MODULE 1

CHAPTER 1 - INTRODUCTION TO OPERATING SYSTEMS

CO – Students will be able to summarize the operating system functions and its structures

Prepared By Mr. EBIN PM, AP, IESCE

OPERATING SYSTEMS

- An operating system is a program that manages the computer hardware.
- It acts as an intermediary between the user of a computer and the computer hardware.
- The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner.
- Some operating systems are designed to be convenient, others to be efficient, and others some combination of the two.

  Efficient - optimum resource utilization

  Convenient - user interaction is simple.

- An operating system is an important part of almost every computer system.

➢A computer system can be divided roughly into four components:

- **Hardware**
- **Operating system**
- **Application programs**
- **Users**

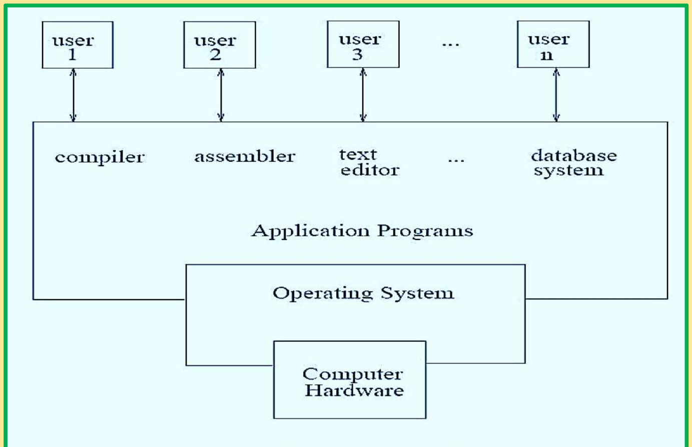
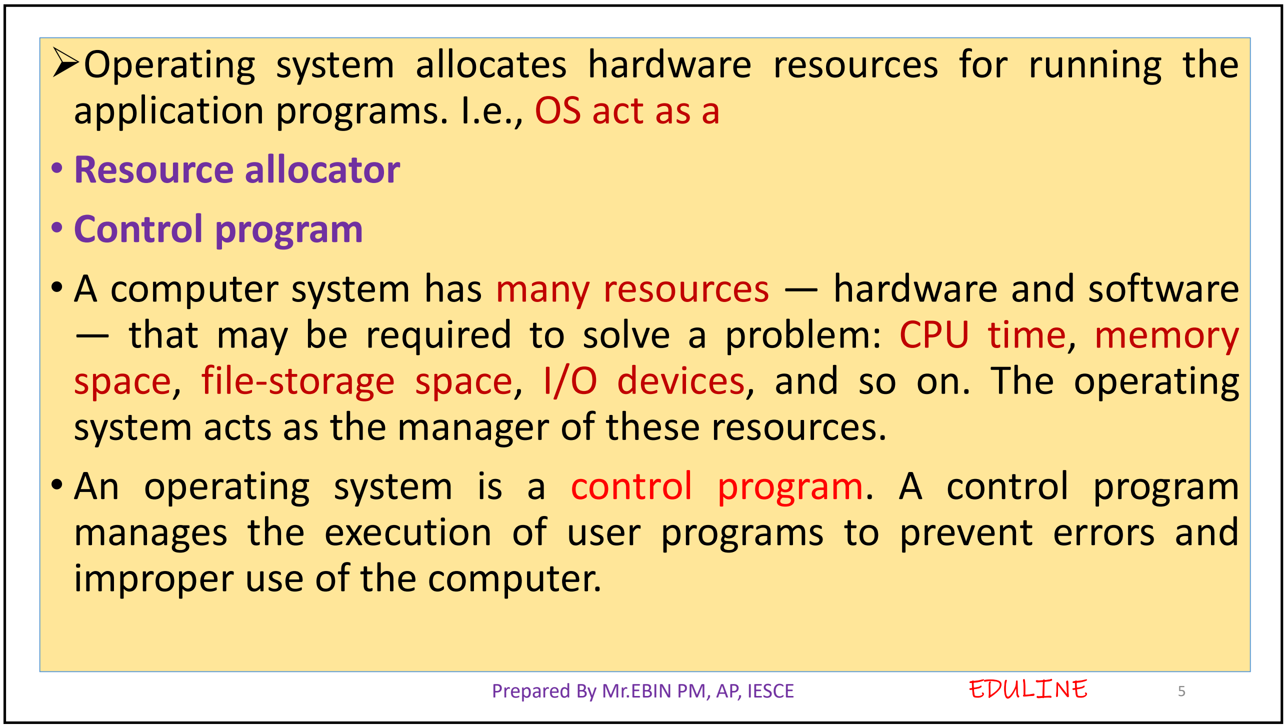


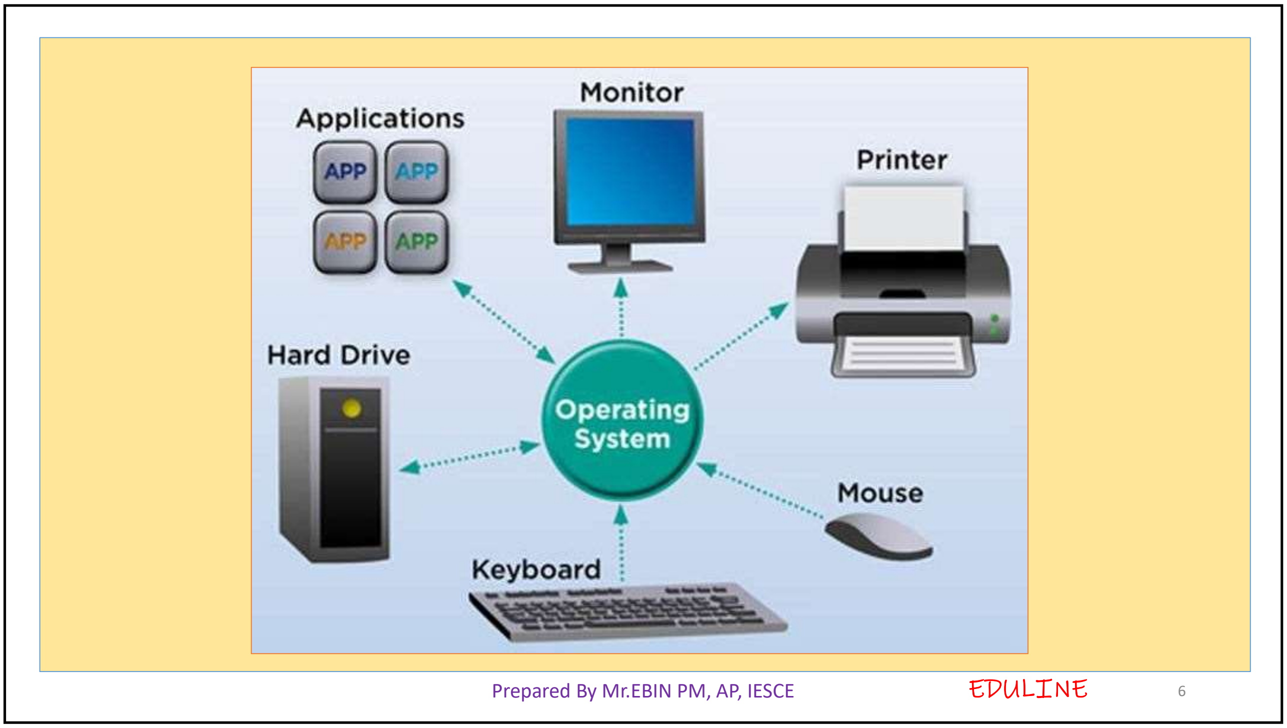
Figure: Abstract view of the components of a computer system.

- **The hardware** — the central processing unit (CPU), the memory, and the input/output (I/O) devices — provides the basic computing resources.
- **The application programs** — such as word processors, spreadsheets, compilers, and web browsers — define the ways in which these resources are used to solve the computing problems of the users.
- **The operating system** controls and coordinates the use of the hardware among the various application programs for the various users.



- ➢Operating system allocates hardware resources for running the application programs. I.e., OS act as a
- **Resource allocator**
- **Control program**
- A computer system has many resources — hardware and software — that may be required to solve a problem: CPU time, memory space, file-storage space, I/O devices, and so on. The operating system acts as the manager of these resources.
- An operating system is a control program. A control program manages the execution of user programs to prevent errors and improper use of the computer.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 5

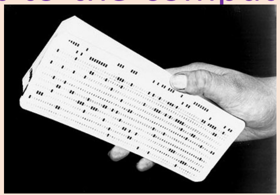


Prepared By Mr.EBIN PM, AP, IESCE EDULINE 6

TYPES OF OPERATING SYSTEMS (OS)

1. BATCH SYSTEMS

- Here the user did not interact directly with the computer systems. Rather, the user prepared a job and submitted it to the computer operator program.
- The job was usually in the form of punch cards.



- At some later time (after minutes, hours, or days), the output appeared.
- Its major task of the OS was to transfer control automatically from one job to the next.

- To speed up processing, operators batched together jobs with similar needs and ran them through the computer as a group.
- The operator would sort programs into batches with similar requirements and, as the computer became available, would run each batch.
- In this execution environment, the CPU is often idle, because the speeds of the mechanical I/O devices are slower than are those of electronic devices.
- Using SPOOLing the idle time of the CPU can be reduced.
- The common input devices are card readers and tape drivers.
- The output devices are line printers, tape drives and card punches.

