PITHAPUR RAJAH'S GOVERNMENT COLLEGE

DEPARTMENT OF COMPUTER APPLICATIONS

II BCOM(MINOR) SEMISTER-IV

SUB:OPERATING SYSTEM **P.V. MAHESWARI**

Lecturer in computer applications

Unit 1

Introduction: What is Operating System? ,History and Evolution of OS, Basic OS Functions, Computer System Architecture, Operating System Structure.

System Structures: Operating System Services, User Operating System Interface, System Calls, Types of System Calls, Overview of UNIX Operating System, Basic Features of Unix Operating System.

Case Study :

1. Understanding and listing the basic differences between UNIX OS and Windows OS in usage, user interface, features etc.

Unit II

Process Management: Process Concept, Operation on Processes, Communication in Client-Server Systems. Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, CPU Scheduling in UNIX.

Case Study:

1. Present your understanding on how CPU Scheduling is different in WINDOWS compared to

UNIX/LINUX.

Unit III

Synchronization: Process Synchronization, Semaphores: Usage, Implementation, The Critical Section Problem,Classic problems of synchronization.

Deadlocks: Introduction, Deadlock Characterization, Necessary and Sufficient conditions for Deadlock,Deadlock Handling Approaches : Deadlock prevention, Deadlock Avoidance and Deadlock detection and

Recovery

Case Study:

1. Present your understanding of Deadlocks and new methodologies available in new Operating Systems

released in the market.

Unit IV

Memory Management: Overview, Swapping, Contiguous Memory Allocation, Paging, Paging Examples, Segmentation, Page Replacement Algorithms, Memory management in UNIX.

Case Study:

1. Present a paper on new methods used in Memory management in the present day Operating Systems

Unit V

Files and Directories in UNIX: Files, Directory Structure, File Operations, File System Implementation:File Allocation Methods, Comparison of UNIX and Windows.

Case Study:

1. Present a Paper on how UNIX treats regular files and directories differently from other operating
systems.

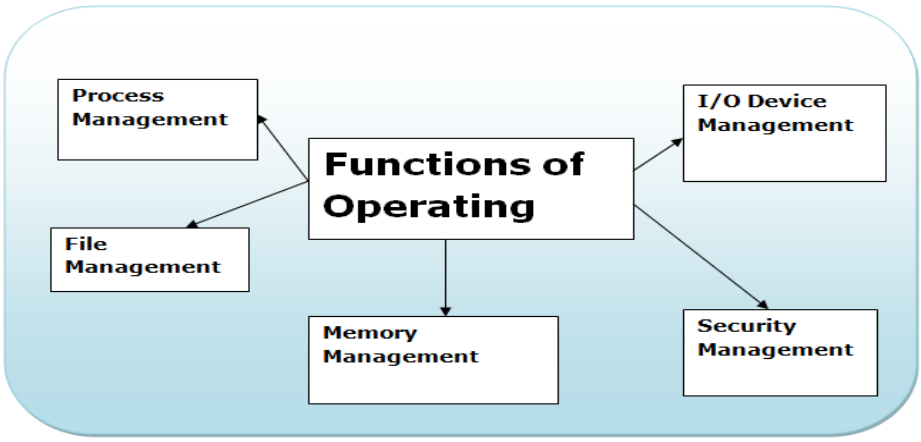
UNIT-1

Essay Questions:

1. What is Operating system? Explain function of Operating System?

A: An Operating System (OS) is an interface between a user and computer hardware. An operating system is a system software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

Functions of Operating System



1. File Management

An operating system's (OS) primary function is to manage files and folders. Operating systems are responsible for managing the files on a computer. Thisincludes creating, opening, closing, and deleting files. The operating system is alsoresponsible for organizing the files on the disk.

2.Device management

Operating systems provide essential functions for managing devices connected to acomputer. These functions include allocating memory, processing input and outputrequests, and managing storage devices. This device could be a keyboard, mouse, printer, or any other devices you may have connected.

3.Process management

The operating system's responsibility is to manage the processes running on your computer. This includes starting and stopping programs, allocating resources, and managing memory usage.

4.Memory management

One of the most critical functions of an operating system is memorymanagement.

- Allocating memory to store programs.
- Deciding the amount of memory that should be allocated to the program.
- Memory distribution while multiprocessing.

5.Job Accounting:

An operating system's (OS) job accounting feature is a powerful tool for tracking how your computer's resources are being used. This information can help you pinpoint and troubleshoot any performance issues and identify unauthorized software installations.

6.Security –

For security, modern operating systems employ a firewall. A firewall is a type of security system that monitors all computer activity and blocks it if it detects a threat.

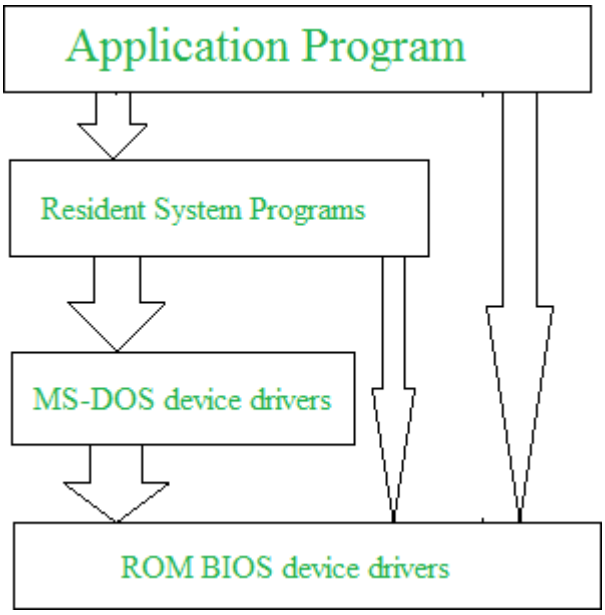
2.Explain about Operating System Structure?

A: Operating system can be implemented with the help of various structures. The structure of the OS depends mainly on how the various common components of the operating system are interconnected and melded into the kernel. Depending on this we have following structures of the operating system:

Simple structure:

Such operating systems do not have well defined structure and are small, simple and limited systems. The interfaces and levels of functionality are not well separated. MS-DOS is an example of such operating system. In MS-DOS application programs are able to access the basic I/O routines. These types of operating system cause the entire system to crash if one of the user programs fails.

Diagram of the structure of MS-DOS is shown below.

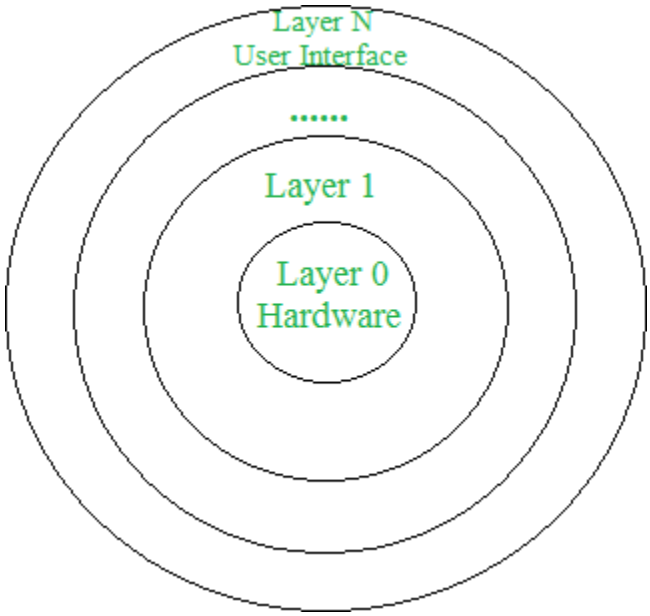


Layered structure:

An OS can be broken into pieces and retain much more control on system. In this structure the OS is broken into number of layers (levels). The bottom layer (layer 0) is the hardware and the topmost layer (layer N) is the user interface. These layers are so designed that each layer uses the functions of the lower level layers only.

This simplifies the debugging process as if lower level layers are debugged and an error occurs during debugging then the error must be on that layer only as the lower level layers have already been debugged.

The main disadvantage of this structure is that at each layer, the data needs to be modified and passed on which adds overhead to the system. Moreover careful planning of the layers is necessary as a layer can use only lower level layers. UNIX is an example of this structure.



