High level Program needs to be converted into machine language so that the computer can understand it. In order to do this a program is either compiled or interpreted.

System Software

Programming Languages:

- Machine languages and assembly languages are also called low-level languages, and are generally used to write the system software.

- Application software is usually written in high-level languages.

- The program written in a programming language is also called the source code.

System Software

Machine Language:

A program written in machine language is a collection of binary digits or bits that the computer reads and interprets. It is a system of instructions and data executed directly by a computer's CPU. It is also referred to as machine code or object code.

➢ The computer can understand the programs written in machine language directly. No translation of the program is needed.

➢ Program written in machine language can be executed very fast (Since no translation is required).

➢ Machine language is defined by the hardware of a computer. It depends on the type of the processor or processor family that the computer uses, and is thus machine-dependent.

➢ A machine- level program written on one computer may not work on another computer with a different processor.

System Software

Machine Language:

- Computers may differ in other details, such as memory arrangement, operating systems, and peripheral devices and a program normally relies on such factors. Different computer may not run the same machine language program, even when the same type of processor is used.

- Most machine-level instructions have opcode fields which specify the basic instruction type (such as arithmetic, logical, jump, etc) and operand, the actual operation to be performed on.

- It is difficult to write a program in machine language as it has to be written in binary code. For e.g., 00010001 11001001. Such programs are also difficult to modify.

- Since writing programs in machine language is very difficult, programs are hardly written in machine language.

System Software

Assembly Language:

➢ A program written in assembly language uses symbolic representation of machine codes needed to program a particular processor (CPU) or processor family. This representation is usually defined by the CPU manufacturer, and is based on abbreviations (called mnemonics) that help the programmer remember individual instructions, registers, etc. Small, English-like representation is used to write the program in assembly language.

Some of the features of a program written in assembly language are as follows:

```
MOV     B, A
MVI     C, 06H
LXI     H, XX50H
ADD     M
JNC     NXTITM
INR     B
INX     H
DCR     C
JNZ     NXTBIT
```

System Software

Assembly Language:

➢ Assembly language programs are easier to write than the machine language programs, since assembly language programs use short, English-like representation of machine code.

➢ The program written in assembly language is the source code, which has to be converted into machine code, also called object code, using translator software, namely, assembler.

➢ Each line of the assembly language program is converted into one or more lines of machine code. Hence assembly language programs are also machine-dependent.

➢ Although assembly language programs use symbolic representation, they are still difficult to write.

➢ Assembly language programs are generally written where the efficiency and the speed of program are the critical issues, i.e. programs requiring high speed and efficiency.

Computer Software

System Software:

➢ The programming language software, translator software, loader and linker are also categorized as system software which are required for application software development.

Computer Software

- **Programming Languages**

- **Translator Software**
  1. Assembler
  2. Compiler
  3. Interpreter

- **Linker**

- **Loader**

Computer Software

Programming Languages:

- A programming language is a vocabulary and set of grammatical rules for instructing a computer to perform specific tasks.

- A programming language is a computer language engineered to create a standard form of commands.

- There are many programming languages -- C, C++, Pascal, BASIC, FORTRAN, COBOL, Java and LISP are just a few. These are all high-level languages.

- Programs can also write in low-level languages called assembly languages.

- Low-level languages are closer to the language used by a computer, while high-level languages are closer to human languages.

Computer Software

Translator Software:

Every program must be translated into a machine language that the computer can understand. This translation is performed by.

1. Compiler   - (Use in High Level Language)
2. Interpreter   - (Use in High Level Language)
3. Assembler   - (Use in Assembly Language)

➢ A program that executes instructions written in a high-level language. There are two ways to run programs written in a high-level language.

➢ The most common is to compile the program; the other method is to pass the program through an interpreter.

➢ Assembly is a programming language that consists of instructions in mnemonic codes and assembler convert the mnemonics codes into machine language instruction.

Computer Software

Translator Software:

➢ An interpreter translates high-level instructions into an intermediate form, which it then executes.

➢ Compiler translates high-level instructions directly into machine language.

➢ The interpreter does not need to go through the compilation stage during which machine instructions are generated.

➢ Interpreters are sometimes used during the development of a program, when a programmer wants to add small sections at a time and test them quickly.

➢ In addition, interpreters are often used in education because they allow students to program interactively.
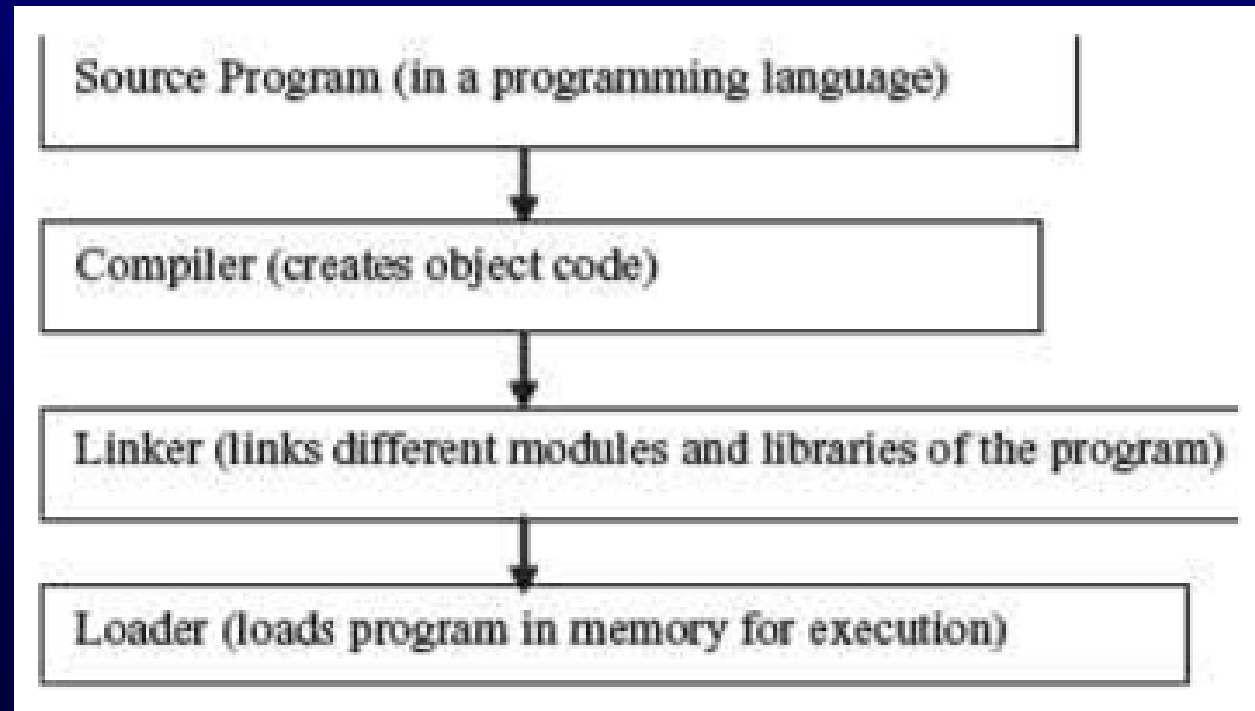
Computer Software

Linker and Loader:

➢ Linker is a program that links several object modules and libraries to a single executable program.

➢ A source code may also include reference to libraries(header) and independent modules(functions) which may not be stored in a single object file. The code is broken down into many independent modules for easy debugging and maintenance. Before execution of the program, these modules and the required libraries are linked together using the linker software.

➢ The compiled and the linked program are called the executable code.
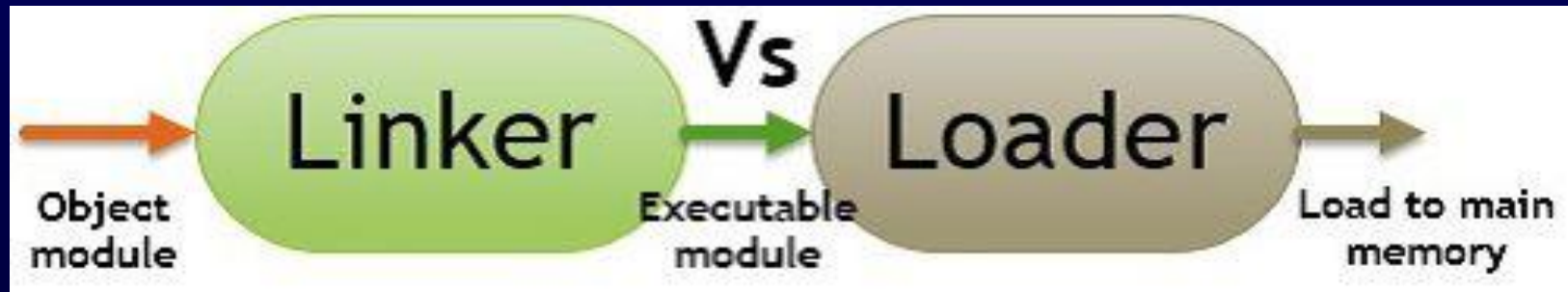
Computer Software

Linker and Loader:

- The loader software is used to load and re-locate the executable program in the main memory. Software has to be loaded into the main memory during execution. Loader assigns storage space to the program in the main memory for execution.

Source Program (in a programming language)

↓

Compiler (creates object code)

↓

Linker (links different modules and libraries of the program)

↓

Loader (loads program in memory for execution)

Computer Software

Linker and Loader:

➢ The key difference between **linker and loader** is that the **linker** generates the executable file of a program whereas, the **loader** loads the executable file obtained from the **linker** into main memory for execution.

➢ The **linker** intakes the object module of a program generated by the assembler.

Computer Software

Application Software:

Software written by user to solve a specific user oriented problem using computer is known as application sw. Software accomplishing a specific task is the application sw.

Examples: Word processor, Spread sheet, Database, Engineering packages(CAD/CAM), etc.

Further grouped into two types as follows:

1. Tailored Software (Customized)
2. Packaged Software

Computer Software

Tailored Software (Customized):

Designed to meet the specific requirements of an organization or individuals and developed on demand of customer by software contractor.