- The introduction of disk technology allowed the operating system to keep all jobs on a disk, rather than in a serial card reader.
- With direct access to several jobs, the operating system could perform job scheduling, to use resources and perform tasks efficiently

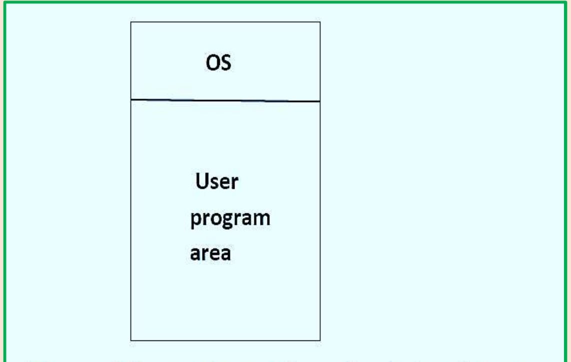
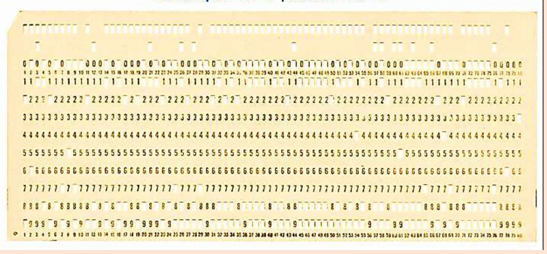


Figure: Memory layout for a simple batch system.

Example of a punch card



2. MULTIPROGRAMMED SYSTEMS

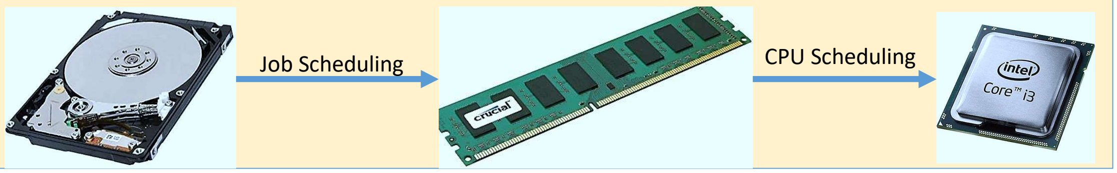
- Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.
- The operating system keeps several jobs in memory simultaneously.
- This set of jobs is a subset of the jobs kept in the job pool.
- The operating system picks and begins to execute one of the jobs in the memory.
- Eventually, the job may have to wait for some task, such as an I/O operation, to complete.

- In a multiprogramming system, the operating system simply switches to, and executes, another job.
- When that job needs to wait, the CPU is switched to another job, and so on.
- Eventually, the first job finishes waiting and gets the CPU back.
- As long as at least one job needs to execute, the CPU is never idle.

➢JOB POOL

- All the jobs that enter the system are kept in the job pool. This pool consists of all processes residing on disk awaiting allocation of main memory.

- If several jobs are ready to be brought into memory, and if there is not enough room for all of them, then the system must choose among them. Making this decision is **job scheduling.**
- When the operating system selects a job from the job pool, it loads that job into memory for execution.
- Having several programs in memory at the same time requires some form of memory management.
- If several jobs are ready to run at the same time, the system must choose among them. Making this decision is **CPU scheduling.**



❖Multi programming requirements are

- Protection and security
- Large memory
- Proper job mixing
- Job Scheduling
- CPU scheduling
- Disk management
- Main memory management

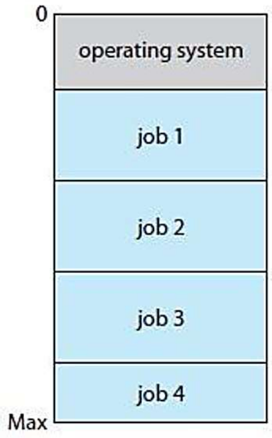
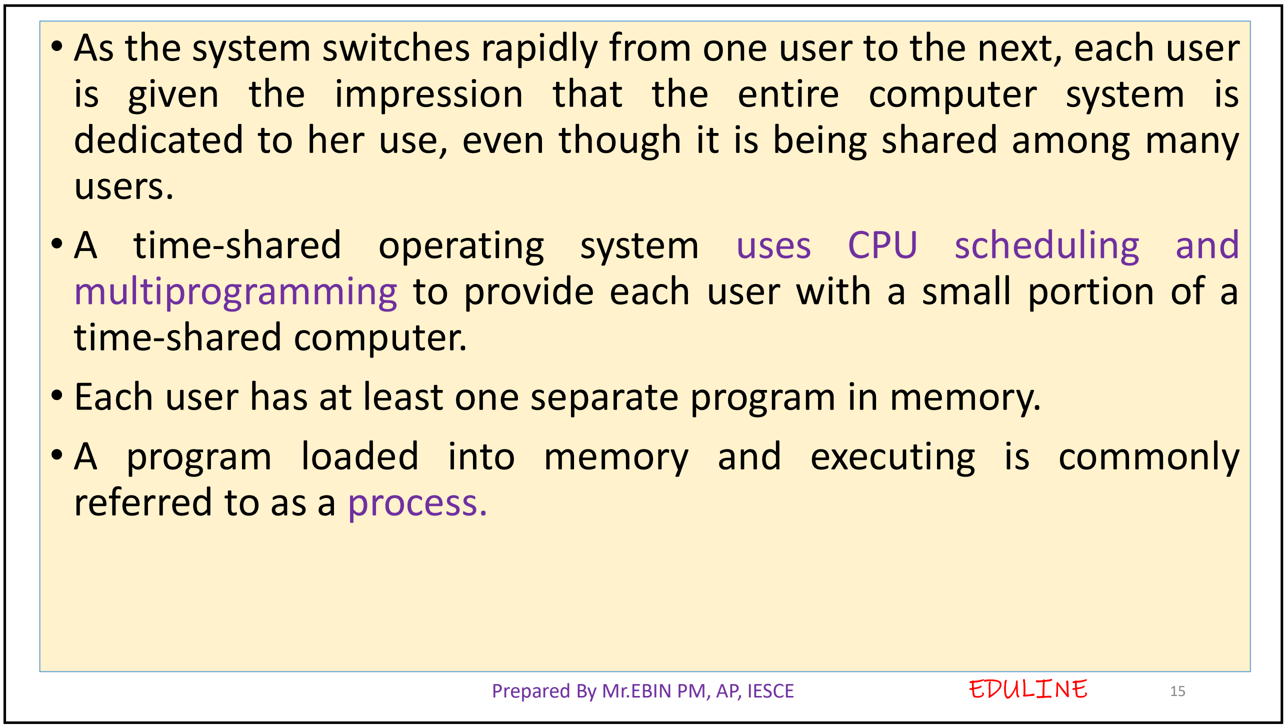


Figure: Memory layout for a multiprogramming system.

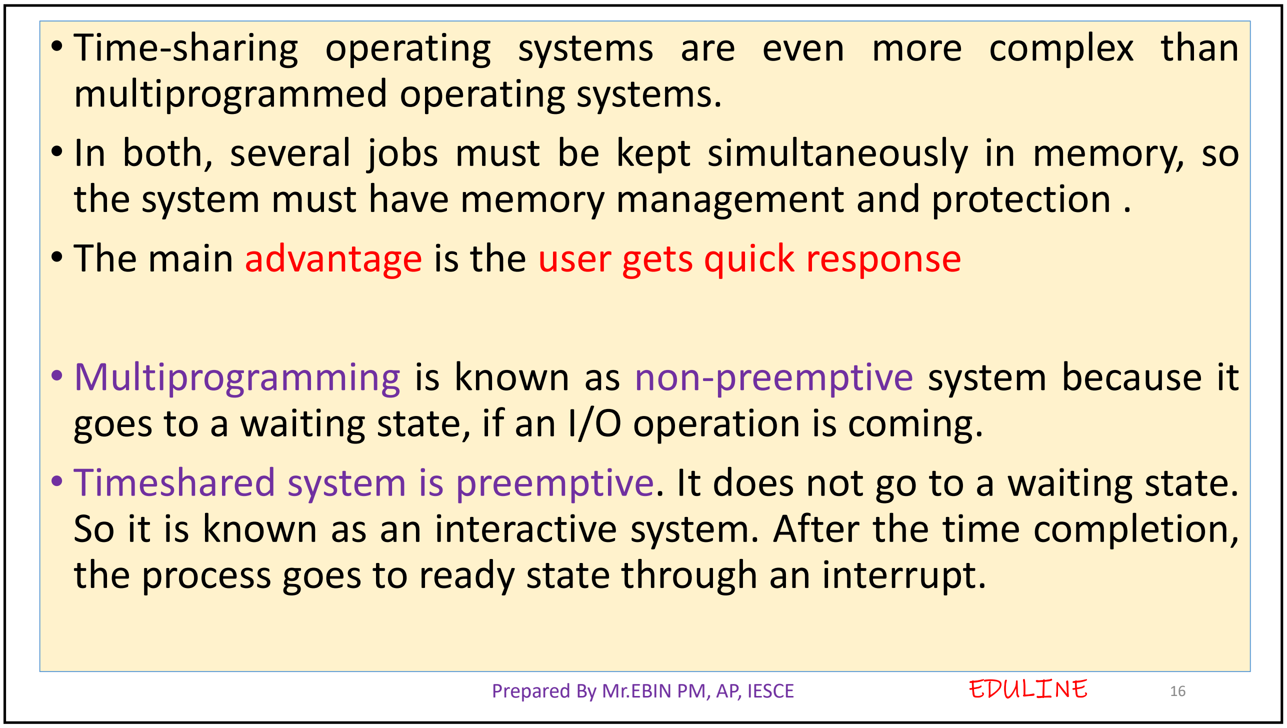
3. TIME-SHARING SYSTEMS (MULTI-TASKING SYSTEM)

- Time sharing (or multitasking) is a logical extension of multiprogramming.
- The CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.
- A time-shared operating system allows many users to share the computer simultaneously.
- Since each action or command in a time-shared system tends to be short, only a little CPU time is needed for each user.