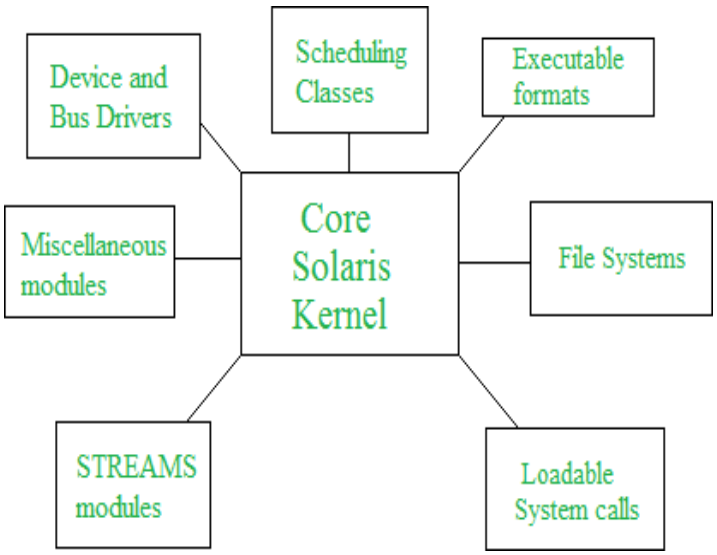
Micro-kernel:

This structure designs the operating system by removing all non-essential components from the kernel and implementing them as system and user programs. This result in a smaller kernel called the micro-kernel. Advantages of this structure are that all new services need to be added to user space and does not require the kernel to be modified. Thus it is more secure and reliable as if a service fails then rest of the operating system remains untouched. Mac OS is an example of this type of OS.

Modular structure or approach:

It is considered as the best approach for an OS. It involves designing of a modular kernel. The kernel has only set of core components and other services are added as dynamically loadable modules to the kernel either during run time or boot time. It resembles layered structure due to the fact that each kernel has defined and protected interfaces but it is more flexible than the layered structure as a module can call any other module.

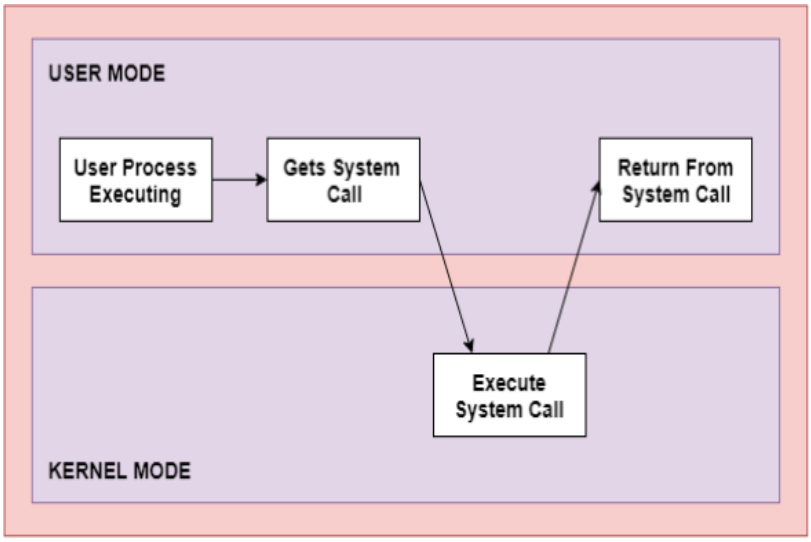
For example Solaris OS is organized as shown in the figure.



3.Explain about System Call in detail?

A: The interface between a process and an operating system is provided by system calls. In general, system calls are available as assembly language instructions. They are also included in the manuals used by the assembly level programmers. System calls are usually made when a process in user mode requires access to a resource. Then it requests the kernel to provide the resource via a system call.

A figure representing the execution of the system call is given as follows −



As can be seen from this diagram, the processes execute normally in the user mode until a system call interrupts this. Then the system call is executed on a priority basis in the kernel mode. After the execution the system call, the control returns to the user mode and execution of user processes can be resumed.

In general, system calls are required in the following situations

- If a file system requires the creation or deletion of files. Reading and writing from files also require a system call.
- Creation and management of new processes.
- Network connections also require system calls. This includes sending and receiving packets.
- Access to a hardware devices such as a printer, scanner etc. requires a system call.

Types of System Calls

There are mainly five types of system calls. These are explained in detail as follows

- **Process Control** These system calls deal with processes such as process creation process termination etc.
- **File Management :** These system calls are responsible for file manipulation such creating a file, reading a file, writing into a file etc.
- **Device Management:** These system calls are responsible for device manipulation such as reading from device buffers, writing into device buffers etc.
- I**nformation Maintenance:** These system calls handle information and its transfer between the operating system and the user program.
- **Communication:** These system calls are useful for interprocess communicatio They also deal with creating and deleting a communication connection.

Short Questions:

1.Write about Operating System Services?

A. 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. Following are a few common services provided by an operating system −
- Program execution
- I/O operations
- File System manipulation
- Communication
- Error Detection
- Resource Allocation
- Protection

Program execution

Operating systems handle many kinds of activities from user programs to system programs like printer spooler, name servers, file server, etc. Each of these activities is encapsulated as a process.

- 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.

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.

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. Examples of storage media include magnetic tape, magnetic disk and optical disk drives like CD, DVD. Each of these media has its own properties like speed, capacity, data transfer rate and data access methods.

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. Multiple processes communicate with one another through communication lines in the network.

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.

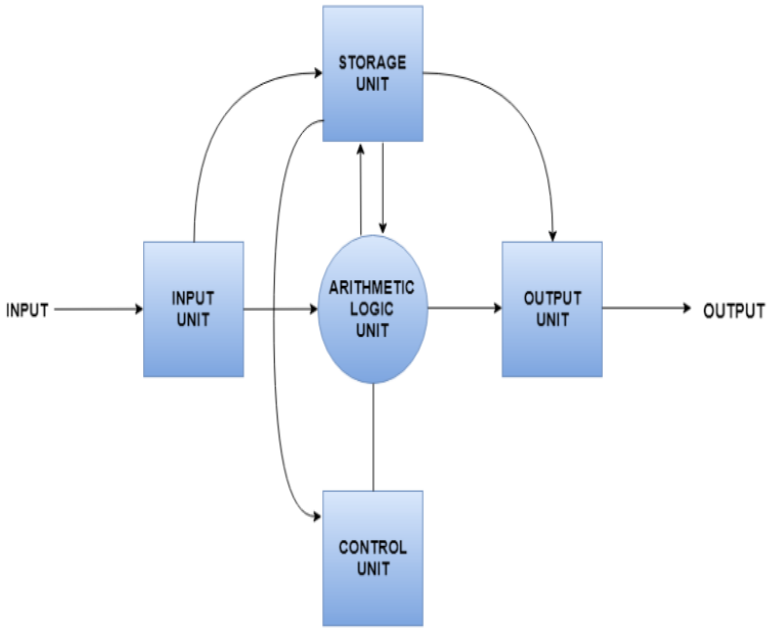
Resource Management

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

2.Explain about Computer System Architecture?

A. A computer system is basically a machine that simplifies complicated tasks. It should maximize performance and reduce costs as well as power consumption. The different components in the Computer System Architecture are Input Unit, Output Unit, Storage Unit, Arithmetic Logic Unit, Control Unit etc.

A diagram that shows the flow of data between these units is as follows −



The input data travels from input unit to ALU. Similarly, the computed data travels from ALU to output unit. The data constantly moves from storage unit to ALU and back again. This is because stored data is computed on before being stored again. The control unit controls all the other units as well as their data.

Input Unit

The input unit provides data to the computer system from the outside. So, basically it links the external environment with the computer. It takes data from the input devices, converts it into machine language and then loads it into the computer system. Keyboard, mouse etc. are the most commonly used input devices.

Output Unit

The output unit provides the results of computer process to the users i.e it links the computer with the external environment. Most of the output data is the form of audio or video. The different output devices are monitors, printers, speakers, headphones etc.

Storage Unit

Storage unit contains many computer components that are used to store data. It is traditionally divided into **primary storage and secondary storage**.

Arithmetic Logic Unit

All the calculations related to the computer system are performed by the arithmetic logic unit. It can perform operations like addition, subtraction, multiplication, division etc.

Control Unit

This unit controls all the other units of the computer system and so is known as its central nervous system. It transfers data throughout the computer as required including from storage unit to central processing unit and vice versa.

