KOSHYS INSTITUTE OF MANAGEMENT STUDIES

DEPARTMENT OF COMPUTER APPLICATIONS

IV SEMESTER BCA

COURSE CODE : DSC12

COURSE TITLE: Operating Systems Concepts

Introduction to Operating Systems - 1

Introduciton to Operating System

Operating System

- OS Concepts
- Functionsof OS
- Applications of OS
- Typesof OS
- Propertiesof OS

Introduction

Operating System:

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

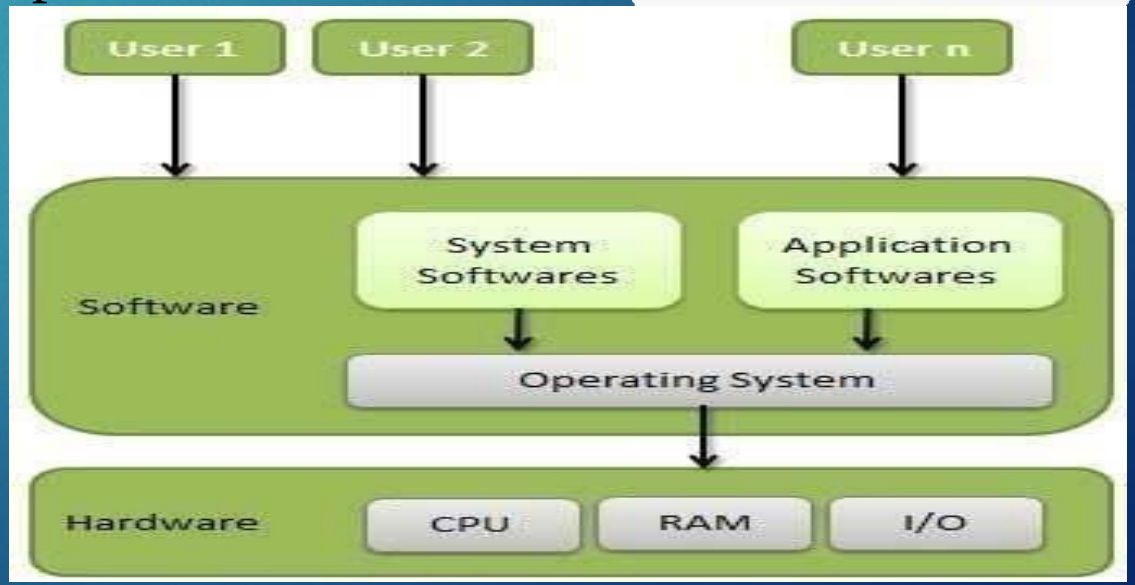
- Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.

Introduction

Operating System:

An Operating System (OS) is an interface between a computer user and computer hardware. .

- An operating system is a software which performs all the basic tasks like:
  - *File management*
  - *Memory management*
  - *Process management*
  - *Handling input and output*
  - *Controlling peripheral devices such as disk drives and printers.*



Introduction

Operating System:

Layered Approach

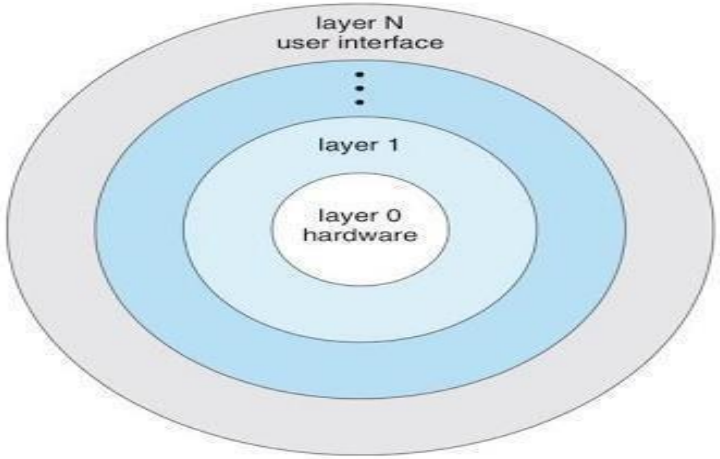
- OS is broken into a no.of.layers to make it modular.
- Each layer uses functions and services

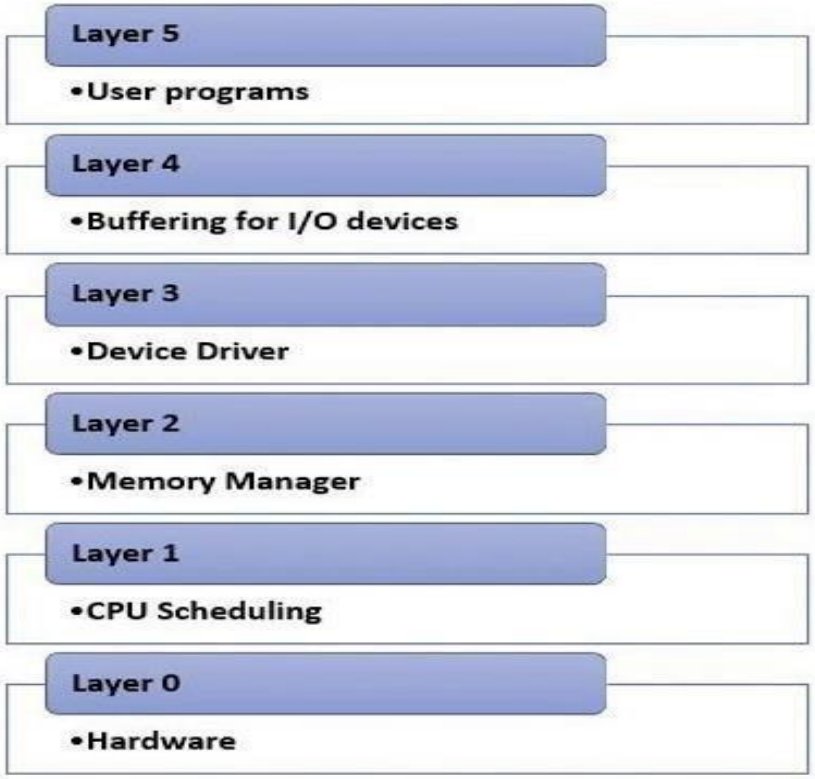
Advantage :

- simplicity of constructio
- debugging.

Disadvantage:

- less efficient-sys call takes long duration





Operating System

Functions of Operating system :

The operating system is a vital component of the system software in a computer system..

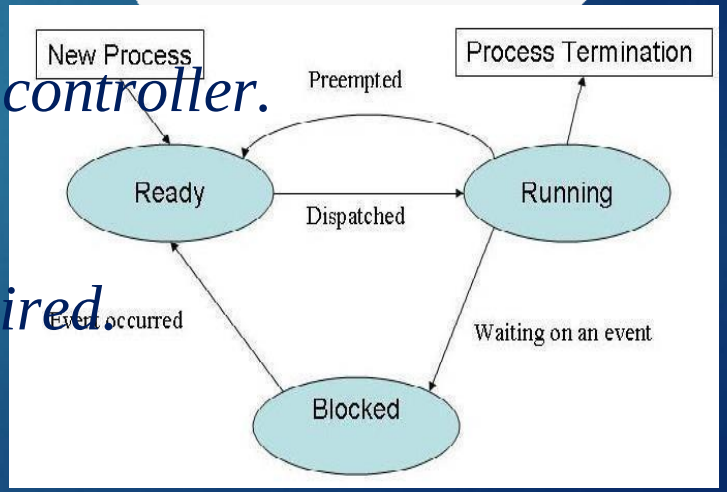
- Following are some of important functions of an operating System.
  - *Processor Management*
  - *Memory Management*
  - *Device Management*
  - *File Management*
  - *Security*
  - *Control over system performance*
  - *Job accounting*
  - *Error detecting aids*
  - *Coordination between other software and users*

Operating System

Processor Management:

In multiprogramming environment, the OS decides which process gets the processor when and for how much time, this function is called process scheduling.

- An Operating System does the following activities for processor management:.
  - *Keeps tracks of processor and status of process.*
  - *The program responsible for this task is known as traffic controller.*
  - *Allocates the processor (CPU) to a process.*
  - *De-allocates processor when a process is no longer required.*



Operating System

Memory Management:

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

- An Operating System does the following activities for memory management:
  - *Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.*
  - *In multiprogramming, the OS decides which process will get memory when and how much.*
  - *Allocates the memory when a process requests it to do so.*

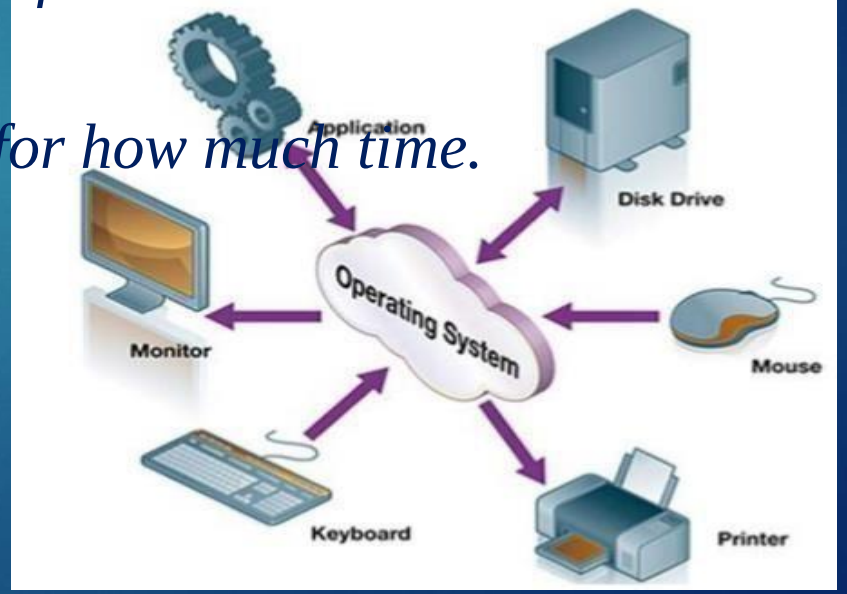
- *De-allocates the memory when a process no longer needs it or has been terminated.*

Operating System

Device Management:

An Operating System manages device communication via their respective drivers.

- An Operating System does the following activities for device management:
  - *Keeps tracks of all devices. Program responsible for this task is known as the I/O controller.*
  - *Decides which process gets the device when and for how much time.*
  - *Allocates the device in the efficient way.*
  - *De-allocates devices.*

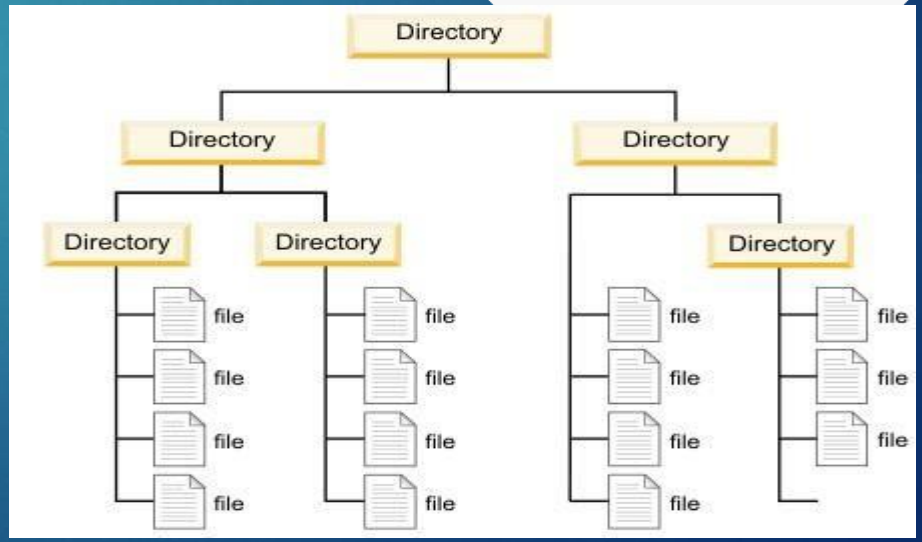


Operating System

File Management:

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

- An Operating System does the following activities for file management:
  - *Keeps track of information, location, uses, status etc. The collective facilities are often known as file system.*
  - *Access to the file for read and write.*
  - *Allocates the file for I/O devices.*
  - *De-allocates the file form storage devices*



Applications of Operating System:

Following are some of the important activities that an Operating System performs.

- **Security:** By means of password and similar other techniques, it prevents unauthorized access to programs and data.
- **Control over system performance:** Recording delays between request for a service and response from the system.
- **Job accounting:** Keeping track of time and resources used by various jobs and users.
- **Error detecting aids:** Production of dumps, traces, error messages, and other debugging and error detecting aids.

- **Coordination between other software and users:** Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

Operating System Types

Types of Operating System:

Operating systems are there from the very first computer generation and they keep evolving with time.

- Various types of operating systems which are most commonly used are:
  - ***Batch operating system***
  - ***Time-sharing operating systems***
  - ***Distributed operating System.***
  - ***Network operating System***
  - ***Real Time operating System***

Operating System Types

Batch operating system:

A batch operating system grabs all programs and data in the batch form and then processes them.

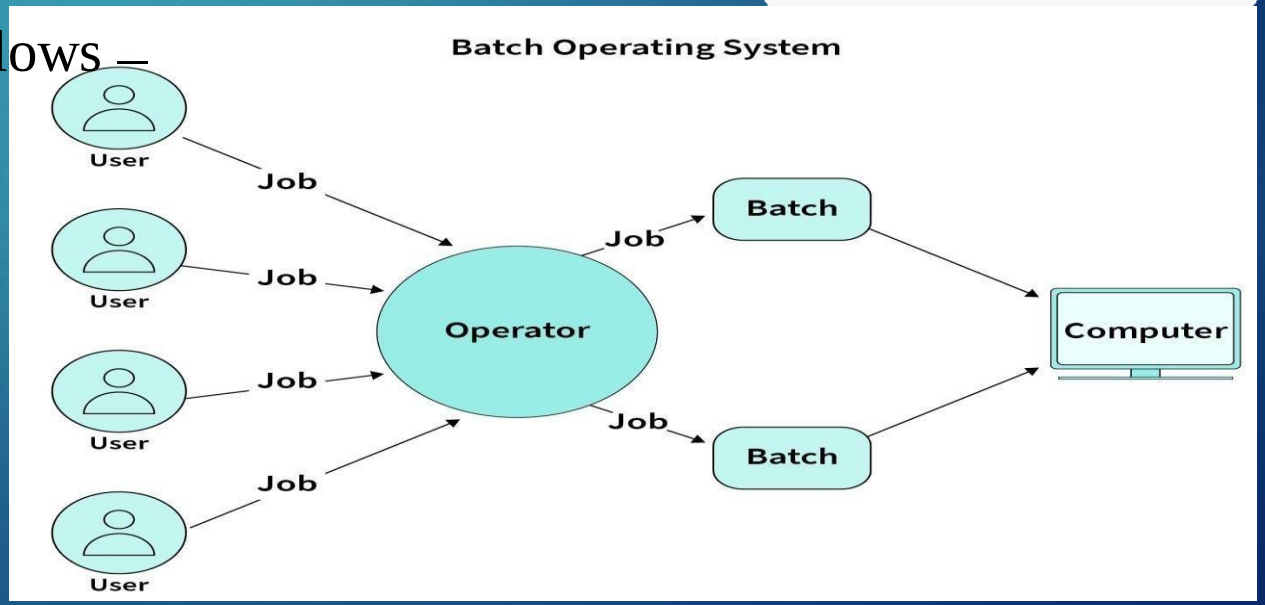
- Each user prepares his job on an off-line device like punch cards and submits it to the computer operator.
- To speed up processing, jobs with similar needs are batched together and run as a group.
- The problems with Batch Systems are as follows –
  - ***Lack of interaction between the user and the job.***
  - ***CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.***
  - ***Difficult to provide the desired priority.***

Operating System Types

Batch operating system:

The users of a batch operating system do not interact with the computer directly.

- The main aim of using a batch processing system is to decrease the setup time while submitting similar jobs to the CPU.
- Batch processing techniques were implemented in the hard disk and card readers as well..
- Examples of Batch Systems are as follows –
  - ***Payroll System***
  - ***Bank Invoice System***
  - ***Transactions Process***
  - ***Daily Report***
  - ***Research Segment***
  - ***Billing System***

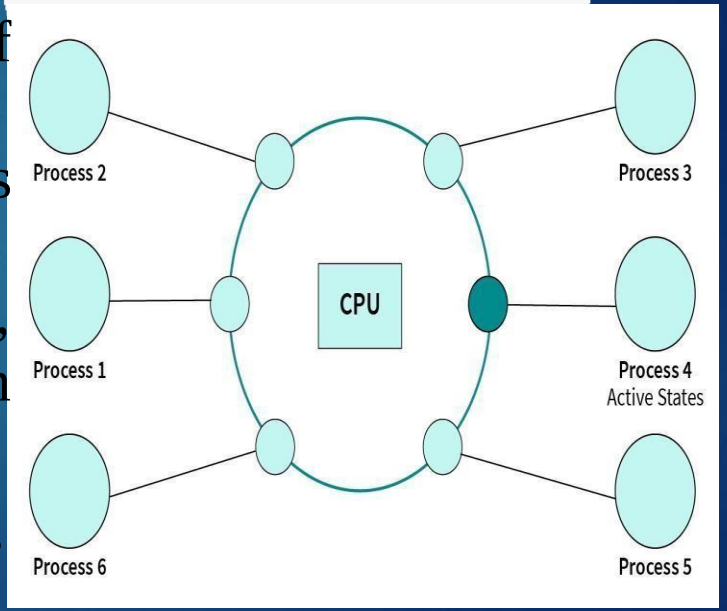


Operating System Types

Time-sharing operating systems:

Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time.

- Time-sharing or multitasking is a logical extension of multiprogramming.
- Processor's time which is shared among multiple users simultaneously is termed as time-sharing.
- Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently enabling, the users to receive an immediate response.
- The operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time.



Operating System Types

Time-sharing operating systems:

Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

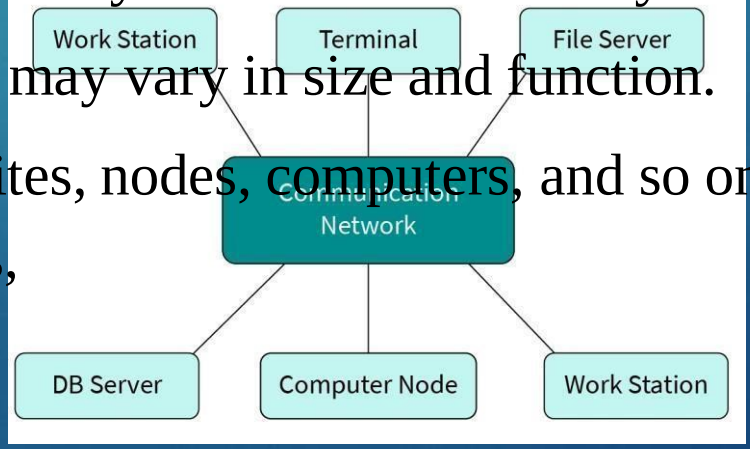
- **Advantages of Timesharing operating systems are as follows:**
  - *Provides the advantage of quick response.*
  - *Avoids duplication of software.*
  - *Reduces CPU idle time.*
- **Disadvantages of Time-sharing operating systems are as follows:**
  - *Problem of reliability.*
  - *Question of security and integrity of user programs and data.*
  - *Problem of data communication.*

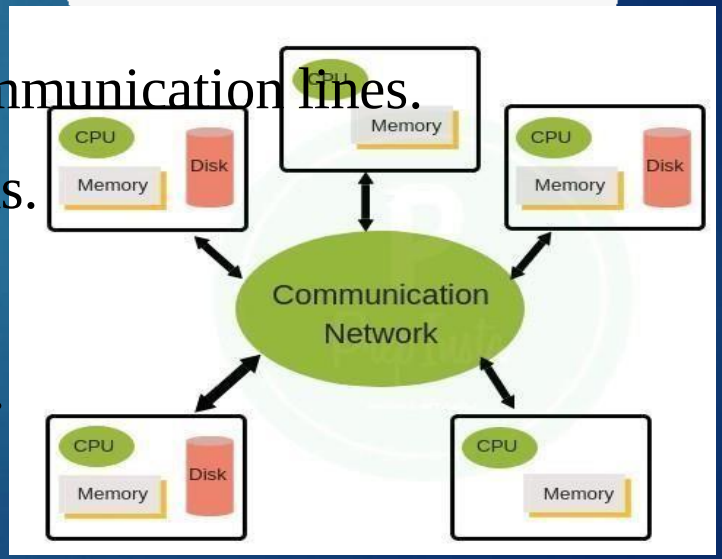
Operating System Types

Distributed operating System:

Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly.

- The processors communicate with one another through various communication lines.
- These are referred as loosely coupled systems or distributed systems.
- Processors in a distributed system may vary in size and function.
- These processors are referred as sites, nodes, computers, and so on.
- Examples: Solaris, OSF/1, Micros, DYNIX, Locus, Mach



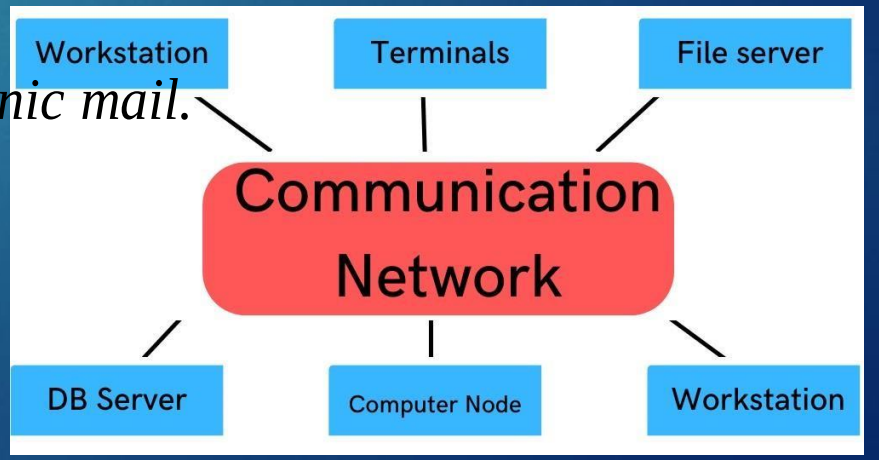


Operating System Types

Distributed operating System:

Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly.

- The advantages of distributed systems are as follows –
  - *With resource sharing facility, a user at one site may be able to use the resources available at another.*
  - *Speedup the exchange of data with one another via electronic mail.*
  - *Better service to the customers.*
  - *Reduction of the load on the host computer.*
  - *Reduction of delays in data processing.*

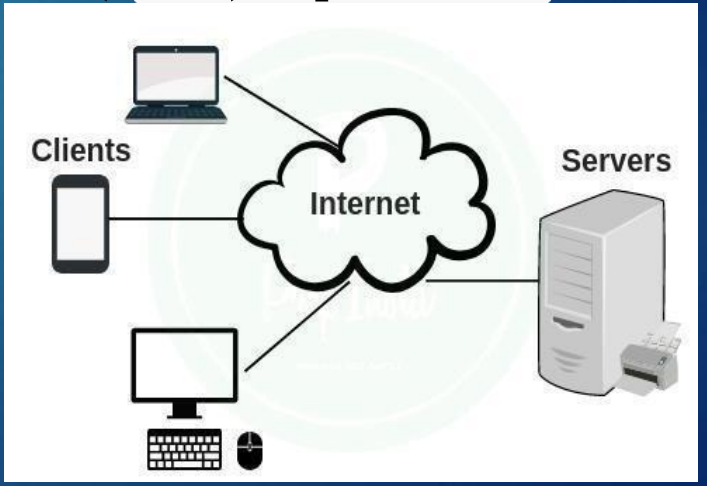


Operating System Types

Network operating System:

A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions.

- The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.
- There are two basic types of network operating systems:
  - *Peer-to-Peer Network Operating Systems*
  - *Client/Server Network Operating Systems* ➢ Examples of network operating systems include:



- *Microsoft Windows Server 2003, Microsoft Windows Server 2008,*
  - *UNIX, Linux, Mac OS X, Novell NetWare, and BSD.*

Operating System Types

Network operating System:

A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions.