- As the system switches rapidly from one user to the next, each user is given the impression that the entire computer system is dedicated to her use, even though it is being shared among many users.
- A time-shared operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer.
- Each user has at least one separate program in memory.
- A program loaded into memory and executing is commonly referred to as a process.

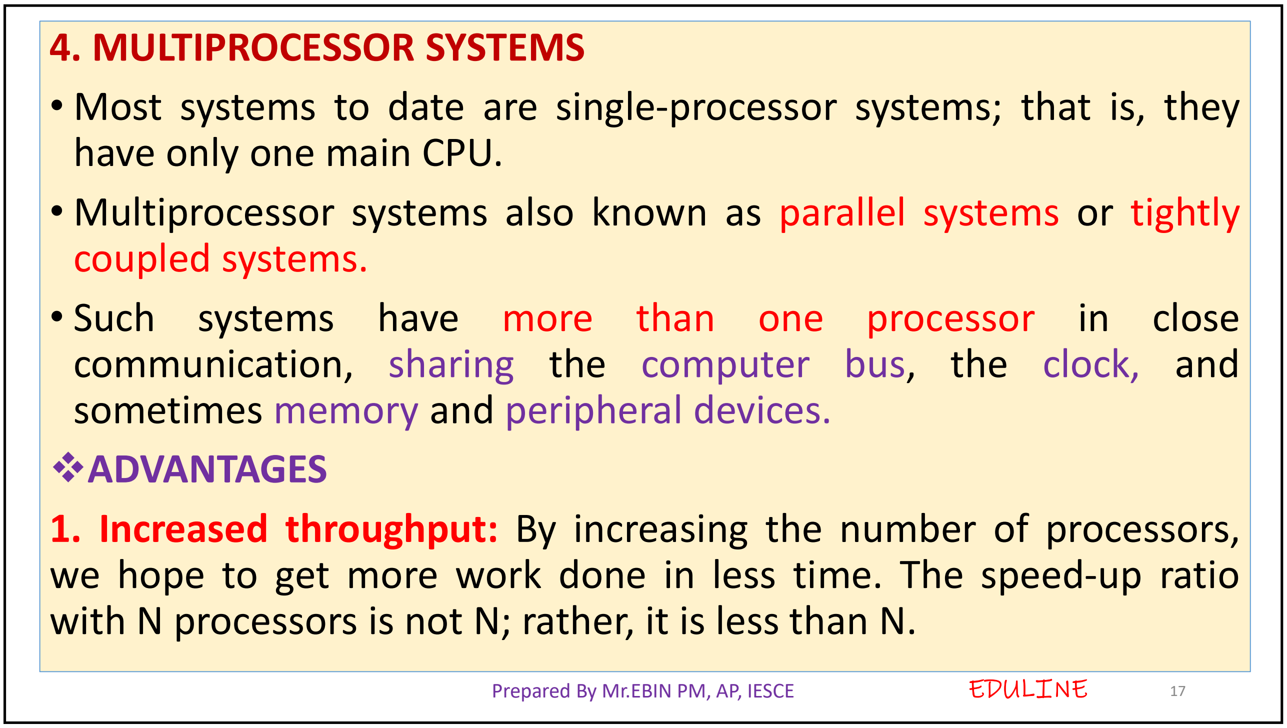
Prepared By Mr.EBIN PM, AP, IESCE EDULINE 15



- Time-sharing operating systems are even more complex than multiprogrammed operating systems.
- In both, several jobs must be kept simultaneously in memory, so the system must have memory management and protection .
- The main advantage is the user gets quick response

- Multiprogramming is known as non-preemptive system because it goes to a waiting state, if an I/O operation is coming.
- Timeshared system is preemptive. It does not go to a waiting state. So it is known as an interactive system. After the time completion, the process goes to ready state through an interrupt.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 16

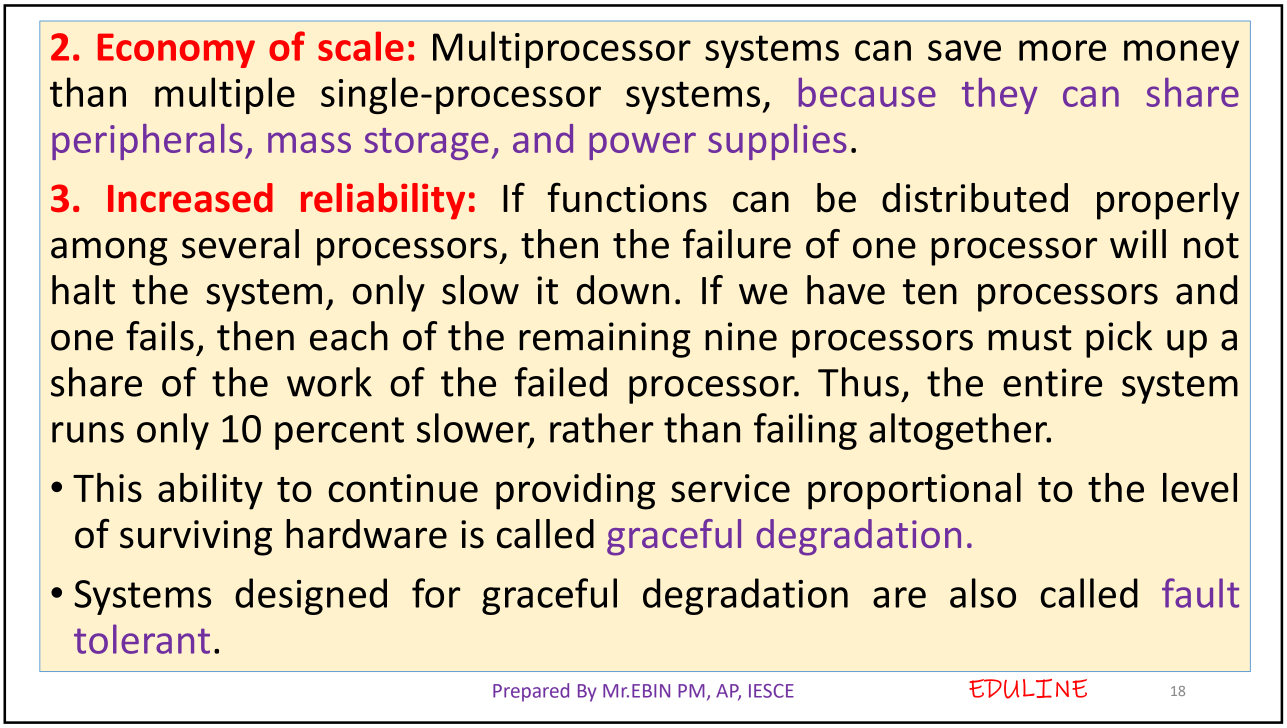


4. MULTIPROCESSOR SYSTEMS

- Most systems to date are single-processor systems; that is, they have only one main CPU.
- Multiprocessor systems also known as parallel systems or tightly coupled systems.
- Such systems have more than one processor in close communication, sharing the computer bus, the clock, and sometimes memory and peripheral devices.

❖**ADVANTAGES**

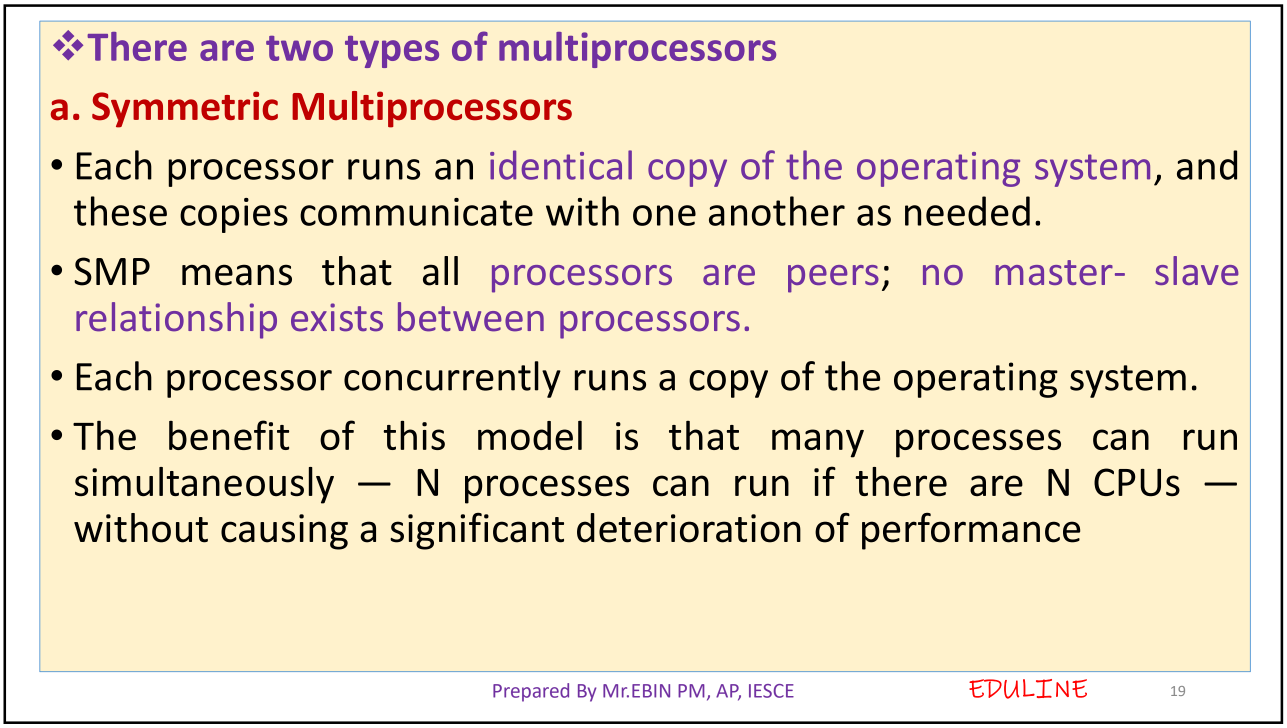
1. Increased throughput: By increasing the number of processors, we hope to get more work done in less time. The speed-up ratio with N processors is not N; rather, it is less than N.