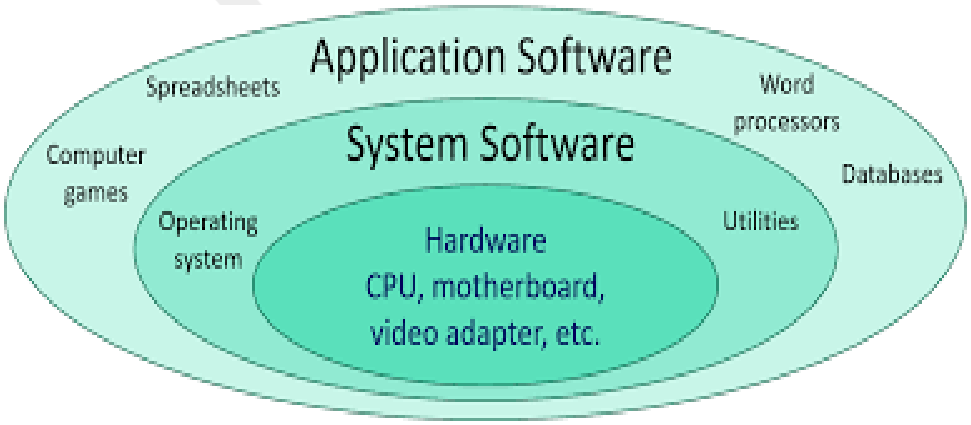
3.What is operating System?

A. An Operating System (OS) is an interface between a user and computer hardware. An operating system is a system software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

- **Generally, a Computer System consists of the following components**:
- **Computer Users** are the users who use the overall computer system.

Application Softwares are the softwares which users use directly to perform different activities. These softwares are simple and easy to use like Browsers, Word, Excel, different Editors, Games etc.

- **System Softwares** are the softwares which are more complex in nature and they are more near to computer hardware.**Operating Systems** (Microsoft Windows, macOS, and Linux).
- **computer Hardware** includes Monitor, Keyboard, CPU, Disks, Memory, etc.



Objectives of Operating System

- To make the computer system convenient to use in an efficient manner
- .To hide the details of the hardware resources from the users.
- To provide users a convenient interface to use the computer system.
- 4.To act as an intermediary between the hardware and its users, making iteasier for the users to access and use other resources.
- 5.To manage the resources of a computer system.
- 6.To provide efficient and fair sharing of resources among users and programs.

UNIT-2

Essay Questions

1.Write about Process Scheduling Algorithms?

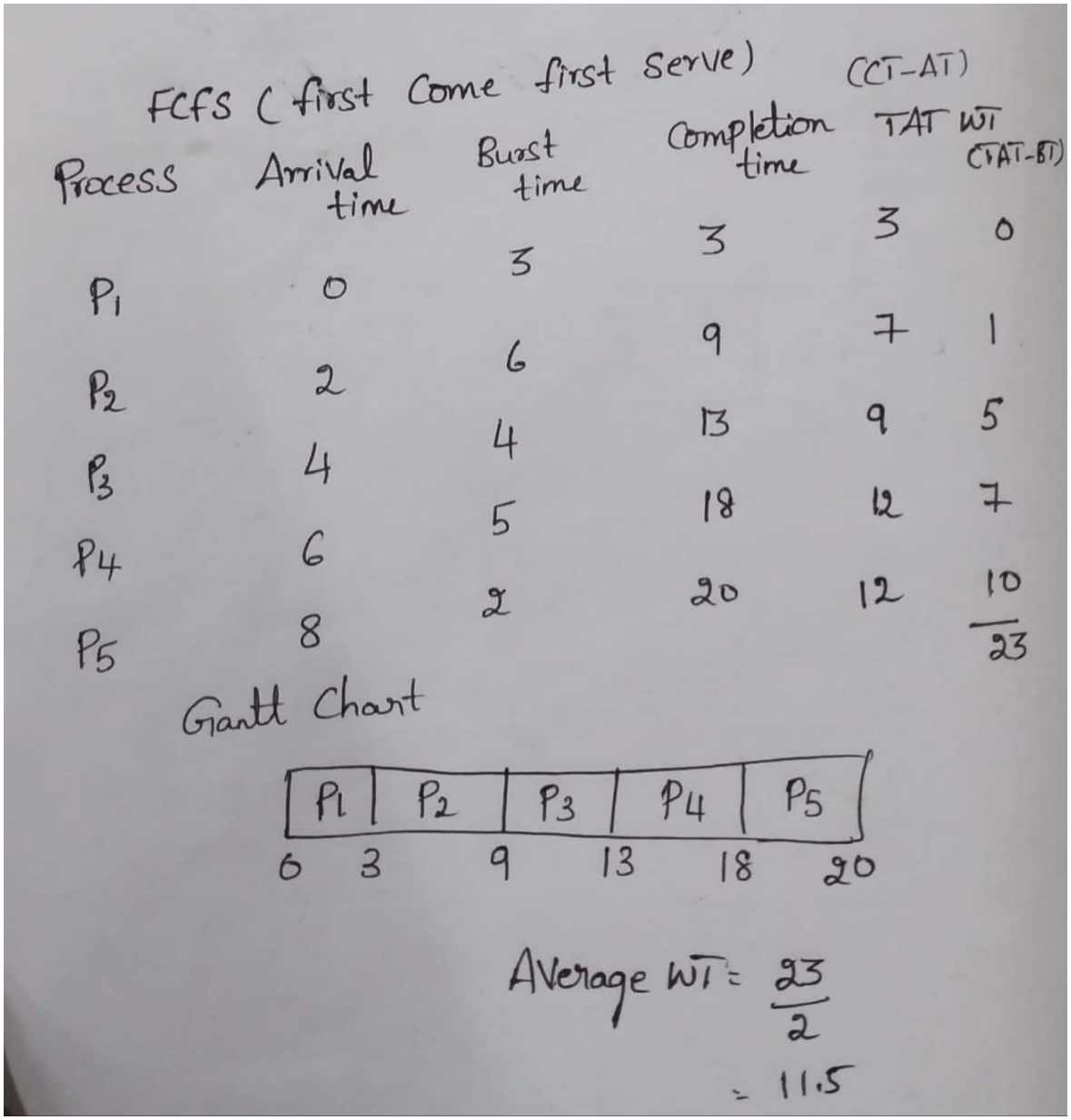
A A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms. There are six popular process scheduling algorithms

- First-Come, First-Served (FCFS) Scheduling Algorithm
- Shortest-Job-Next (SJN) Scheduling Algorithm
- Priority Scheduling Algorithm
- Round Robin(RR) Scheduling Algorithm

These algorithms are either **non-preemptive or preemptive**. Non-preemptive algorithms are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time, whereas the preemptive scheduling is based on priority where a scheduler may preempt a low priority running process anytime when a high priority process enters into a ready state.

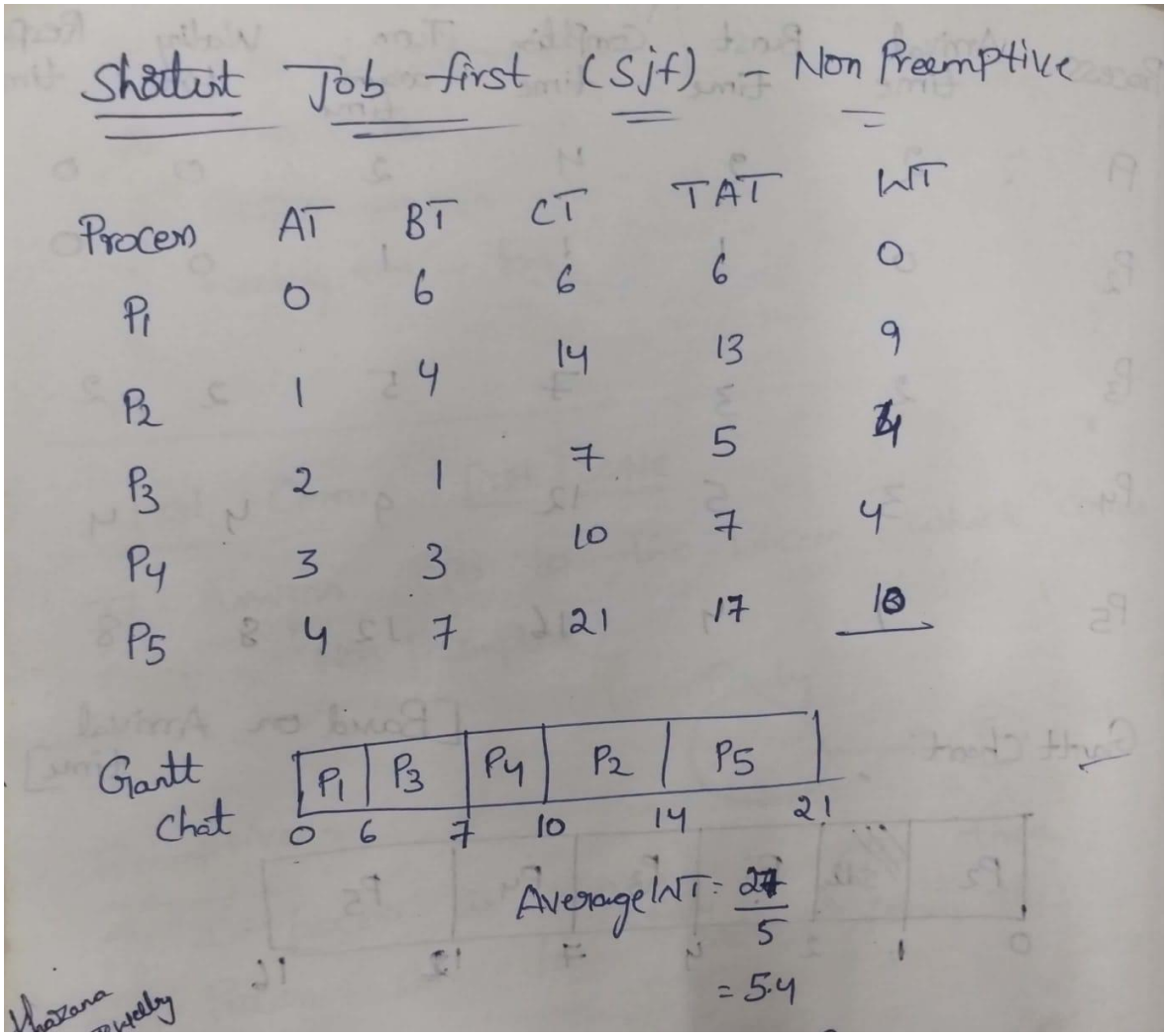
First Come First Serve (FCFS)