- **The advantages of network operating systems are as follows:**
  - *Centralized servers are highly stable.*
  - *Security is server managed.*
  - *Upgrades to new technologies and hardware can be easily integrated into the system.*
  - *Remote access to servers is possible from different locations and types of systems.*

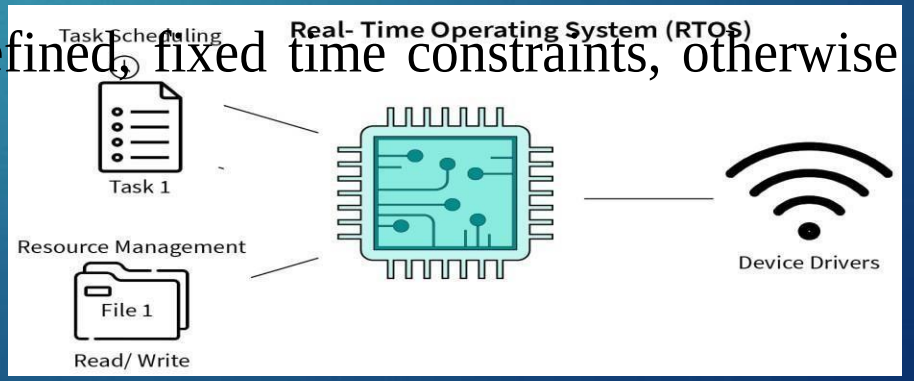
- **The disadvantages of network operating systems are as follows:**
  - *High cost of buying and running a server.*
  - *Dependency on a central location for most operations.*
  - *Regular maintenance and updates are required..*

Real Time operating System:

A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment.

- The time taken by the system to respond to an input and display of required updated information is termed as the response time.
- In this method, the response time is very less as compared to online processing.

- Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application.
- A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail.

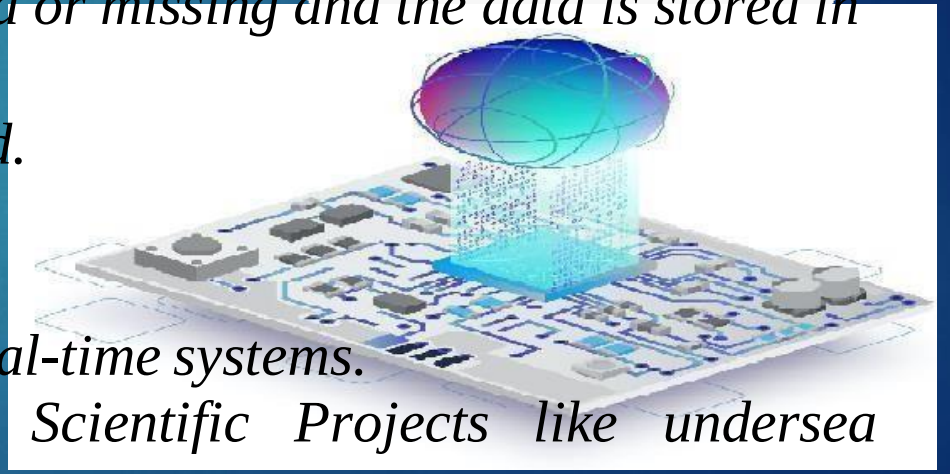


Operating System Types

Real Time operating System:

There are two types of real-time operating systems.

- **Hard real-time systems**
  - *Hard real-time systems guarantee that critical tasks complete on time.*
  - *In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM.*
  - *In these systems, virtual memory is almost never found.*
- **Soft real-time systems**
  - *Soft real-time systems are less restrictive.*
  - *Soft real-time systems have limited utility than hard real-time systems.*
  - ***Example:*** *Multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers, etc.*



Operating System- Services

Operating System - Services:

An Operating System provides services to both the users and to the programs.

- It provides programs an environment to execute.
- It provides users the services to execute the programs in a convenient manner.
- Common services provided by an operating System are:
  - *Program execution*
  - *I/O operations*
  - *File System manipulation*
  - *Communication*
  - *Error Detection*
  - *Resource Allocation*
  - *Protection*

Operating System- Services

Program execution:

Operating systems handle many kinds of activities from user programs to system programs, these activities are encapsulated as a process.

- A process includes the complete execution context (code to execute, data to manipulate, registers, OS resources in use).
- Following are the activities of an operating system with respect to program management:
  - *Loads a program into memory.*
  - *Executes the program.*
  - *Handles program's execution.*
  - *Provides a mechanism for process synchronization.*
  - *Provides a mechanism for process communication.*
  - *Provides a mechanism for deadlock handling.*

Operating System - Services

File system manipulation:

A file represents a collection of related information. Computers can store files on the disk (secondary storage), for long-term storage purpose.

- Afile system is normally organized into directories for easy navigation and usage.
- These directories may contain files and other directions.
- Following are the major activities of an operating system with respect to file management:
  - *Program needs to read a file or write a file.*
  - *The operating system gives the permission to the program for operation on file.*
  - *Permission varies from read-only, read-write, denied and so on.*
  - *Operating System provides an interface to the user to create/delete files.*

- *Operating System provides an interface to the user to create/delete directories.*
  Operating System provides an interface to create the backup of file system.

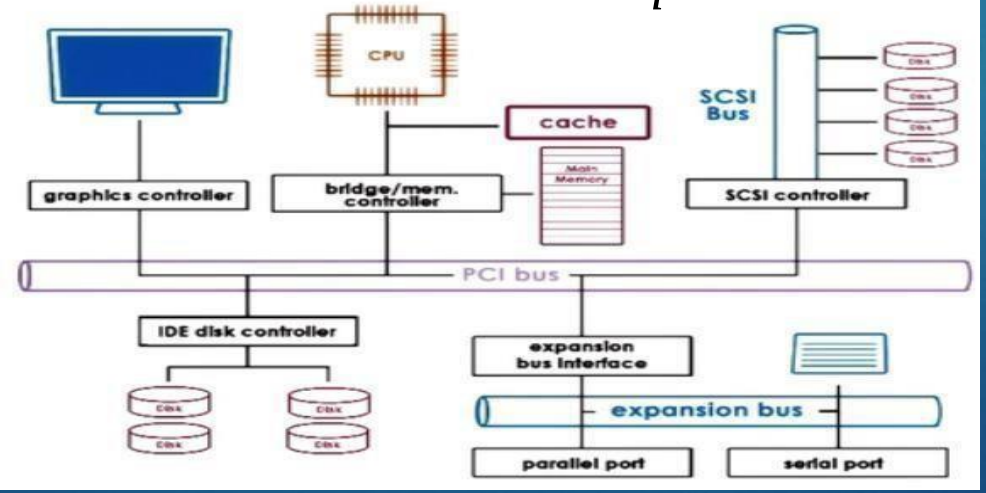
-

Operating System- Services

I/O Operation:

An I/O subsystem comprises of I/O devices and their corresponding driver software. Drivers hide the peculiarities of specific hardware devices from the users.

- An Operating System manages the communication between user and device drivers:
  - *I/O operation means read or write operation with any file or any specific I/O device.*
  - *Operating system provides the access to the required I/O device when required.*



Operating System - Services

Communication:

In case of distributed systems which are a collection of processors that do not share memory, peripheral devices, or a clock, the operating system manages communications between all the processes.

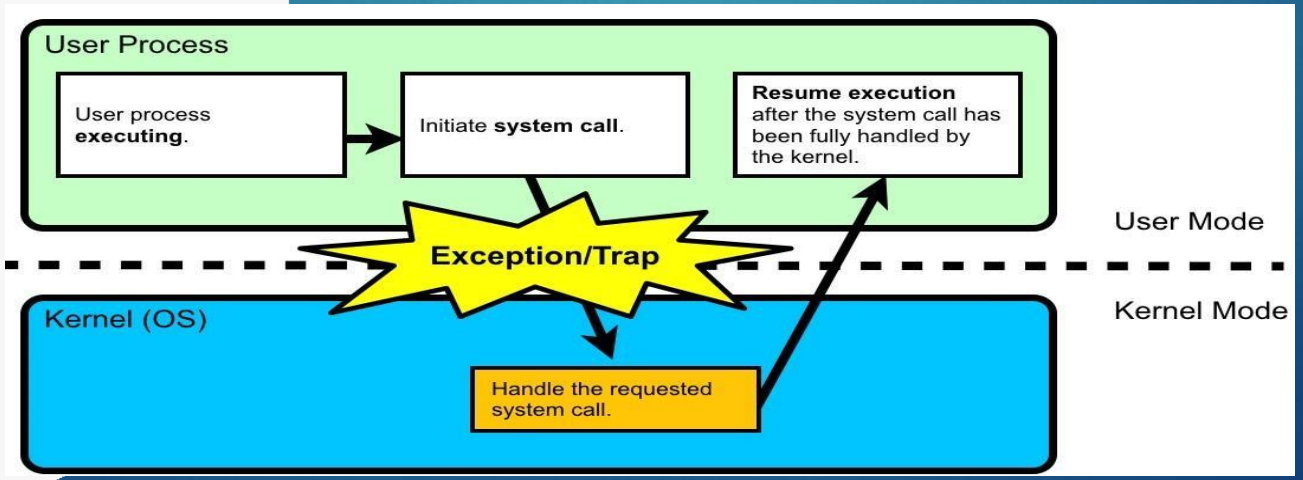
- The OS handles routing and connection strategies, and the problems of contention and security.
- Following are the major activities of an operating system with respect to communication:
  - *Two processes often require data to be transferred between them*
  - *Both the processes can be on one computer or on different computers, but are connected through a computer network.*
  - *Communication may be implemented by two methods, either by Shared Memory or by Message Passing..*

Operating System- Services

Error handling:

Errors can occur anytime and anywhere. An error may occur in CPU, in I/O devices or in the memory hardware.

- Following are the major activities of an operating system with respect to error handling:
  - *The OS constantly checks for possible errors.*
  - *The OS takes an appropriate action to ensure correct and consistent computing.*



Operating System- Services

Resource Management:

In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job.

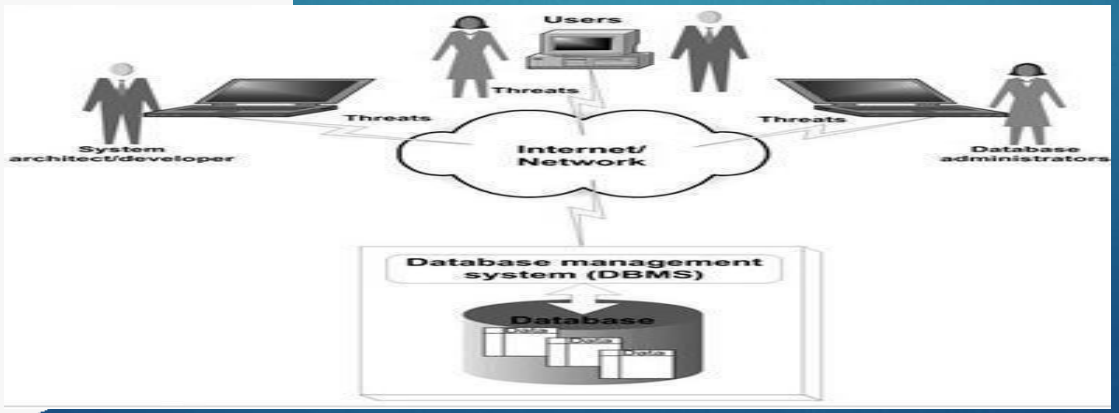
- Following are the major activities of an operating system with respect to resource management:
  - *The OS manages all kinds of resources using schedulers.*
  - *CPU scheduling algorithms are used for better utilization of CPU.*

Operating System- Services

Protection:

Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system.

- Following are the major activities of an operating system with respect to protection:
  - *The OS ensures that all access to system resources is controlled.*
  - *The OS ensures that external I/O devices are protected from invalid access attempts.*
  - *The OS provides authentication features for each user by means of passwords.*

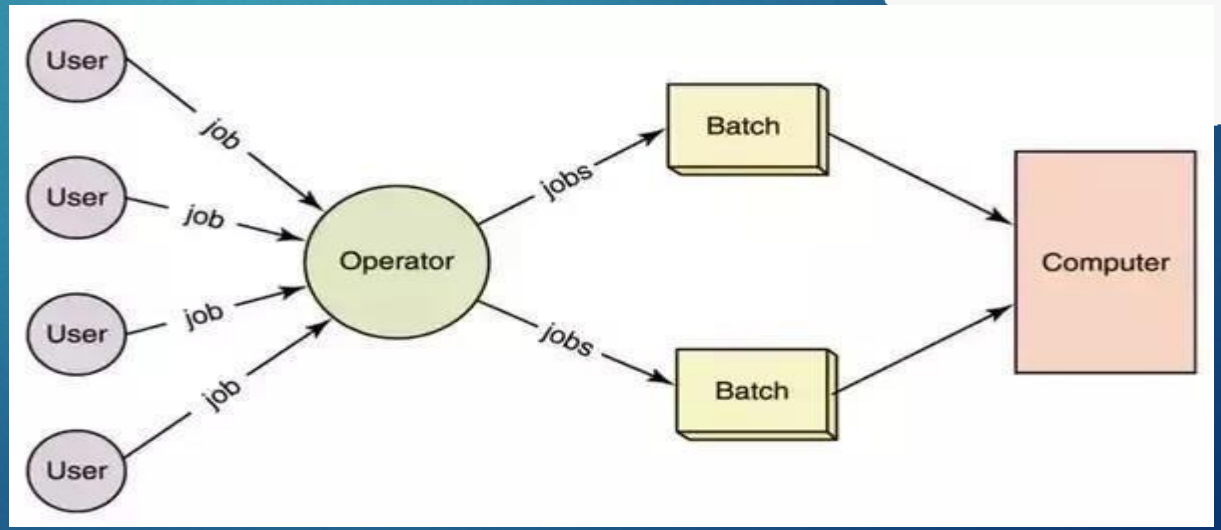


Operating System- Properties

Properties of Operating System:

Operating System refers to a mechanism to access the programs, processes, or users to the resources defined by a computer system.

- Following are the properties of Operating System:
  - *Batch processing.*
  - *Multitasking*
  - *Multiprogramming*
  - *Interactivity*
  - *Real Time System*
  - *Distributed Environment*
  - *Spooling*

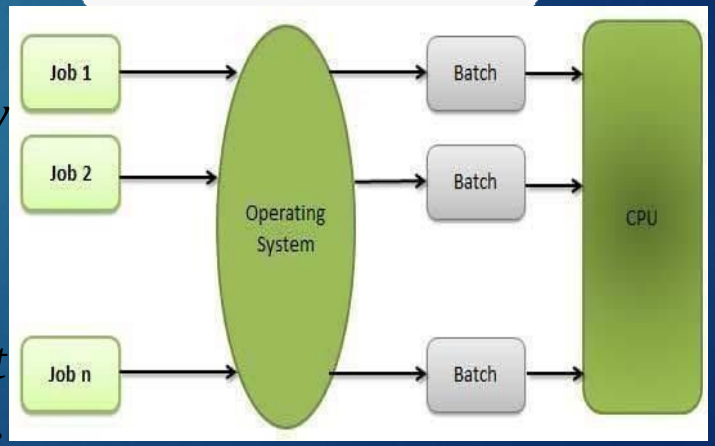


Operating System- Properties

Batch processing:

Batch processing is a technique in which an Operating System collects the programs and data together in a batch before processing starts.

- An operating system does the following activities related to batch processing:
  - *The OS defines a job which has predefined sequence of commands, programs and data as a single unit.*
  - *The OS keeps a number a jobs in memory and executes them without any manual information.*
  - *Jobs are processed in the order of submission, i.e., first come first served fashion.*
  - *When a job completes its execution, its memory is released and the output for the job gets copied into an output spool for later printing or processing.*



Operating System- Properties

Batch processing:

Batch processing is a technique in which an Operating System collects the programs and data together in a batch before processing starts.

- **Advantages**
  - Batch processing takes much of the work of the operator to the computer.
  - Increased performance as a new job get started as soon as the previous job is finished, without any manual intervention.

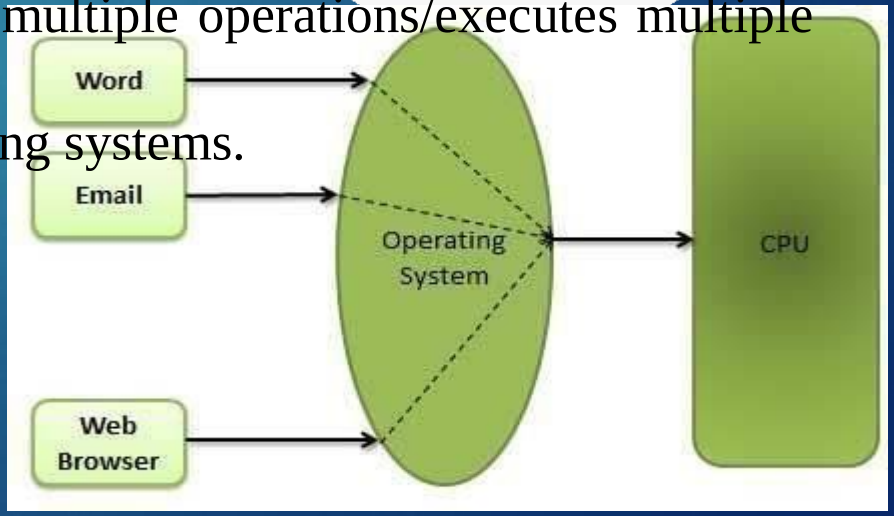
- **Disadvantages**
  - Difficult to debug program.
  - A job could enter an infinite loop.
  - Due to lack of protection scheme, one batch job can affect pending jobs.:

Operating System- Properties

Multitasking:

Multitasking is when multiple jobs are executed by the CPU simultaneously by switching between them.

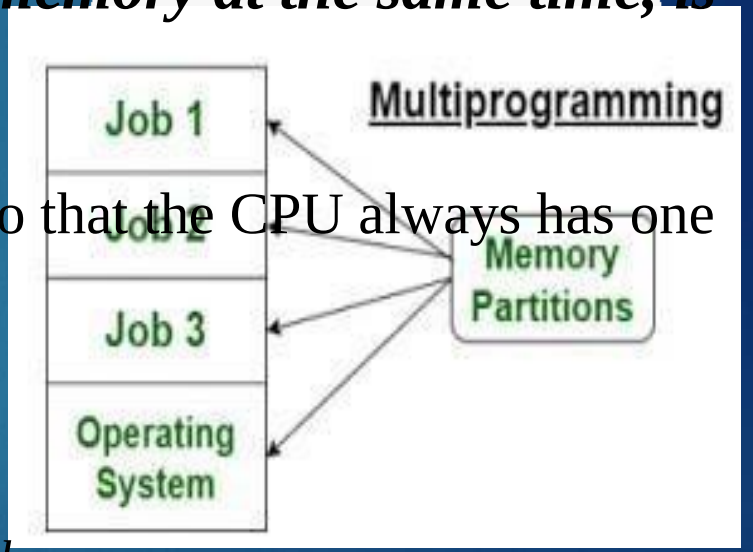
- An Operating System does the following activities related to multitasking:
  - The user gives instructions to the operating system or to a program directly, and receives an immediate response.
  - The OS handles multitasking in the way that it can handle multiple operations/executes multiple programs at a time.
  - Multitasking Operating Systems are also known as Time-sharing systems.
  - Each user has at least one separate program in memory.



35 Operating System - Properties

Multiprogramming:

Sharing the processor, when two or more programs reside in memory at the same time, is referred as multiprogramming.



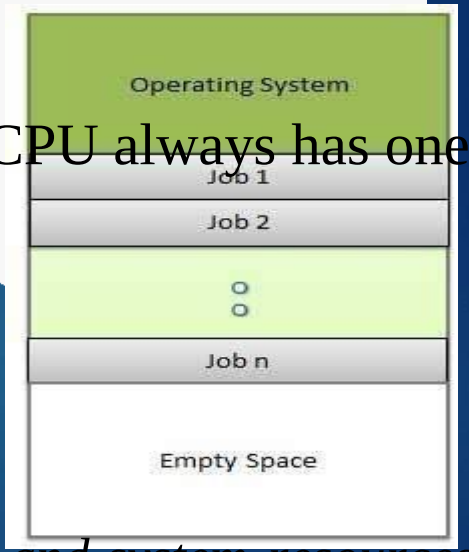
- Multiprogramming assumes a single shared processor.
- Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.
- An OS does the following activities related to multiprogramming:
  - *The operating system keeps several jobs in memory at a time.*
  - *This 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.*
  - *Multiprogramming operating systems monitor the state of all active programs and system resources using memory management programs to ensures that the CPU is never idle, unless there are no jobs to process*

35 Operating System - Properties

Multiprogramming:

Sharing the processor, when two or more programs reside in memory at the same time, is referred as multiprogramming.

- Multiprogramming assumes a single shared processor.
- Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.
- An OS does the following activities related to multiprogramming:
  - *The operating system keeps several jobs in memory at a time.*
  - *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.*
  - *Multiprogramming operating systems monitor the state of all active programs and system resources using memory management programs to ensures that the CPU is never idle, unless there are no jobs to process*



35 Operating System - Properties

Operating System - Properties

Real Time System & Interactivity:

Real-time systems are usually dedicated, embedded systems.

- An operating system does the following activities related to real-time system activity..
  - *In such systems, Operating Systems typically read from and react to sensor data.*
  - *The Operating system must guarantee response to events within fixed periods of time to ensure correct performance.*

Interactivity

- Interactivity refers to the ability of users to interact with a computer system.
- An Operating system does the following activities related to interactivity − ❖ *Provides the user an interface to interact with the system.*
  - *Manages input devices to take inputs from the user. For example, keyboard.*

- *Manages output devices to show outputs to the user. For example, Monitor.* The response time of the OS needs to be short, since the user submits and waits for the result.

Operating System- Properties

Distributed Environment:

A distributed environment refers to multiple independent CPUs or processors in a computer system.

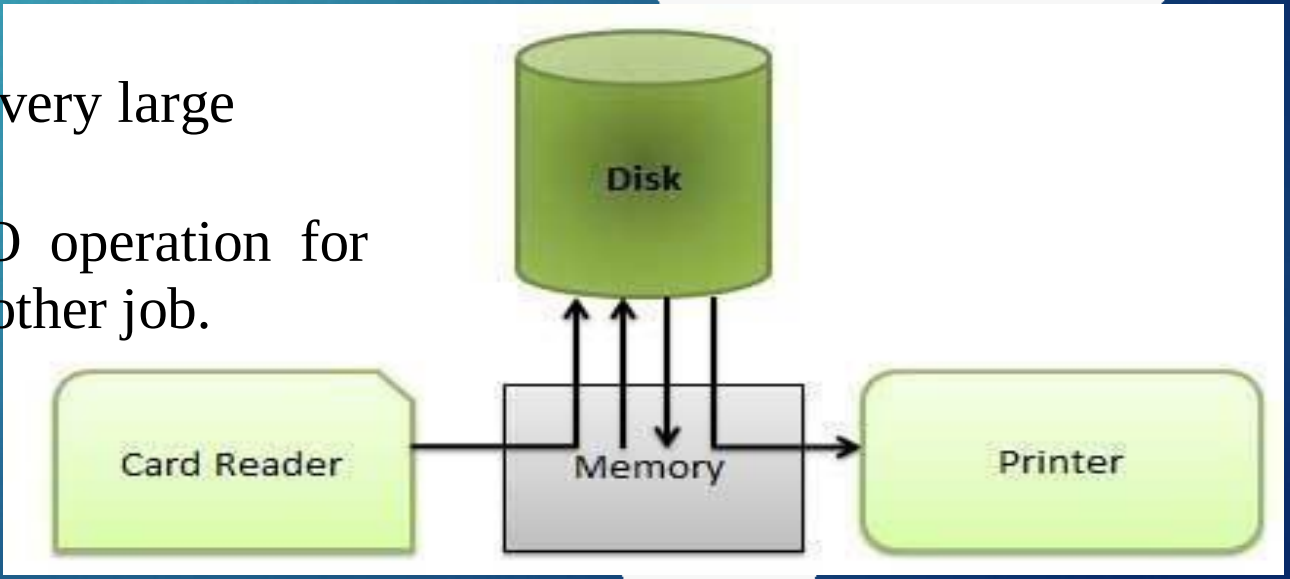
- An operating system does the following activities related to distributed environment:
  - *The OS distributes computation logics among several physical processors.*
  - *The processors do not share memory or a clock. Instead, each processor has its own local memory.*
  - *The OS manages the communications between the processors.*
  - *They communicate with each other through various communication lines.*

Operating System- Properties

Spooling:

Spooling is an acronym for simultaneous peripheral operations on line.

- Spooling refers to putting data of various I/O jobs in a buffer.
- This buffer is a special area in memory or hard disk which is accessible to I/O devices.
- **Advantages**
  - The spooling operation uses a disk as a very large buffer.
  - Spooling is capable of overlapping I/O operation for one job with processor operations for another job.