➢ Written using high level programming language such as c, C++, Java, VB etc.

➢ Examples: Air traffic control system, school billing system etc.

Computer Software

Packaged Software:

➢ Generalized set of programs designed and developed for general purpose.

➢ Produced by development organization and sold on the open market to any customer who is able to buy and use.

➢ Also called universal software as it can be used by users and organizations all over the world.

➢ Examples: Word processor, Spread sheet, Database, Engineering packages(CAD/CAM), etc.

Computer Software

Program Vs Software:

➢ **Programs** are developed by individuals for their personal use. They are generally, small in size and have limited functionality. The author of a program himself uses and maintains his program, these usually do not have a good user interface and lack of proper documentation.

➢ Whereas **Software products** have multiple users and therefore should have a good user interface, proper operating procedures, and good documentation support.

Computer Software

Program Vs Software:

- **Programs** are developed by individuals for their personal use. They are generally, small in size and have limited functionality. The author of a program himself uses and maintains his program, these usually do not have a good user interface and lack of proper documentation.

- Whereas **Software products** have multiple users and therefore should have a good user interface, proper operating procedures, and good documentation support.
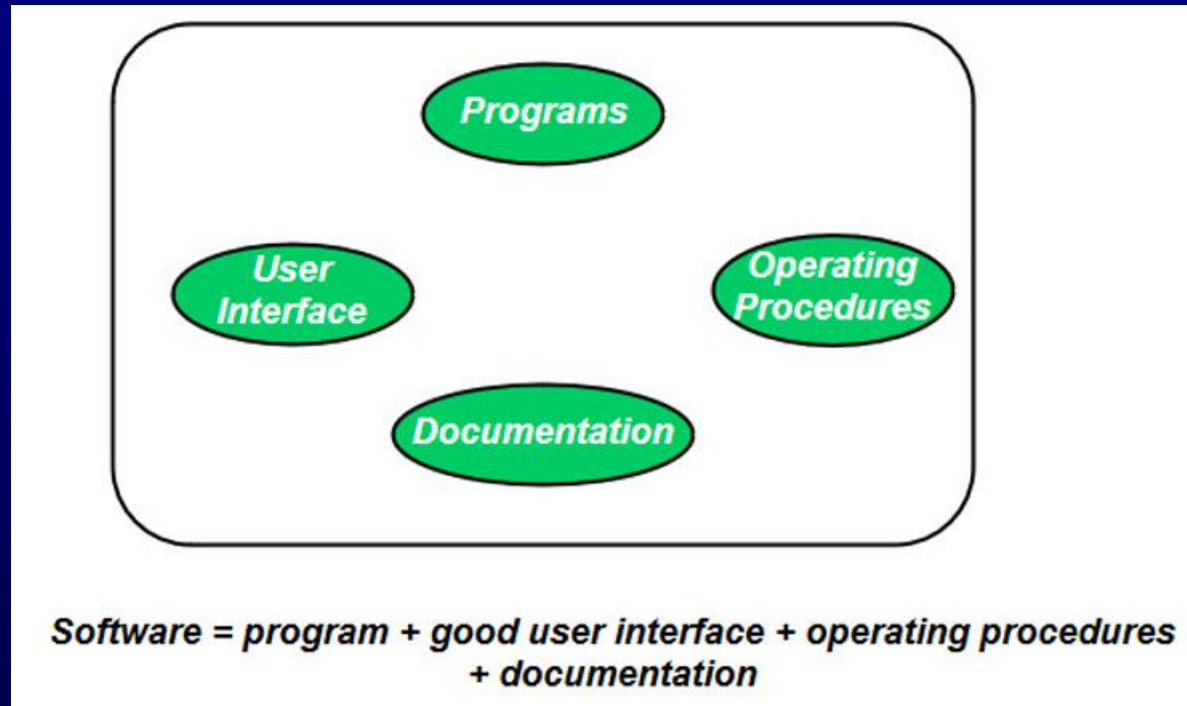
Computer Software

Program Vs Software:

Software = program + good user interface + operating procedures + documentation

Computer Software

Software Acquisition:

The Software Acquisition is a computer-aided system that supports the improvement of an organization's software acquisition process capability and performance.

- **Retail software**
- **OEM** (Original Equipment Manufacturer) **software**
- **Shareware**
- **Freeware**
- **Open source software**
- **Public Domain software**
- **Demo software**

Computer Software

Software Acquisition:

- **Retail software:** is off-the-shelf software sold in retail stores. It comes with printed manuals and installation instructions.

- **OEM:** "Original Equipment Manufacturer" refers to software which is sold, and bundled with hardware. Microsoft sells its operating system as OEM software to hardware dealers. OEM software is sold at reduced price, without the manuals, packaging and installation instructions.

- **Shareware :** is a program that the user is allowed to try for free, for a specified period of time, as defined in the license. It is downloadable from the Internet. When the trial period ends, the software must be purchased or uninstalled.

Computer Software

Software Acquisition:

➢ **Freeware :** is software that is free for personal use. It is downloadable from the Internet. The commercial use of this software may require a paid license. The author of the freeware software is the owner of the software, though others may use it for free. The users abide by the license terms, where the user cannot make changes to it, or sell it to someone else.

➢ **Open source software :** is software whose source code is available and can be customized and altered within the specified guidelines laid down by the creator. Unlike public domain software, open-source software has restrictions on their use and modification, redistribution limitations, and copyrights. Linux, Apache, Firefox, OpenOffice are some examples of open-source software.

Computer Software

Software Acquisition:

➢ **Public Domain Software:** is free software. Unlike freeware, public domain software does not have a copyright owner or license restrictions. The source code is publicly available for anyone to use. Public domain software can be modified by the user.

➢ **Demo Software:** is designed to demonstrate what a purchased version of the software is capable of doing and provides a restricted set of features. To use the software, the user must buy a fully- functional version.

Course Contents

Unit-06: Computer Software

➢ **Introduction**

➢ **Types of Software: System Software Vs Application Software**

➢ **Software Acquisition**

➢ **Operating System**

- **Introduction**

- **Objectives of Operating System**

- **Types of OS**

- **Functions of OS (Process Management, Memory Management, File Management, Device Management, Protection and Security, User Interface)**

➢ **Examples of Operating Systems**

Course Contents

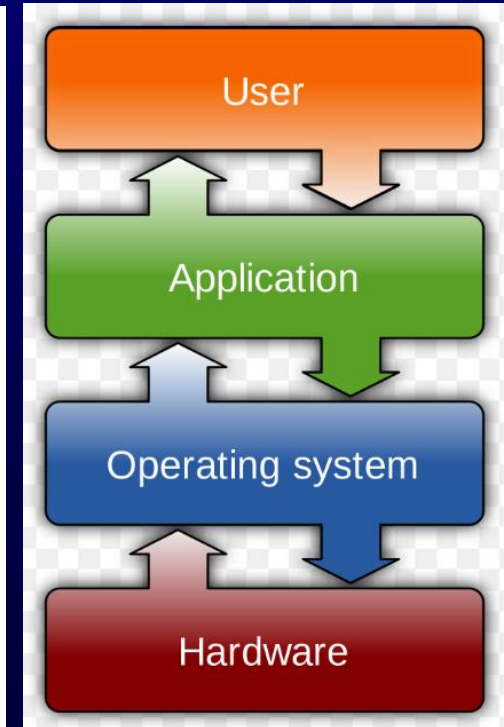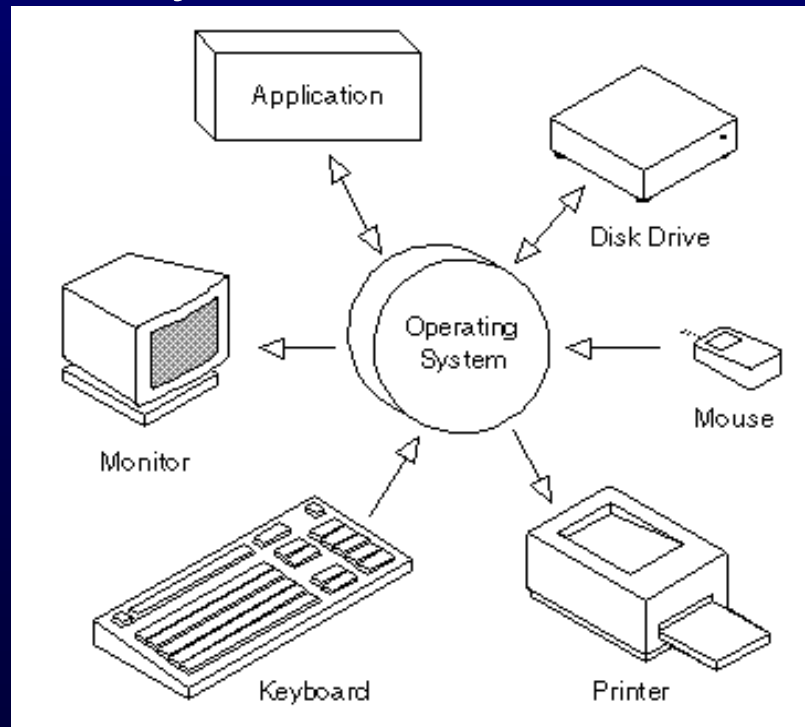## Operating System

Course Contents

Operating system

Operating system:

An **operating system** (**OS**) is software that allows a user to run other applications on a computing device. The **operating system** manages a computer's hardware resources including I/O devices, CPU, memory etc.

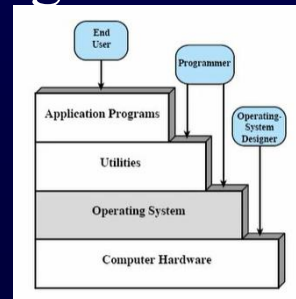# Operating system

Operating system:

- An **operating system** is the **most important software** that runs on a computer.

- OS manages the computer's **memory** and **processes**, as well as all of its **software** and **hardware**.