- 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.



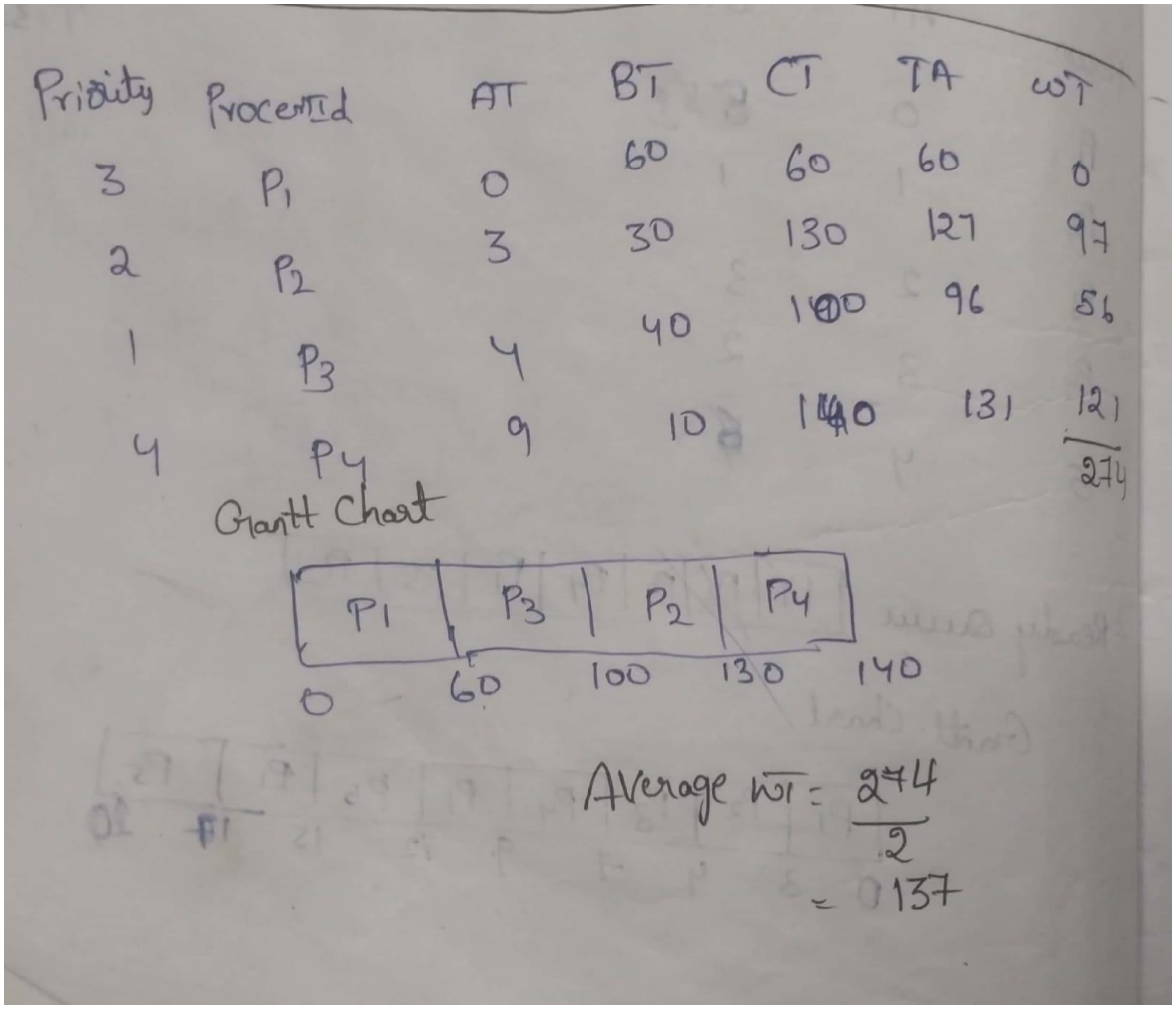
Shortest Job First(SJf)

- This is also known as **shortest job first**, or SJF
- 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.



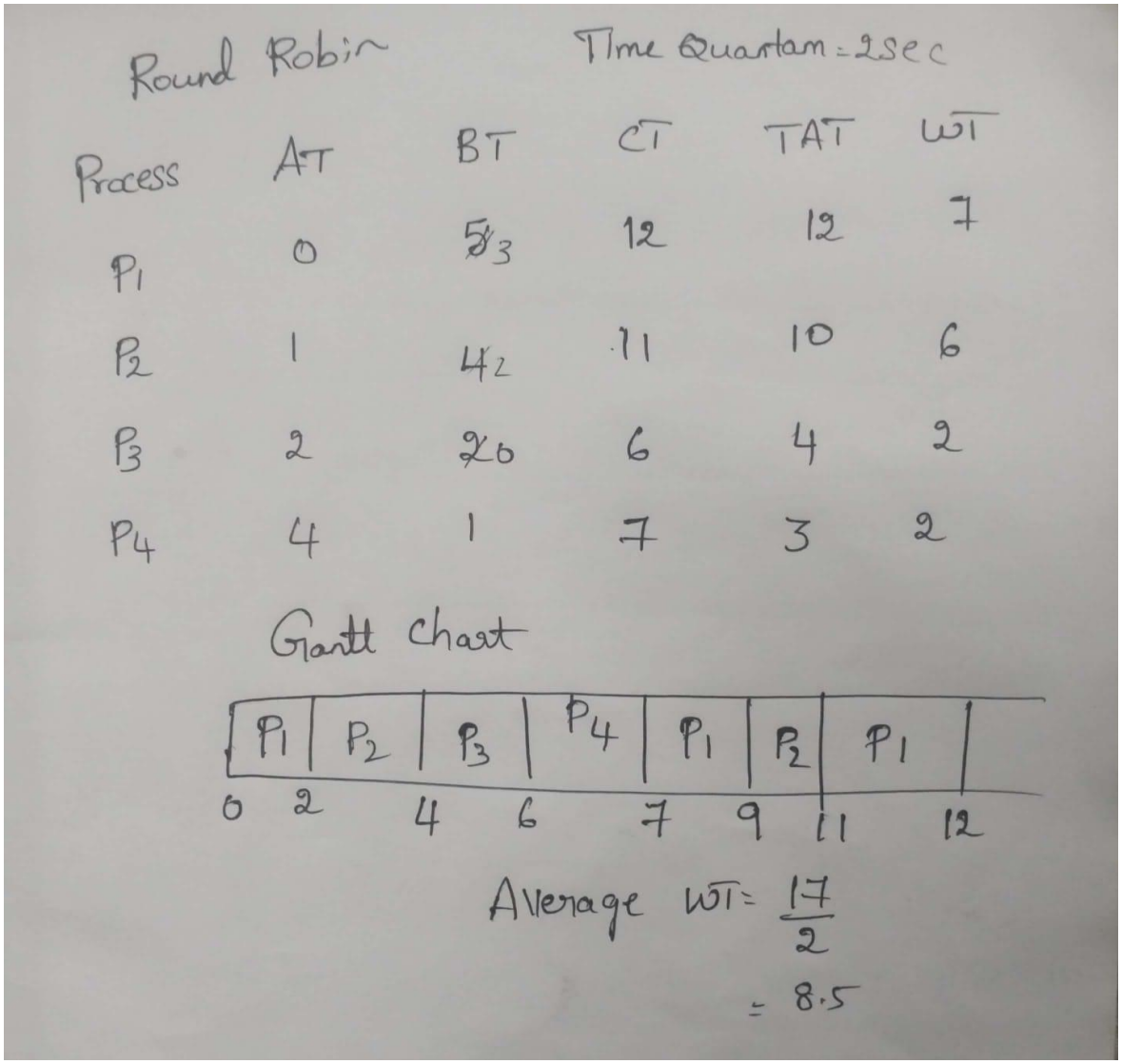
Priority Scheduling Algorithm

- Priority scheduling is a non-preemptive 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