2. Economy of scale: Multiprocessor systems can save more money than multiple single-processor systems, because they can share peripherals, mass storage, and power supplies.

3. Increased reliability: If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, only slow it down. If we have ten processors and one fails, then each of the remaining nine processors must pick up a share of the work of the failed processor. Thus, the entire system runs only 10 percent slower, rather than failing altogether.

- This ability to continue providing service proportional to the level of surviving hardware is called graceful degradation.
- Systems designed for graceful degradation are also called fault tolerant.

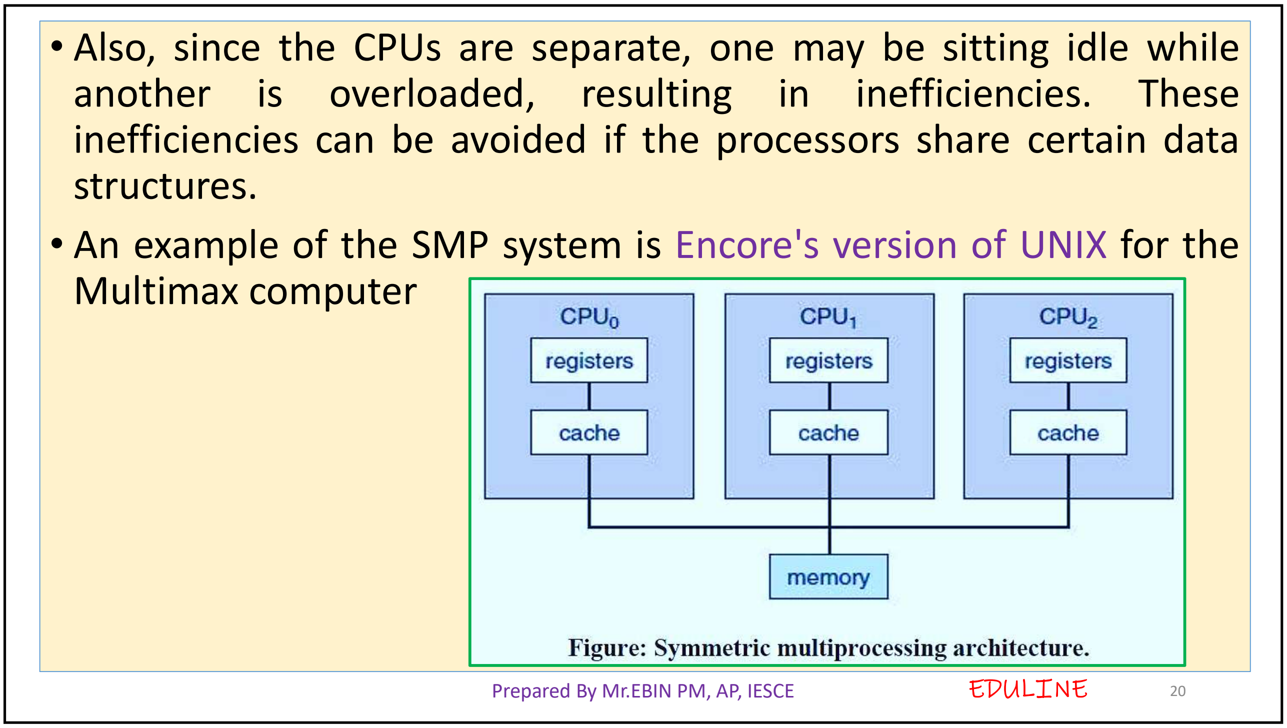


❖**There are two types of multiprocessors**

a. Symmetric Multiprocessors

- Each processor runs an identical copy of the operating system, and these copies communicate with one another as needed.
- SMP means that all processors are peers; no master- slave relationship exists between processors.
- Each processor concurrently runs a copy of the operating system.
- The benefit of this model is that many processes can run simultaneously — N processes can run if there are N CPUs — without causing a significant deterioration of performance

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 19

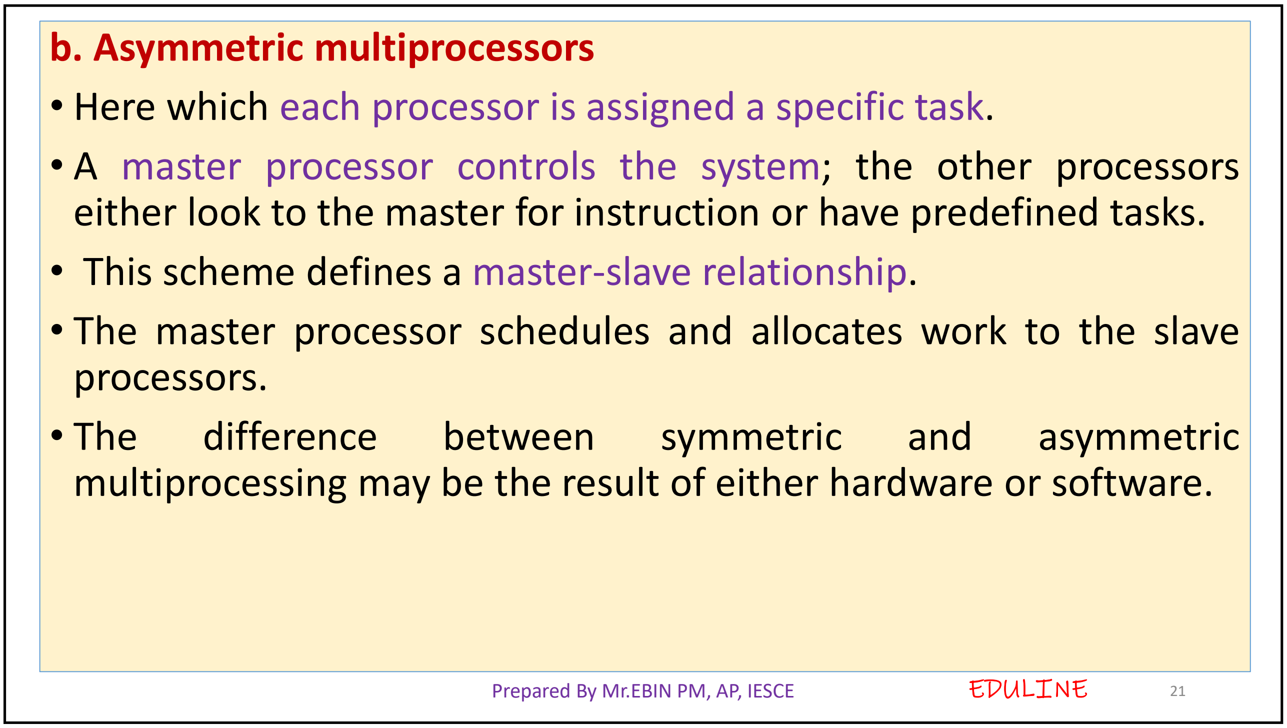


- Also, since the CPUs are separate, one may be sitting idle while another is overloaded, resulting in inefficiencies. These inefficiencies can be avoided if the processors share certain data structures.
- An example of the SMP system is Encore's version of UNIX for the Multimax computer