40 Operating System - Properties

Operating System - Properties

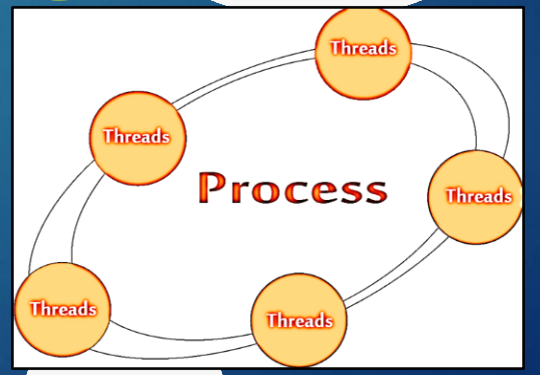
Spooling:

Spooling is an acronym for simultaneous peripheral operations on line.

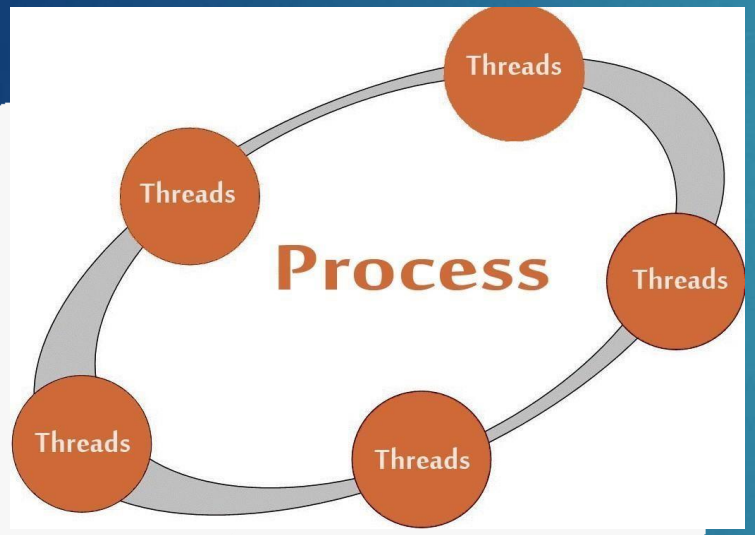
- Spooling refers to putting data of various I/O jobs in a buffer.
- An operating system does the following activities related to distributed environment .
  - *Handles I/O device data spooling as devices have different data access rates.*
  - *Maintains the spooling buffer which provides a waiting station where data can rest while the slower device catches up.*
  - *Maintains parallel computation because of spooling process as a computer can perform I/O in parallel fashion.*
  - *It becomes possible to have the computer read data from a tape, write data to disk and to write out to a tape printer while it is doing its computing task*

Introduction to Operating Systems -2

Processmanagement



Process management



Processmanagement

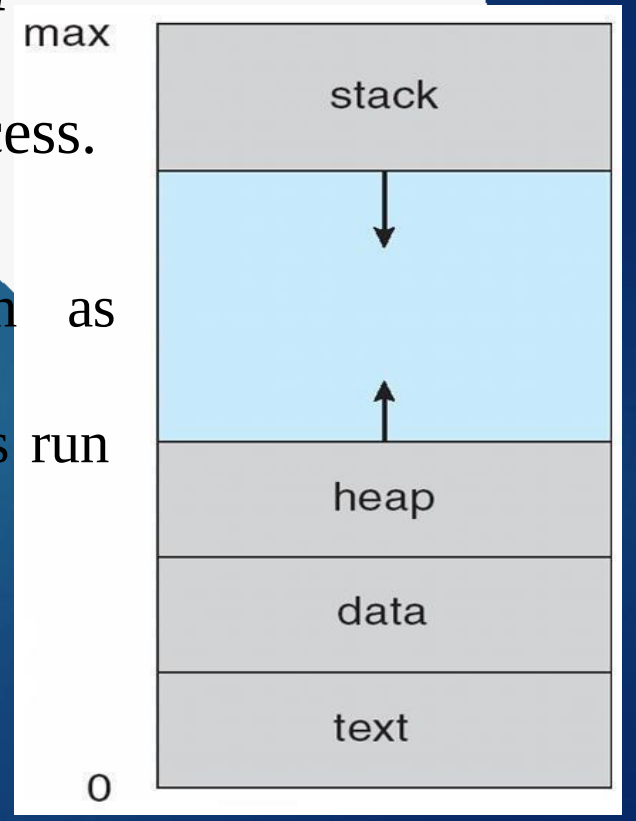
- ~~ProcessConcept~~
- ~~ProcessScheduling~~
- ~~Inter process Communication~~
- ~~Threads~~
- ~~CPU Scheduling Algorithm~~

Introduction

Introduction to Process:

A process is basically a program in execution. The execution of a process must progress in a sequential fashion.

- When a program is loaded into the main memory it becomes a process.
- It can be divided into four sections —
  - **Stack:** The process Stack contains the temporary data such as method/function parameters, return address and local variables
  - **Heap:** This is dynamically allocated memory to a process during its run time
  - **Text:** This includes the current activity represented by the value of Program Counter and the contents of the processor's registers.
  - **Data:** This section contains the global and static variables.

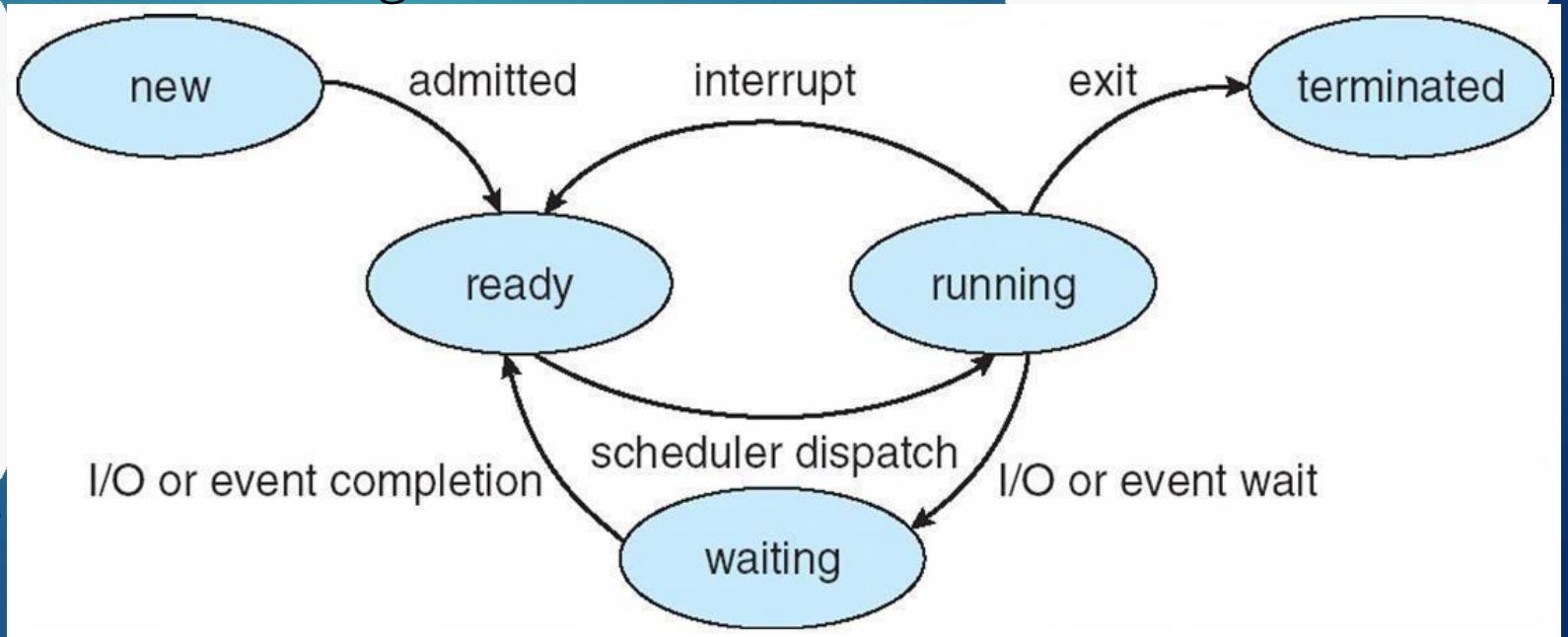


Introduction

Process Life Cycle:

When a process executes, it passes through different states. These stages may differ in different operating systems, and the names of these states are also not standardized.

- A process can have one of the following five states at a time.
  - **Start**
  - **Ready**
  - **Running**
  - **Waiting**
  - **Terminated or Exit**



Process Life Cycle:

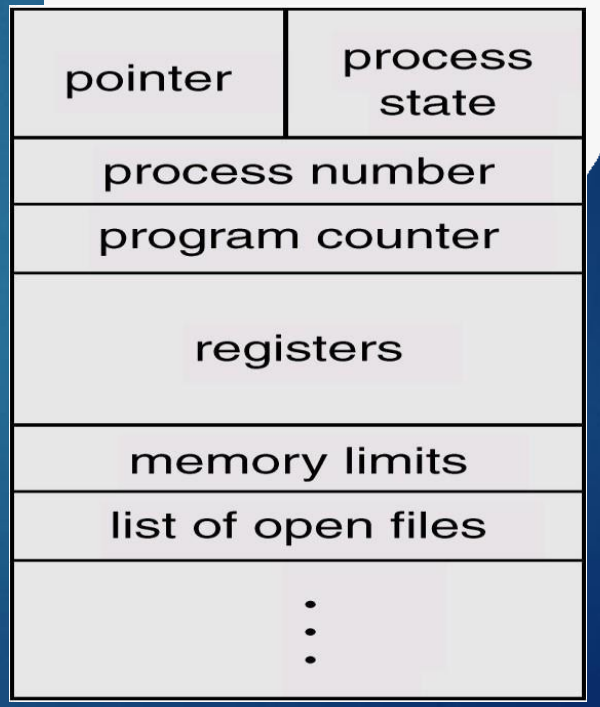
- **Start:** *This is the initial state when a process is first started/created.*
- **Ready:** *The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the OS so that they can run.*
- **Running:** *Once the process has been assigned to a processor by the OS scheduler, the process state is set to running and the processor executes its instructions.*
- **Waiting:** *Process moves into the waiting state if it needs to wait for a resource, such as waiting for user input, or waiting for a file to become available.*
- **Terminated or Exit:** *Once the process finishes its execution, or it is terminated by the operating system, it is moved to the terminated state where it waits to be removed from main memory.*

ProcessControl Block

Process Control Block (PCB):

A Process Control Block is a data structure maintained by the Operating System for every process.

- The PCB is identified by an integer process ID (PID).
- A PCB keeps all the information needed to keep track of a process.
- The PCB is maintained for a process throughout its lifetime, and is deleted once the process terminates.

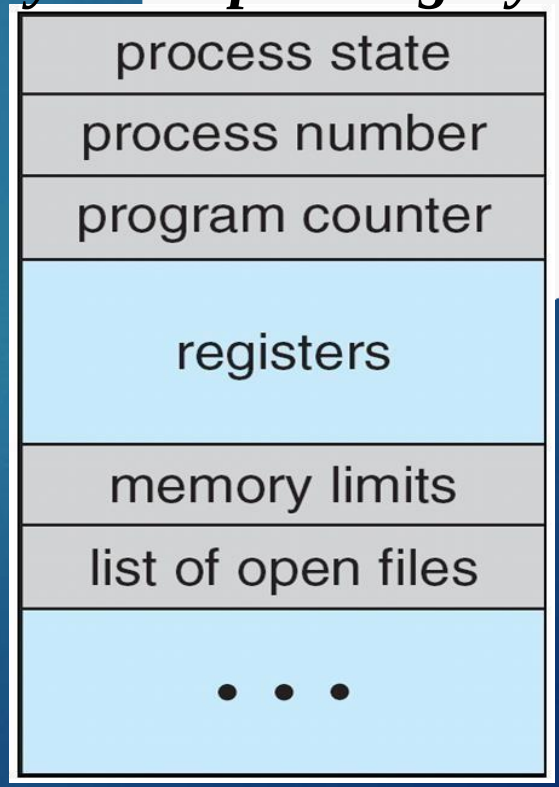


ProcessControl Block

Process Control Block (PCB):

A Process Control Block is a data structure maintained by the Operating System for every process.

- Information associated with each process
  - *Process state*
  - *Program counter*
  - *CPU registers*
  - *CPU scheduling information*
  - *Memory-management information*
  - *Accounting information*
  - *I/O status information*



ProcessControl Block

CPU Switch From Process to Process:

Layered Approach

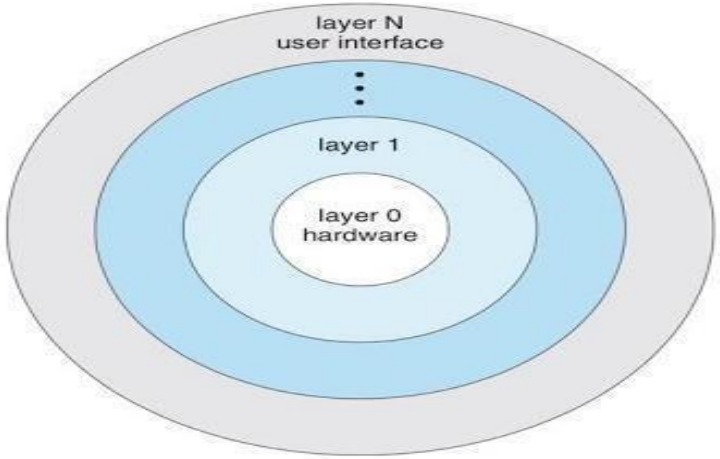
- OS is broken into a no.of.layers to make it modular.
- Each layer uses functions and services

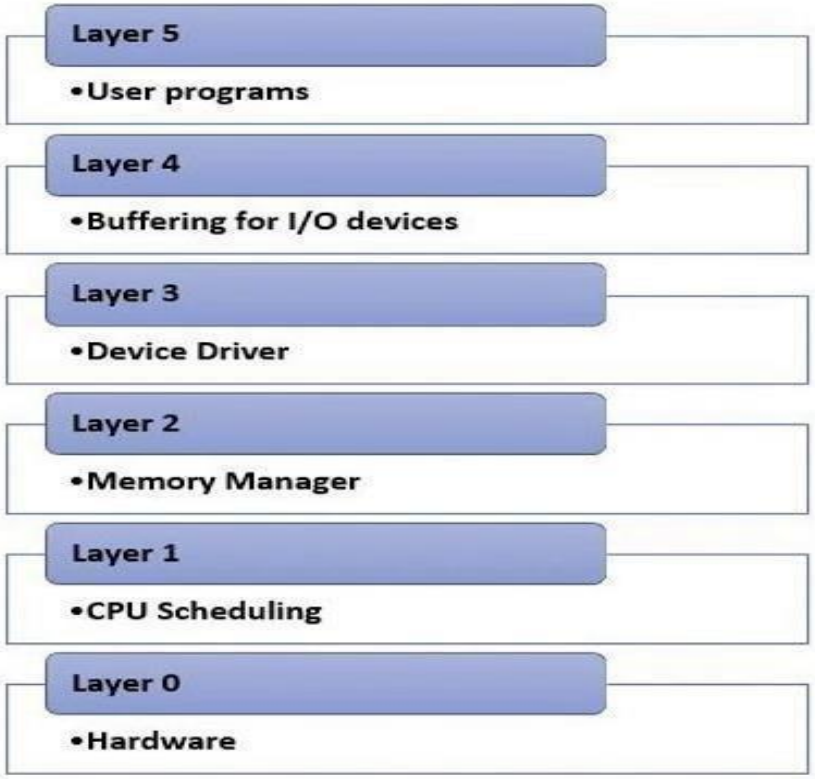
Advantage :

- simplicity of constructio
- debugging.

Disadvantage:

- less efficient-sys call takes long duration





ProcessCommunication

Context Switch:

When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process via a context switch.

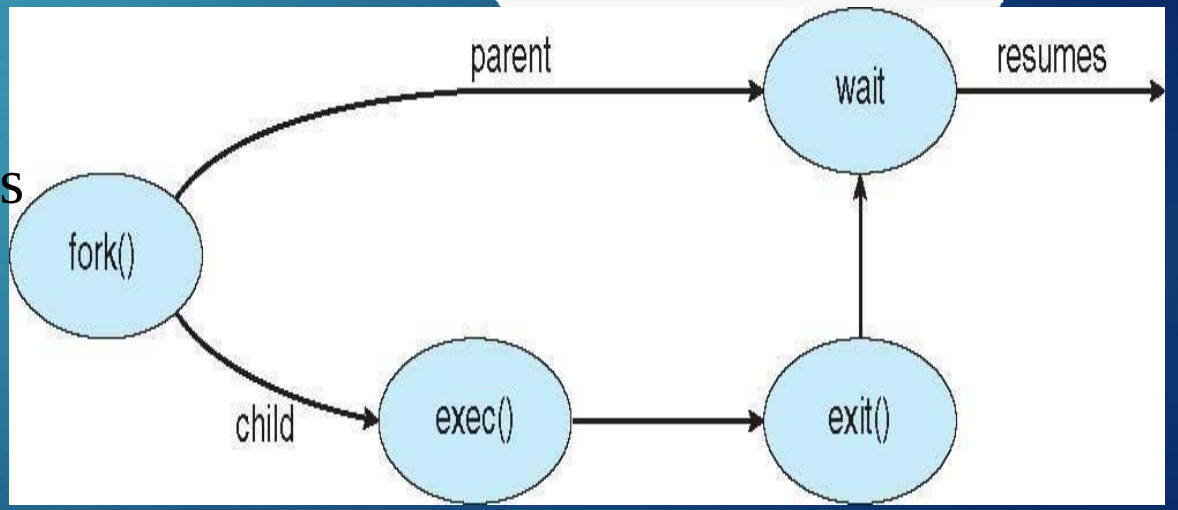
- Context of a process represented in the PCB
- Context-switch time is overhead; the system does no useful work while switching
  - The more complex the OS and the PCB -> longer the context switch
- Time dependent on hardware support
  - Some hardware provides multiple sets of registers per CPU -> multiple contexts loaded at once

ProcessCommunication

Process Creation:

Parent process create children processes, which, in turn create other processes, forming a tree of processes.

- Generally, process identified and managed via a process identifier (pid)
- Resource sharing
  - Parent and children share all resources
  - Children share subset of parent's resources
  - Parent and child share no resources
- Execution
  - Parent and children execute concurrently
  - Parent waits until children terminate



ProcessCommunication

Process Termination:

Process executes last statement and asks the operating system to delete it (exit).

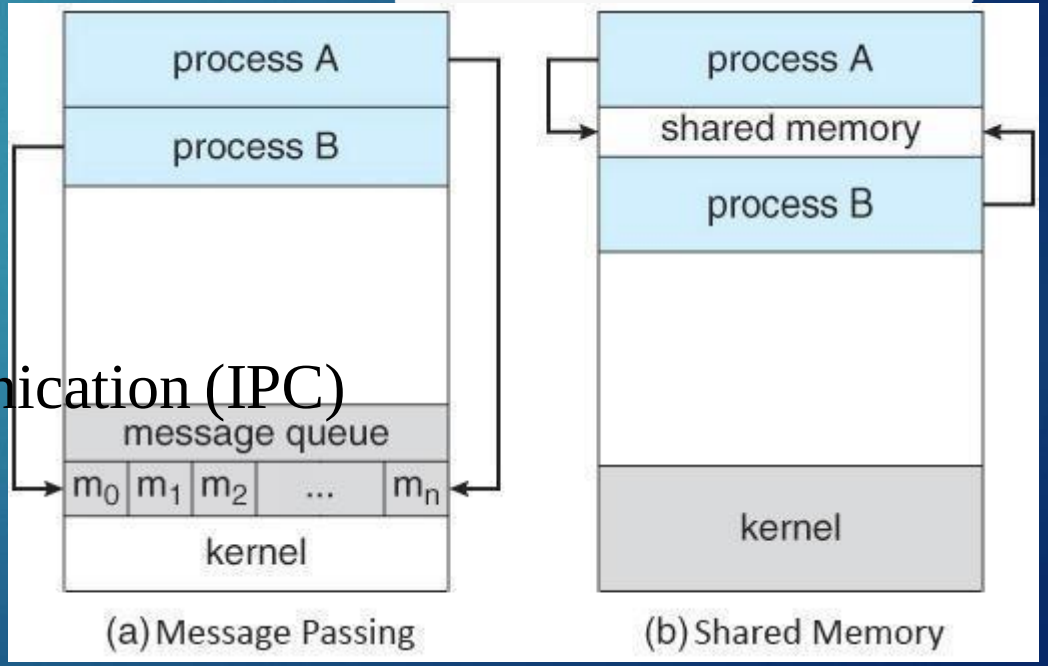
- Output data from child to parent (via wait)
- Process's resources are deallocated by operating system
- Parent may terminate execution of children processes (abort)
  - Child has exceeded allocated resources
  - Task assigned to child is no longer required
  - If parent is exiting
    - Some operating systems do not allow child to continue if its parent terminates
    - All children terminated - cascading termination

ProcessCommunication

Inter process Communication (IPC):

Processes within a system may be independent or cooperating, Cooperating process can affect or be affected by other processes, including sharing data.

- Reasons for cooperating processes:
  - *Information sharing*
  - *Computation speedup*
  - *Modularity*
  - *Convenience*
- Cooperating processes need interprocess communication (IPC)
- Two models of IPC
  - *Shared memory*
  - *Message passing*

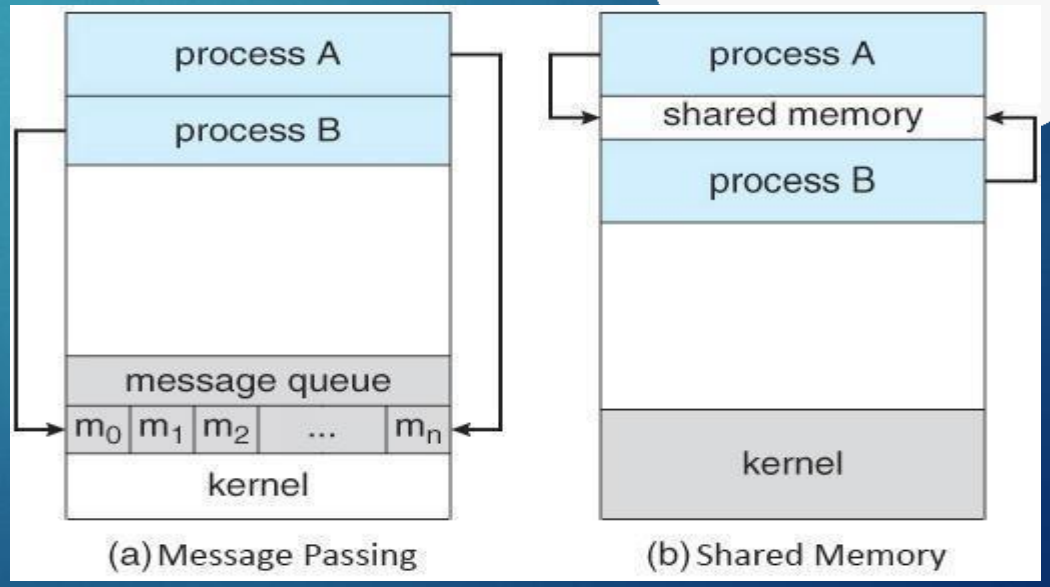


ProcessCommunication

Inter process Communication (IPC):

Inter process communication is a mechanism which allows processes to communicate with each other and synchronize their actions.

- The communication between these processes can be seen as a method of co-operation between them.
- Some of the methods to provide IPC:
  - *Message Queue.*
  - *Shared Memory.*
  - *Signal.*
  - *Shared Files and Pipe*
  - *Socket*



Threading

Thread:

Athread is a single sequential flow of execution of tasks of a process so it is also known as thread of execution or thread of control.