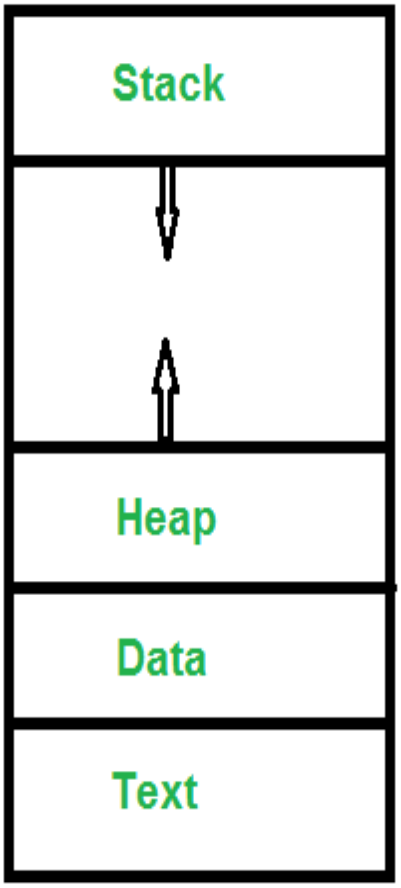
Round Robin Scheduling

- 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.



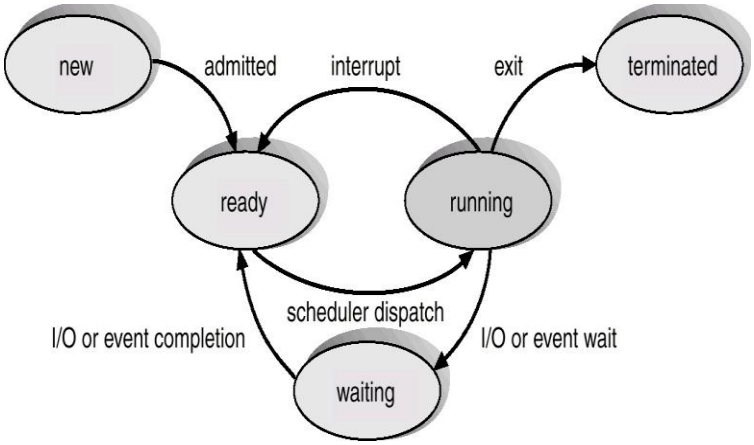
2.Explain about Process Concept in detail?

A. A process is an activity of executing a program. It is a program under execution. Every process needs certain resources to complete its task. Processes are the programs that are dispatched from the ready state and are scheduled in the CPU for execution. PCB (Process Control Block) holds the context of the process. A process can create other processes which are known as Child Processes. The process takes more time to terminate, and it is isolated means it does not share the memory with any other process. The process can have the following states new, ready, running, waiting, terminated, and suspended.



- **Text** : A Process, sometimes known as the Text Section, also includes the current activity represented by the value of the Program Counter .
- **Stack** : The stack contains temporary data, such as function parameters, returns addresses, and local variables.
- **Data** : Contains the global variable.
- **Heap** : Dynamically memory allocated to process during its run time.
- **operations on process:**
- **New:** This state represents a newly created process that hasn't started running yet. It has not been loaded into the main memory, but its process control block (PCB) has been created, which holds important information about the process.
- **Ready:** A process in this state is ready to run as soon as the CPU becomes available. It is waiting for the operating system to give it a chance to execute.

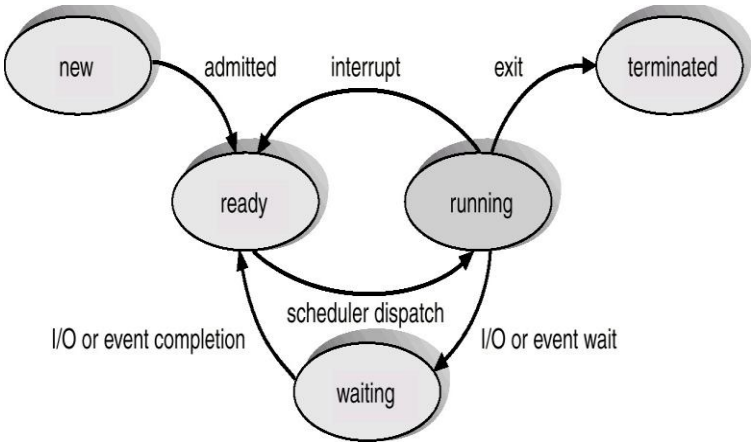
- **Running**: This state means the process is currently being executed by the CPU. Since we're assuming there is only one CPU, at any time, only one process can be in this state.
- **Blocked/Waiting:** This state means the process cannot continue executing right now. It is waiting for some event to happen, like the completion of an input/output operation (for example, reading data from a disk).
- **Exit/Terminate:** A process in this state has finished its execution or has been stopped by the user for some reason. At this point, it is released by the operating system and removed from memory.



Short Answer Questions

1.What is Process? Explain Process State diagram in OS?

A. a process in an operating system (OS) is a program that's currently running, like a specific instance of an application or task, rather than just the program code itself.



- **New:** This state represents a newly created process that hasn't started running yet. It has not been loaded into the main memory, but its process control block (PCB) has been created, which holds important information about the process.
- **Ready:** A process in this state is ready to run as soon as the CPU becomes available. It is waiting for the operating system to give it a chance to execute.
- **Running**: This state means the process is currently being executed by the CPU. Since we're assuming there is only one CPU, at any time, only one process can be in this state.
- **Blocked/Waiting:** This state means the process cannot continue executing right now. It is waiting for some event to happen, like the completion of an input/output operation (for example, reading data from a disk).
- **Exit/Terminate:** A process in this state has finished its execution or has been stopped by the user for some reason. At this point, it is released by the operating system and removed from memory.
- **2.Write about Process Scheduling?**

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

  Process scheduling is an essential part of a Multiprogramming operating systems. Such operating systems allow more than one process to be loaded into the executable memory at a time and the loaded process shares the CPU using time multiplexing.

Process Scheduling Queues

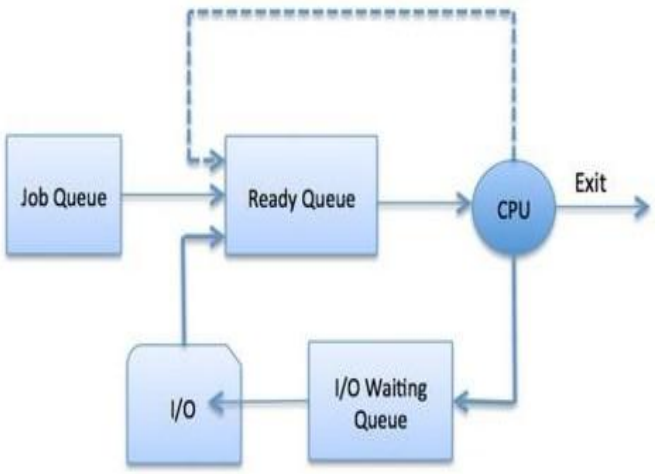
The OS maintains all PCBs in Process Scheduling Queues. The OS maintains a separate queue for each of the process states and PCBs of all processes in the same execution state are placed in the same queue. When the state of a process is changed, its PCB is unlinked from its current queue and moved to its new state queue.

The Operating System maintains the following important 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. A new process is always put in this queue.

Device queues − The processes which are blocked due to unavailability of an I/O device constitute this queue.



The OS can use different policies to manage each queue (FIFO, Round Robin, Priority, etc.). The OS scheduler determines how to move processes between the ready and run queues which can only have one entry per processor core on the system; in the above diagram, it has been merged with the CPU.