CPU_0
registers
cache
CPU_1
registers
cache
CPU_2
registers
cache
memory

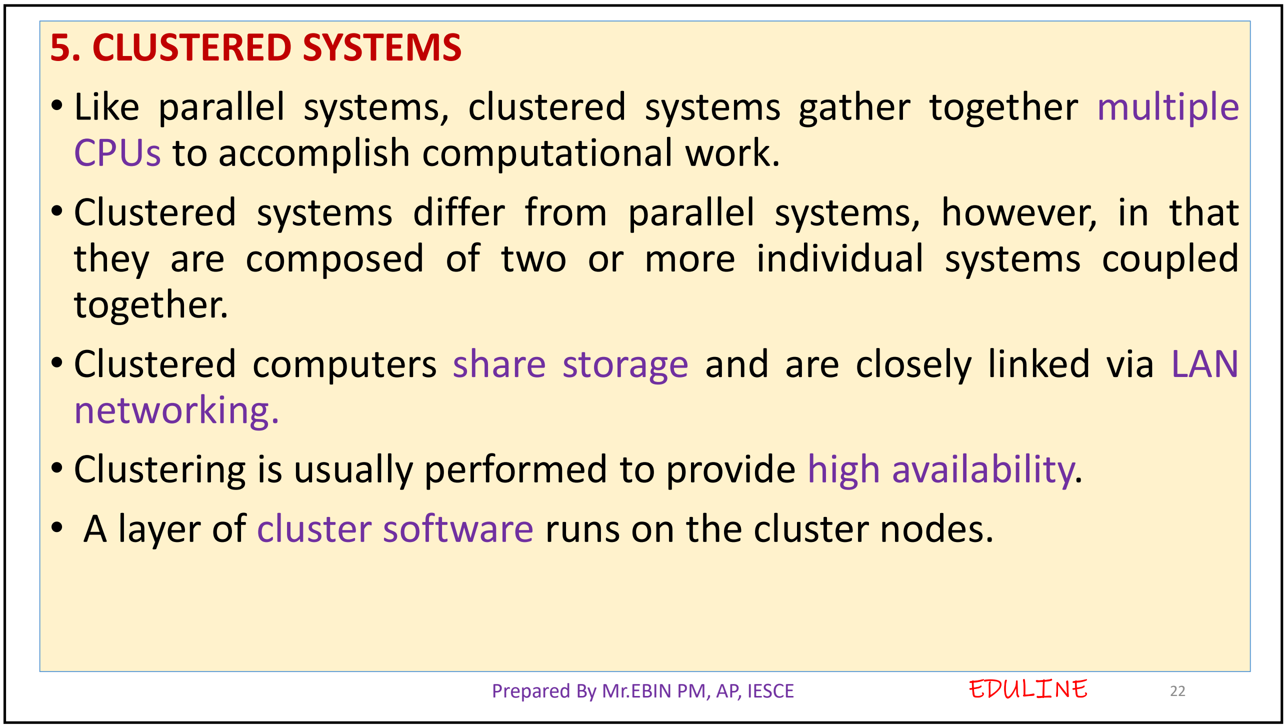
Figure: Symmetric multiprocessing architecture.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 20



b. Asymmetric multiprocessors

- Here which each processor is assigned a specific task.
- A master processor controls the system; the other processors either look to the master for instruction or have predefined tasks.
- This scheme defines a master-slave relationship.
- The master processor schedules and allocates work to the slave processors.
- The difference between symmetric and asymmetric multiprocessing may be the result of either hardware or software.



5. CLUSTERED SYSTEMS

- Like parallel systems, clustered systems gather together multiple CPUs to accomplish computational work.
- Clustered systems differ from parallel systems, however, in that they are composed of two or more individual systems coupled together.
- Clustered computers share storage and are closely linked via LAN networking.
- Clustering is usually performed to provide high availability.
- A layer of cluster software runs on the cluster nodes.

- Each node can monitor one or more of the others (over the LAN).
- If the monitored machine fails, the monitoring machine can take ownership of its storage, and restart the application(s) that were running on the failed machine.
- The failed machine can remain down, but the users and clients of the application would only see a brief interruption of service.

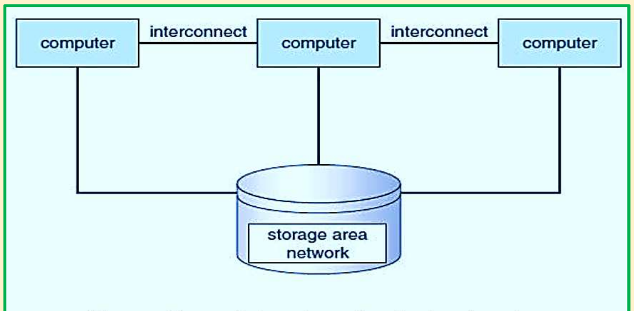


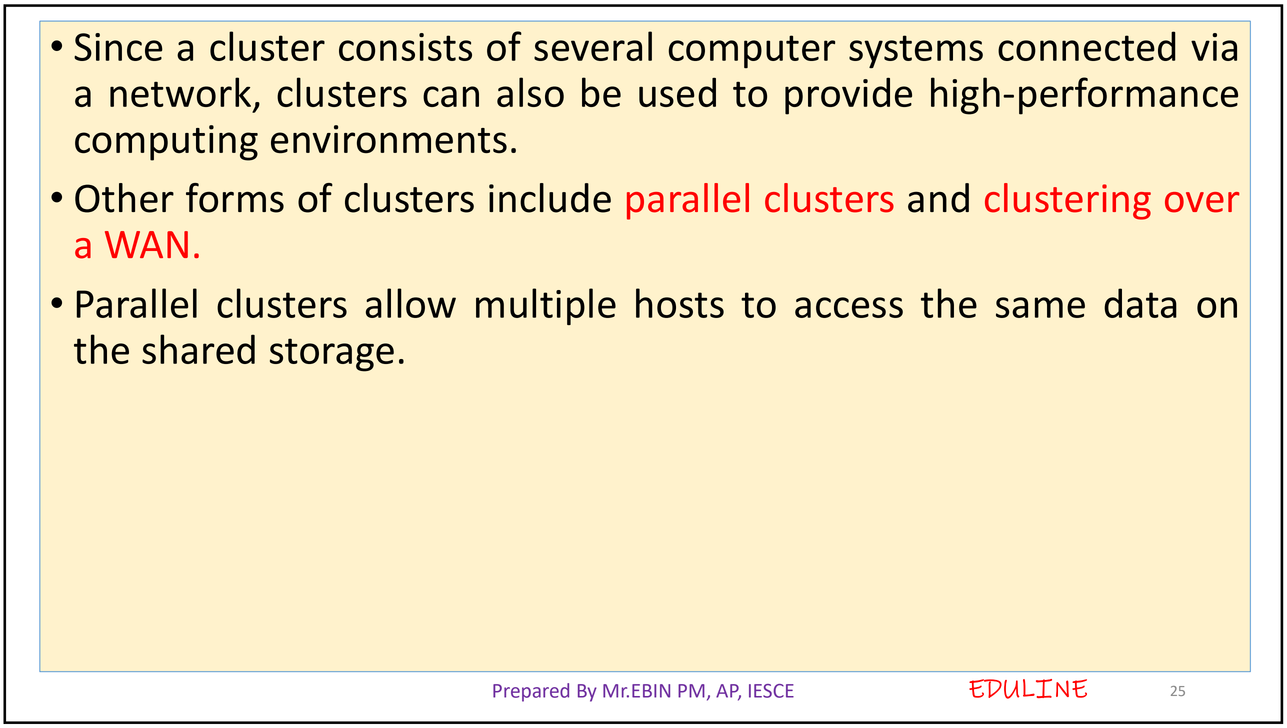
Figure: General structure of a clustered system.

a. Symmetric clustering

- In symmetric mode, two or more hosts are running applications, and they are monitoring each other.
- This mode is obviously more efficient, as it uses all of the available hardware. It does require that more than one application be available to run.

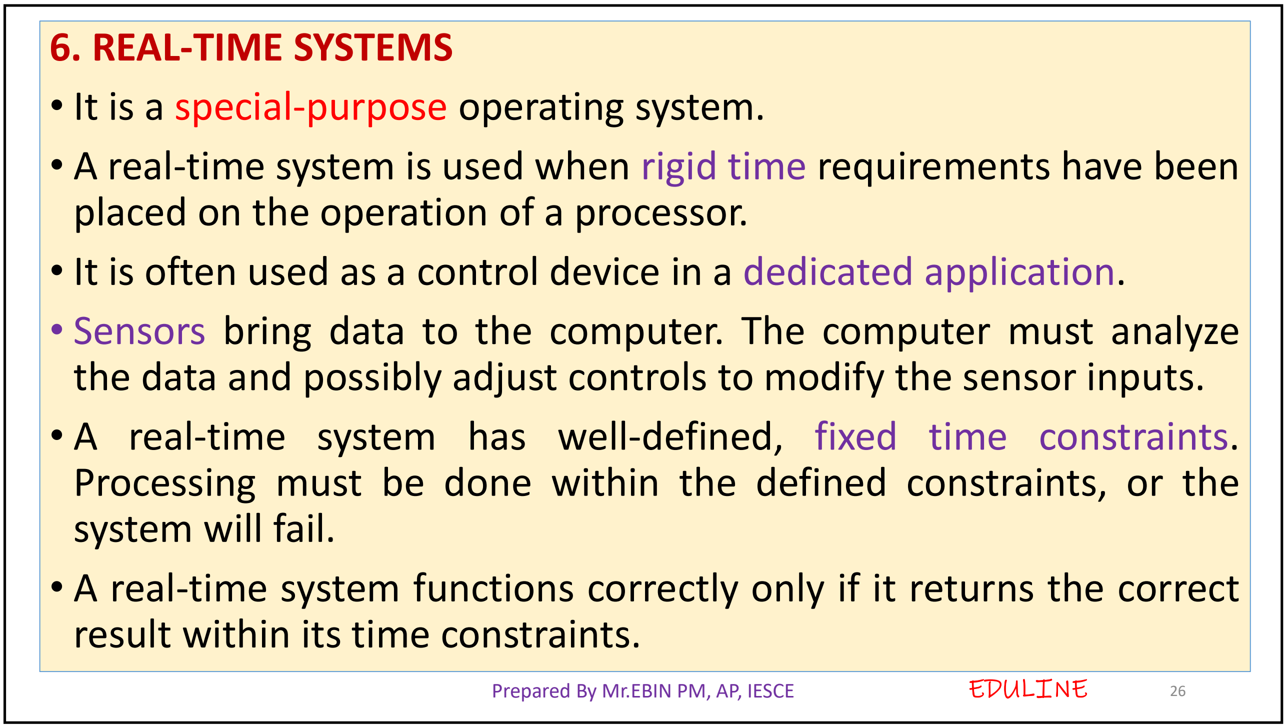
b. Asymmetric clustering

- In asymmetric clustering, one machine is in hot-standby mode while the other is running the applications.
- The hot-standby host (machine) does nothing but monitor the active server. If that server fails, the hot standby host becomes the active server



- Since a cluster consists of several computer systems connected via a network, clusters can also be used to provide high-performance computing environments.
- Other forms of clusters include parallel clusters and clustering over a WAN.
- Parallel clusters allow multiple hosts to access the same data on the shared storage.