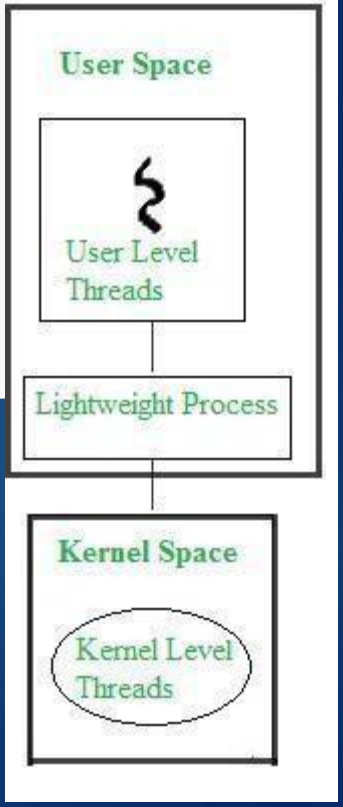
- Each thread of the same process makes use of a separate program counter and a stack of activation records and control blocks.
- Thread is often referred to as a lightweight process.
- Need of Thread:–
  - It takes far less time to create a new thread in an existing process than to create a new process.
  - Threads can share the common data, they do not need to use Inter- Process communication.
  - Context switching is faster when working with threads.
  - It takes less time to terminate a thread than a process.

Threading

Thread:

Threads are implemented in following two ways :

- **User Level Threads** – User managed threads.
  - The operating system does not recognize the user-level thread.
  - User threads can be easily implemented and it is implemented by the user.
  - If a user performs a user-level thread blocking operation, the whole process is blocked.
- **Kernel Level Threads** – The kernel thread recognizes the operating system.
  - There is a thread control block and process control block in the system for each thread and process in the kernel-level thread.
  - The kernel-level thread is implemented by the operating system.
  - The kernel knows about all the threads and manages them.

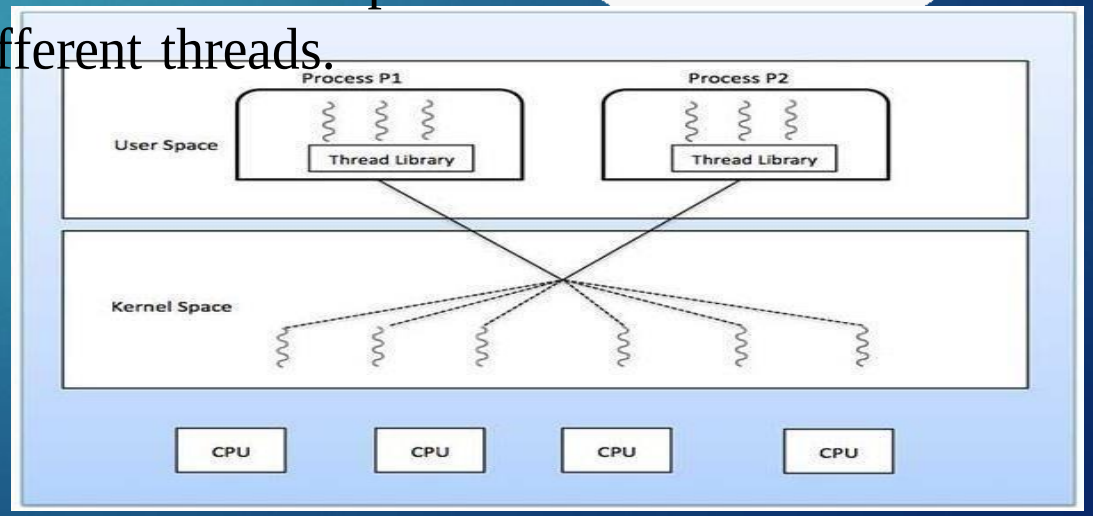


Threading

Multithreading Models:

Some operating system provide a combined user level thread and Kernel level thread facility.

- In a combined system, multiple threads within the same application can run in parallel on multiple processors and a blocking system call need not block the entire process.
- The idea is to achieve parallelism by dividing a process into multiple threads.
- For example, in a browser, multiple tabs can be different threads.
- Multithreading models are three types
  - Many to many relationship
  - Many to one relationship
  - One to one relationship



Threading

Thread:

Difference between User-Level & Kernel-Level Thread.

User-Level Threads	Kernel-Level Thread
User-level threads are faster to create and manage.	Kernel-level threads are slower to create and manage.
Implementation is by a thread library at the user level.	Operating system supports creation of Kernel threads.
User-level thread is generic and can run on any operating system.	Kernel-level thread is specific to the operating system.
Multi-threaded applications cannot take advantage of multiprocessing.	Kernel routines themselves can be multithreaded.

Threading

Advantages of Thread:

The various advantages of using Thread are:

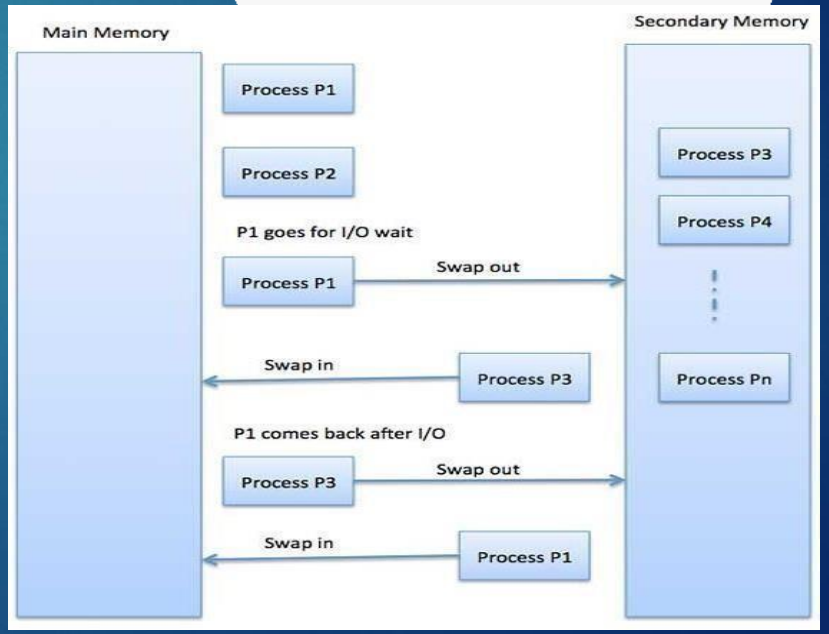
- Threads minimize the context switching time.
- Use of threads provides concurrency within a process.
- Efficient communication.
- It is more economical to create and context switch threads.
- Threads allow utilization of multiprocessor architectures to a greater scale and efficiency..

Swapping

Swapping:

Swapping is a mechanism in which a process can be swapped temporarily out of main memory to secondary storage and make that memory available to other processes.

- Swapping is also known as a technique for memory compaction.
- The total time taken by swapping process includes the time it takes to move the entire process to a secondary disk and then to copy the process back to memory



ProcessScheduling

Process Scheduling:

Maximize CPU use, quickly switch processes onto CPU for time sharing.

- Process scheduler selects among available processes for next execution on CPU
- Maintains scheduling queues of processes
  - *Job queue – set of all processes in the system*
  - *Ready queue – set of all processes residing in main memory, ready and waiting to execute*
  - *Device queues – set of processes waiting for an I/O device*
  - *Processes migrate among the various queues*

Process Scheduling

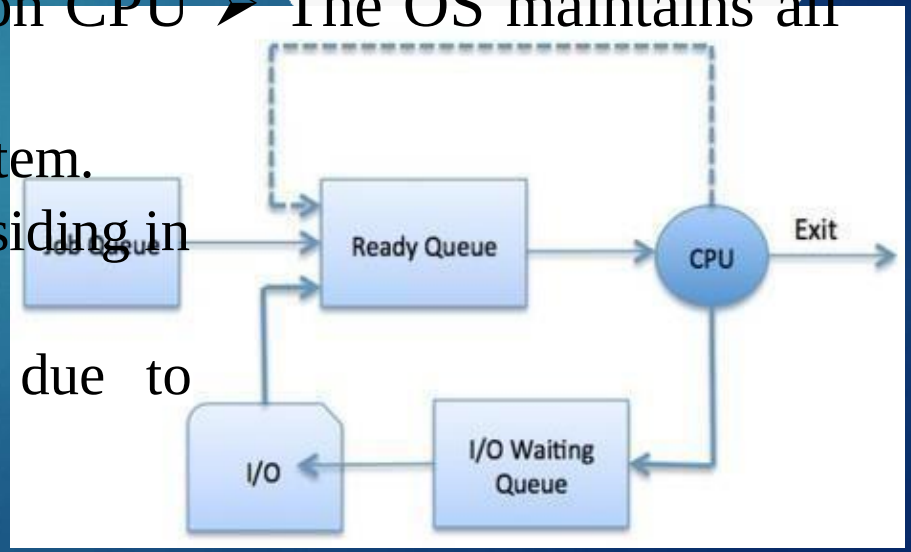
21 Process Scheduling

Process Scheduling:

The process scheduling is the activity of the process manager that handles the removal of the running process from the CPU on the basis of a particular strategy.

➢ Process scheduling is an essential part of a Multiprogramming operating systems used to select among available processes for next execution on CPU ➢ The OS maintains all PCBs in Process Scheduling Queues.

- **Job queue:** This queue keeps all the processes in the system.
- **Ready queue:** This queue keeps a set of all processes residing in main memory, ready and waiting to execute.
- **Device queues:** The processes which are blocked due to unavailability of an I/O device constitute this queue

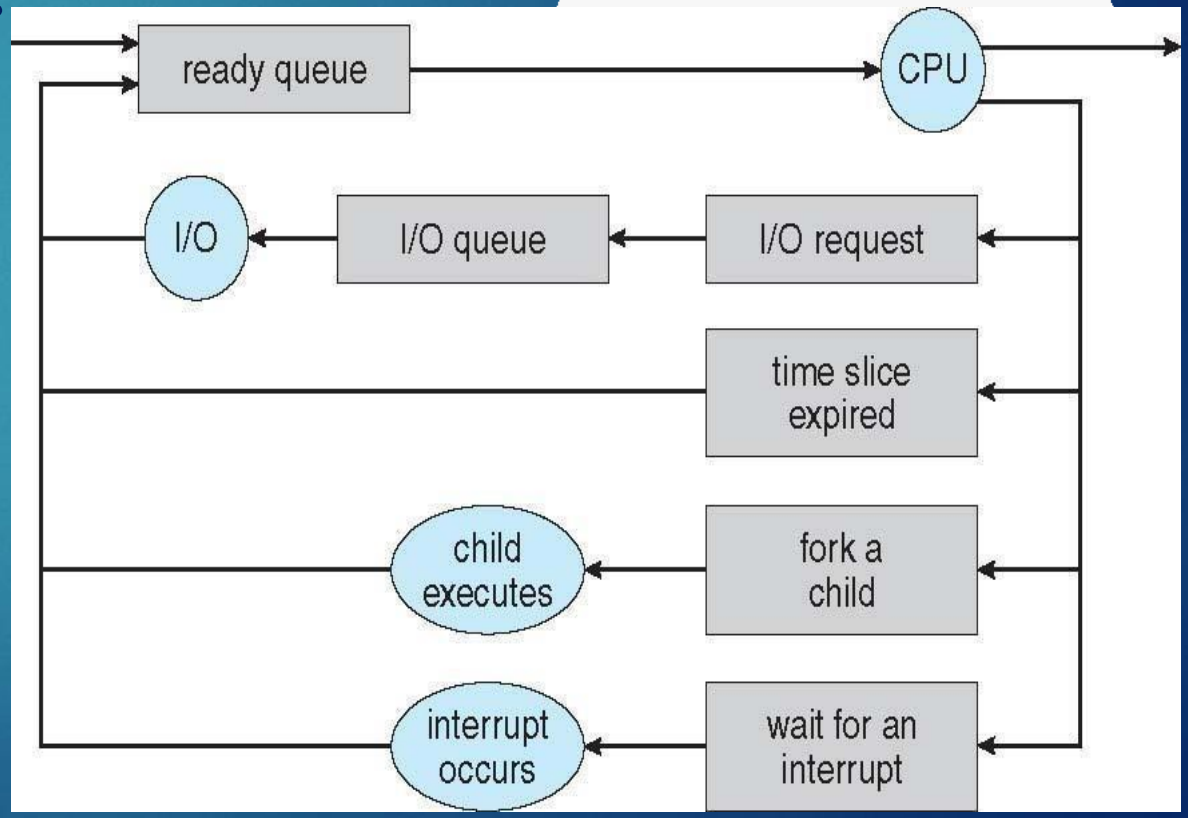


ProcessScheduling

Representation of Process Scheduling:

Processes can be described as either:

- **I/O-bound process** – spends more time doing I/O than computations, many short CPU bursts
- **CPU-bound process** – spends more time doing computations; few very long CPU bursts



ProcessScheduling

Schedulers:

Schedulers are special system software which handle process scheduling in various ways.

- The main task is to select the jobs to be submitted into the system and to decide which process to run.
- Schedulers are of three types:
  - **Long-Term Scheduler:**
    - It is also called a job scheduler.
    - A long-term scheduler determines which programs are admitted to the system for processing.
  - **Short-Term Scheduler:**
    - It is also called as CPU scheduler.
    - Its main objective is to increase system performance in accordance with the chosen set of criteria.
  - **Medium-Term Scheduler:**
    - Medium-term scheduling is a part of swapping.
    - It removes the processes from the memory. It reduces the degree of multiprogramming.

ProcessScheduling

Context Switch:

When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process via a context switch.

- Context of a process represented in the PCB
- Context-switch time is overhead; the system does no useful work while switching
  - The more complex the OS and the PCB -> longer the context switch

- Time dependent on hardware support
  - Some hardware provides multiple sets of registers per CPU -> multiple contexts loaded at once

25 Process Scheduling

Process Scheduling

Scheduling Criteria:

Schedulers selects the processes in memory that are ready to execute, and allocates the CPU based on certain scheduling Criterias.

- Scheduling Criteria are based on:
  - ***CPU utilization*** *– keep the CPU as busy as possible*
  - ***Throughput*** *– No. of processes that complete their execution per time unit*
  - ***Turnaround time*** *– amount of time to execute a particular process*
  - ***Waiting time*** *– amount of time a process has been waiting in the ready queue*
  - ***Response time*** *– amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment)*

ProcessScheduling

Scheduling algorithms:

A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms.

- These algorithms are either non-preemptive or preemptive
- There are popular process scheduling algorithms:
  - ***First-Come, First-Served (FCFS) Scheduling***
  - ***Shortest-Job-Next (SJN) Scheduling***
  - ***Priority Scheduling***
  - ***Round Robin(RR) Scheduling***
  - ***Multiple-Level Queues Scheduling.***

Scheduling Algorithms

First Come First Serve (FCFS):

In First Come First Serve (FCFS) scheduling, Jobs are executed on first come, first serve basis.

- It is a non-preemptive, pre-emptive scheduling algorithm.
- Easy to understand and implement.
- Its implementation is based on FIFO queue.
- Poor in performance as average wait time is high..

Scheduling Algorithms

First Come First Serve (FCFS):

In First Come First Serve (FCFS) scheduling, Jobs are executed on first come, first serve basis.

Example: FCFS Scheduling

Process	Duration	Order	Arrival Time
P1	24	1	0
P2	3	2	0
P3	4	3	0

The final schedule (Gantt chart):

P1 waiting time: 0
P2 waiting time: 24
P3 waiting time: 27

The average waiting time:
(0+24+27)/3 = 17

Scheduling Algorithms

Shortest Job Next (SJN):

This is also known as shortest job first, associate with each process the length of its next CPU burst. Use these lengths to schedule the process with the shortest time.

- This is a non-preemptive, pre-emptive scheduling algorithm.
- Best approach to minimize waiting time.
- Easy to implement in Batch systems where required CPU time is known in advance.
- Impossible to implement in interactive systems where required CPU time is not known.
- The processer should know in advance how much time process will take..

Scheduling Algorithms

Shortest Job Next (SJN):

This is also known as shortest job first, associate with each process the length of its next CPU burst. Use these lengths to schedule the process with the shortest time.

- Example:

Process	Duration	Order	Arrival Time
P1	6	1	0
P2	8	2	0
P3	7	3	0
P4	3	4	0

Do it yourself

P4 waiting time: 0
P1 waiting time: 3
P3 waiting time: 9
P2 waiting time: 16

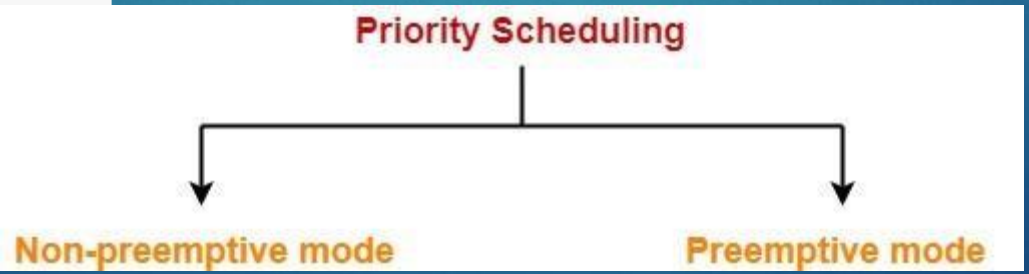
The total time is: 24
The average waiting time (AWT):
(0+3+9+16)/4 = 7

Scheduling Algorithms

Priority Scheduling:

Priority scheduling is a priority based algorithm and one of the most common scheduling algorithms in batch systems.

- Each process is assigned a priority. Process with highest priority is to be executed first and so on.
- Processes with same priority are executed on first come first served basis.
- Priority can be decided based on memory requirements, time requirements or any other resource requirement.



Scheduling Algorithms

Priority Based Scheduling:

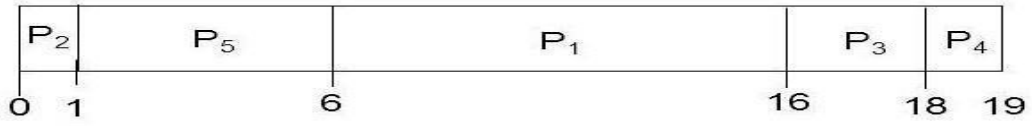
Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems.

- Example:

Non-Preemptive Priority

Process	Burst Time	Priority	arrival time
P_1	10	3	0
P_2	1	1	0
P_3	2	4	0
P_4	1	5	0
P_5	5	2	0

- Priority Scheduling (non-preemptive)



- Average waiting time = (0 + 1 + 6 + 16 + 18)/5 = 8.2

Scheduling Algorithms

Round Robin Scheduling:

Each process gets a small unit of CPU time (time quantum), after this time has elapsed, the process is preempted and added to the end of the ready queue.

- Round Robin is the preemptive process scheduling algorithm.
- Each process is provided a fix time to execute, it is called a quantum.
- Once a process is executed for a given time period, it is preempted and other process executes for a given time period.
- Context switching is used to save states of preempted processes.

Scheduling Algorithms

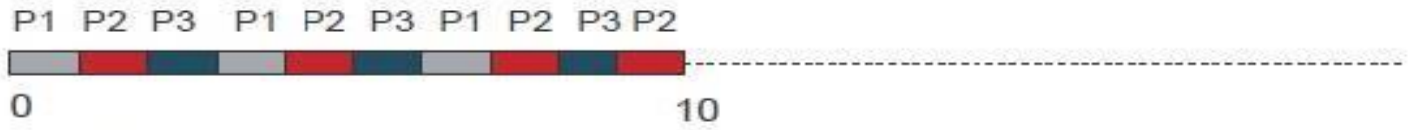
Round Robin Scheduling:

Each process gets a small unit of CPU time (time quantum), after this time has elapsed, the process is preempted and added to the end of the ready queue.

- Example:

Process	Duration	Order	Arrival Time
P1	3	1	0
P2	4	2	0
P3	3	3	0

Suppose time quantum is: 1 unit, P1, P2 & P3 never block



P1 waiting time: 4
P2 waiting time: 6
P3 waiting time: 6

The average waiting time (AWT):
(4+6+6)/3 = 5.33

Scheduling Algorithms

Multiple-Level Queues Scheduling:

Multiple-level queues are not an independent scheduling algorithm.

- They make use of other existing algorithms to group and schedule jobs with common characteristics.
  - Multiple queues are maintained for processes with common characteristics.
  - Each queue can have its own scheduling algorithms.
  - Priorities are assigned to each queue.

ProcessSynchronization

Process Synchronization:

Process Synchronization means sharing system resources by processes in a such a way that, Concurrent access to shared data is handled thereby minimizing the chance of inconsistent data.

- Process Synchronization ensures a perfect co-ordination among the process.
- Maintaining data consistency demands mechanisms to ensure synchronized execution of cooperating processes.
- Process Synchronization can be provided by using several different tools like:
  - ***Semaphores***
  - ***Mutual Exclusion or Mutex***
  - ***Monitor***

37 Process Synchronization

Process Synchronization:

Process Synchronization means sharing system resources by processes in a such a way that, Concurrent access to shared data is handled thereby minimizing the chance of inconsistent data.

- Process synchronization problem arises in the case of Cooperative process also because resources are shared in Cooperative processes.
- On the basis of synchronization, processes are categorized as one of the following two types:
  - ***Independent Process:*** *Execution of one process does not affects the execution of other processes.*
  - ***Cooperative Process:*** *Execution of one process affects the execution of other processes.*

ProcessSynchronization

Process Synchronization:

Race Condition:

- When serval process access and manipulates the same data at the same time, they may enter into a race condition
- Race condition occurs among process that share common storage for read and write.
- Race condition occurs due to improper synchronization of shared memory access.

Critical section problem:

- Critical section is a code segment that can be accessed by only one process at a time.
- Critical section contains shared variables which need to be synchronized to maintain consistency of data variables.
- *Any solution to the critical section problem must satisfy three requirements:*
  - ***Mutual Exclusion***
  - ***Progress***
  - ***Bounded Waiting***

ProcessSynchronization

Semaphores:

A semaphore is a signaling mechanism and a thread that is waiting on a semaphore can be signaled by another thread.

- A semaphore uses two atomic operations, wait and signal for process synchronization.
- Classical problems of Synchronization with Semaphore Solution:
  - ***Bounded-buffer (or Producer-Consumer) Problem***
  - ***Dining- Philosophers Problem***
  - ***Readers and Writers Problem***
  - ***Sleeping Barber Problem***

ProcessSynchronization

Bounded-buffer (or Producer-Consumer) Problem:

Bounded Buffer problem is also called producer consumer problem. This problem is generalized in terms of the Producer-Consumer problem.

- Solution to this problem is, creating two counting semaphores “full” and “empty” to keep track of the current number of full and empty buffers respectively.

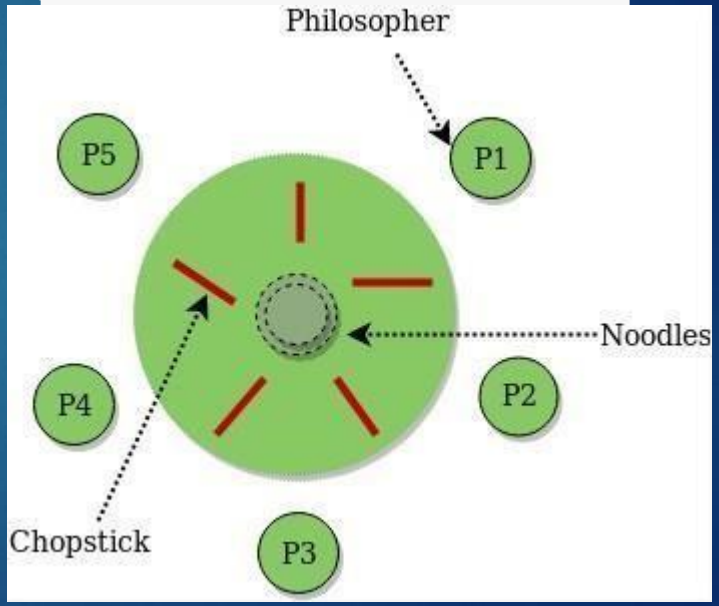
- Producers produce a product and consumers consume the product, but both use of one of the containers each time.

ProcessSynchronization

Dining- Philosophers Problem:

The Dining Philosopher Problem states that K philosophers seated around a circular table with one chopstick between each pair of philosophers.

- There is one chopstick between each philosopher.
- A philosopher may eat if he can pickup the two chopsticks adjacent to him.
- One chopstick may be picked up by any one of its adjacent followers but not both.
- This problem involves the allocation of limited resources to a group of processes in a deadlock-free and starvation-free manner.



Readers and Writers Problem:

Suppose that a database is to be shared among several concurrent processes. We distinguish between these two types of processes by referring to the former as readers and to the latter as writers.