- It also allows use to **communicate** with the computer without knowing how to speak the computer's language.

- Computer is useless without an operating system.

Course Contents

Objectives of Operating System:

➢ Operating system is system software that controls and coordinates the use of hardware among the different application software and users.

➢ OS intermediates between the user of computer and the computer hardware.

➢ The user gives a command and the OS translates the command into a form that the machine can understand and execute.

➢ OS has two main objectives

1. to make the computer system convenient and easy to use, for the user, and

2. to use the computer hardware in an efficient way, by handling the details of the operations of the hardware.

Operating system

Types of Operating System:

- OS are classified into different types depending on their capability of processing

  1. Single user
  2. Multiuser
  3. Multitasking
  4. Multiprocessing
  5. Real time
  6. Embedded
  7. Distributed

Operating system

Types of Operating System:

➢ **Single User and Single Task OS:** As the name implies, this operating system is designed to manage the computer so that one user can effectively do one thing at a time. Operating system for Personal Computers (PC) are single user OS. MS-DOS is an example of single user OS.

➢ **Single-user, multi-tasking:** allows execution of more than one task or process concurrently. For this, the processor time is divided amongst different tasks. This division of time is also called time sharing. The processor switches rapidly between processes. The user can switch between the applications and also transfer data between them. Windows 95 and all later versions of Windows are examples of multitasking OS.

➢ **Multi-user OS**: A multi-user operating system allows many different users to take advantage of the computer's resources simultaneously. Multi-user is used in computer networks that allow same data and applications to be accessed by multiple users at the same time. The users can also communicate with each other. Linux, UNIX, and Windows 7 are examples of multiuser OS.

Operating system

Types of Operating System:

➢ **Multiprocessing OS:** have two or more processors for a single running process. Processing takes place in parallel and is also called parallel processing. Each processor works on different parts of the same task, or, on two or more different tasks. Since execution takes place in parallel, they are used for high speed execution, and to increase the power of computer. Linux, UNIX and Windows 7 are examples of multiprocessing OS.

➢ **Real-time operating system(RTOS):** Real Time OS are designed to respond to an event within a predetermined time. These operating systems are used to control processes. Processing is done within a time constraint. OS monitors the events that affect the execution of process and respond accordingly. LynxOS is an example of real time OS. RTOS is used to control machinery, scientific instruments and industrial systems.

Operating system

Types of Operating System:

- **Embedded:** Embedded OS is embedded in a device in the ROM. They are specific to a device and are less resource intensive. They are used in appliances like microwaves, washing machines, traffic control systems etc.

- **Distributed:** A distributed operating system manages a group of independent computers and makes them appear to be a single computer.

Operating System

Functions of Operating System:

1. Process Management
2. Memory Management
3. File Management
4. Device Management
5. Protection and Security
6. User Interface

Course Contents

Functions of Operating System:

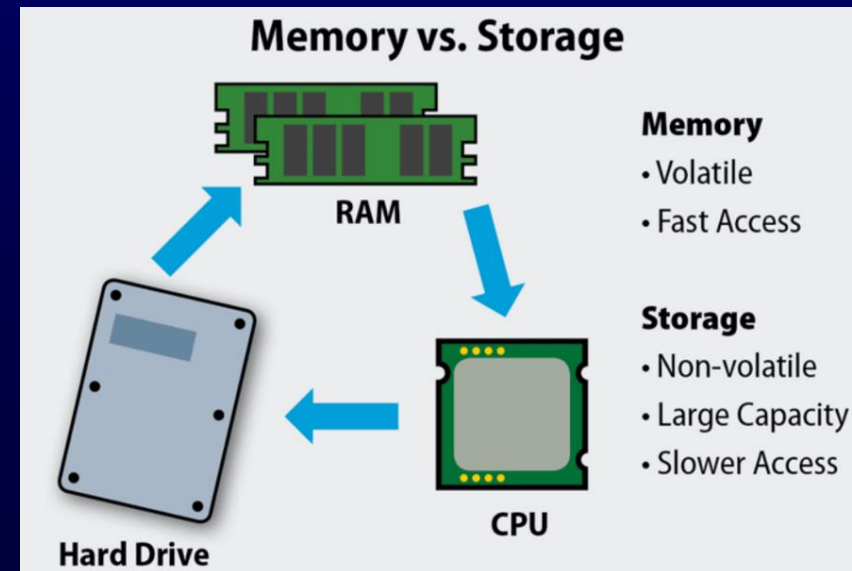1. **Process Management:** The process management activities handled by the OS are
    1) Control access to shared resources like file, memory, I/O and CPU,
    2) Control execution of applications
    3) Create, execute and delete a process (system process or user process),
    4) Cancel or resume a process
    5) Schedule a process, and
    6) Synchronization, communication and deadlock handling for processes.

Course Contents

Functions of Operating System:

2. Memory Management : The activities of memory management handled by OS are

1) allocate memory,
2) free memory,
3) re-allocate memory to a program when a used block is freed, and
4) keep track of memory usage.

Memory vs. Storage

RAM

Hard Drive

CPU

Memory
- Volatile
- Fast Access

Storage
- Non-volatile
- Large Capacity
- Slower Access

Course Contents

Functions of Operating System:

3. File Management: **The file management tasks include**

    1)   create and delete both files and directories,

    2)   provide access to files,

    3)   allocate space for files,

    4)   keep back-up of files, and

    5)   secure files.

4. Device Management: **The device management tasks handled by OS are**

    1)   open, close and write device drivers, and

    2)   communicate, control and monitor the device driver.

Course Contents

Functions of Operating System:

5. Protection and Security:

➢ OS protects the resources of system. User authentication, file attributes like read, write, encryption, and back-up of data are used by OS to provide basic protection.

6. User Interface or Command Interpreter:

➢ Operating system provides an interface between the computer user and the computer hardware. The user interface is a set of commands or a graphical user interface via which the user interacts with the applications and the hardware.

Process Management

1. **Process Management:**

a. Program **Vs** Process

b. CPU Scheduling

- What is scheduling?
    1. Non-pre-emptive scheduling
    2. Pre-emptive scheduling
- Scheduling algorithms are?
    1. First Come First Served (FCFS) Scheduling
    2. Shortest Job First (SJF) Scheduling
    3. Round Robin (RR) Scheduling

c. Process Synchronization

d. Deadlock

- What is Deadlock?
- What are the reasons for deadlock?
    1. Mutual Exclusion
    2. No Pre-emption
    3. Hold and Wait
    4. Circular Wait

Process Management

1. **Process Management:**

Program Vs Process

➢ A program is an executable file which contains a certain set of instructions written to complete the specific job on your computer. Programs are never stored on the primary memory in your computer. Instead, they are stored on a disk or secondary memory and are read from the primary memory and executed by the OS.

➢ A process is an execution of any specific program. It is considered an active entity that actions the purpose of the application. Multiple processes may be related to the same program.

Process Management

1. Process Management:

Program Vs Process

➢ A process is a program in a state of execution. It is a unit of work for the operating system. A process can be created, executed, and stopped.

➢ In contrast, a program is always static and does not have any state. A program may have two or more processes running. A process and a program are, thus, two different entities.

Process Management

1. Process Management:

Program Vs Process

Example of Program and Process:

Program to transfer Rs10 from account X to Y and add Rs5 to account X.

Process-1	Process-2
read_item(X);	
X:=X-10;	
write_item(X);	
read_item(Y);	
Y:=Y+10;	
write_item(Y);	
	read_item(X);
	X:=X+5;
	write_item(X);

Process-1	Process-2
	read_item(X);
	X:=X+5;
	write_item(X);
read_item(X);	
X:=X-10;	
write_item(X);	
read_item(Y);	
Y:=Y+10;	
write_item(Y);	

Process-1	Process-2
read_item(X);	
X:=X-10;	
write_item(X);	
	read_item(X);
	X:=X+5;
	write_item(X);
read_item(Y);	
Y:=Y+10;	
write_item(Y);	

Process Management

1. Process Management:

BASIS FOR COMPARISON	PROGRAM	PROCESS
Basic	Program is a set of instruction.	When a program is executed, it is known as process.
Nature	Passive	Active
Lifespan	Longer	Limited
Required resources	Program is stored on disk in some file and does not require any other resources.	Process holds resources such as CPU, memory address, disk, I/O etc.

Process Management

1. **Process Management:**

Program Vs Process

➢ To accomplish a task, a process needs to have access to different system resources like I/O devices, CPU, memory etc. The process management function of an operating system handles allocation of resources to the processes in an efficient manner. The allocation of resources required by a process is made during process creation and process execution.

➢ A process changes its state as it is executed. The various states that a process changes during execution are as follows in Figure next slide…..

Process Management

1. Process Management:

Program Vs Process

Process states are:

New: process is in a new state when it is created,

Ready: process is in ready state when it is waiting for a processor,

Running: process is in running state if processor is executing the process,

Waiting: process is in waiting state when it waits for some event to happen (I/O etc), and

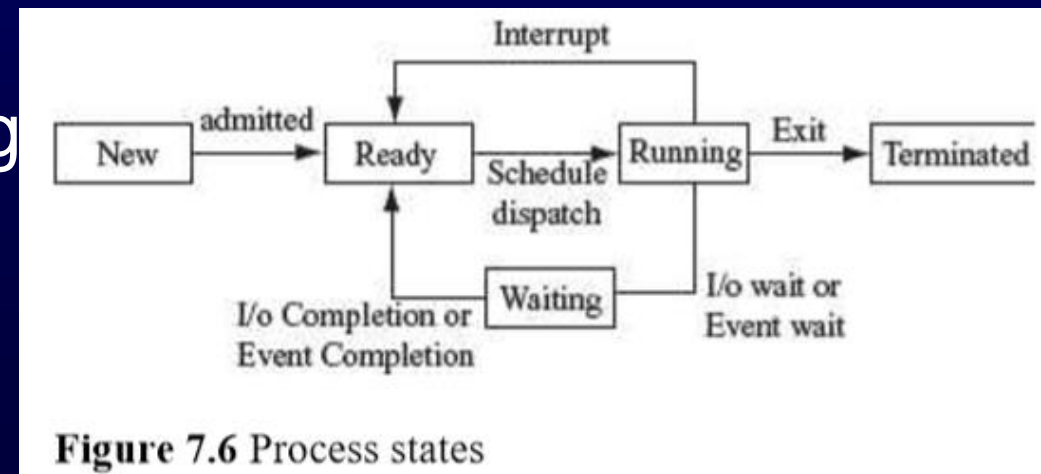Terminated: process that has finished execution is in terminated state.