3.Describe CPU Scheduling?

A. CPU scheduling is a process used by the operating system to decide which task or process gets to use the CPU at a particular time. This is important because a CPU can only handle one task at a time, but there are usually many tasks that need to be processed. The following are different purposes of a CPU scheduling time.

- Maximize the CPU utilization

- Minimize the response and waiting time of the process.

Terminologies Used in CPU Scheduling

- **Arrival Time:** The time at which the process arrives in the ready queue.
- **Completion Time:** The time at which the process completes its execution.
- **Burst Time:** Time required by a process for CPU execution.
- **Turn Around Time:** Time Difference between completion time and arrival time.

Different Types of CPU Scheduling Algorithms

There are mainly two types of scheduling methods:

Preemptive Scheduling: Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to the ready state.

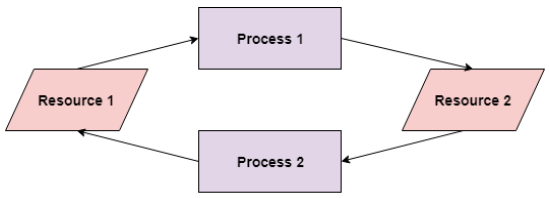
Non-Preemptive Scheduling: Non-Preemptive scheduling is used when a process terminates , or when a process switches from running state to waiting state.

UNIT-3

Essay Questions

1 Explain about Deadlock concept in detail?

A. A deadlock happens in operating system when two or more processes need some resource to complete their execution that is held by the other process.



Deadlock in Operating System

In the above diagram, the process 1 has resource 1 and needs to acquire resource 2. Similarly process 2 has resource 2 and needs to acquire resource 1. Process 1 and process 2 are in deadlock as each of them needs the other's resource to complete their execution but neither of them is willing to relinquish their resources.

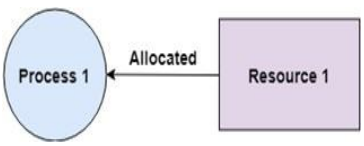
Coffman Conditions

A deadlock occurs if the four Coffman conditions hold true. But these conditions are not mutually exclusive.

The Coffman conditions are given as follows

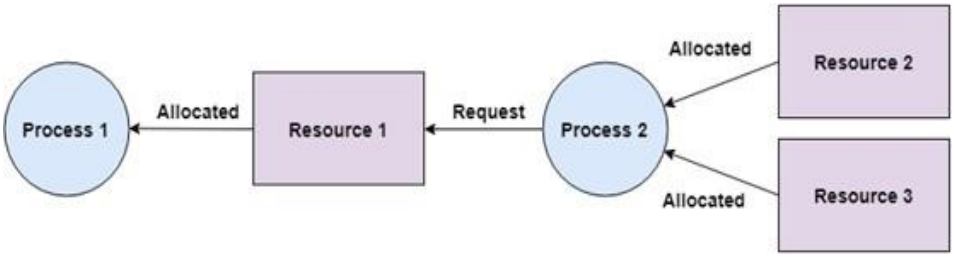
Mutual Exclusion

There should be a resource that can only be held by one process at a time. In the diagram below, there is a single instance of Resource 1 and it is held by Process 1 only.



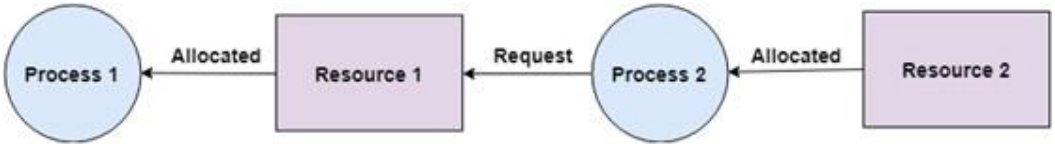
Hold and Wait

A process can hold multiple resources and still request more resources from other processes which are holding them. In the diagram given below, Process 2 holds Resource 2 and Resource 3 and is requesting the Resource 1 which is held by Process 1.



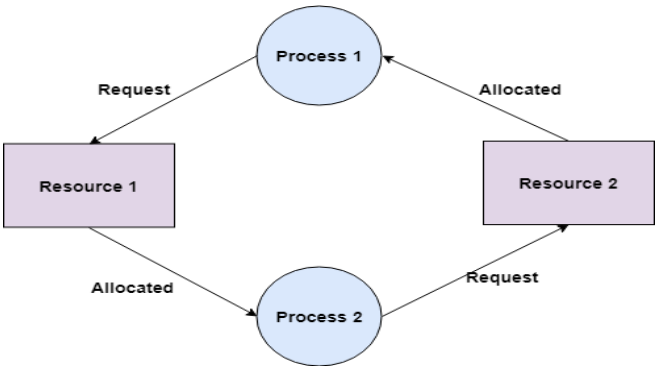
No Preemption

A resource cannot be preempted from a process by force. A process can only release a resource voluntarily. In the diagram below, Process 2 cannot preempt Resource 1 from Process1. It will only be released when Process 1 relinquishes it voluntarily after its execution is complete.



Circular Wait

All the processes must be waiting for the resources in a cyclic manner so that the last process is waiting for the resource which is being held by the first process.



Deadlock Detection

A deadlock can be detected by a resource scheduler as it keeps track of all the resources that are allocated to different processes. After a deadlock is detected, it can be resolved using the following methods −

- All the processes that are involved in the deadlock are terminated. This is not a good approach as all the progress made by the processes is destroyed.
- Resources can be preempted from some processes and given to others till the deadlock is resolved.

Deadlock Prevention

It is very important to prevent a deadlock before it can occur. So, the system checks each transaction before it is executed to make sure it does not lead to deadlock. If there is even a slight chance that a transaction may lead to deadlock in the future, it is never allowed to execute.

Deadlock Avoidance

It is better to avoid a deadlock rather than take measures after the deadlock has occurred. The wait for graph can be used for deadlock avoidance. This is however only useful for smaller databases as it can get quite complex in larger databases.

Topic: 2.2.1

2. Explain about Classical Problems of Synchronization?

A Semaphore can be used in other synchronization problems besides Mutual Exclusion.

Below are some of the classical problem depicting flaws of process synchronaization in systems where cooperating processes are present.

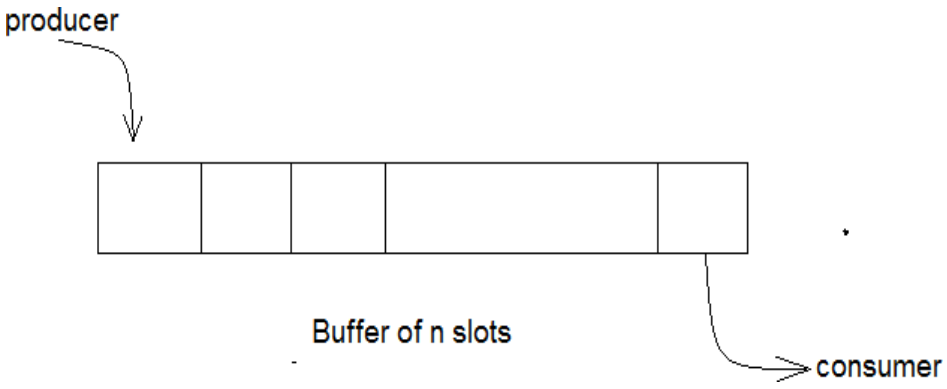
We will discuss the following three problems:

- Bounded Buffer (Producer-Consumer) Problem
- Dining Philosophers Problem
- The Readers Writers Problem

Bounded Buffer Problem

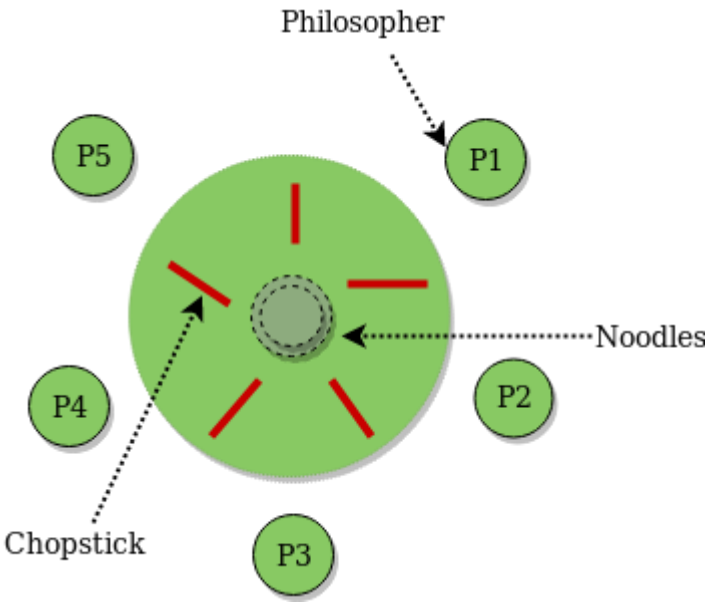
- This problem is generalised in terms of the **Producer Consumer problem**, where a **finite** buffer pool is used to exchange messages between producer and consumer processes.