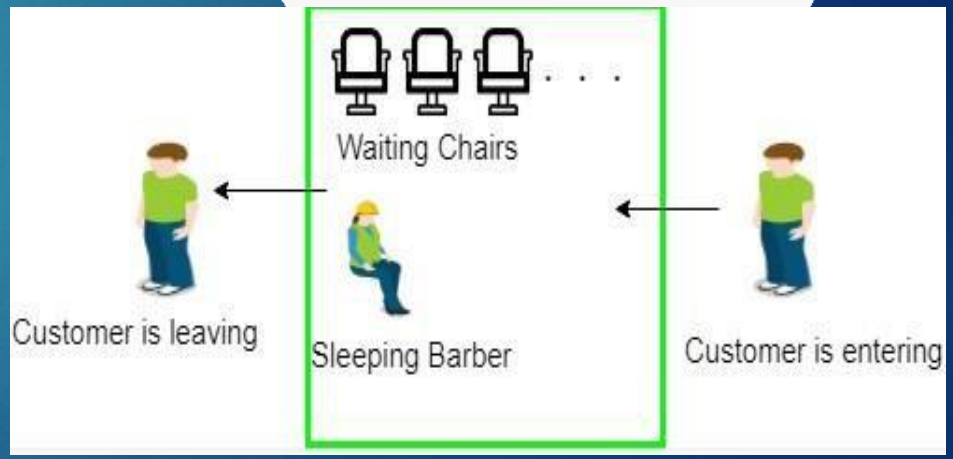
- Precisely in OS we call this situation as the readers-writers problem.
- Problem parameters:
  - One set of data is shared among a number of processes.
  - Once a writer is ready, it performs its write. Only one writer may write at a time.
  - If a process is writing, no other process can read it.
  - If at least one reader is reading, no other process can write.
  - Readers may not write and only read.

ProcessSynchronization

Sleeping Barber Problem:

The Dining Philosopher Problem states that K philosophers seated around a circular table with one chopstick between each pair of philosophers.

- Barber shop with one barber, one barber chair and N chairs to wait in.
- When no customers the barber goes to sleep in barber chair and must be woken when a customer comes in.
- When barber is cutting hair new customers take empty seats to wait, or leave if no vacancy.

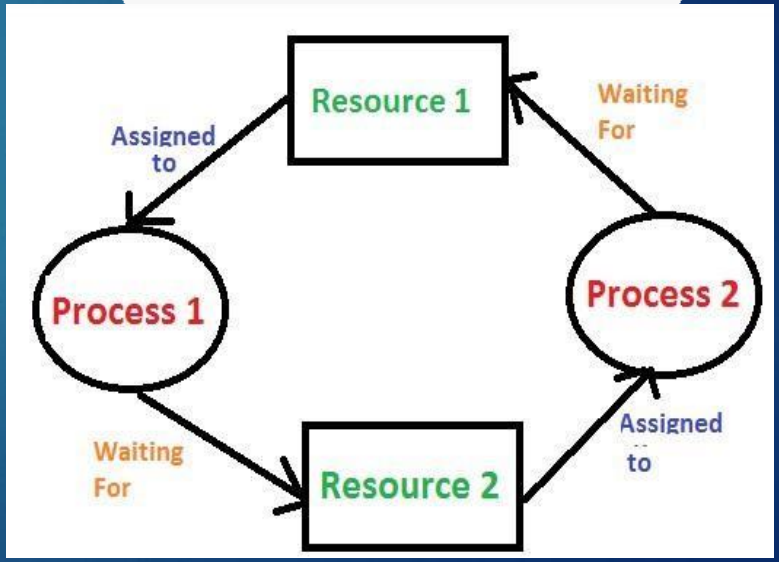


Deadlocks

Deadlock:

Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.

- In operating systems when there are two or more processes that hold some resources and wait for resources held by other(s).
- Example:
  - Process 1 is holding Resource 1 and waiting for resource 2 which is acquired by process 2, and process 2 is waiting for resource 1.



Deadlocks

Deadlock:

Methods for handling deadlock .

- There are three ways to handle deadlock:
  - ***Deadlock prevention or avoidance****: The idea is to not let the system into a deadlock state. by using strategy of "Avoidance", we have to make an assumption.*
  - ***Deadlock detection and recovery****: Let deadlock occur, then do preemption to handle it once occurred.*
  - ***Ignore the problem altogether****: If deadlock is very rare, then let it happen and reboot the system. This is the approach that both Windows and UNIX take.*

Introduction to Operating Systems – 3

MemoryManagement

Introduction to Operating System

Memory Management

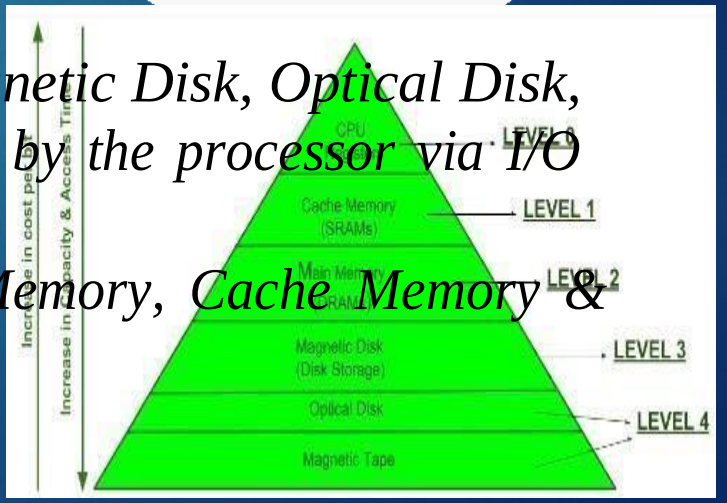
- Memory Concepts
- Memory Allocation
- Memory Partitions
- Paging
- Virtual Memory

Introduction

Memory Hierarchy Design:

In the Computer System Design, Memory Hierarchy is an enhancement to organize the memory such that it can minimize the access time.

- The Memory Hierarchy was developed based on a program behavior known as locality of references.
- This Memory Hierarchy Design is divided into 2 main types:
  - **External Memory or Secondary Memory:** *Comprising of Magnetic Disk, Optical Disk, Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.*
  - **Internal Memory or Primary Memory:** *Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.*
- ,



Memory Management:

Memory management is the functionality of an operating system which handles or manages primary memory and moves processes back and forth between main memory and disk during execution.

- Main Memory refers to a physical memory that is the internal memory to the computer.
- Memory management keeps track of each and every memory location, regardless of either it is allocated to some process or it is free.
- It checks how much memory is to be allocated to processes.
- It decides which process will get memory at what time.
- It tracks whenever some memory gets freed or unallocated and correspondingly it updates the status.

Introduction

Process Address Space:

The process address space is the set of logical addresses that a process references in its code.

- The operating system takes care of mapping the logical addresses to physical addresses at the time of memory allocation to the program.
- There are three types of addresses used in a program before and after memory is allocated .
  - **Symbolic addresses:** The addresses used in a source code. The variable names, constants, and instruction labels are the basic elements of the symbolic address space.
  - **Relative addresses:** At the time of compilation, a compiler converts symbolic addresses into relative addresses.
  - **Physical addresses:** The loader generates these addresses at the time when a program is loaded into main memory.

MemoryLoading

Memory Loading:

All the programs are loaded in the main memory for execution. Sometimes complete program is loaded into the memory, but some times a certain part or routine of the program is loaded into the main memory only when it is called by the program.

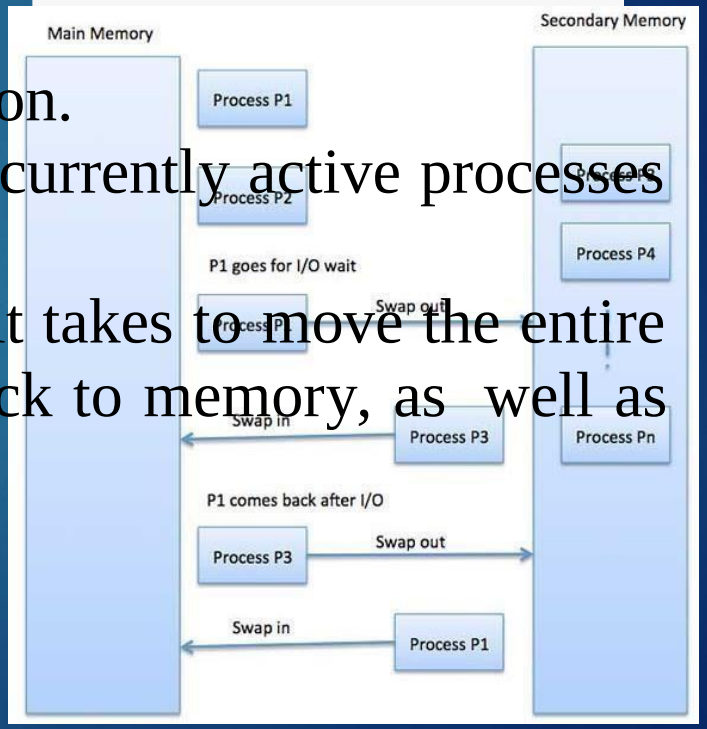
- There are two types of Loading techniques:
  - **Static Loading:** *The absolute program (and data) is loaded into memory in order for execution to start.*
  - **Dynamic Loading:** *dynamic routines of the library are stored on a disk in relocatable form and are loaded into memory only when they are needed by the program.*

MemoryLoading

Swapping:

Swapping is the process of bringing in each process in main memory, running it for a while and then putting it back to the disk.

- Swapping is also known as a technique for memory compaction.
- sometimes there is not enough main memory to hold all the currently active processes in a timesharing system.
- The total time taken by swapping process includes the time it takes to move the entire process to a secondary disk and then to copy the process back to memory, as well as the time the process takes to regain main memory.



8 MemoryAllocation

Memory Allocation

MemoryAllocation:

In the operating system, the following are four common memory management techniques.

- ***Single contiguous allocation:*** *Simplest allocation method used by MS-DOS. All memory (except some reserved for OS) is available to a process.*
- ***Partitioned allocation:*** *Memory is divided into different blocks or partitions. Each process is allocated according to the requirement.*
- ***Paged memory management:*** *Memory is divided into fixed-sized units called page frames, used in a virtual memory environment.*
- ***Segmented memory management:*** *Memory is divided into different segments. In this management, allocated memory doesn't have to be contiguous.*

..

9 MemoryAllocation

Memory Allocation

PartitionAllocation :

In Partition Allocation, when there is more than one partition freely available to accommodate a process's request, a partition must be selected.

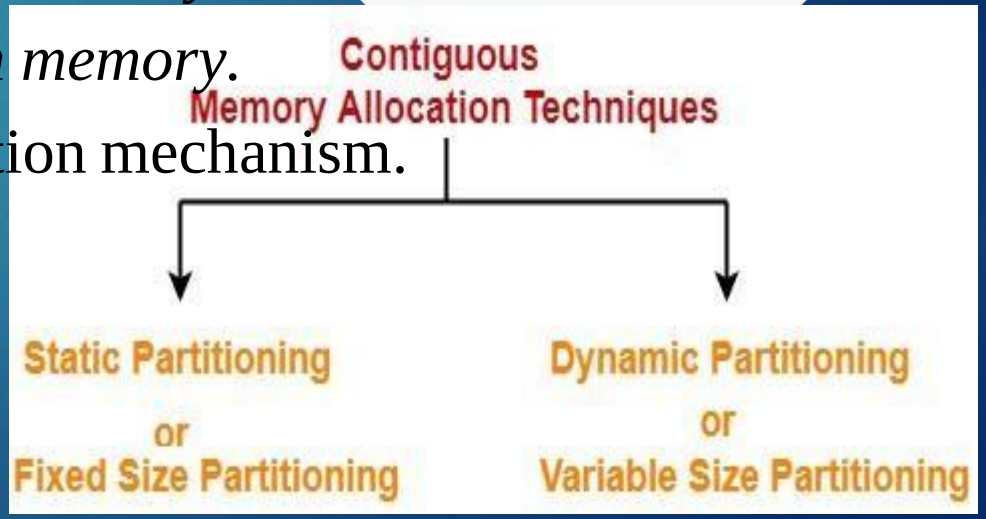
- To choose a particular partition, a partition allocation method is needed.
- When it is time to load a process into the main memory and if there is more than one free block of memory of sufficient size then the OS decides which free block to allocate.
- There are different PlacementAlgorithm:
  - **First Fit:** *The first hole that is big enough is allocated to program.*
  - **Best Fit:** *The smallest hole that is big enough is allocated to program.*
  - **Worst Fit:** *The largest hole that is big enough is allocated to program.*

MemoryAllocation

Memory Partitions :

Memory allocation is a process by which computer programs are assigned memory or space.

- Main memory usually has two partitions –
  - ***Low Memory*** – *Operating system resides in this memory.*
  - ***High Memory*** – *User processes are held in high memory.*
- Operating system uses the following memory allocation mechanism.
- Memory ***Partitioning*** types are:
  - ***Fixed / Static-partition allocation***
  - ***Dynamic/ Multiple-partition allocation***

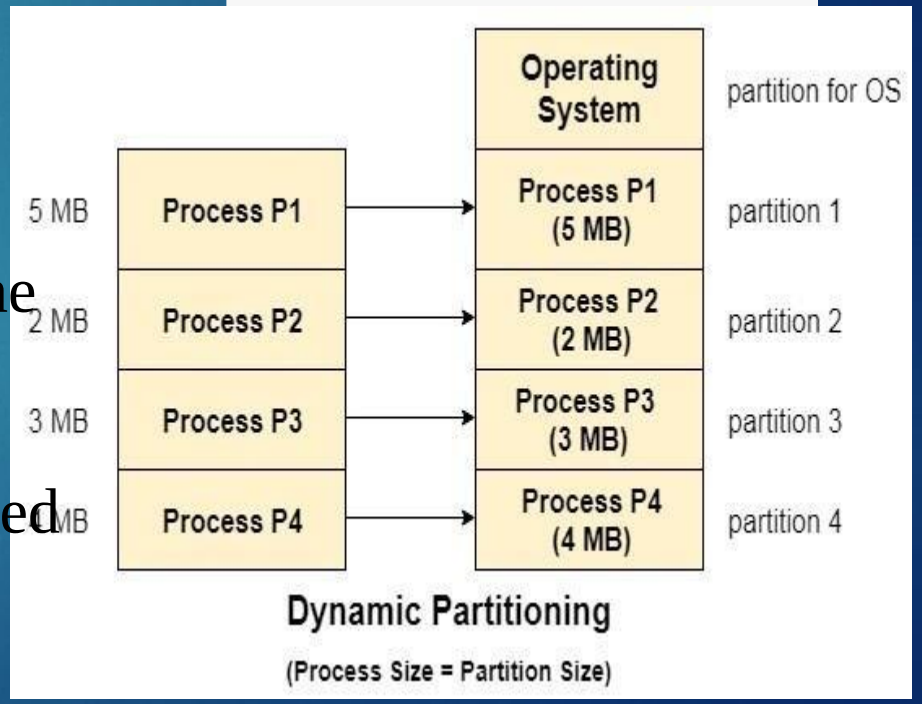


MemoryAllocation

Dynamic Memory Partitioning:

In this technique, the partition size is not declared initially. It is declared at the time of process loading

- The first partition is reserved for the operating system.
- The remaining space is divided into parts.
- The size of each partition will be equal to the size of the process.
- The partition size varies according to the need of the process so that the internal fragmentation can be avoided

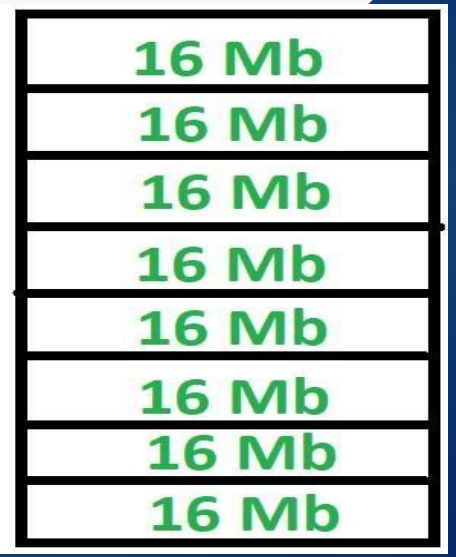


MemoryAllocation

Fixed / Contiguous Memory Partitions :

In this type of allocation, main memory is divided into a number of fixed-sized partitions where each partition should contain only one process.

- In this technique, the main memory is divided into partitions of equal or different sizes.
- The operating system always resides in the first partition while the other partitions can be used to store user processes.
- The memory is assigned to the processes in contiguous way.
- The partitions cannot overlap.
- A process must be contiguously present in a partition for the execution

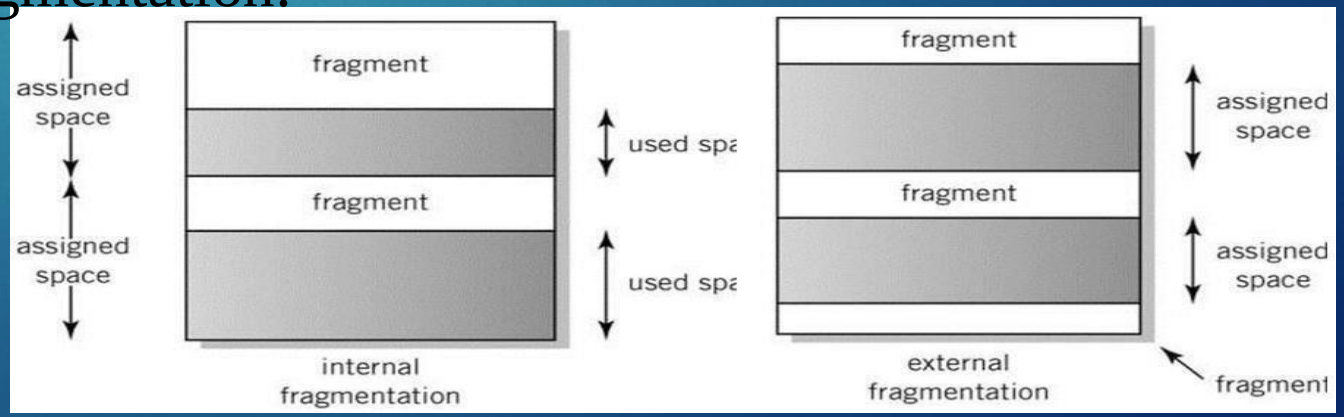


MemoryAllocation

Fragmentation:

Fragmentation is an unwanted problem where the memory blocks cannot be allocated to the processes due to their small size and the blocks remain unused.

- The process with the size greater than the size of the largest partition could not be executed due to the lack of sufficient contiguous memory blocks cannot be allocated to new upcoming processes and results in inefficient use of memory.
- Basically, there are two types of fragmentation:
  - Internal Fragmentation
  - External Fragmentation

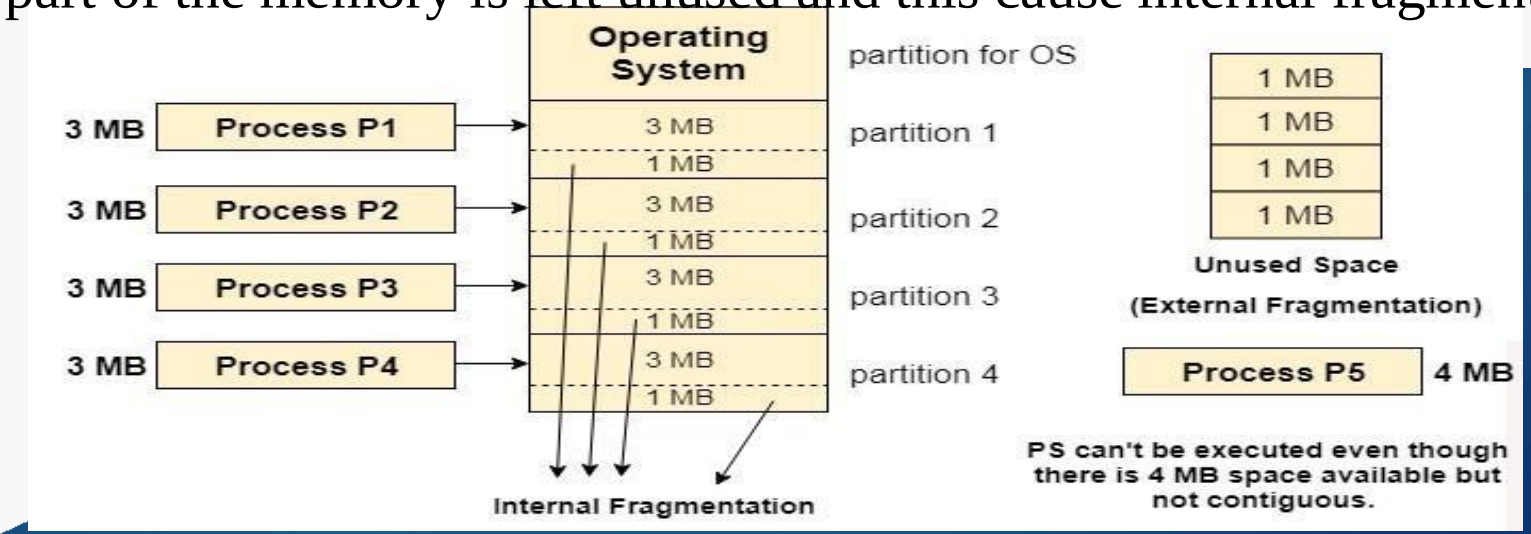


MemoryAllocation

Internal Fragmentation:

Memory block assigned to process is bigger. Some portion of memory is left unused, as it cannot be used by another process.

- In this fragmentation, the process is allocated a memory block of size more than the size of that process.
- Due to this some part of the memory is left unused and this cause internal fragmentation.

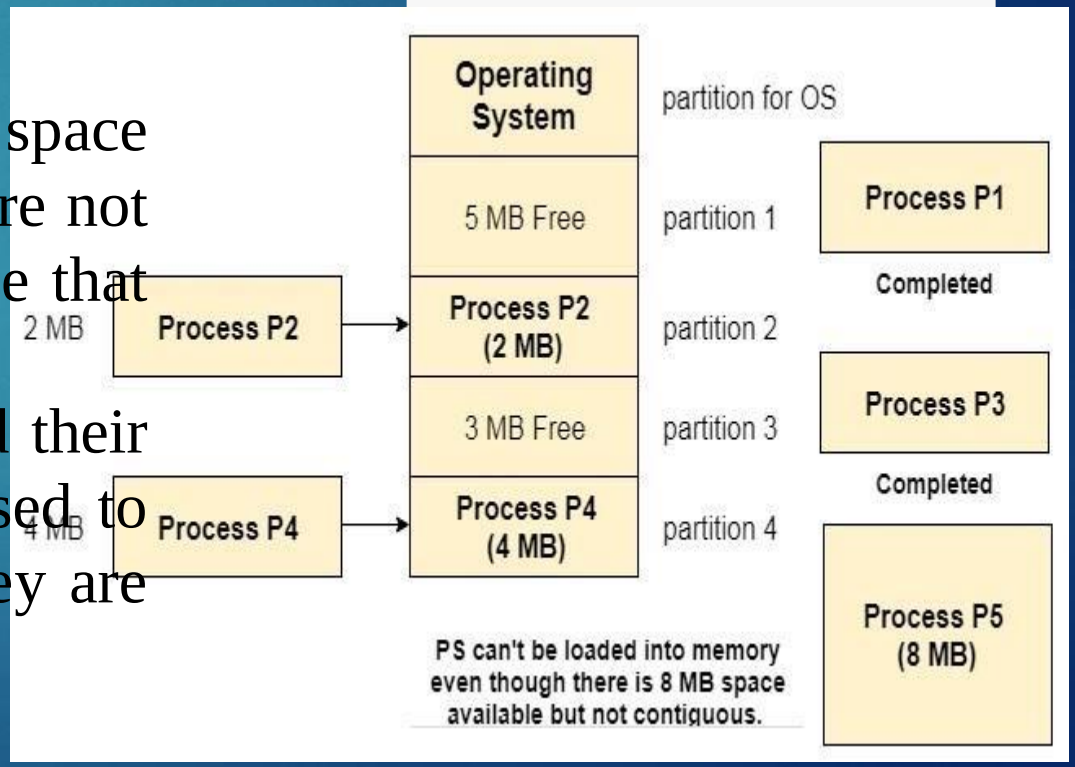


MemoryAllocation

External Fragmentation:

Total memory space is enough to satisfy a request or to reside a process in it, but it is not contiguous, so it cannot be used.