Figure 7.6 Process states

Process Management

1. Process Management:

Program Vs Process

➤ The concurrent execution of the process requires process synchronization and CPU scheduling and deadlock situations of processes which are described in the following slides.

Process Management

1. **Process Management:**

b. **CPU Scheduling**

- **What is scheduling?**
    1. **Non-pre-emptive scheduling**
    2. **Pre-emptive scheduling**

- **Scheduling algorithms are?**
    1. **First Come First Served (FCFS) Scheduling**
    2. **Shortest Job First (SJF) Scheduling**
    3. **Round Robin (RR) Scheduling**

Process Management

1. **Process Management:**

b.   **CPU Scheduling**

   - **What is scheduling?**

➢ All computer resources like I/O, memory, and CPU are to be scheduled for use.

➢ In a multiprogramming and time sharing system, the processor executes multiple processes by switching the CPU among the processes, so that no user has to wait for long for a program to execute. To enable running of several concurrent processes, the processor time has to be distributed amongst all the processes efficiently.

Process Management

1. Process Management:

b.   CPU Scheduling

- **What is scheduling?**

➢ Scheduler is a component of the operating system that is responsible for scheduling transition of processes. At any one time, only one process can be in running state and the rest are in ready or waiting state. The scheduler assigns the processor to different processes in a manner so that no one process is kept waiting for long.

➢ Scheduling can be of two types:
  1. **Non-pre-emptive scheduling**
  2. **Pre-emptive scheduling**

Process Management

1. Process Management:

b.   CPU Scheduling

➢ In non-preemptive scheduling, the processor executes a process till termination without any interruption. Hence the system resources are not used efficiently.

➢ In pre-emptive scheduling, a running process may be interrupted by another process that needs to execute. Pre-emption allows the operating system to interrupt the executing task and handle any important task that requires immediate action. In pre-emptive scheduling, the system resources are used efficiently.

Process Management:
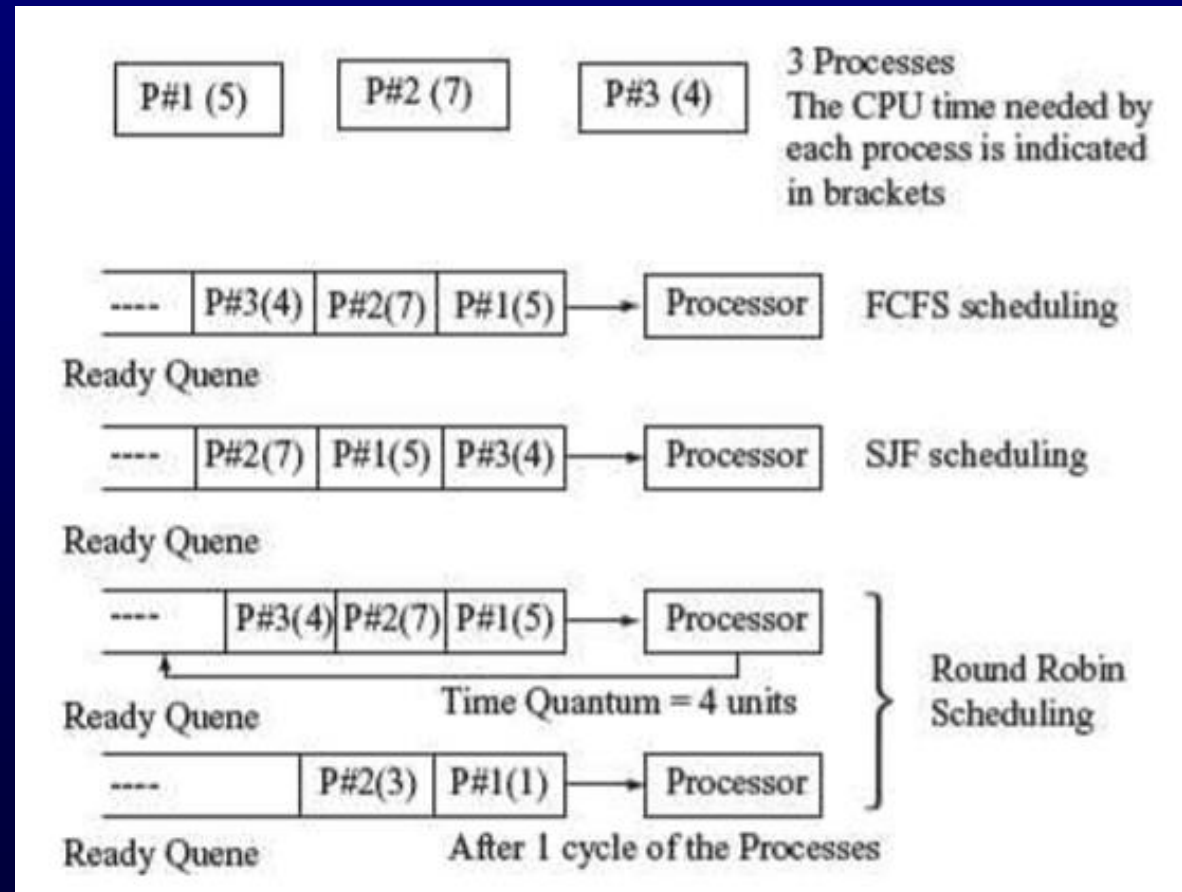
b.   CPU Scheduling : Scheduling algorithms are?

➢ **First Come First Served (FCFS) Scheduling:** the process that requests for the CPU first, gets the CPU first. A queue is maintained for the processes requesting the CPU. The process first in the queue is allocated the CPU first. FCFS scheduling is non-pre-emptive. The drawback of this scheduling algorithm is that the process that is assigned to the CPU may take long time to complete, keeping all other processes waiting in the queue, even if they require less CPU time.

➢ **Shortest Job First (SJF) Scheduling:** The process that requires the least CPU time is allocated the CPU first. SJF scheduling is non-pre-emptive. The drawback of this scheduling is that a process that requires more CPU time may have to wait for long time, since processes requiring less CPU time will be assigned the CPU first.

➢ **Round Robin (RR) Scheduling:** It is designed for time-sharing systems. RR scheduling is pre-emptive. In this scheduling, a small quantum of time (10—100 ms) is defined, and each process in the queue is assigned the CPU for this quantum of time circularly. New processes are added at the tail of the queue and the process that has finished execution is removed from the queue. RR scheduling overcomes the disadvantage of FCFS and SJF scheduling. A process does not have to wait for long, if it is not the first one in the queue, or, if it requires CPU for a long period of time.

Process Management:

b.   CPU Scheduling : Scheduling algorithms are?

1.   **First Come First Served (FCFS) Scheduling**

2.   **Shortest Job First (SJF) Scheduling**

3.   **Round Robin (RR) Scheduling**

Example: P#1 with CPU time requirement of 5 units, P#2 with 7 units and P#3 with 4 units, are scheduled, using the different CPU scheduling algorithms

Process Management

1. **Process Management:**

a. **Program Vs Process**

b. **CPU Scheduling**

- **What is scheduling?**
    1. **Non-pre-emptive scheduling**
    2. **Pre-emptive scheduling**

- **Scheduling algorithms are?**
    1. **First Come First Served (FCFS) Scheduling**
    2. **Shortest Job First (SJF) Scheduling**
    3. **Round Robin (RR) Scheduling**

c. **Process Synchronization**

d. **Deadlock**

- **What is Deadlock?**
- **What are the reasons for deadlock?**
    1. **Mutual Exclusion**
    2. **No Pre-emption**
    3. **Hold and Wait**
    4. **Circular Wait**

Process Management:

C. Process Synchronization:

➢ In a computer, multiple processes are executing at the same time. The processes that share the resources have to communicate with one another to prevent a situation where one process disrupts another process.

➢ When two or more processes execute at the same time, independent of each other, they are called concurrent processes.

➢ A situation where multiple processes access and manipulate the same data concurrently, in which the final result depends on the order of process execution, is called a race condition. To handle such situations, synchronization and coordination of the processes is required.

Process Management

C. Process Synchronization:

Program: Program to transfer Rs10 from account X to Y and add Rs5 to account X.

What will happen when process-2 items starts after X:=X-10 and before write_item(X) of process-1?

Process-1	Process-2
read_item(X);	
X:=X-10;	
write_item(X);	
read_item(Y);	
Y:=Y+10;	
write_item(Y);	
	read_item(X);
	X:=X+5;
	write_item(X);

Process-1	Process-2
	read_item(X);
	X:=X+5;
	write_item(X);
read_item(X);	
X:=X-10;	
write_item(X);	
read_item(Y);	
Y:=Y+10;	
write_item(Y);	

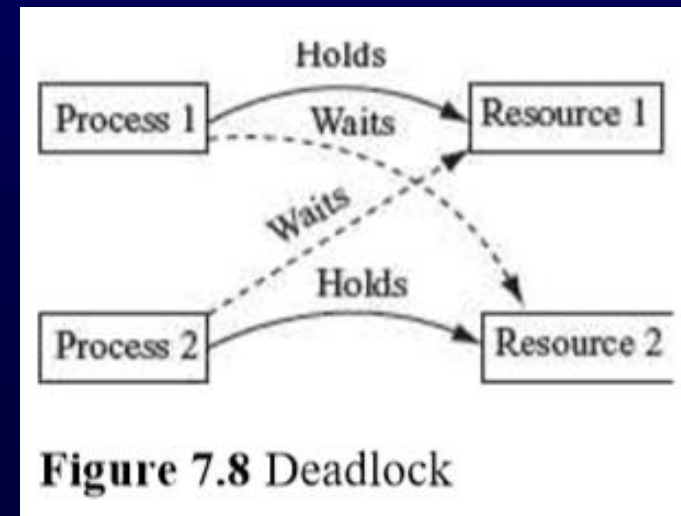| Process-1	Process-2
read_item(X);	
X:=X-10;	
write_item(X);	
	read_item(X);
	X:=X+5;
	write_item(X);
read_item(Y);	
Y:=Y+10;	
write_item(Y);	

Process Management:

D. Deadlock:

- ➢ **What is Deadlock?**
- ➢ **What are the reasons for deadlock?**
  1. Mutual Exclusion
  2. No Pre-emption
  3. Hold and Wait
  4. Circular Wait
- ➢ **Deadlock handling?**

Process Management:

D. Deadlock:

➤ In concurrent process environment, processes may try to access the same resource or data. A deadlock is a situation when a process waits endlessly for a resource and the requested resource is being used by another process that is waiting for some other resource.