- Because the buffer pool has a maximum size, this problem is often called the **Bounded buffer 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.



Dining Philosopher Problem

The Dining Philosopher Problem states that 5 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-Writers Problem

Suppose that a database is to be shared among several concurrent processes. Some of these processes may want only to read the database, whereas others may want to update (that is, to read and write) the database. 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.

3.Explain about Deadlock Detection and recovery?

Deadlock Detection and Recovery is the mechanism of detecting and resolving deadlocks in an operating system. In operating systems, deadlock recovery is important to keep everything running smoothly. A deadlock occurs when two or more processes are blocked, waiting for each other to release the resources they need.

Deadlock Detection

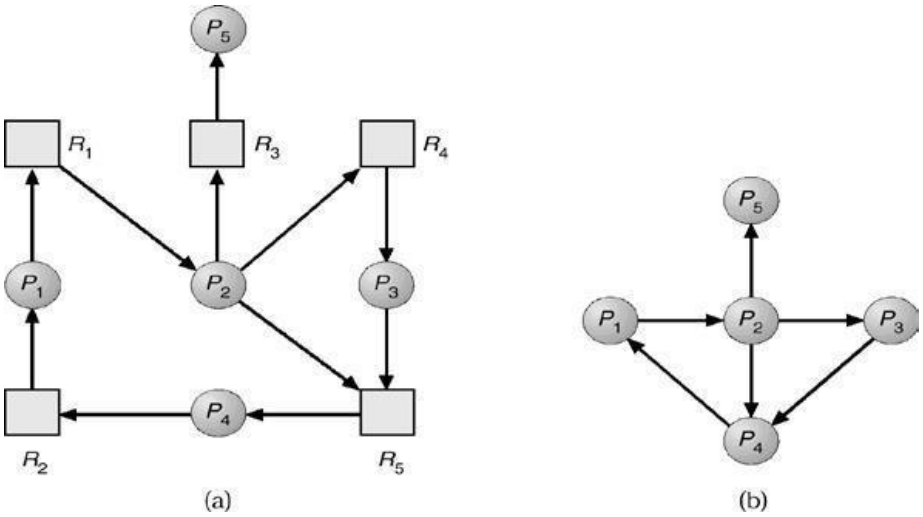
If a system does not employ either a deadlock-prevention or a deadlock avoidance algorithm, then a deadlock situation may occur. In this environment, the system must provide:

- An algorithm that examines the state of the system to determine whether a deadlock has occurred.
- An algorithm to recover from the deadlock.

According to number of instances in each resource type, the Deadlock Detection algorithm can be classified into two categories as follows:

1. **Single Instance of Each Resource Type:** If all resources have only a single instance, then it can define a deadlock detection algorithm that uses a variant of the resource-allocation graph (is called a *wait-for* graph).

A wait–for graph can be draw by removing the nodes of type resource and collapsing the appropriate edges from the resource-allocation graph.



2.Multiple Instances of Resources

Detection of the cycle is necessary but not a sufficient condition for deadlock detection, in this case, the system may or may not be in deadlock varies according to different situations.

For systems with multiple instances of resources, algorithms like Banker's Algorithm can be adapted to periodically check for deadlocks.

Detection recovery

When a detection algorithm determines that a deadlock exists, then the system or operator is responsible for handling deadlock problem. There are two options for breaking a deadlock.

- Process Termination
- Resource preemption

Process Termination

There are two method to eliminate deadlocks by terminating a process as follows:

- **Abort all deadlocked processes:** This method will break the deadlock cycle clearly by terminating all process. This method is cost effective. And it removes the partial computations completed by the processes.
- **Abort one process at a time until the deadlock cycle is eliminated:** This method

terminates one process at a time, and invokes a deadlock-detection algorithm to determine whether any processes are still deadlocked.

Resource Preemption

In resource preemption, the operator or system preempts some resources from processes and give these resources to other processes until the deadlock cycle is broken.If preemption is required to deal with deadlocks, then three issues need to be addressed:

- **Selecting a victim:** The system or operator selects which resources and which processes are to be preempted based on cost factor.
- **Rollback:** The system or operator must roll back the process to some safe state and restart it from that state.
- **Starvation:** The system or operator should ensure that resources will not always be preempted from the same process?

Short Answer Questions

1.Write about Semaphores?

A Semaphores are integer variables that are used to solve the critical section problemby using two atomic operations, wait and signal that are used for process synchronization.

The definitions of wait and signal are as follows −

- ***Wait***

  The wait operation decrements the value of its argument S, if it is positive. If S is negative or zero, then no operation is performed.

- ***Signal***

  The signal operation increments the value of its argument S.signal(S)

Types of Semaphores

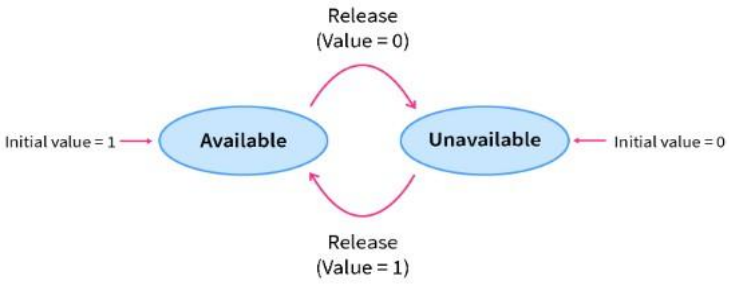
1.Binary Semaphores

2.Counting Semaphores

Binary Semaphores

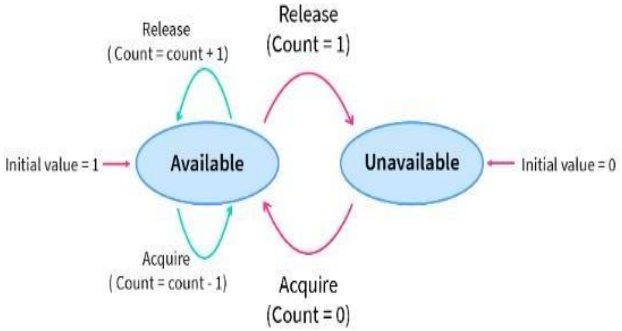
In these types of Semaphores the integer value of the semaphore can only beeither 0 or 1. If the value of the Semaphore is 1, it means that the process can proceed to the critical section.

if the value of the binary semaphore is 0, then theprocess cannot continue to the critical section of the code.



Counting Semaphores

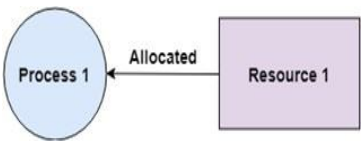
Counting semaphores are signaling integers that can take on any integer value



2.Explain about Deadlock Characterization?

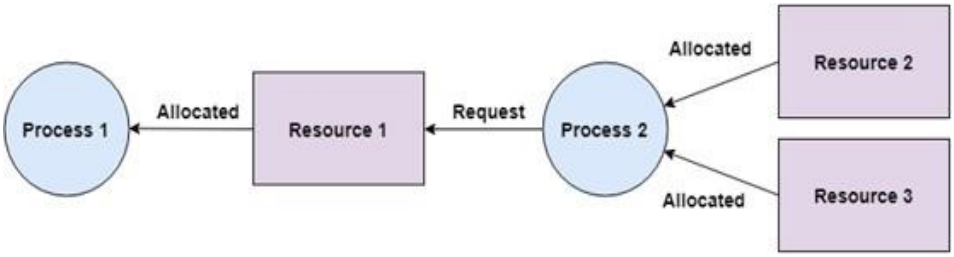
A Mutual Exclusion

There should be a resource that can only be held by one process at a time. In the diagram below, there is a single instance of Resource 1 and it is held by Process 1 only.