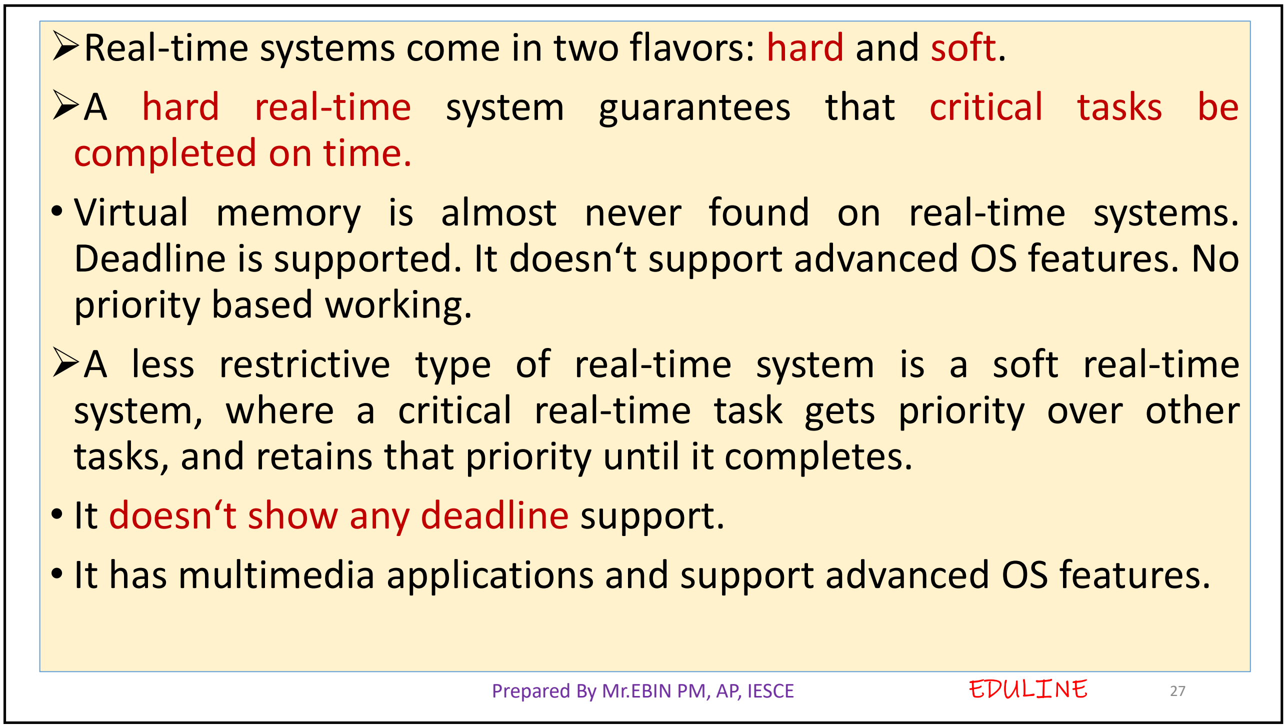
Prepared By Mr.EBIN PM, AP, IESCE EDULINE 25



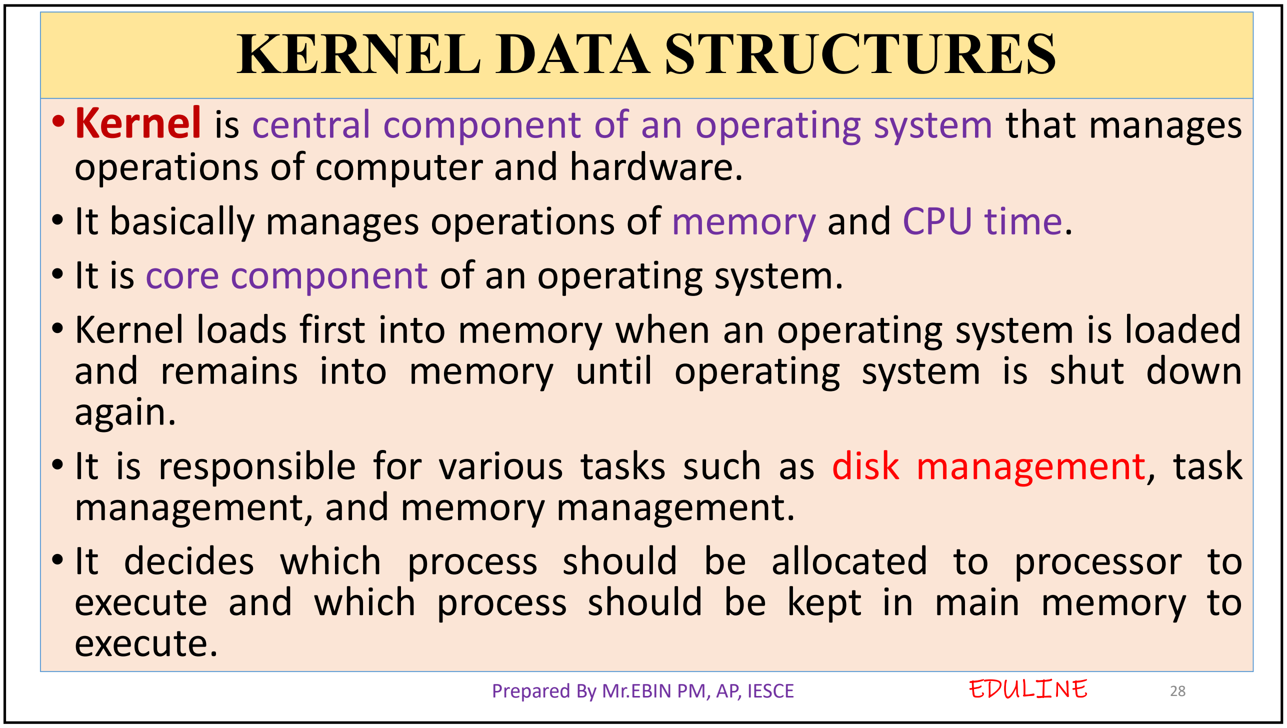
6. REAL-TIME SYSTEMS

- It is a special-purpose operating system.
- A real-time system is used when rigid time requirements have been placed on the operation of a processor.
- It is often used as a control device in a dedicated application.
- Sensors bring data to the computer. The computer must analyze the data and possibly adjust controls to modify the sensor inputs.
- A real-time system has well-defined, fixed time constraints. Processing must be done within the defined constraints, or the system will fail.
- A real-time system functions correctly only if it returns the correct result within its time constraints.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 26

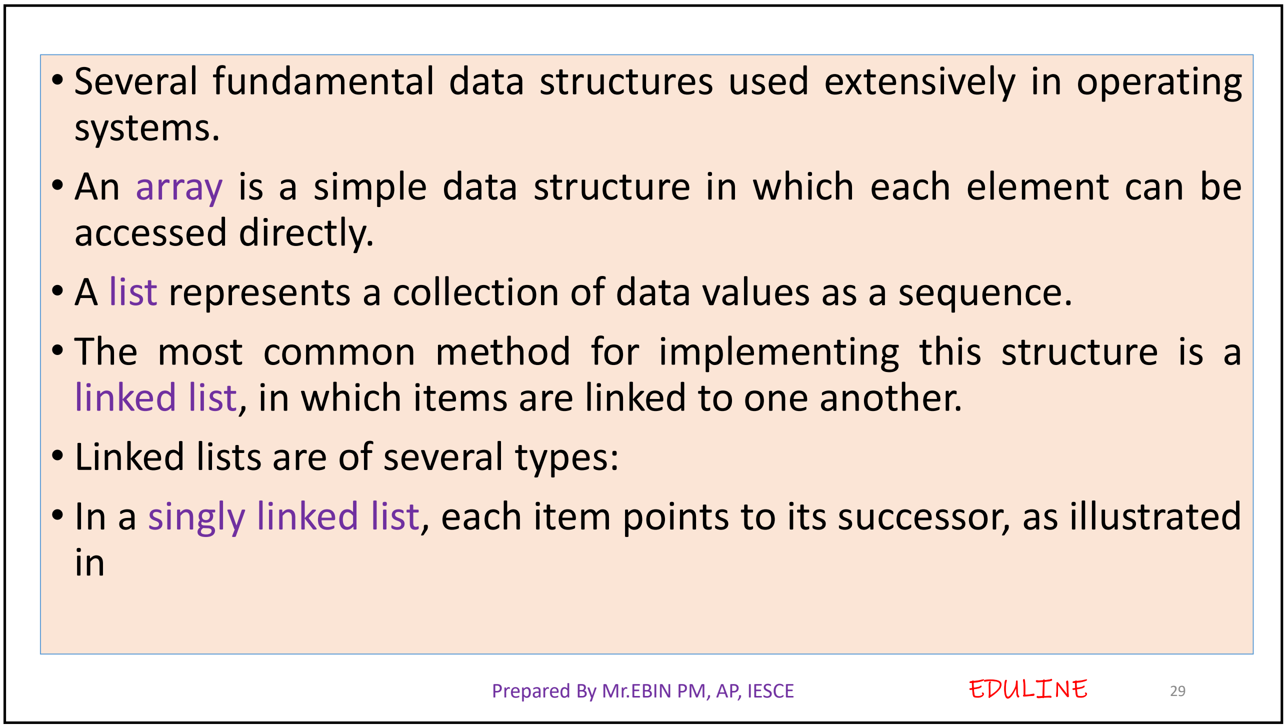


- ➢Real-time systems come in two flavors: hard and soft.
- ➢A hard real-time system guarantees that critical tasks be completed on time.
- Virtual memory is almost never found on real-time systems. Deadline is supported. It doesn‘t support advanced OS features. No priority based working.
- ➢A less restrictive type of real-time system is a soft real-time system, where a critical real-time task gets priority over other tasks, and retains that priority until it completes.
- It doesn‘t show any deadline support.
- It has multimedia applications and support advanced OS features.