➤ A deadlock arises when the four necessary conditions hold true simultaneously in a system. These conditions are as follows:

Figure 7.8 Deadlock

Process Management:

D. Deadlock:

A deadlock arises when the four necessary conditions hold true simultaneously in a system. These conditions are as follows:

Mutual Exclusion: Only one process at a time can use the resource. Any other process requesting the resource has to wait until the resource is released.

No Pre-emption: A process releases the resource by itself. A process cannot remove the resource from another process.

Hold and Wait: A process holds a resource while requesting another resource, which may be currently held by another process.

Circular Wait: In this situation, a process P1 waits for a resource held by another process P2, and the process P2 waits for a resource held by process P1.

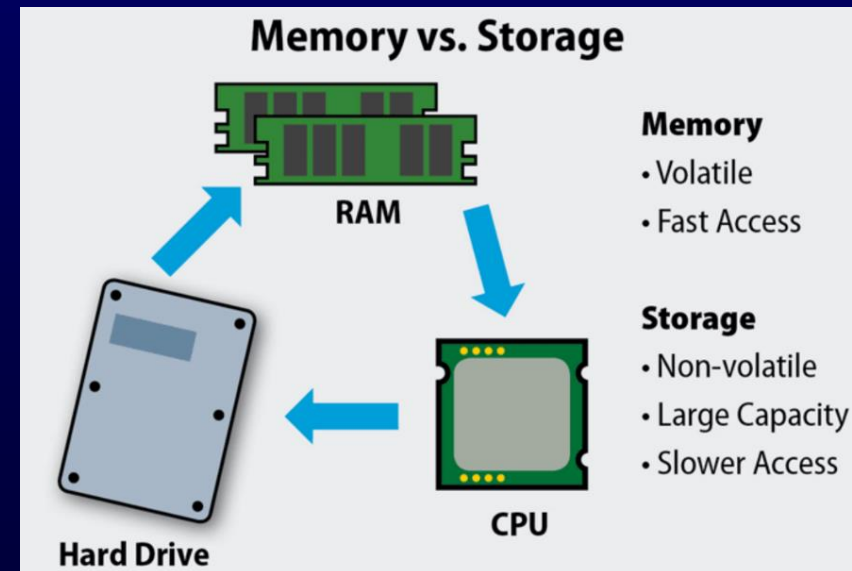# Process Management:

D. Deadlock

Deadlock Handling?

➢ **Deadlock Prevention - is a set of method that ensures that at least one of the above four necessary conditions required for deadlock, does not hold true.**

➢ **Deadlock Avoidance - requires that the operating system be given information in advance regarding the resources a process will request and use. This information is used by the operating system to schedule the allocation of resources so that no process waits for a resource.**

Functions of Operating System

2. Memory Management : The activities of memory management handled by OS are

1) allocate memory,

2) free memory,

3) re-allocate memory to a program when a used block is freed, and

4) keep track of memory usage.

Memory vs. Storage

RAM

Memory
- Volatile
- Fast Access

Storage
- Non-volatile
- Large Capacity
- Slower Access

Hard Drive

CPU

Functions of Operating System

2. Memory Management(MM) :

➢ The task of MM is to handle the allocation of memory to different processes and on completion of process execution, the memory is de-allocated and made available to another process.

➢ Additionally, different processes that have been allocated memory should not interfere into each other's memory space. This requires some memory protection and sharing mechanism.

➢ Memory allocation, de-allocation, reallocation of free memory, and memory protection & sharing are the jobs of OS and is called memory management.

Functions of Operating System

2. Memory Management :

Memory Allocation

- In single-user and single-task operating system like MS-DOS, only one process can execute at a time and one process is allocated to memory and allocated memory is freed after the termination of the process which is made available to any other process.

- In a multiprogramming system, in addition to allocation and de-allocation of memory, more tasks are involved like keeping track of processes allocated to the memory, memory protection and sharing etc.

Functions of Operating System

2. Memory Management :

Memory Allocation

The different memory allocation schemes are

- ➢ Multiple Partition Allocation
- ➢ Paging
- ➢ Virtual Memory

Functions of Operating System

2. Memory Management : Multiple Partition Allocation

The OS keeps track of blocks of memory which are free and those which are unavailable. The single block of available memory is called a hole.

- Hole – block of available memory; holes of various size are scattered throughout memory

- When a process arrives, it is allocated memory from a hole large enough to accommodate it.

- Operating system maintains information about:
  - a) allocated partitions   b) free partitions (hole)

Functions of Operating System

2. Memory Management : Multiple Partition Allocation

During allocation of memory, the set of holes is searched to determine which hole is to be allocated. For this, 3 hole allocation strategies are used.

➢ First-fit: Allocate the first hole that is big enough

➢ Best-fit: Allocate the smallest hole that is big enough; must search entire list, unless ordered by size

- Produces the smallest leftover hole

➢ Worst-fit: Allocate the largest hole; must also search entire list

- Produces the largest leftover hole.

First-fit and best-fit better than worst-fit in terms of speed and storage utilization

Functions of Operating System

2. Memory Management :

Multiple Partition Allocation: Fragmentation

As processes are loaded and removed from memory, the free memory space is broken into little pieces. It happens after sometimes that processes cannot be allocated to memory blocks considering their small size and memory blocks remains unused. This problem is known as Fragmentation
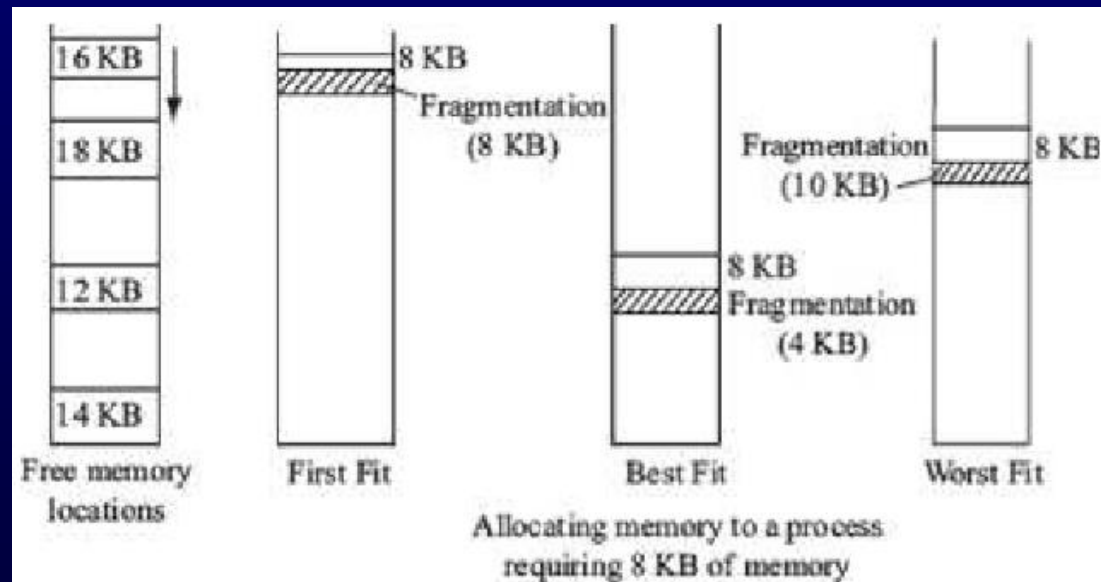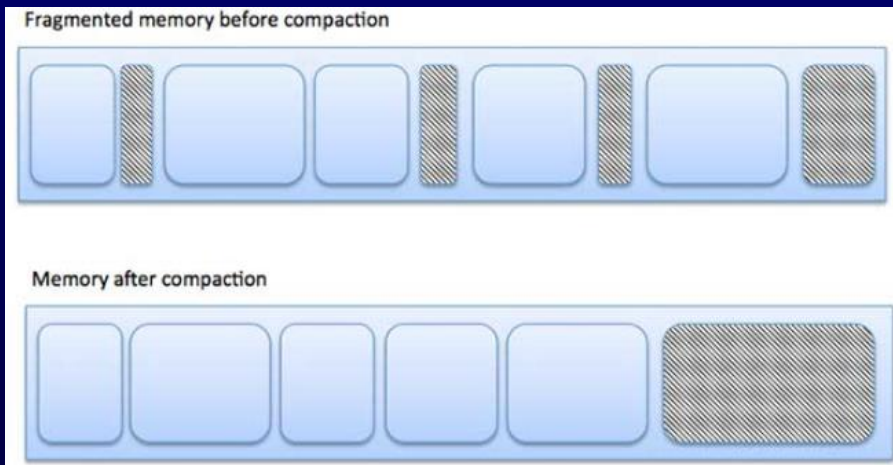

Figure 7.9 Multiple partition memory allocation

2. Memory Management : Paging

➤ Paging is a memory management technique in which process address space is broken into blocks of the same size called pages. The size of the process is measured in the number of pages.

➤ Similarly, main memory is divided into small fixed-sized blocks of (physical) memory called frames and the size of a frame is kept the same as that of a page to have optimum utilization of the main memory and to avoid external fragmentation.