- In this fragmentation, although we have total space available that is needed by a process still we are not able to put that process in the memory because that space is not contiguous.
- After some time P1 and P3 got completed and their assigned space is freed, but they cannot be used to load a 2 MB process in the memory since they are not contiguously located

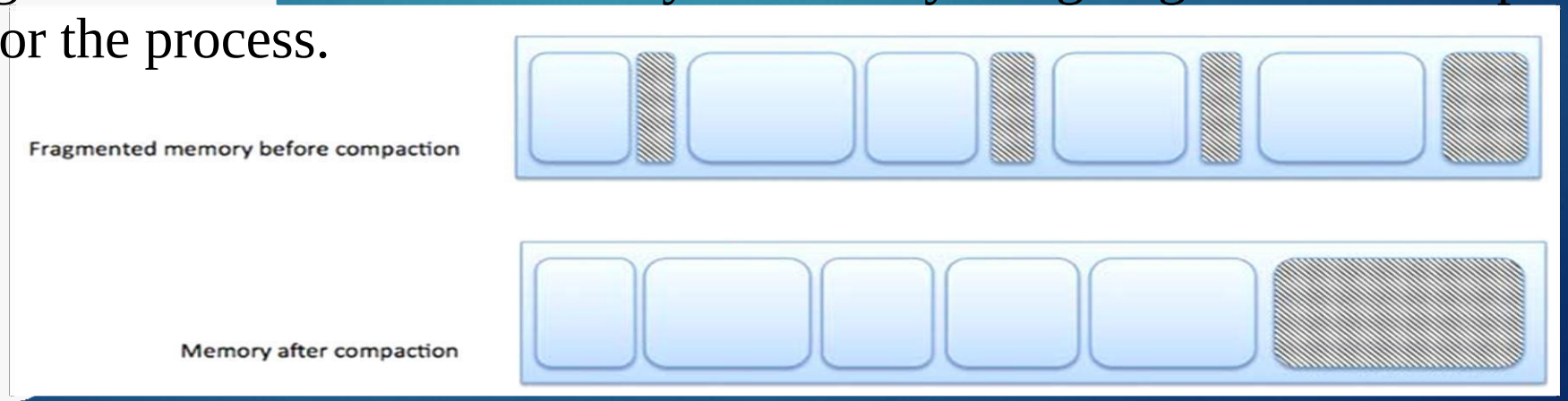


MemoryAllocation

Compaction:

Compaction technique can be used to create more free memory out of fragmented memory.

- External fragmentation can be reduced by compaction or shuffle memory contents to place all free memory together in one large block. To make compaction feasible, relocation should be dynamic.
- The internal fragmentation can be reduced by effectively assigning the smallest partition but large enough for the process.

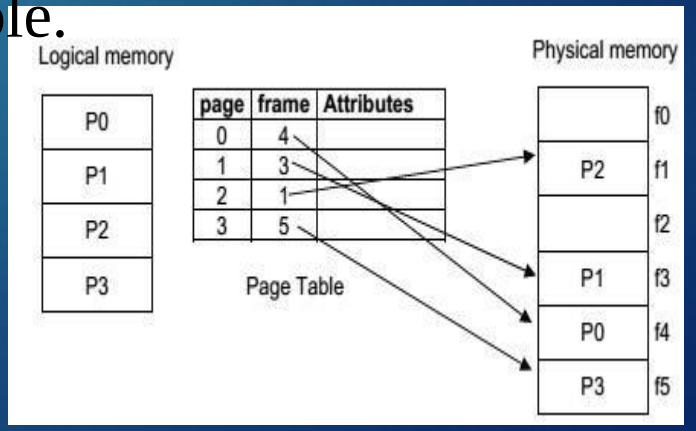


MemoryAllocation

Paging:

Pages of the process are brought into the main memory only when they are required otherwise they reside in the secondary storage.

- Considering the fact that the pages are mapped to the frames in Paging, page size needs to be as same as frame size.
- Different operating system defines different frame sizes.
- The pages belonging to a certain process are loaded into available.
- The sizes of each frame must be equal.
- **Pages** size is power of 2, between 512 bytes and 8192 bytes.
- The size of the process is measured in the number of pages..



MemoryAllocation

Paging:

Address Translation.

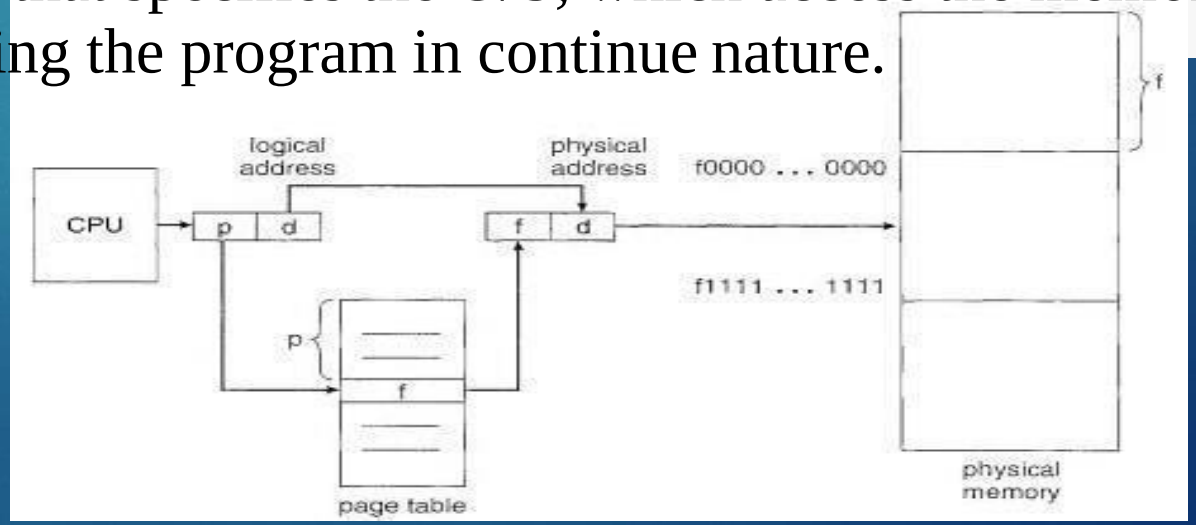
- Page address is called **logical address** and represented by **page number** and the **offset.**
  - Logical Address = Page number + page offset
- Frame address is called **physical address** and represented by a **frame number** and the **offset**.
  - Physical Address = Frame number + page offset
- A data structure called **page map table** is used to keep track of the relation between a page of a process to a frame in physical memory.

MemoryAllocation

Paging Faults:

A page fault occurs when a program attempts to access a block of memory that is not stored in the physical memory, or RAM.

- The fault notifies the operating system that it must locate the data in virtual memory, then transfer it from the storage device to the system RAM.
- Page fault arise the exception, that specifies the O/S, which access the memory slots from virtual memory for running the program in continue nature.

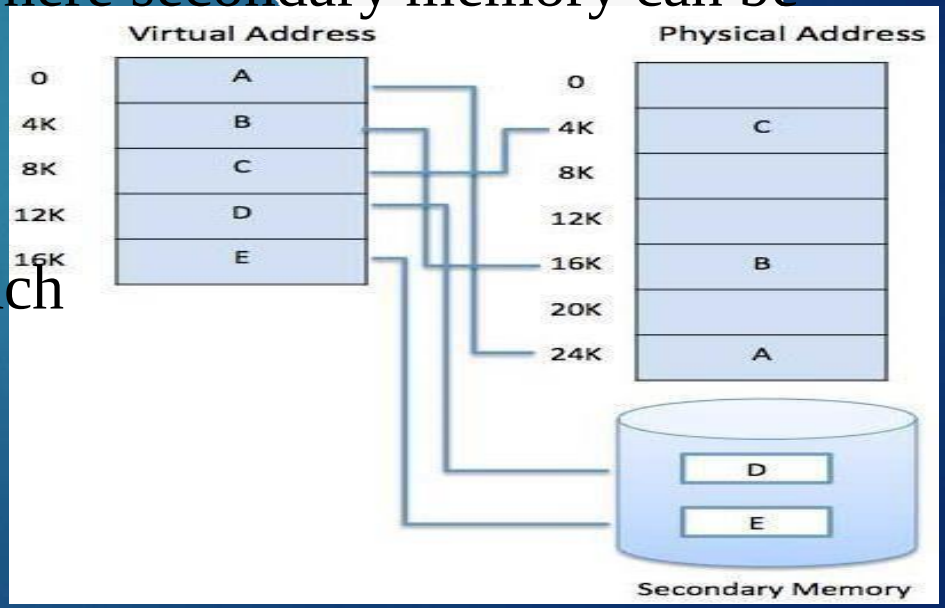


Virtulamemory

Definition:

Virtual memory is a feature of an operating system that enables a computer to be able to compensate shortages of physical memory by transferring pages of data from random access memory to disk storage.

- Virtual memory is a memory management technique where secondary memory can be used as if it were a part of the main memory
- Virtual memory serves two purposes.
  - It allows us to extend the use of physical memory.
  - It allows us to have memory protection, because each virtual address is translated to a physical address.

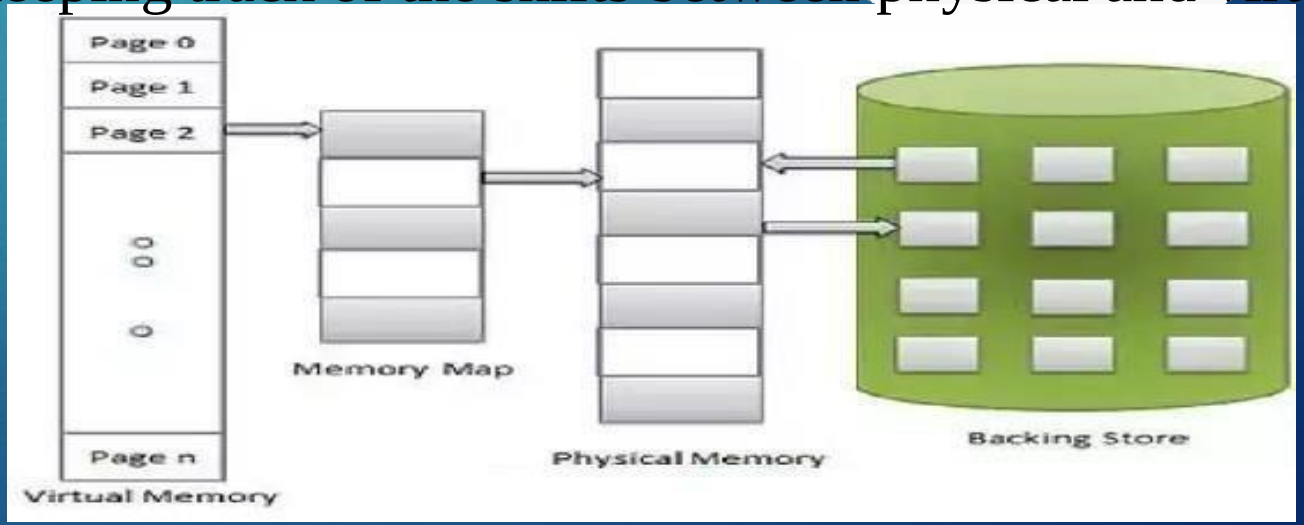


Virtulamemory

Virtual memory:

Virtual memory allows a computer to treat secondary memory as though it were the main memory.

- Virtual memory uses hardware and software to allow a computer to compensate for physical memory shortages.
- Memory manager is in charge of keeping track of the shifts between physical and virtual memory.
- Types of virtual memory are:
  - ***Demand Paging***
  - ***Segmentation***

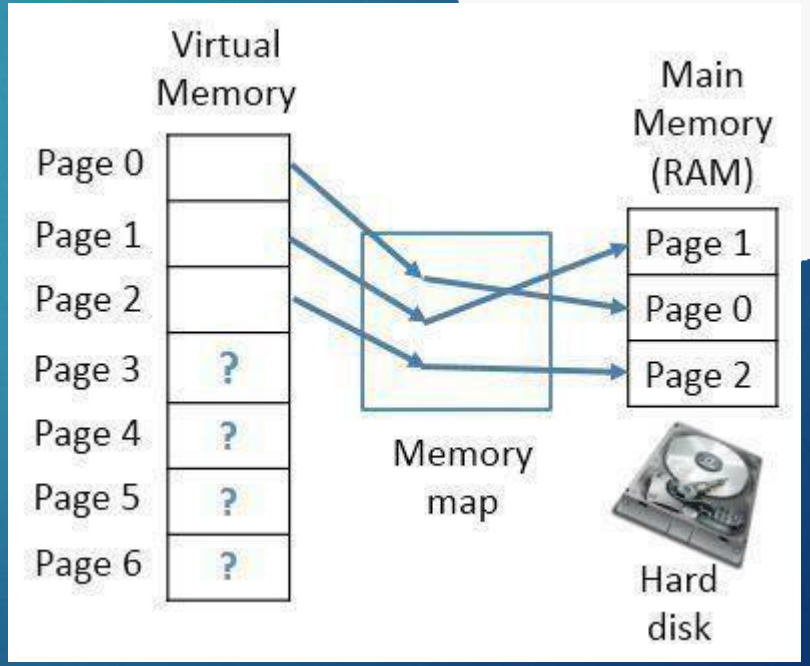


Virtulamemory

Demand Paging:

A demand paging system is quite similar to a paging system with swapping where processes reside in secondary memory and pages are loaded only on demand, not in advance.

- Types of Page Replacement Methods are:
  - *FIFO*
  - *Optimal Algorithm*
  - *LRU Page Replacement*

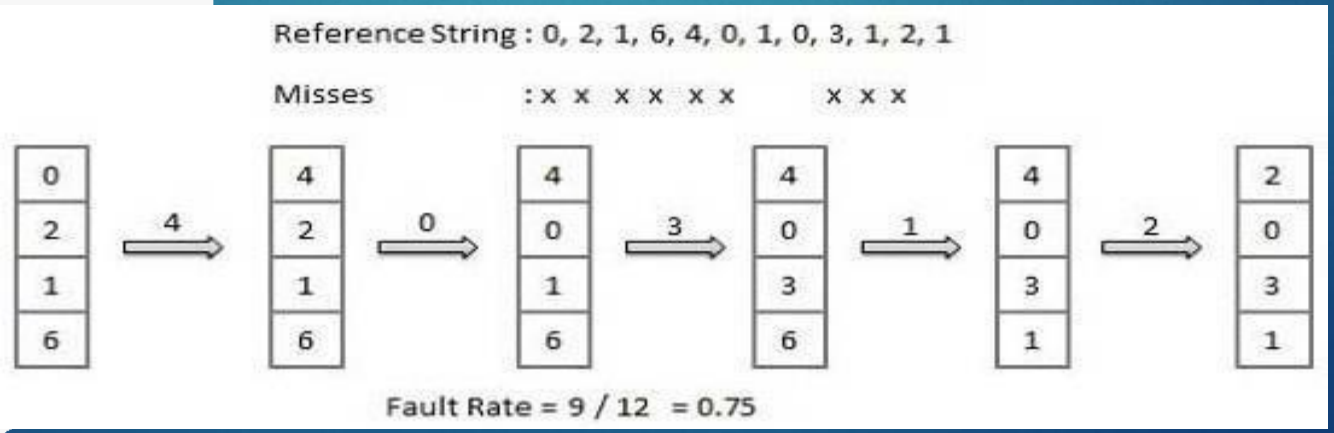


Virtulamemory

FIFO Page Replacement:

FIFO (First-in-first-out) is a simple implementation method, the process selects the page for a replacement that has been in the virtual address of the memory for the longest time.

- Features of FIFO Page Replacement Methods are:
  - *Whenever a new page loaded, the page recently comes in the memory is removed.*
  - *The oldest page in the main memory is one that should be selected for replacement first*

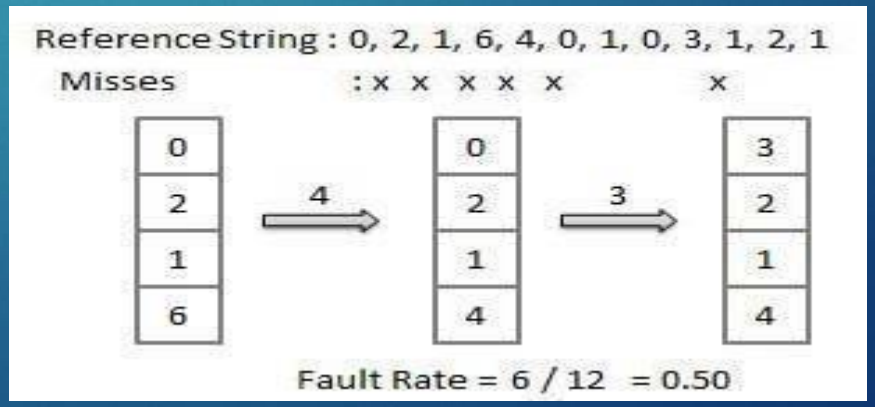


Virtulamemory

Optimal Algorithm:

The optimal page replacement method selects that page for a replacement for which the time to the next reference is the longest.

- Features of Optimal algorithm are:
  - *Optimal algorithm results in the fewest number of page faults. This algorithm is difficult to implement.*
  - *Replace the page which unlike to use for a longer period of time. It only uses the time when a page needs to be used.*



LRU Page Replacement:

Least Recently Used (LRU) page, this method helps OS to find page usage over a short period of time.

- This algorithm should be implemented by associating a counter with an even- page.
- Features of LRU Page Replacement are:
  - *The LRU replacement method has the highest count. This counter is also called aging registers, which specify their age and how much their associated pages should also be referenced.*
  - *The page which hasn't been used for the longest time in the main memory is the one that should be selected for replacement.*
  - *It also keeps a list and replaces pages by looking back into time*

Virtulamemory

Virtual memory:

Virtual memory is important for improving system performance, multitasking, using large programs and flexibility.

- Benefits of using virtual memory
  - *Frees applications from managing shared memory and saves users from having to add memory modules when RAM space runs out.*
  - *Increased security because of memory isolation.*
  - *Multiple larger applications can be run simultaneously.*
  - *Doesn't need external fragmentation.*
  - *Effective CPU use.*
  - *Data can be moved automatically.*

Operating System - Properties

Distributed Environment:

A distributed environment refers to multiple independent CPUs or processors in a computer system.

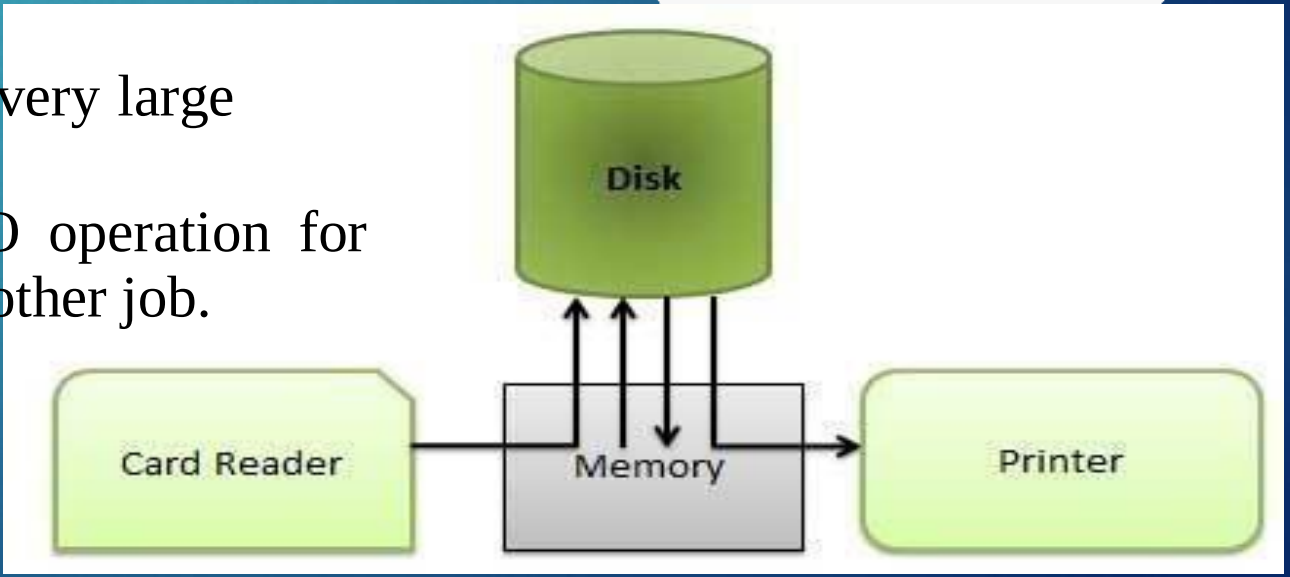
- An operating system does the following activities related to distributed environment:
  - *The OS distributes computation logics among several physical processors.*
  - *The processors do not share memory or a clock. Instead, each processor has its own local memory.*
  - *The OS manages the communications between the processors.*
  - *They communicate with each other through various communication lines.*

Operating System - Properties

Spooling:

Spooling is an acronym for simultaneous peripheral operations on line.

- Spooling refers to putting data of various I/O jobs in a buffer.
- This buffer is a special area in memory or hard disk which is accessible to I/O devices.
- **Advantages**
  - The spooling operation uses a disk as a very large buffer.
  - Spooling is capable of overlapping I/O operation for one job with processor operations for another job.



29 OperatingSystem-Properties

Operating System - Properties

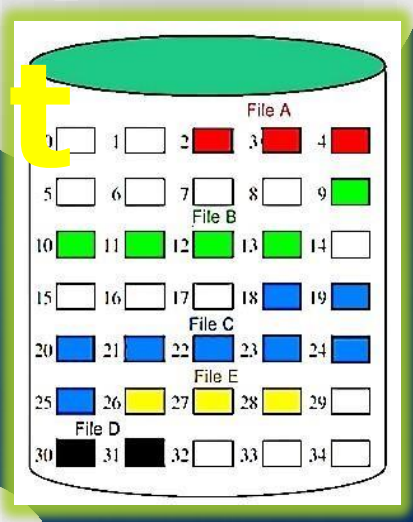
Spooling:

Spooling is an acronym for simultaneous peripheral operations on line.

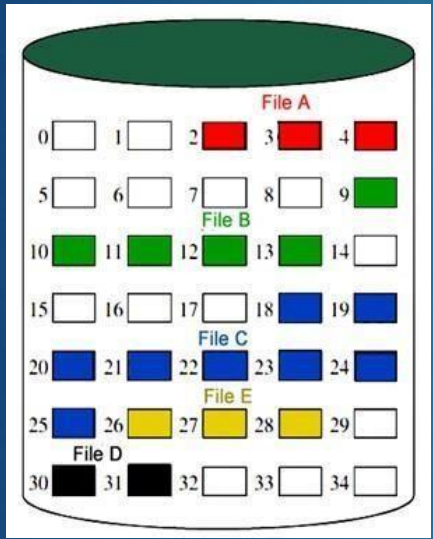
- Spooling refers to putting data of various I/O jobs in a buffer.
- An operating system does the following activities related to distributed environment .
  - *Handles I/O device data spooling as devices have different data access rates.*
  - *Maintains the spooling buffer which provides a waiting station where data can rest while the slower device catches up.*
  - *Maintains parallel computation because of spooling process as a computer can perform I/O in parallel fashion.*
  - *It becomes possible to have the computer read data from a tape, write data to disk and to write out to a tape printer while it is doing its computing task*

Introduction to Operating Systems -4

Filemanagement



Filemanagement



Filemanagement

- FileConcepts
- Accessmethods
- Allocatiomethods
- Disk Management
- Schedulingmethods

3 Introduction

Introduction

Introduction to File:

A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks.

- File is a collection of logically related entities
  - Afile has a certain defined structure according to its type.
  - Atext file is a sequence of characters organized into lines.
  - Asource file is a sequence of procedures and functions.
  - An object file is a sequence of bytes organized into blocks that are understandable by the machine.

- When operating system defines different file structures, it also contains the code to support these file structure. Unix, MS-DOS support minimum number of file structure

Introduction

File Type:

File type refers to the ability of the operating system to distinguish different types of file such as text files source files and binary files etc.

- Operating system like MS-DOS and UNIX have the following types of files.
  - **Ordinary files**
    - These are the files that contain user information.
    - These may have text, databases or executable program.
  - **Directory files**
    - These files contain list of file names and other information related to these files.
  - **Special files**
    - These files are also known as device files.
    - These files represent physical device like disks, terminals, printers, networks, tape drive etc.

Introduction

File Attributes:

file is a collection of logically related entities.

- **Name** – only information kept in human-readable form
- **Identifier** – unique tag (number) identifies file within file system
- **Type** – needed for systems that support different types
- **Location** – pointer to file location on device
- **Size** – current file size
- **Protection** – controls who can do reading, writing, executing
- **Time, date, and user identification** – data for protection, security, and usage monitoring Information about files are kept in the directory structure, which is maintained on the disk.

FileOperation

Operation on a File:

The file contains information about the files name, including attributes, location and ownership.