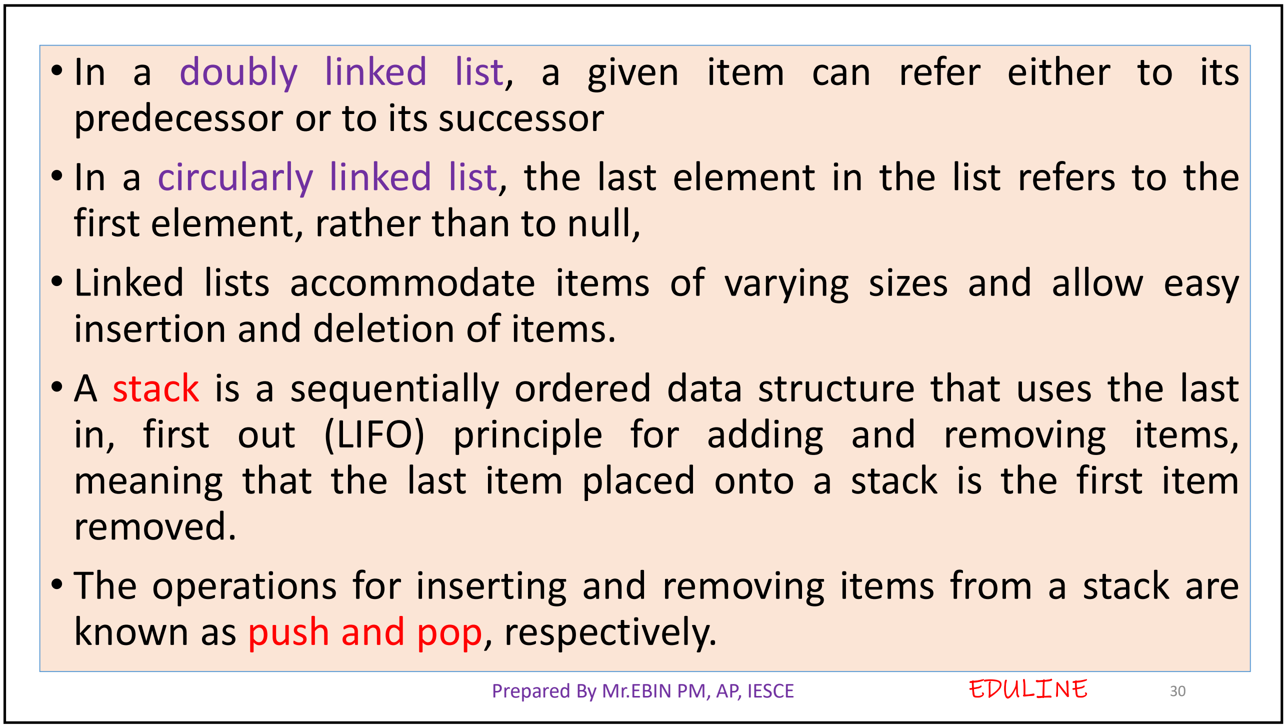
KERNEL DATA STRUCTURES

- **Kernel** is central component of an operating system that manages operations of computer and hardware.
- It basically manages operations of memory and CPU time.
- It is core component of an operating system.
- Kernel loads first into memory when an operating system is loaded and remains into memory until operating system is shut down again.
- It is responsible for various tasks such as disk management, task management, and memory management.
- It decides which process should be allocated to processor to execute and which process should be kept in main memory to execute.



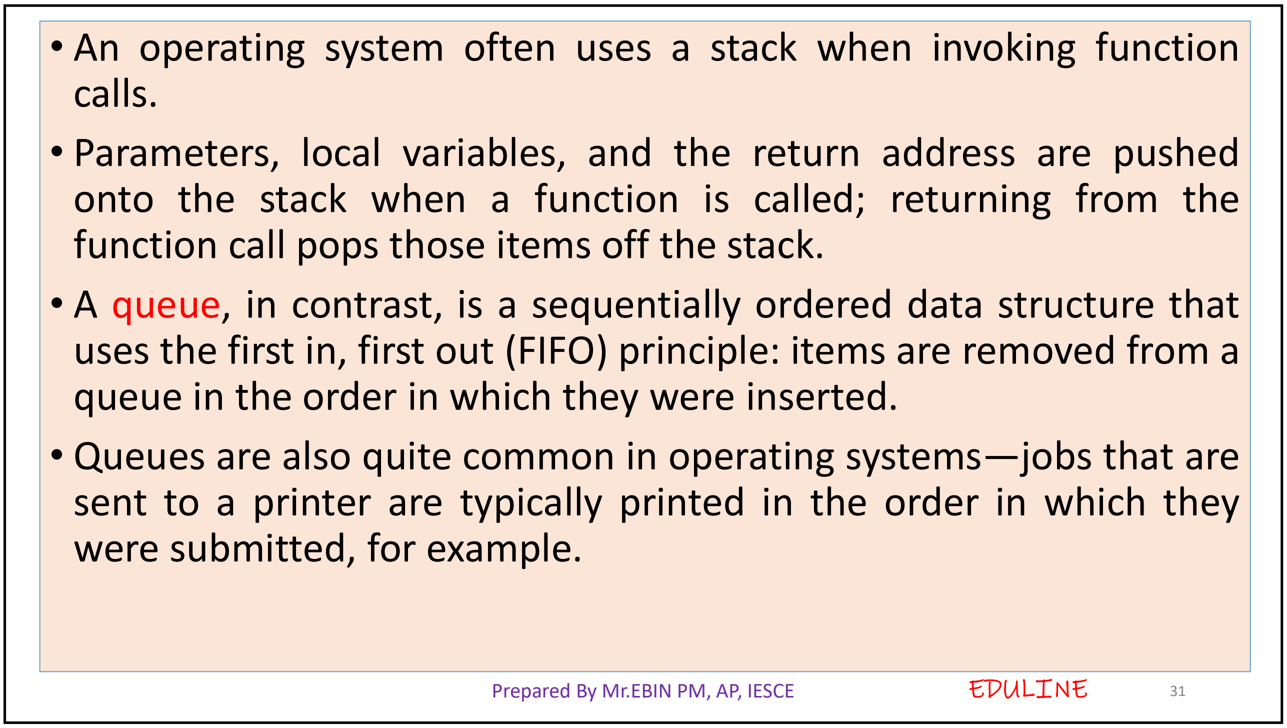
- Several fundamental data structures used extensively in operating systems.
- An array is a simple data structure in which each element can be accessed directly.
- A list represents a collection of data values as a sequence.
- The most common method for implementing this structure is a linked list, in which items are linked to one another.
- Linked lists are of several types:
- In a singly linked list, each item points to its successor, as illustrated in