➤ When a process is executed, its pages are loaded into the frames.

➤ An address generated by CPU has two parts - page number and page offset. A page table is maintained by the operating system that maps the page number to the frame number. The page number is used to index the page table and get the frame number. The page offset is added to the page frame number to get the physical memory address
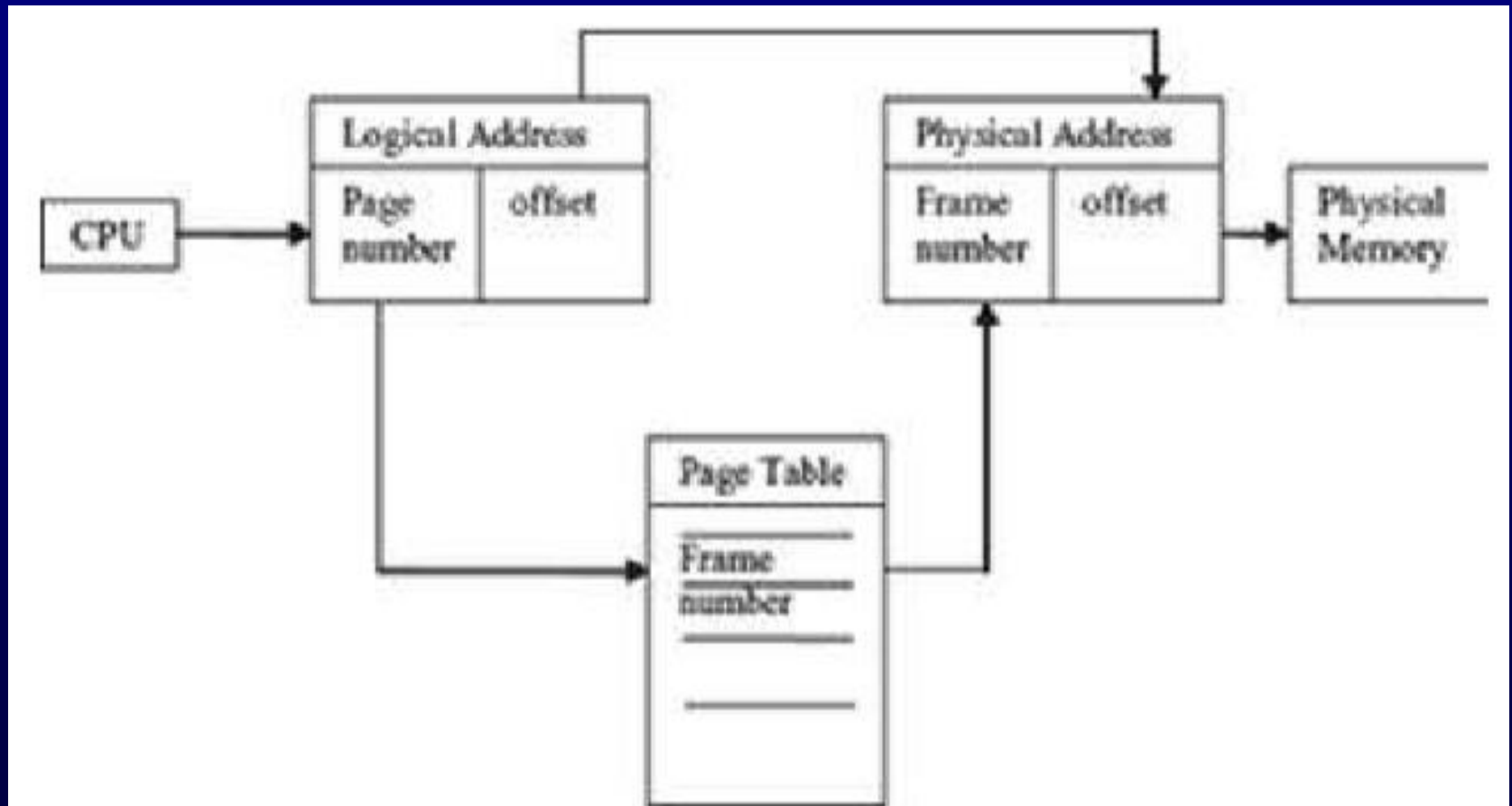
2. Memory Management : Paging

Figure 7.10 Paging

Functions of Operating System

2. Memory Management : Virtual Memory

➢ A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard disk that's set up to emulate the computer's RAM.

➢ In the memory management schemes discussed in above, the whole process is kept in memory before the execution starts. However, for some applications, large memory is required to run the applications, and the whole program cannot be loaded into the memory.

➢ Virtual memory allows the execution of those processes that are not completely in memory.

➢ Virtual memory is commonly implemented by demand paging. Demand paging is similar to paging with swapping. Swapping is transferring of block of data from the on-line secondary storage like hard disk to the memory and vice versa.

➢ In demand paging, the processes reside in the online secondary memory. When a process executes and a page is required, that page is swapped-in into the memory. This allows execution of large-sized programs without loading them completely into the memory.

Functions of Operating System

2. Memory Management : Virtual Memory

➢ In the memory management schemes discussed above, the whole process is kept in memory before the execution starts. However, for some applications, large memory is required to run the applications, and the whole program cannot be loaded into the memory.

➢ A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of a hard disk that's set up to emulate the computer's RAM.

➢ Virtual memory allows the execution of those processes that are not completely in memory.

➢ Virtual memory is commonly implemented by demand paging. Demand paging is similar to paging with swapping. Swapping is transferring of block of data from the on-line secondary storage like hard disk to the memory and vice versa. ☐ In demand paging, the processes reside in the online secondary memory. When a process executes and a page is required, that page is swapped-in into the memory. This allows execution of large-sized programs without loading them completely into the memory.

Course Contents

Functions of Operating System:

3. File Management: The file management tasks include

1) create and delete both files and directories,
2) provide access to files,
3) allocate space for files,
4) keep back-up of files, and
5) secure files.

4. Device Management: The device management tasks handled by OS are

1) open, close and write device drivers, and
2) communicate, control and monitor the device driver.

Course Contents

Functions of Operating System:

3. File Management:

➢ The file management function of the operating system involves handling the file system which consists of two parts -a set of files, and a directory structure.

➢ The operating system manages the storage media like the disk and implements the abstract concept of the file. System calls are an interface between the process and the operating system. Operating system provides system calls for creating, reading, writing, deleting, repositioning, and truncating a file. Some of the operations that can be performed on a directory are—search for a file, create, delete and rename a file, list a directory, and traverse the file system within the directory. The user simply uses the system calls like "dir", "list" to perform operation on a file or directory, without going into the details of its working.

Course Contents

Functions of Operating System:

4. Device Management: The device management tasks handled by OS are

1) open, close and write device drivers, and
2) communicate, control and monitor the device driver.

Course Contents

Functions of Operating System:

4. Device Management:

➢ OS manages and controls the devices attached to the computer. OS provides appropriate functionality to the application programs for controlling different aspects of the devices.

➢ The OS communicates with the I/O hardware via the device driver software. The device driver software comes along with each device.

➢ In addition to managing the peripheral devices, OS also provides various services related to I/O like I/O scheduling, buffering, spooling, and error handling.

Functions of Operating System: 4. Device Management:

➢ **Scheduling of I/O requests** involves ordering the requests to improve performance of the system and provide fair access to all processes. A queue of request is maintained for each device and I/O scheduler re-arranges the queue to improve the efficiency of the overall system.

➢ **Buffer** is a memory area that stores the data, while it is being transferred between two devices or between a device and an application. The speed at which the I/O device can transfer data is different from the speed at which the data is processed. Buffering handles the speed mismatch by storing the data in a buffer till the complete data has arrived and then writing it in a single write operation.

➢ **Spool** (Simultaneous Peripheral Operation On-Line) is a buffer in memory area or disk. Spooling stores the jobs in a spool where the device can access it when it is ready. Spooling is commonly used for printers. Users may give several print commands, and continue working with other operations. However, the printer can print only one job at a time. The rest of the jobs are stored in the spool in a queue, and the printer accesses the spool when it is ready to print the next job.

Functions of Operating System: 4. Device Management:

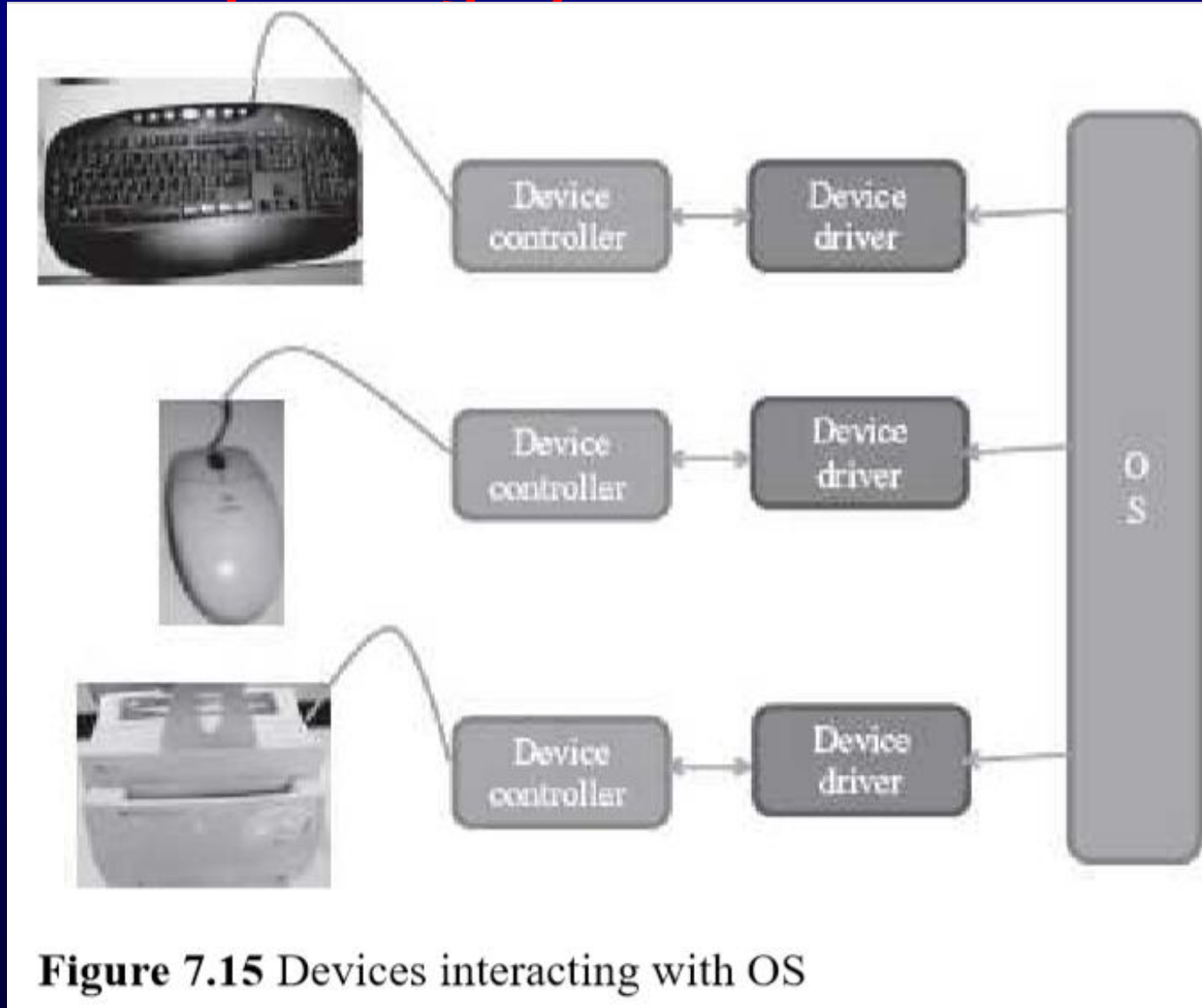**Figure 7.15** Devices interacting with OS

Course Contents

Functions of Operating System:

5. Protection and Security:

➢ OS protects the resources of system. User authentication, file attributes like read, write, encryption, and back-up of data are used by OS to provide basic protection.

Course Contents

- Security mechanism prevents unauthorized access to the computer. Security concerns include - security of software, security of data stored in the computer, and security of physical resources of the computer.

- In a personal computer, security can be ensured using

  - (1) user accounts—individual accounts for each user, (2) user authentication—using password protection, (3) access rights—define rights for access of different kind of information for different people, (4) data encryption (BitLocker Drive Encryption in windows10)—store data in computer in encrypted form, and (5) data backup—storing data on a peripheral device other than the hard disk.

- In a networked environment, only trusted computers should be able to share data. Some of the common security threats occur due to hacking, viruses etc.

Course Contents

Functions of Operating System:

6. User Interface or Command Interpreter:

➢ Operating system provides an interface between the computer user and the computer hardware. The user interface is a set of commands or a graphical user interface via which the user interacts with the applications and the hardware.

Course Contents

Functions of Operating System:

6. User Interface or Command Interpreter:

➢ The primary goal of operating system is to make the computer convenient for use by its user. It should allow users to easily access and communicate with the applications and the hardware.

➢ The users can interact with the computer by using mainly two kinds of interfaces—(1) Command Line Interface (CLI), and (2) Graphical User Interface (GUI).

➢ CLI requires the user to interact with operating system in the form of text keyed in from the keyboard. In this, the user has to learn and remember the different commands required for copying, deleting, opening a file or folder etc. MS-DOS and Linux shell are examples of command line mode of interfaces.

Operating system

Examples Operating system:

The three most common operating systems for personal computers are

Microsoft Windows,

macOS, and

Linux.

Operating system

Examples Operating system: Microsoft Windows

Microsoft created the **Windows** operating system in the mid-1980s. There have been many different versions of Windows, but the most recent ones are **Windows 10** (released in 2015), **Windows 8** (2012), **Windows 7** (2009), and **Windows Vista** (2007).

MD-DOS (Microsoft Disk Operating System) [1981]

Windows 1.0 - 2.0 [1985 - 1992]

Windows 3.0 - 3.1 [1990 - 1994]

Windows 95 [1995]

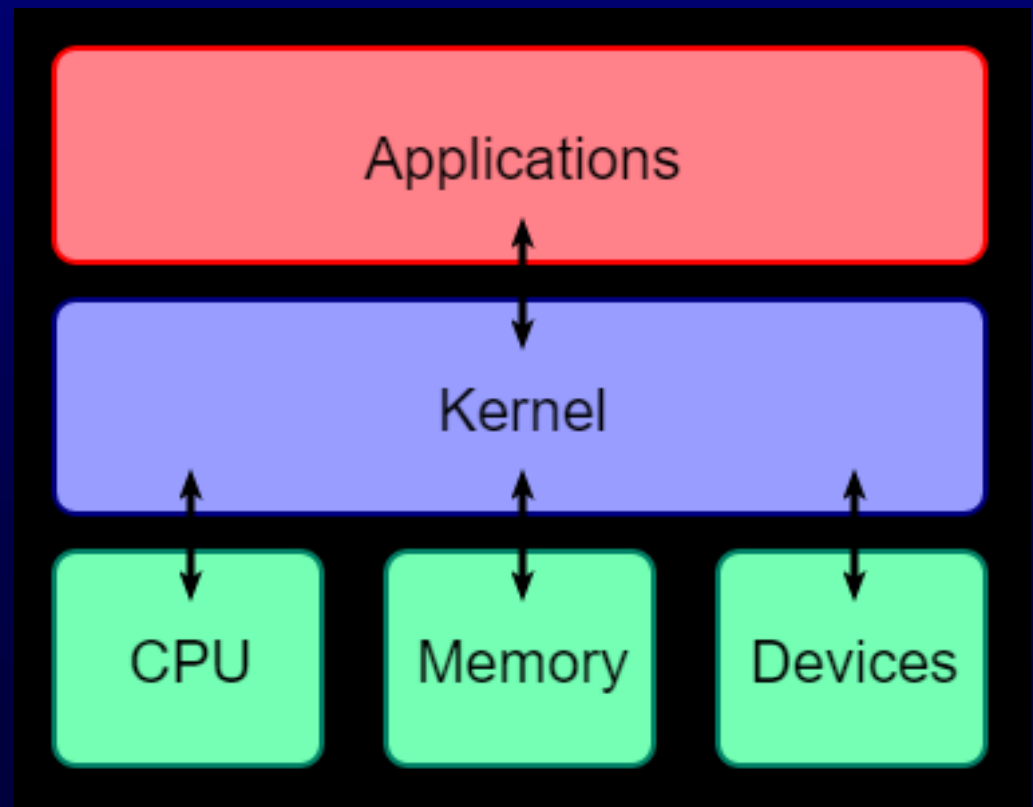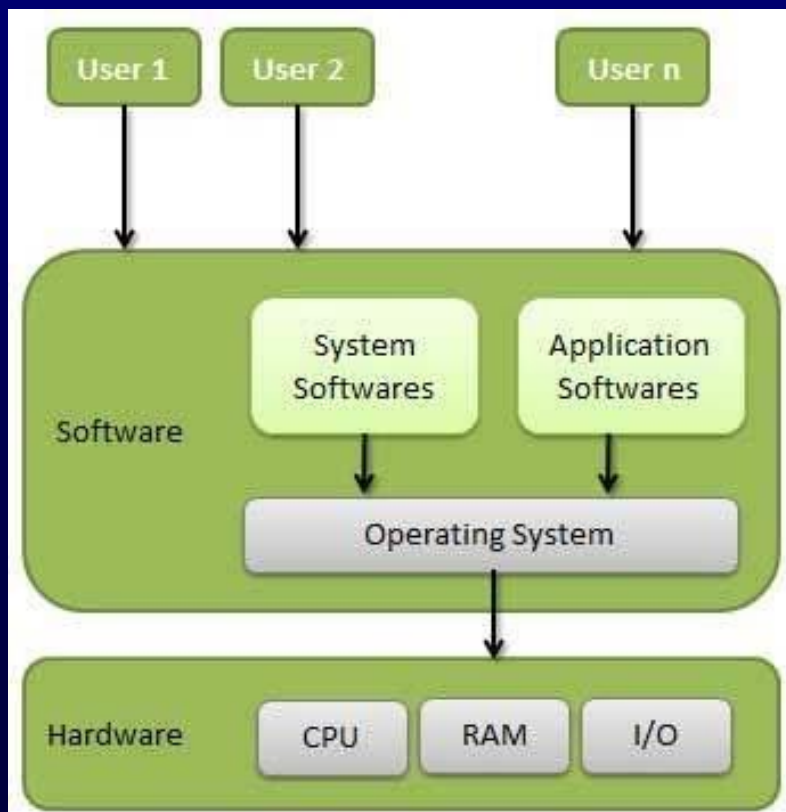Windows 98 [1998]

Windows ME (Millennial Edition) [2000]

Windows NT 31. - 4.0 [1993 - 1996]

Windows 2000 [2000]

Windows comes pre-loaded on most new PCs.