- Operation performed on file are.
  - Create a file
  - Delete a file
  - Open a file
  - Close a file
  - Write _ at write pointer location
  - Read _ at read pointer location
  - Reposition within file - seek
  - Truncate the file system

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes com-pressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

FileAccessMethods

File Structure:

There are several types of files structures depending on its size and allocation.

- **None -** sequence of words, bytes
- **Simple record structure**
  - Lines
  - Fixed length
  - Variable length
- **Complex Structures**
  - Formatted document
  - Relocatable load file

FileAccessMethods

File Access Mechanisms:

There are several ways to access files.

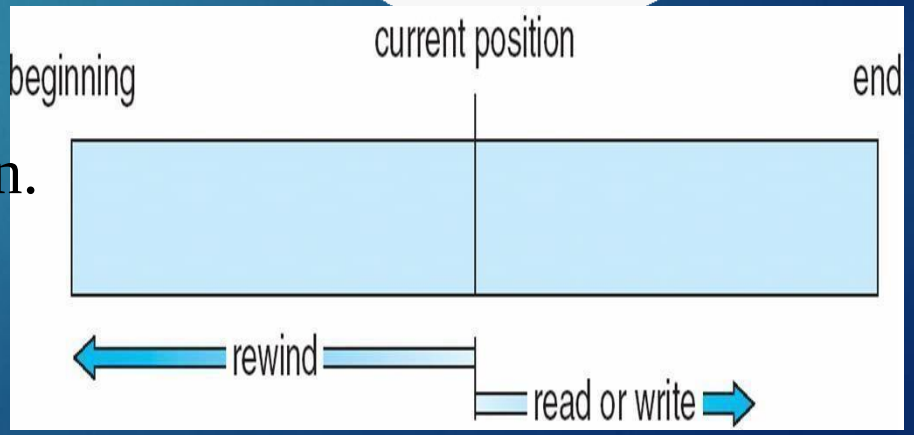
- Operating system like MS-DOS and UNIX have the following types of files.
  - **Sequential access**
  - **Direct/Random access**
  - **Indexed sequential access**.

FileAccessMethods

Sequential access:

A sequential access is that in which the data in the file are accessed in a sequential order in which there are stored.

- A sequential access is that in which the records are accessed in some sequence, i.e., the information in the file is processed in order, one record after the other.
- This access method is the most primitive one.
- Example: Compilers usually access files in this fashion.



FileAccessMethods

Direct/Random access:

Direct access is that in which the data in the file are accessed in a Random order in which there are stored.

- Random access file organization provides, accessing the records directly.
- Each record has its own address on the file with by the help of which it can be directly accessed for reading or writing.
- The records need not be in any sequence within the file and they need not be in adjacent locations on the storage medium.

FileAccessMethods

Indexed sequential access:

Indexed sequential access is that in which the data in the file are accessed using pointers to search sequentially in which there are stored.

- This mechanism is built up on base of sequential access.
- An index is created for each file which contains pointers to various blocks.
- Index is searched sequentially and its pointer is used to access the file directly.

FileAllocation

File Space Allocation:

Directories need to be fast to search, insert, and delete, with a minimum of wasted disk space.

- Files are allocated disk spaces by operating system..
  - Contiguous Allocation
  - Linked Allocation
  - Indexed Allocation

FileAllocation

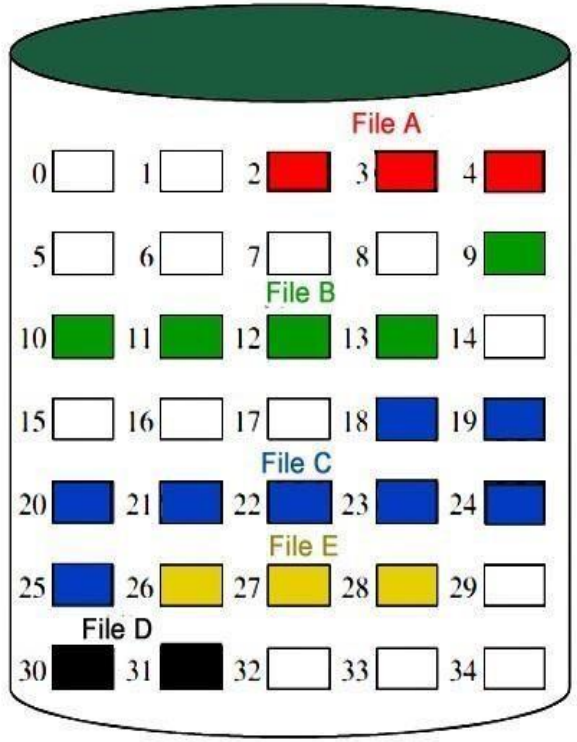
Contiguous Allocation:

In Contiguous file allocation, files are allocated disk spaces by operating system:

- Each file occupies a contiguous address space on disk.
- Assigned disk address is in linear order.
- Easy to implement.
- External fragmentation is a major issue with this type of allocation technique

File allocation table

File name	Start block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

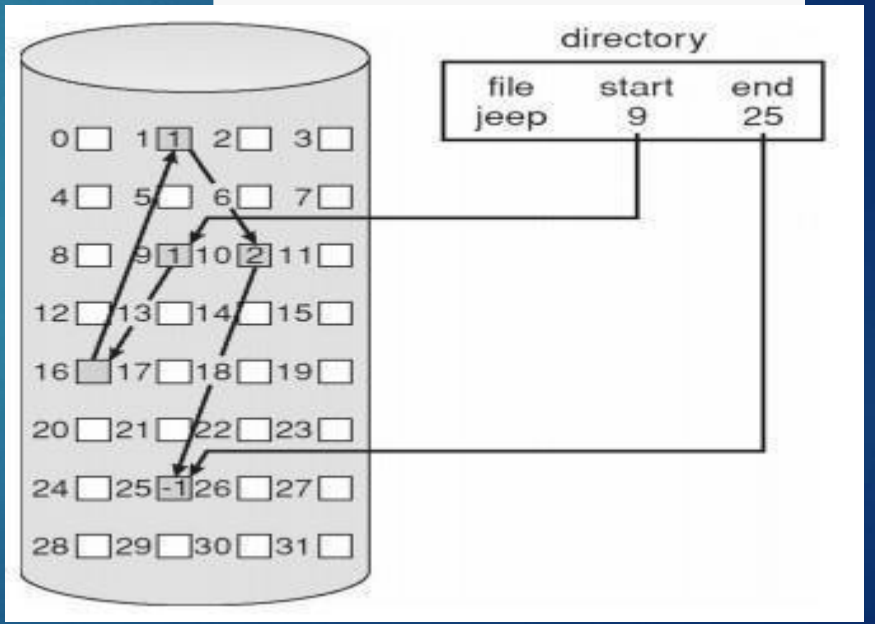


FileAllocation

Linked Allocation:

In Linked file allocation, files are allocated disk spaces by operating system:

- Each file carries a list of links to disk blocks.
- Directory contains link / pointer to first block of a file.
- No external fragmentation
- Effectively used in sequential access file.
- Inefficient in case of direct access file.

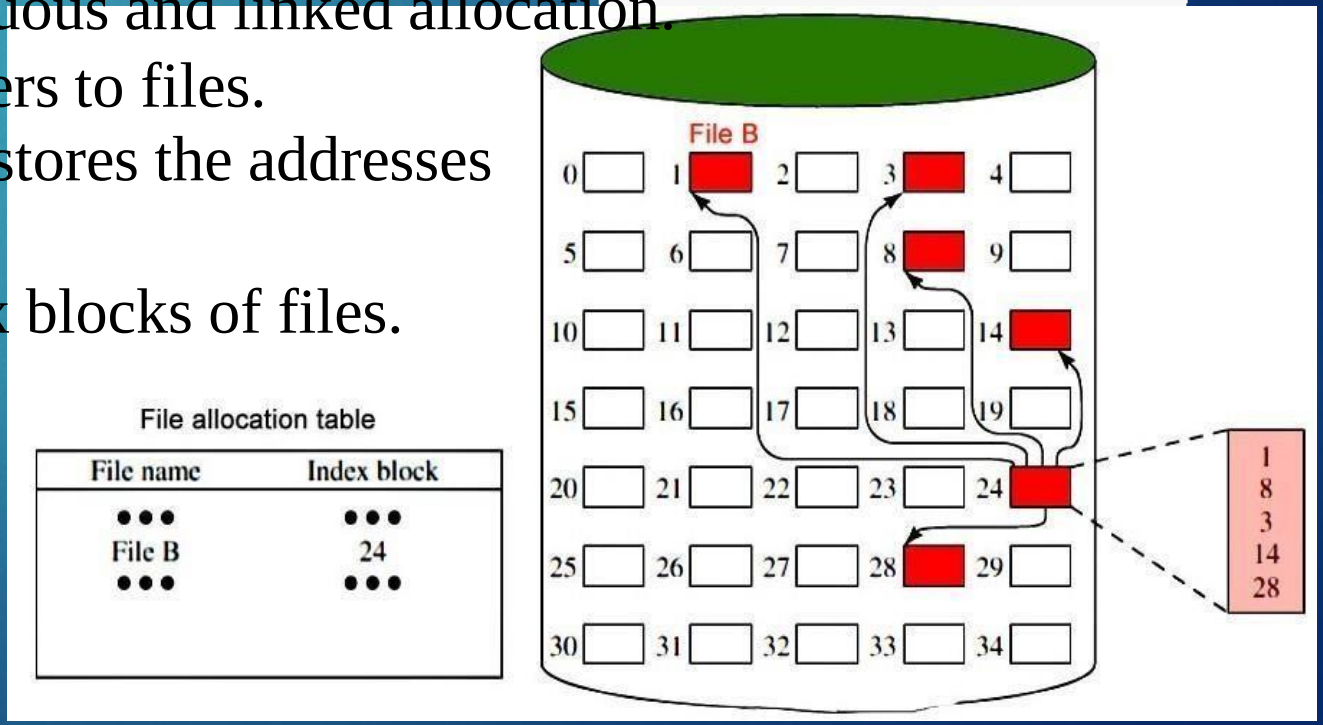


FileAllocation

Indexed Allocation:

In Indexed file allocation, files are allocated disk spaces by operating system:

- Provides solutions to problems of contiguous and linked allocation.
- A index block is created having all pointers to files.
- Each file has its own index block which stores the addresses of disk space occupied by the file.
- Directory contains the addresses of index blocks of files.



Space Management

Free-Space Management:

Another important aspect of disk management is keeping track of and allocating free space:

- The optimal allocation method is different for sequential access files than for random access files, and is also different for small files than for large files.
- Various methods for keeping track of and allocating free space are:
  - ***Bit Vector***
  - ***Linked List***
  - ***Grouping***
  - ***Counting***
  - ***Space Maps***

Space Management

Free-Space Management:

Bit Vector:

- One simple approach is to use a bit vector, in which each bit represents a disk block, set to 1 if free or 0 if allocated.
- Fast algorithms exist for quickly finding contiguous blocks of a given size
- The down side is that a 40GB disk requires over 5MB just to store the bitmap. (For
- example.)

0	0	0	1	0	1	1	0	-----	1

0- if disk block is allocated

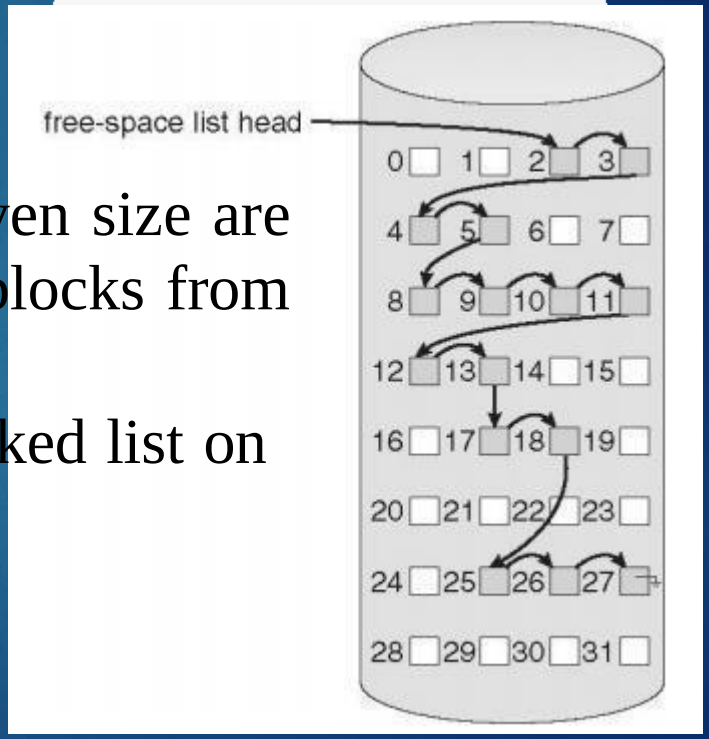
1- if disk block is free

Space Management

Free-Space Management:

Linked List:

- A linked list can also be used to keep track of all free blocks.
- Traversing the list and/or finding a contiguous block of a given size are tracked by the system just by adding and removing single blocks from the beginning of the list.
- The FAT table keeps track of the free list as just one more linked list on the table



Space Management
SpaceManagement

***Free-Space Management:* Grouping:**

- Avariation on linked list free lists is to use links of blocks of indices of free blocks.
- If a block holds up to N addresses, then the first block in the linked-list contains up to N-1 addresses of free blocks and a pointer to the next block of free addresses.

Counting:

- When there are multiple contiguous blocks of free space then the system can keep track of the starting address of the group and the number of contiguous free blocks.
- As long as the average length of a contiguous group of free blocks is greater than two this offers a savings in space needed for the free list.

FileDirectories

File Directories:

Collection of files is a file directory. The directory contains information about the files, including attributes, location and ownership.

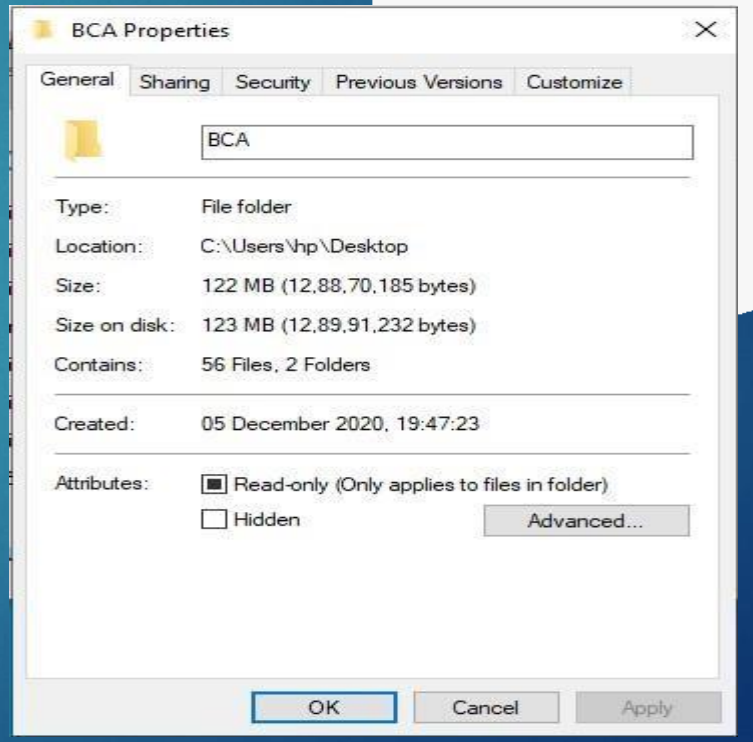
- Advantages of maintaining directories are:
  - **Efficiency:** A file can be located more quickly.
  - **Naming:** It becomes convenient for users as two users can have same name for different files or may have different name for same file.
  - **Grouping:** Logical grouping of files can be done by properties e.g. all java programs, all games etc.

FileDirectories

File Directories:

Collection of files is a file directory. The directory contains information about the files, including attributes, location and ownership.

- file is a collection of logically related entities.
  - Name
  - Type
  - Address
  - Current length
  - Maximum length
  - Date last accessed
  - Date last updated
  - Owner id
  - Protection information

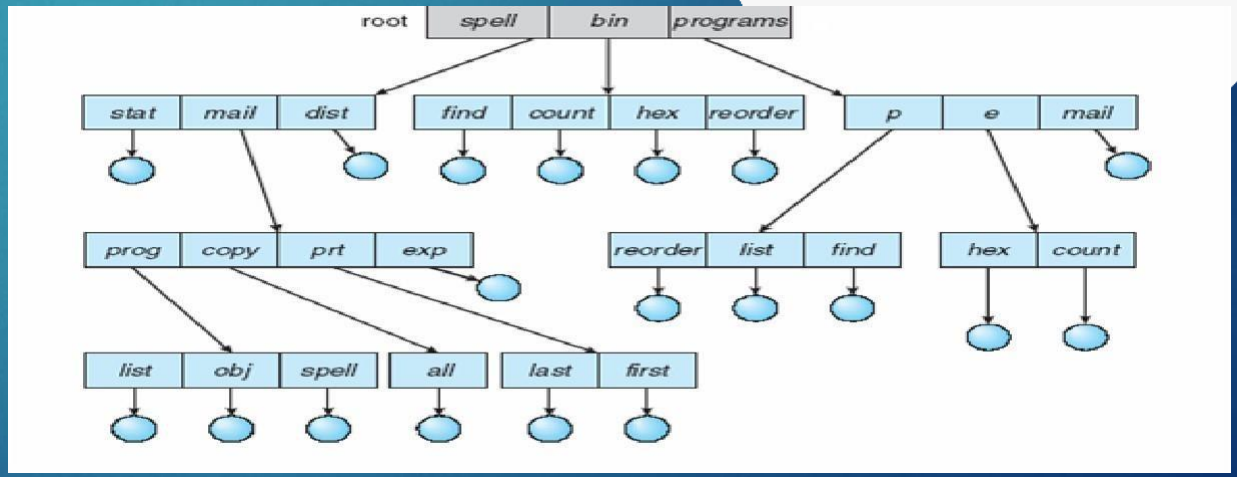


FileDirectories

File Directories Types:

Collection of files is a file directory. The directory contains information about the files, including attributes, location and ownership.

- Various types of directories are:
  - Single-level Directory.
  - Two-level Directory.
  - Tree-structured Directory.
  - Graph Directory.

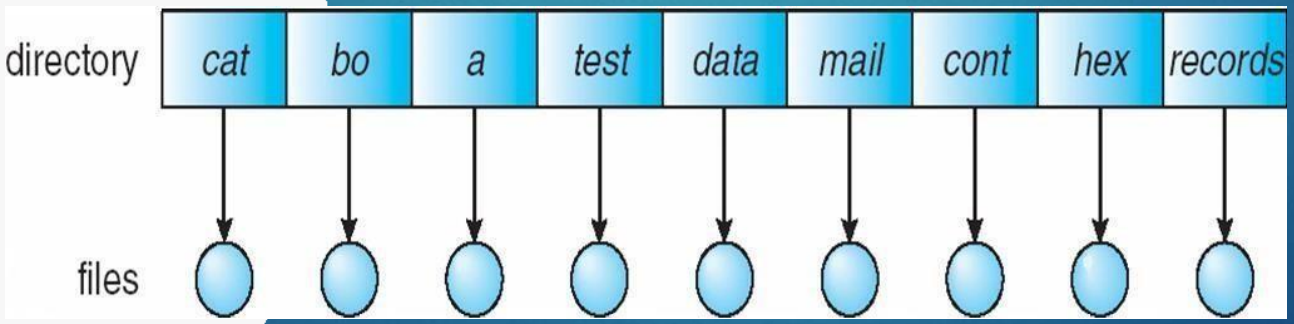


FileDirectories

Single-level Directory:

In this a single directory is maintained for all the users.

- Naming problem: Users cannot have same name for two files.
- Grouping problem: Users cannot group files according to their need:

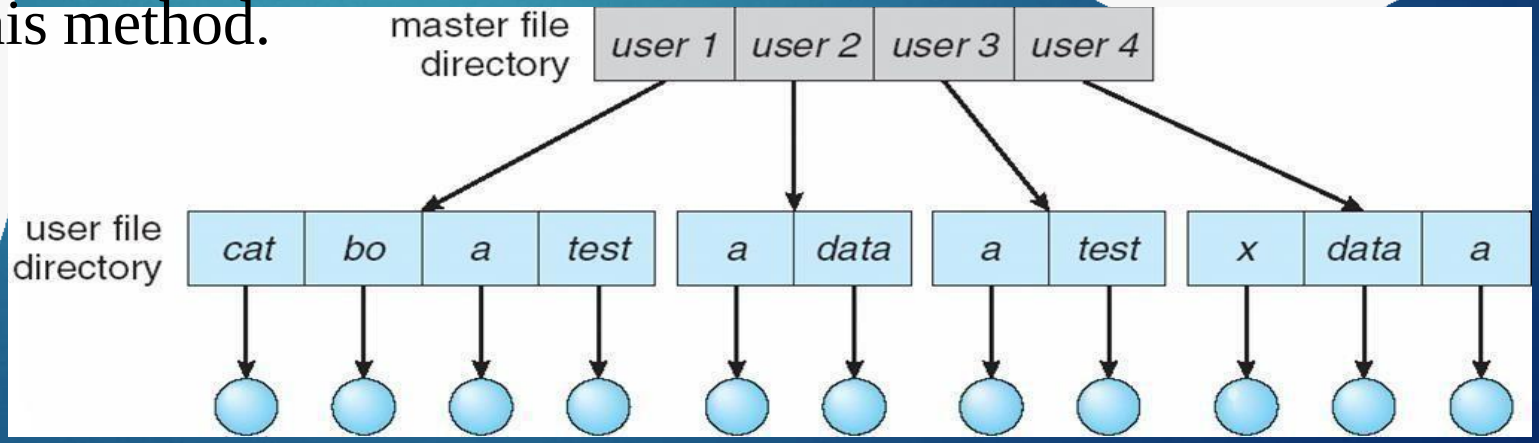


FileDirectories

Two-level Directory:

In this separate directories for each user is maintained.

- Path name: Due to two levels there is a path name for every file to locate that file.
- Now, we can have same file name for different user.
- Searching is efficient in this method.
- No grouping capability

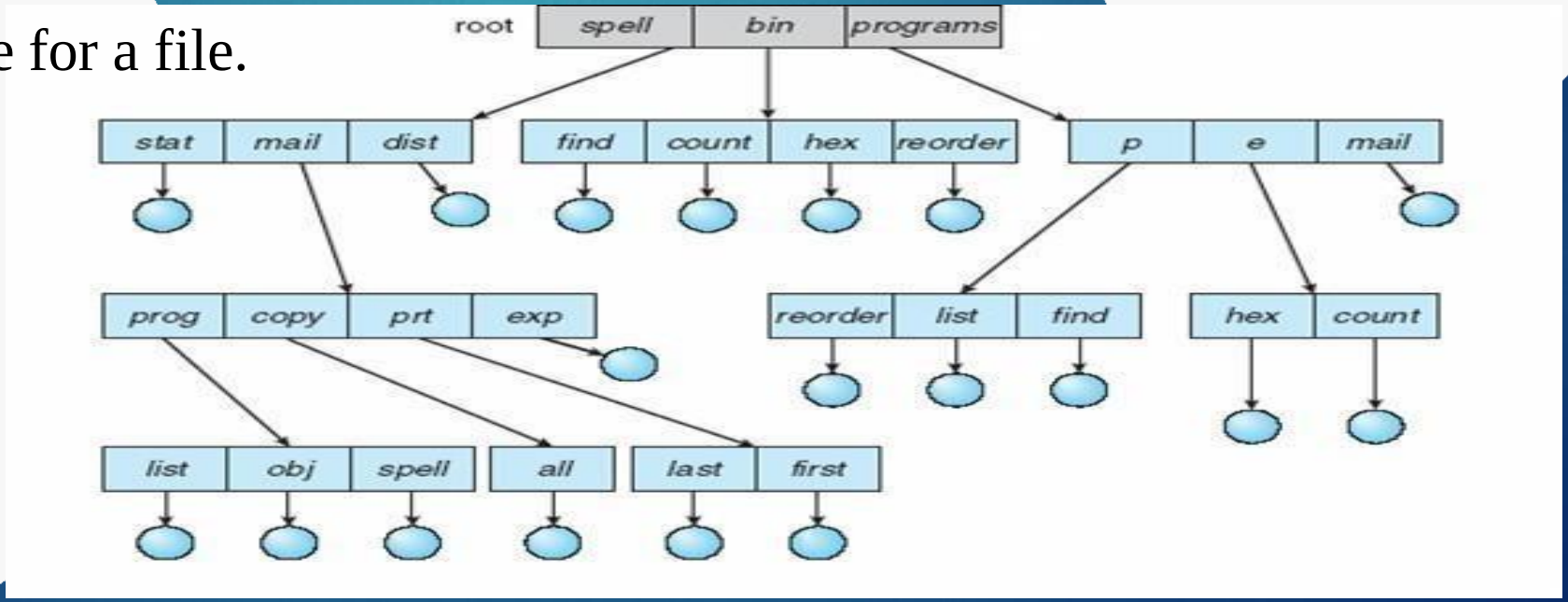


FileDirectories

Tree-structured Directory:

Directory is maintained in the form of a tree.