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 29



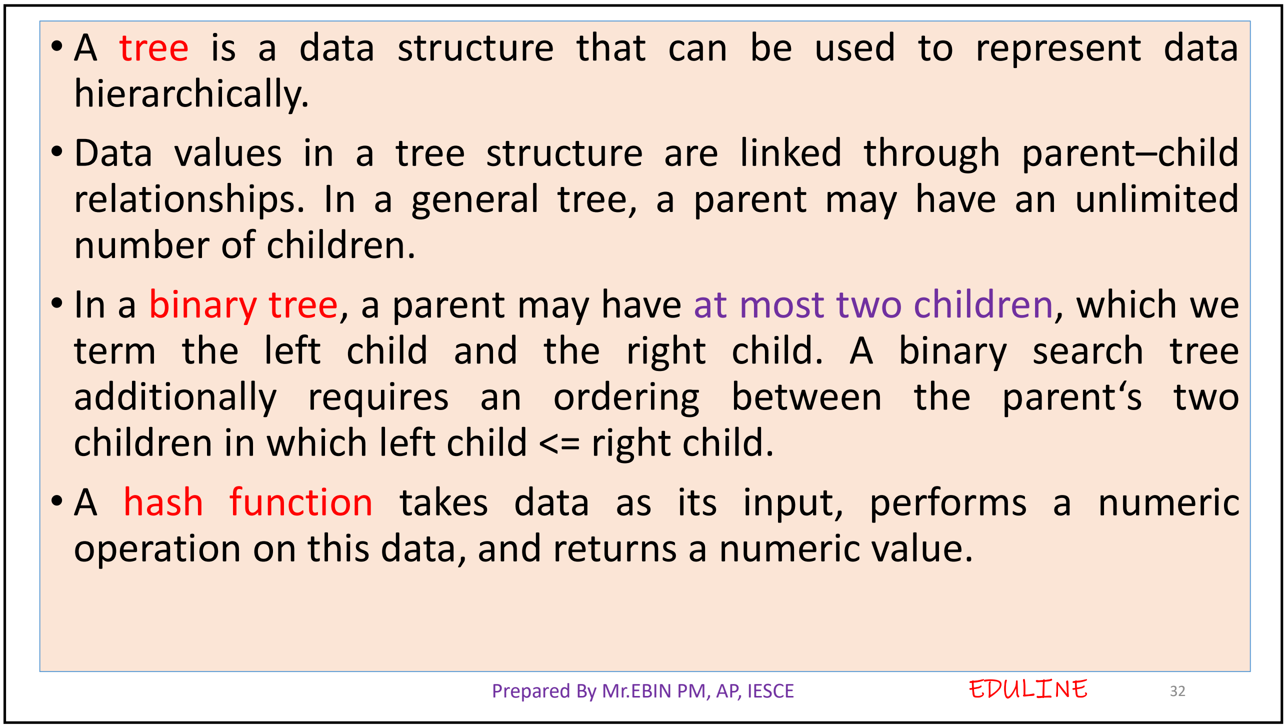
- In a doubly linked list, a given item can refer either to its predecessor or to its successor
- In a circularly linked list, the last element in the list refers to the first element, rather than to null,
- Linked lists accommodate items of varying sizes and allow easy insertion and deletion of items.
- A stack is a sequentially ordered data structure that uses the last in, first out (LIFO) principle for adding and removing items, meaning that the last item placed onto a stack is the first item removed.
- The operations for inserting and removing items from a stack are known as push and pop, respectively.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 30



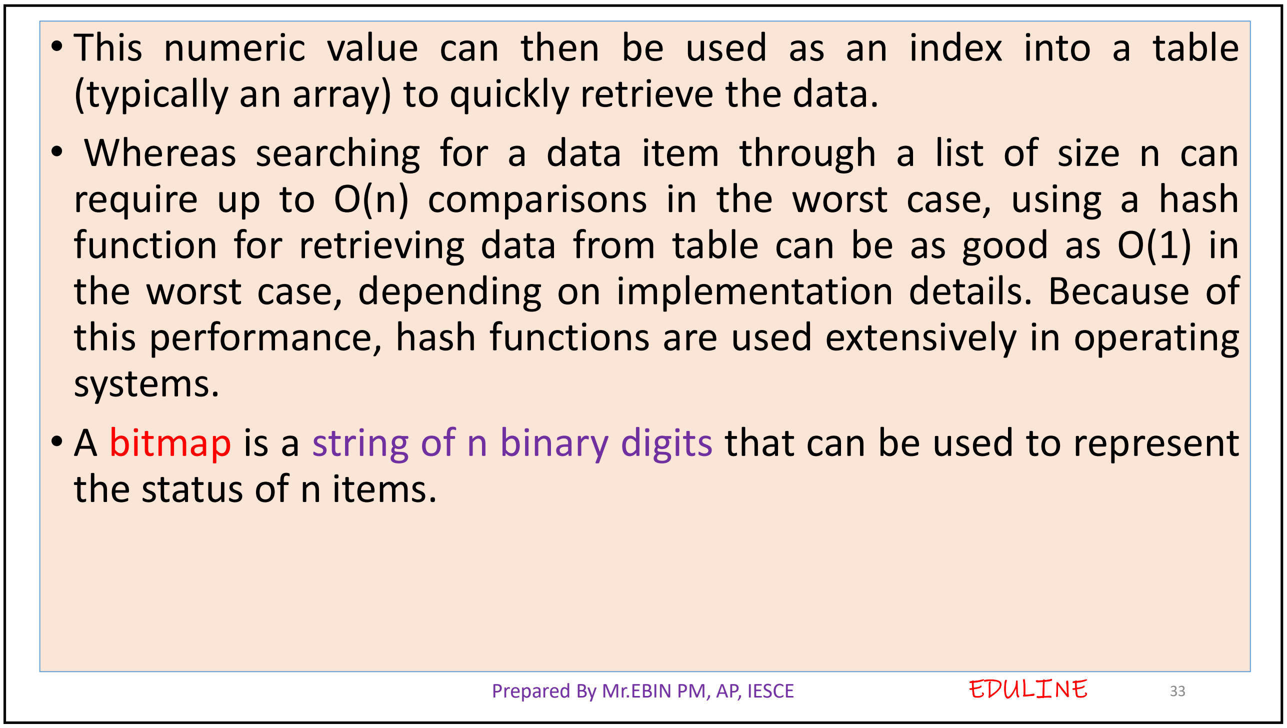
- An operating system often uses a stack when invoking function calls.
- Parameters, local variables, and the return address are pushed onto the stack when a function is called; returning from the function call pops those items off the stack.
- A queue, in contrast, is a sequentially ordered data structure that uses the first in, first out (FIFO) principle: items are removed from a queue in the order in which they were inserted.
- Queues are also quite common in operating systems—jobs that are sent to a printer are typically printed in the order in which they were submitted, for example.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 31

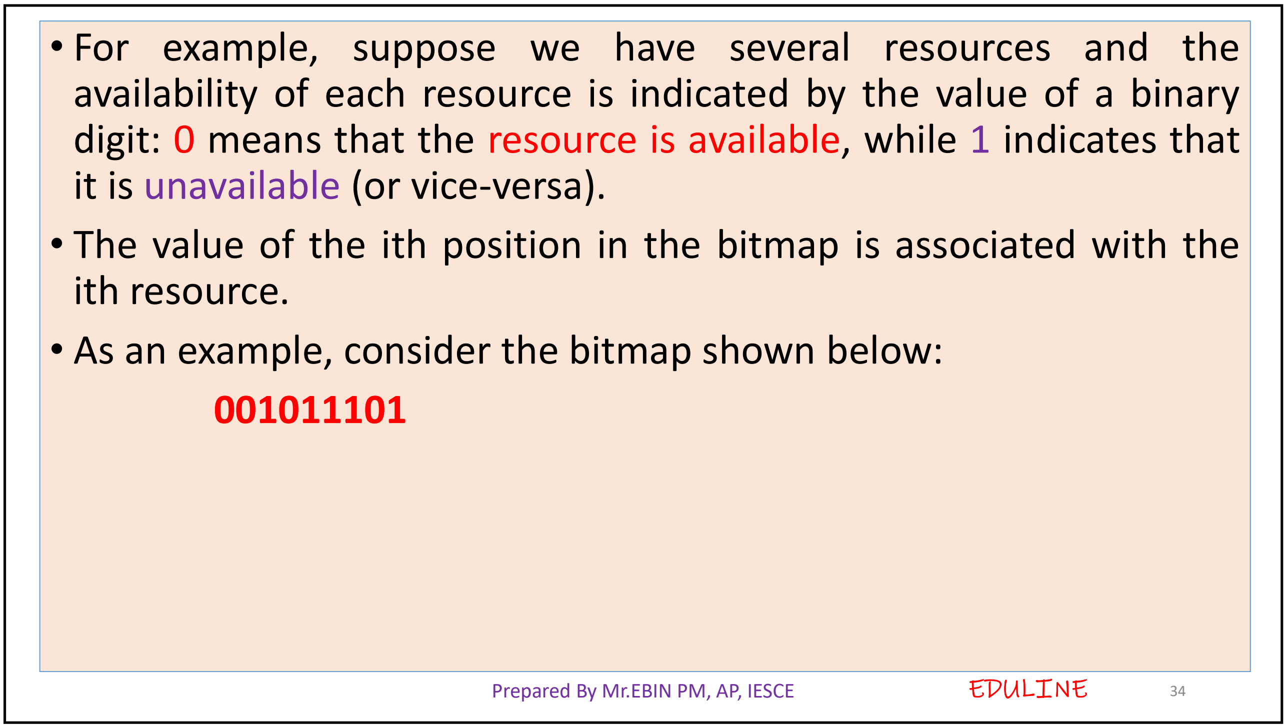


- A tree is a data structure that can be used to represent data hierarchically.
- Data values in a tree structure are linked through parent–child relationships. In a general tree, a parent may have an unlimited number of children.
- In a binary tree, a parent may have at most two children, which we term the left child and the right child. A binary search tree additionally requires an ordering between the parent‘s two children in which left child <= right child.
- A hash function takes data as its input, performs a numeric operation on this data, and returns a numeric value.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 32



- This numeric value can then be used as an index into a table (typically an array) to quickly retrieve the data.
- Whereas searching for a data item through a list of size n can require up to O(n) comparisons in the worst case, using a hash function for retrieving data from table can be as good as O(1) in the worst case, depending on implementation details. Because of this performance, hash functions are used extensively in operating systems.
- A bitmap is a string of n binary digits that can be used to represent the status of n items.



- For example, suppose we have several resources and the availability of each resource is indicated by the value of a binary digit: 0 means that the resource is available, while 1 indicates that it is unavailable (or vice-versa).
- The value of the ith position in the bitmap is associated with the ith resource.
- As an example, consider the bitmap shown below:

001011101