Operating system

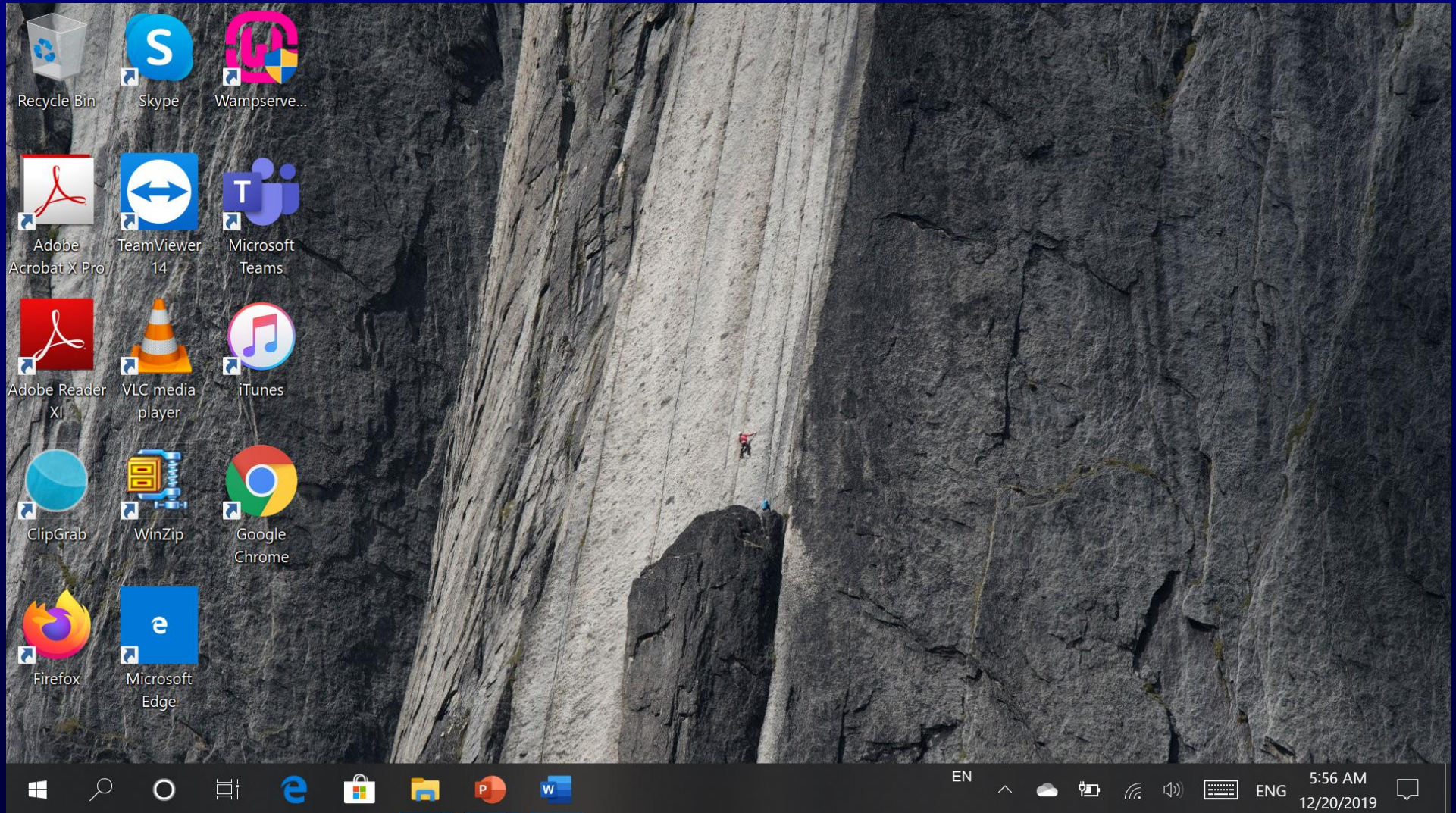## Operating system Vs Kernel (Linux operating system)

Operating system

Examples Operating system: Microsoft Windows

Operating system

Examples Operating system: Microsoft Windows

Operating system

Examples Operating system: macOS

macOS (previously called **OS X**) is a line of operating systems created by Apple. It comes preloaded on all Macintosh computers, or Macs. Some of the specific versions include **Mojave** (released in 2018), **High Sierra** (2017), and **Sierra** (2016).

Operating system

Examples Operating system: macOS

List of Mac OS versions

- OS X 10 beta: Kodiak - 13 September 2000
- OS X 10.0: Cheetah - 24 March 2001
- OS X 10.1: Puma - 25 September 2001
- OS X 10.2: Jaguar - 24 August 2002
- OS X 10.3 Panther (Pinot) - 24 October 2003
- OS X 10.4 Tiger (Merlot) - 29 April 2005
- [OS X 10.4.4 Tiger (Chardonnay)]
- OS X 10.5 Leopard (Chablis) - 26 October 2007
- OS X 10.6 Snow Leopard - 28 August 2009
- OS X 10.7 Lion (Barolo) - 20 July 2011
- OS X 10.8 Mountain Lion (Zinfandel) - 25 July 2012

- OS X 10.9 Mavericks (Cabernet) - 22 October 2013
- OS X 10.10: Yosemite (Syrah) - 16 October 2014
- OS X 10.11: El Capitan (Gala) - 30 September 2015
- macOS 10.12: Sierra (Fuji) - 20 September 2016
- macOS 10.13: High Sierra (Lobo) - 25 September 2017
- macOS 10.14: Mojave (Liberty) - 24 September 2018
- macOS 10.15: Catalina - 7 October 2019

Operating system

Examples Operating system: macOS

Operating system

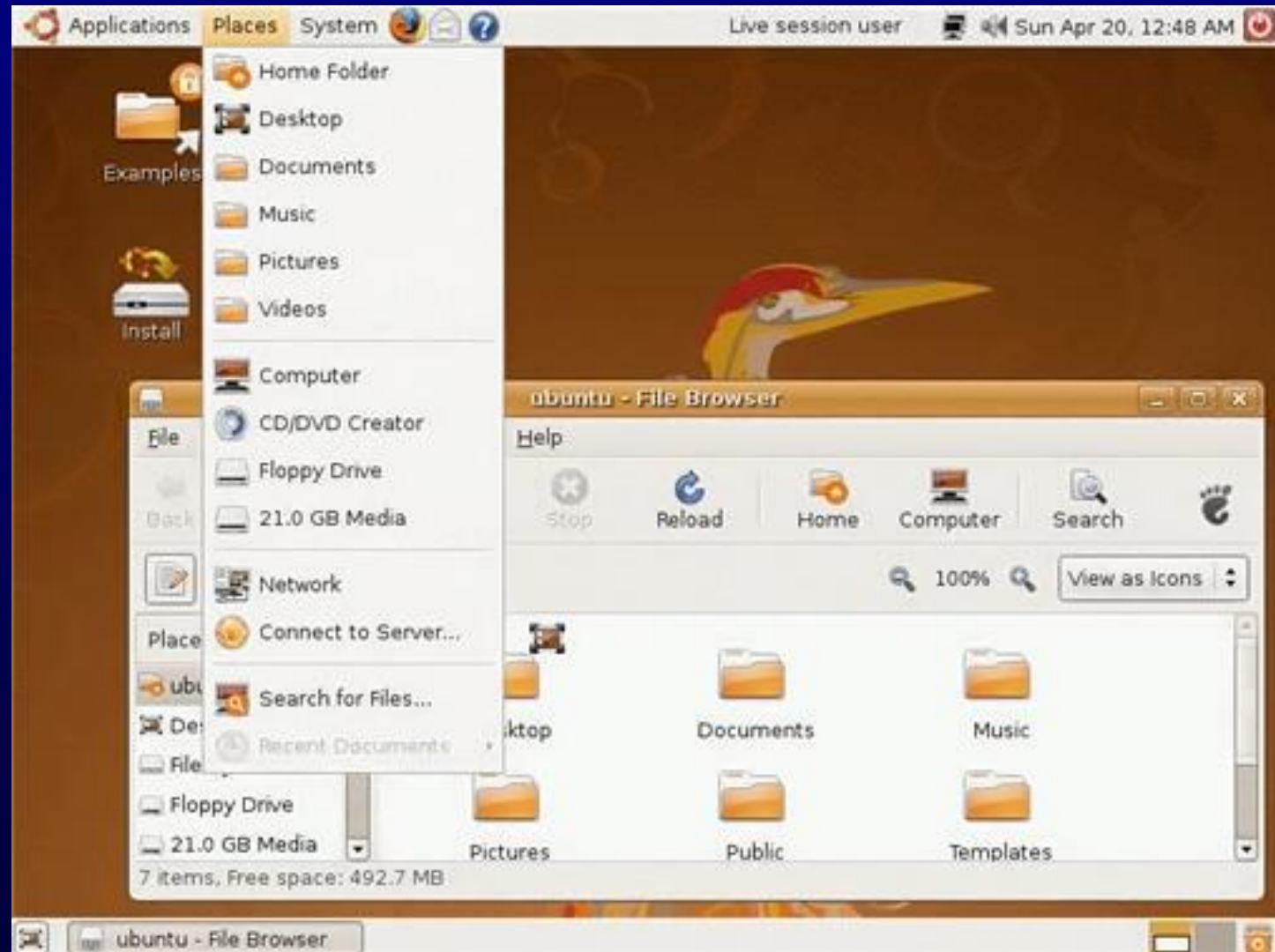**Examples Operating system:** Linux

Linux (pronounced **LINN-ux**) is a family of **open-source** operating systems, which means they can be modified and distributed by anyone around the world.

This is different from **proprietary software** like Windows, which can only be modified by the company that owns it.

The advantages of Linux are that it is **free**, and there are many different **distributions**—or versions.

Operating system

Examples Operating system: Linux (Ubuntu)

Operating system

Examples Operating system: Mobile OS

Mobile devices such as **phones**, **tablet computers**, and **MP3 players** are different from desktop and laptop computers, so they run operating systems that are designed specifically for mobile devices.

Examples of mobile operating systems include

- Apple iOS
- Google Android
- Windows 10 Mobile by Microsoft.

Operating system

Examples Operating system: Mobile OS

Operating system

Examples Operating system: Mobile OS

Top Mobile OS are:

- **Symbian (**property of Nokia)
- **Android (**from Google)
- **Apple iOS (From Apple,** iOS has been used in all iPhones, iPod & iPad.)
- **Blackberry OS (**Blackberry line of smartphones from RIM (Research In Motion))

- **Windows OS**
- **BADA (**Samsung owns BADA)
- **Palm OS (Garnet OS) -** for PDAs (Personal Digital Assistance)
- **Open WebOS (**by Palm Inc but own by Hewlett-Packard)
- **Maemo**
- **MeeGo**
- **Verdict**