- Searching is efficient and also there is grouping capability. We have absolute or relative path name for a file.

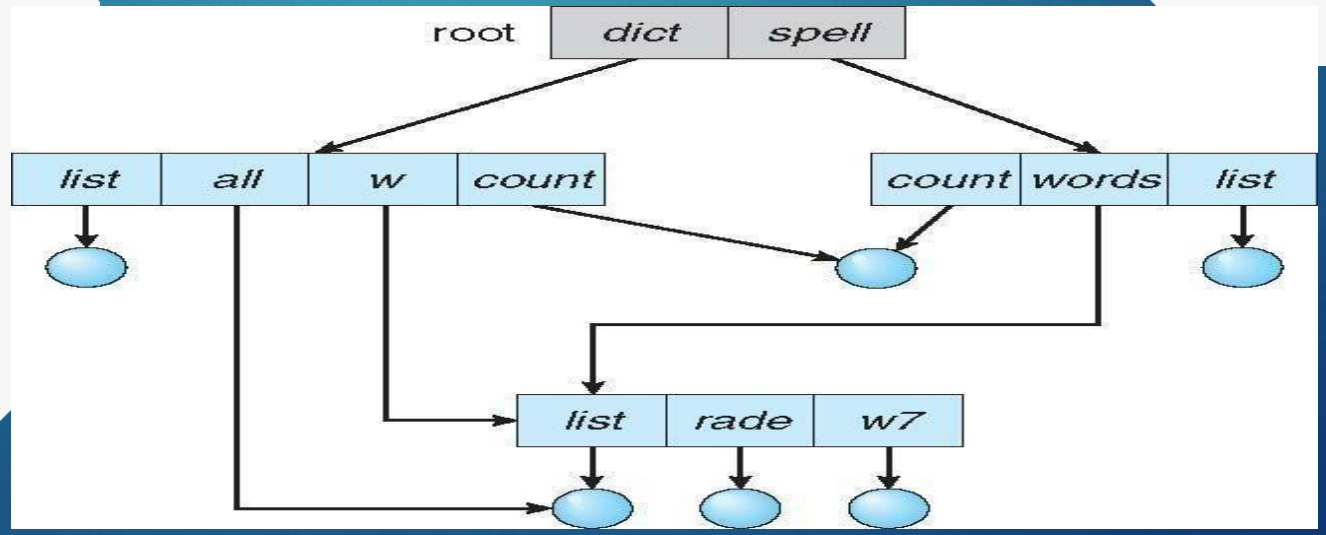


FileDirectories

Graph Directory:

Directory is maintained in the form of a graph.

- Searching is efficient and also there is grouping capability.
- Allow only links to file not subdirectories
- Every time a new link is added use a cycle detection algorithm to determine whether it is OK
- Garbage collection



FileSharing

File Sharing:

Specify how multiple users are to access a shared file simultaneously.

- Sharing of files on multi-user systems is desirable
- Sharing may be done through a protection scheme
- On distributed systems, files may be shared across a network
- Network File System (NFS) is a common distributed file-sharing method
- If multi-user system
  - User IDs identify users, allowing permissions and protections to be per-user Group IDs allow users to be in groups, permitting group access rights
  - Owner of a file / directory
  - Group of a file / directory

File Sharing

File Sharing – Remote File Systems:

Specify how multiple users on networking to allow file system access between systems..

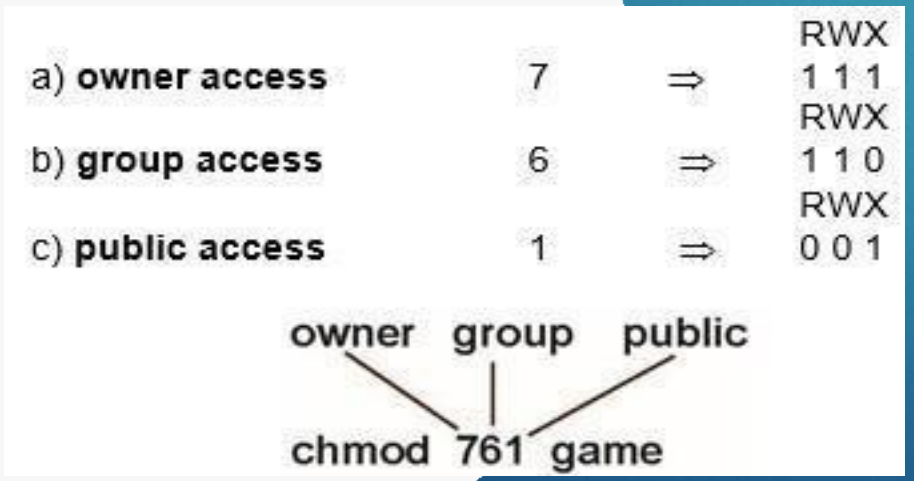
- Manually via programs like FTP automatically, seamlessly using distributed file systems
- Semi automatically via the world wide web
- Client-server model allows clients to mount remote file systems from servers
- Server can serve multiple clients
- Client and user-on-client identification is insecure or complicated
- Standard operating system file calls are translated into remote calls
- Distributed Information Systems such as LDAP, DNS, NIS, Active Directory implement unified access to information needed for remote computing

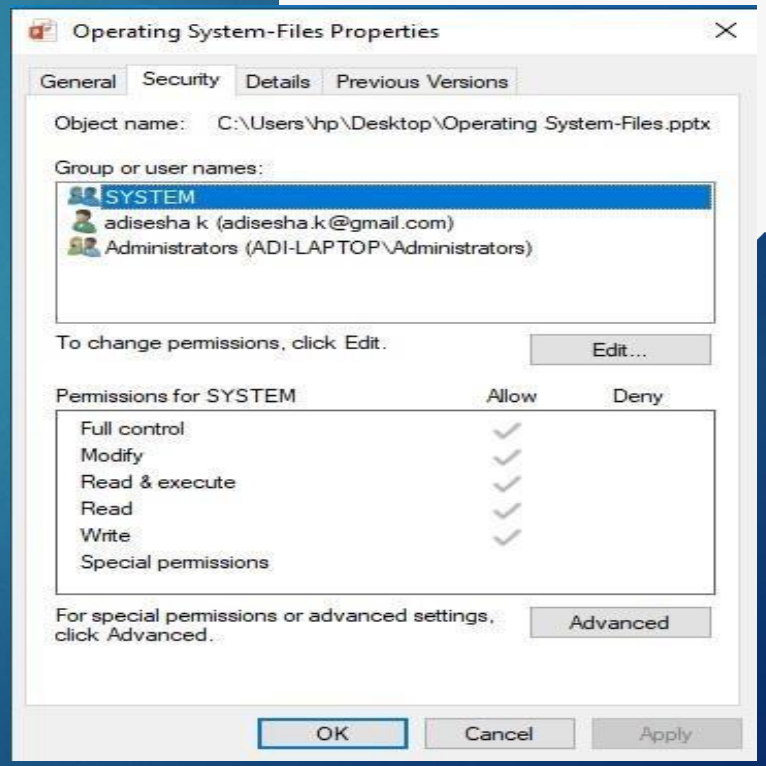
FileProtection

File Protection:

Specify how file owner/creator should be able to specify control the usage of files by other users such as what can be done and by whom.

- Mode of access: read, write, execute
- Three classes of users on Unix / Linux





Storage management

Storage management:

Storage management deals with the storage procedures in the computer system using an operating system.

- Disks are the mainly used mass storage devices. They provide the bulk of secondary storage in operating systems today
- Various mass devices are:
  - Magnetic Tapes
  - Magnetic Disks
  - Solid-State Disks

Storage management

Magnetic Tapes:

Magnetic tapes were once used for common secondary storage before the days of hard disk drives, but today are used primarily for backups.

- Accessing a particular spot on a magnetic tape can be slow, but once reading or writing commences, access speeds are comparable to disk drives.
- Capacities of tape drives can range from 20 to 200 GB, and compression can double that capacity.

32 Storage management

Storage management

Magnetic Disks:

One or more platters in the form of disks covered with magnetic media. Hard disk platters are made of rigid metal, while "floppy" disks are made of more flexible plastic.

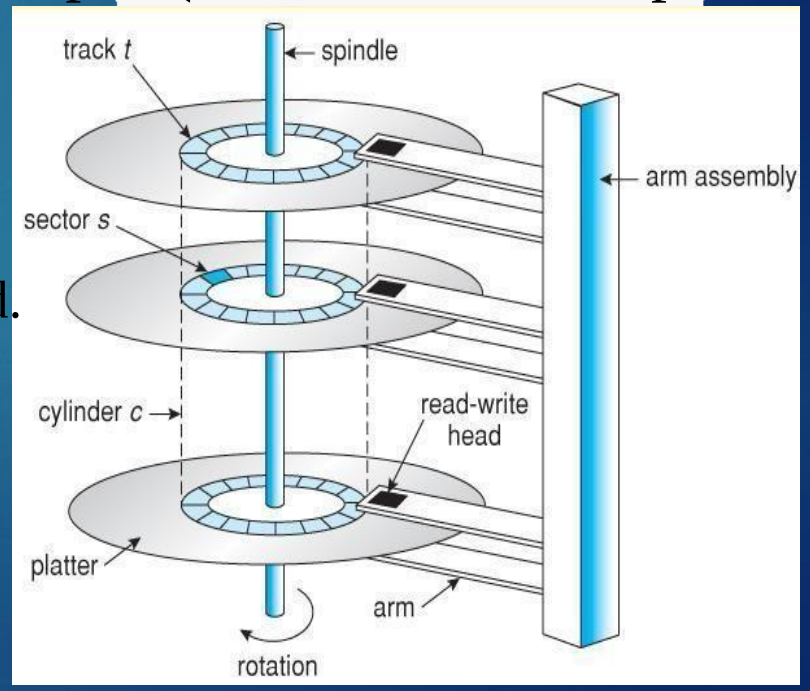
- Floppy disks and Hard drives are normally removable mass devices.
- Traditional magnetic disks have the following basic structure:
  - Each platter has two working surfaces.
  - Each working surface is divided into a number of concentric rings called tracks.
  - Each track is further divided into sectors, traditionally containing 512 bytes of data each.
  - The data on a hard drive is read by read-write heads.
  - The storage capacity of a traditional disk drive is equal to the number of heads.

Storage management

Magnetic Disks:

Traditional magnetic disks have the following basic structure.

- In operation the disk rotates at high speed, such as 7200 rpm (120 revolutions per second.).
  - The seek time or random access time is the time required to move the heads from one cylinder to another.
  - The rotational latency is the amount of time required for the desired sector to rotate around and come under the read-write head.
  - The transfer rate, which is the time required to move the data electronically from the disk to the computer.



34 Storagemanagement

Solid-State Disks (SSD):

SSDs use memory technology as a small fast hard disk. Specific implementations may use either flash memory or DRAM chips protected by a battery to sustain the information through power cycles.

- Because SSDs have no moving parts they are much faster than traditional hard drives, and certain problems such as the scheduling of disk accesses simply do not apply.
- SSDs are especially useful as a high-speed cache of hard-disk information that must be accessed quickly.
- They are more expensive than hard drives, generally not as large, and may have shorter life spans.
- SSDs are also used in laptops to make them smaller, faster, and lighter.

Disk Scheduling

Disk Scheduling:

The operating system is responsible for using hardware efficiently - for the disk drives, this means having a fast access time and disk bandwidth.

- Various Disk-Scheduling Algorithm are:
  - FCFS Scheduling
  - SSTF Scheduling
  - SCAN Scheduling
  - C-SCAN Scheduling
  - LOOK Scheduling

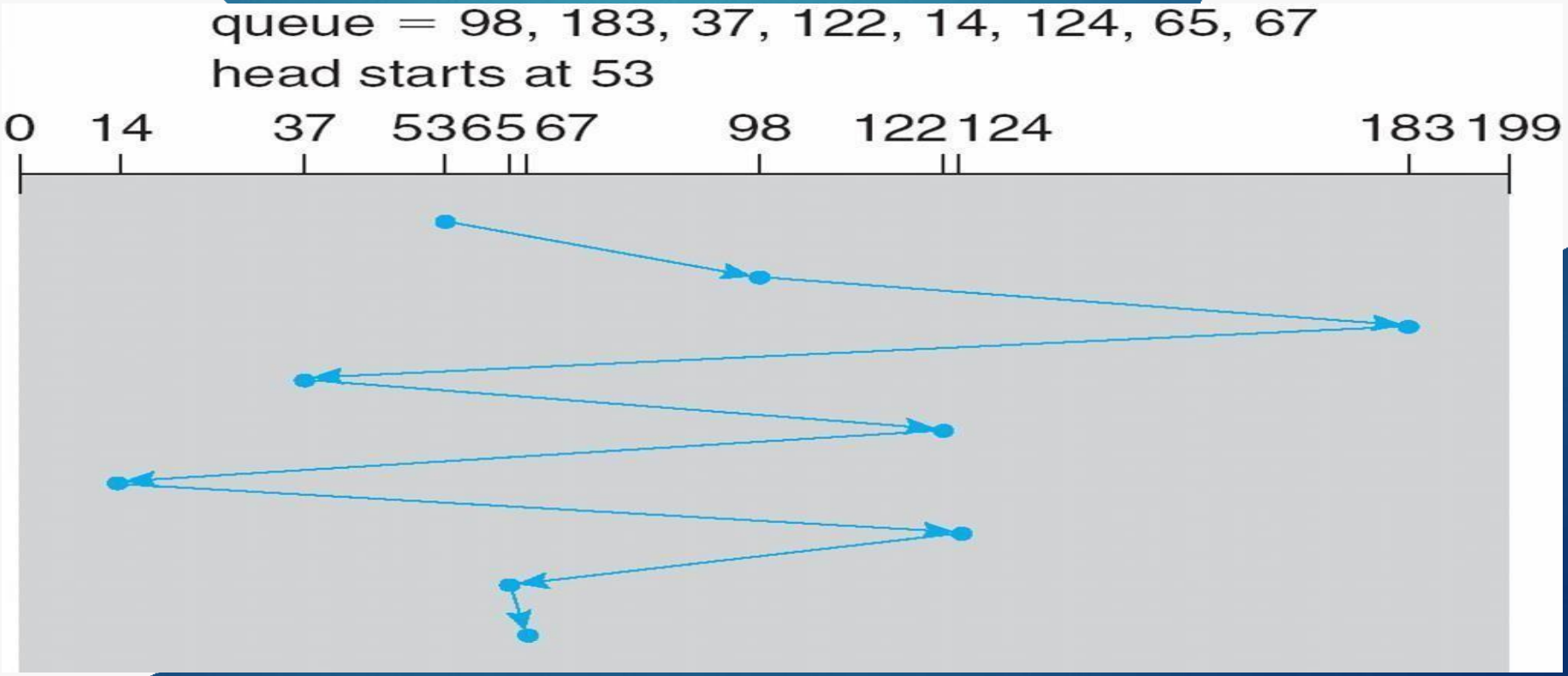
We illustrate scheduling algorithms with a request queue (0-199)

98, 183, 37, 122, 14, 124, 65, 67
Head pointer 53

DiskScheduling

FCFS Scheduling:

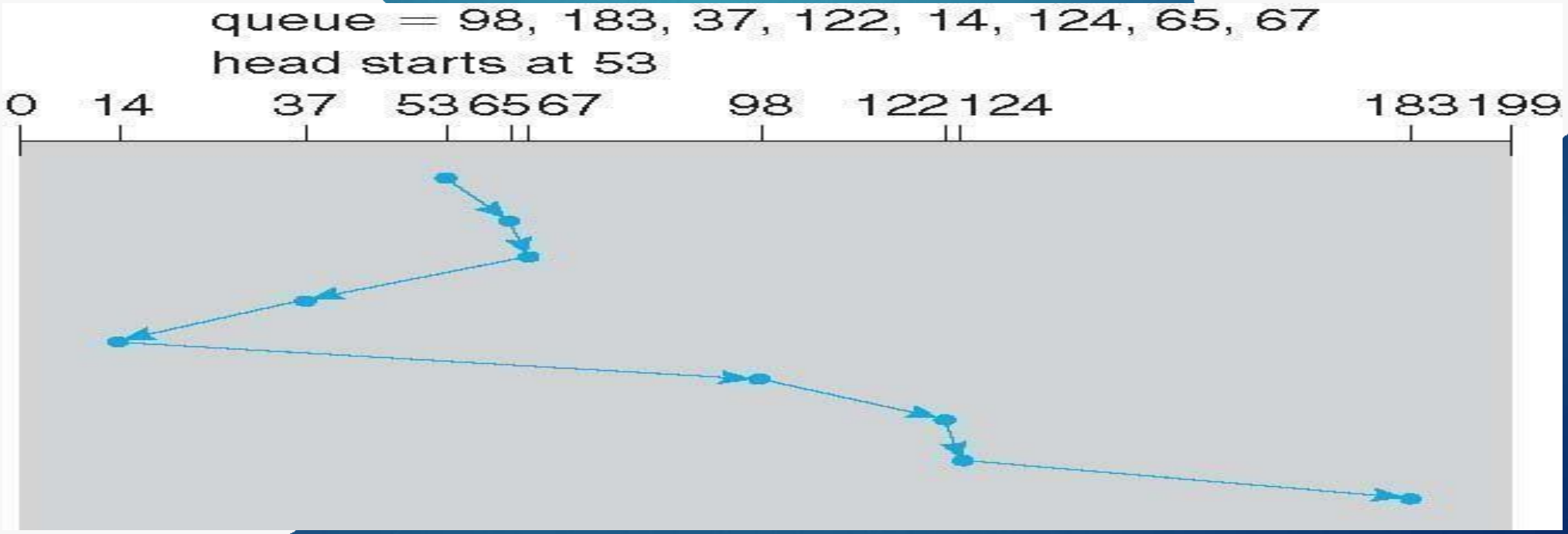
First-Come First-Serve is simple and intrinsically fair, but not very efficient..



DiskScheduling

SSTF Scheduling:

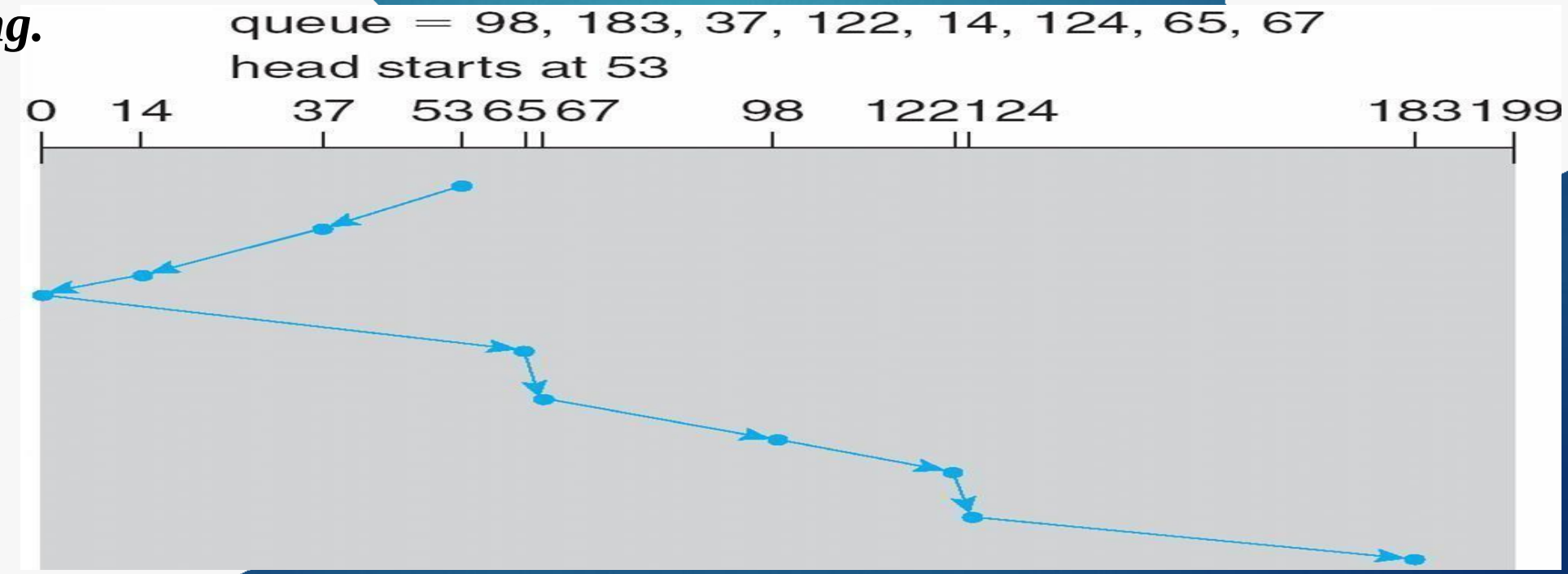
Shortest Seek Time First scheduling is more efficient, but may lead to starvation if a constant stream of requests arrives for the same general area of the disk.



DiskScheduling

SCAN Scheduling:

SCAN algorithm Sometimes called the elevator algorithm moves back and forth from one end of the disk to the other, similarly to an elevator processing requests in a tall building.

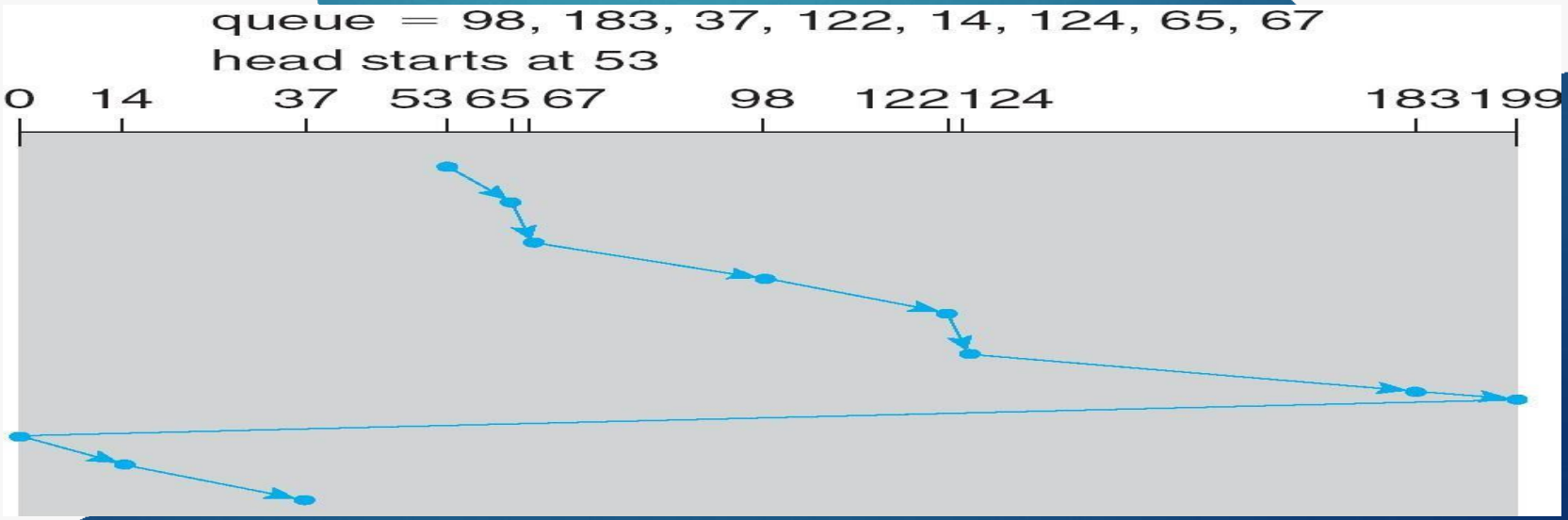


DiskScheduling

C-SCAN Scheduling:

The Circular-SCAN algorithm improves upon SCAN by treating all requests in a circular queue fashion .

- Once the head reaches the end of the disk, it returns to the other end without processing any requests, and then starts again from the beginning of the disk.



DiskScheduling

LOOK Scheduling:

LOOK scheduling improves upon SCAN by looking ahead at the queue of pending requests, and not moving the heads any farther towards the end of the disk than is necessary.