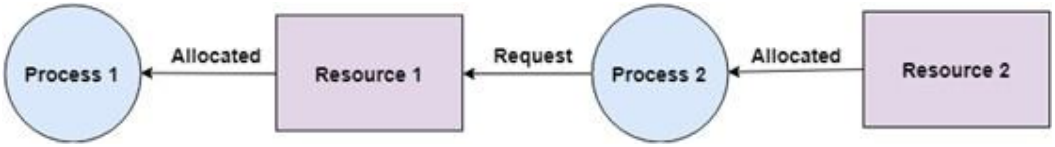
Hold and Wait

A process can hold multiple resources and still request more resources from other processes which are holding them. In the diagram given below, Process 2 holds Resource 2 and Resource 3 and is requesting the Resource 1 which is held by Process 1.



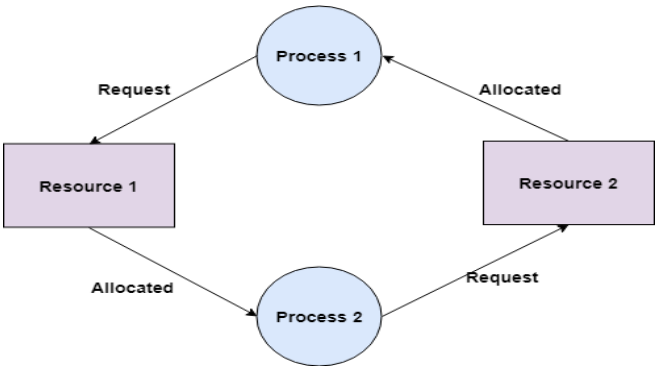
No Preemption

A resource cannot be preempted from a process by force. A process can only release a resource voluntarily. In the diagram below, Process 2 cannot preempt Resource 1 from Process1. It will only be released when Process 1 relinquishes it voluntarily after its execution is complete.



Circular Wait

All the processes must be waiting for the resources in a cyclic manner so that the last process is waiting for the resource which is being held by the first process.



3.Describe about Reader-Writer problem?

A Suppose that a database is to be shared among several concurrent processes. Some of these processes may want only to read the database, whereas others may want to update (that is, to read and write) the database. 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.

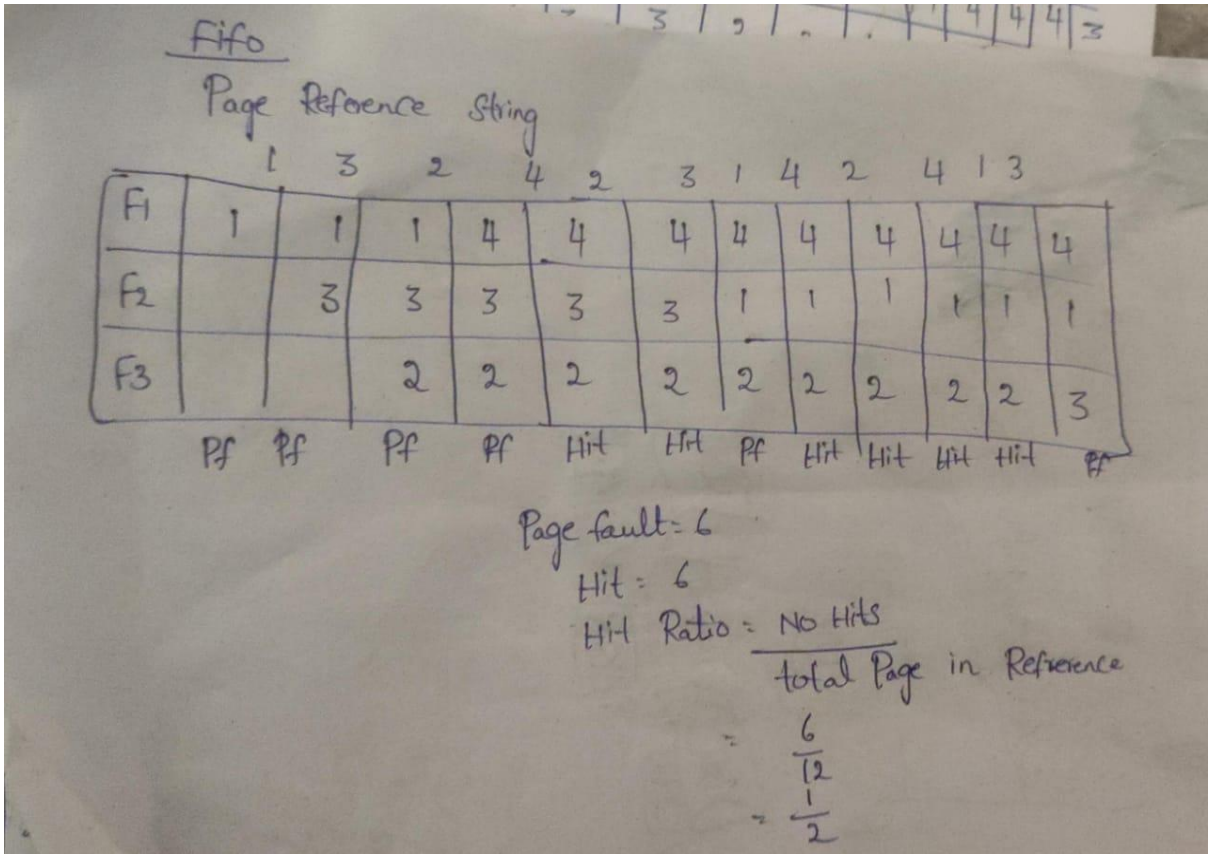
UNIT-4

1.Explain about Page replacement Algorithm?

A. Pagereplacement algorithms are techniques used in operating systems to manage memory efficiently when the physical memory is full. When a new page needs to be loaded into physical memory, and there is no free space, these algorithms determine which existing page to replace**.**

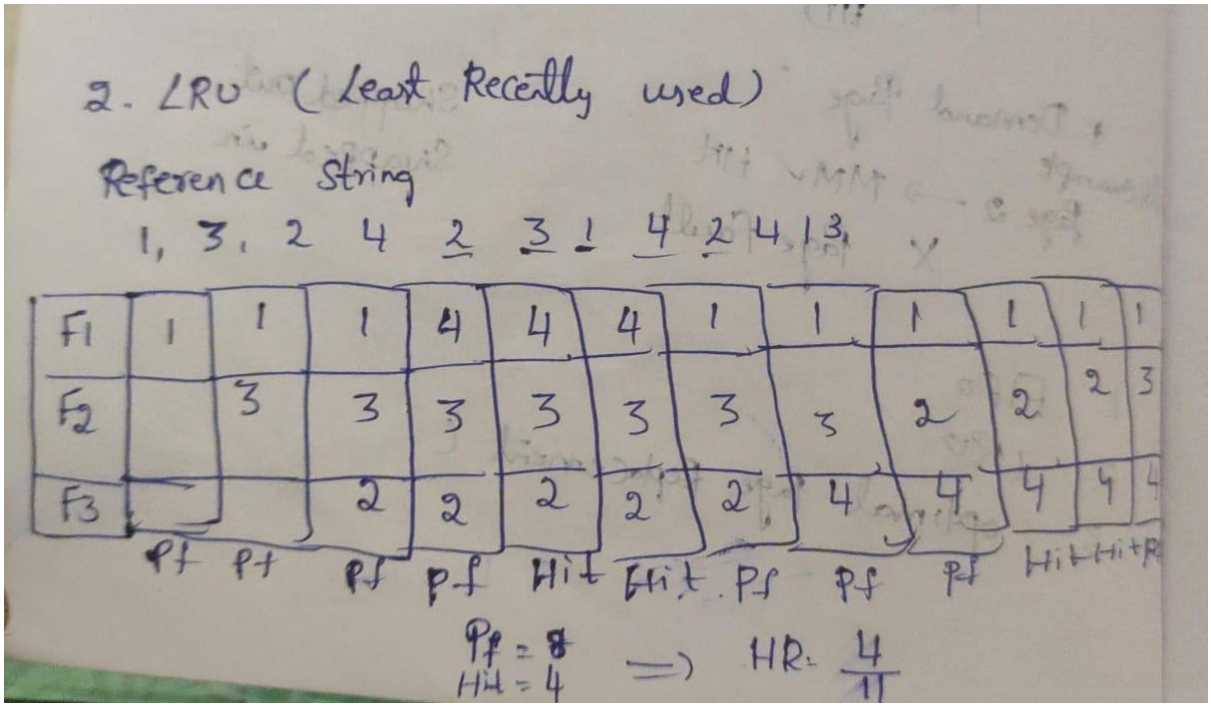
common Page Replacement Techniques

- First In First Out (FIFO)
- Least Recently Used (LRU)
- Optimal Page replacement
- **First In First Out (FIFO)**
  This is the simplest page replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a page needs to be replaced page in the front of the queue is selected for removal.



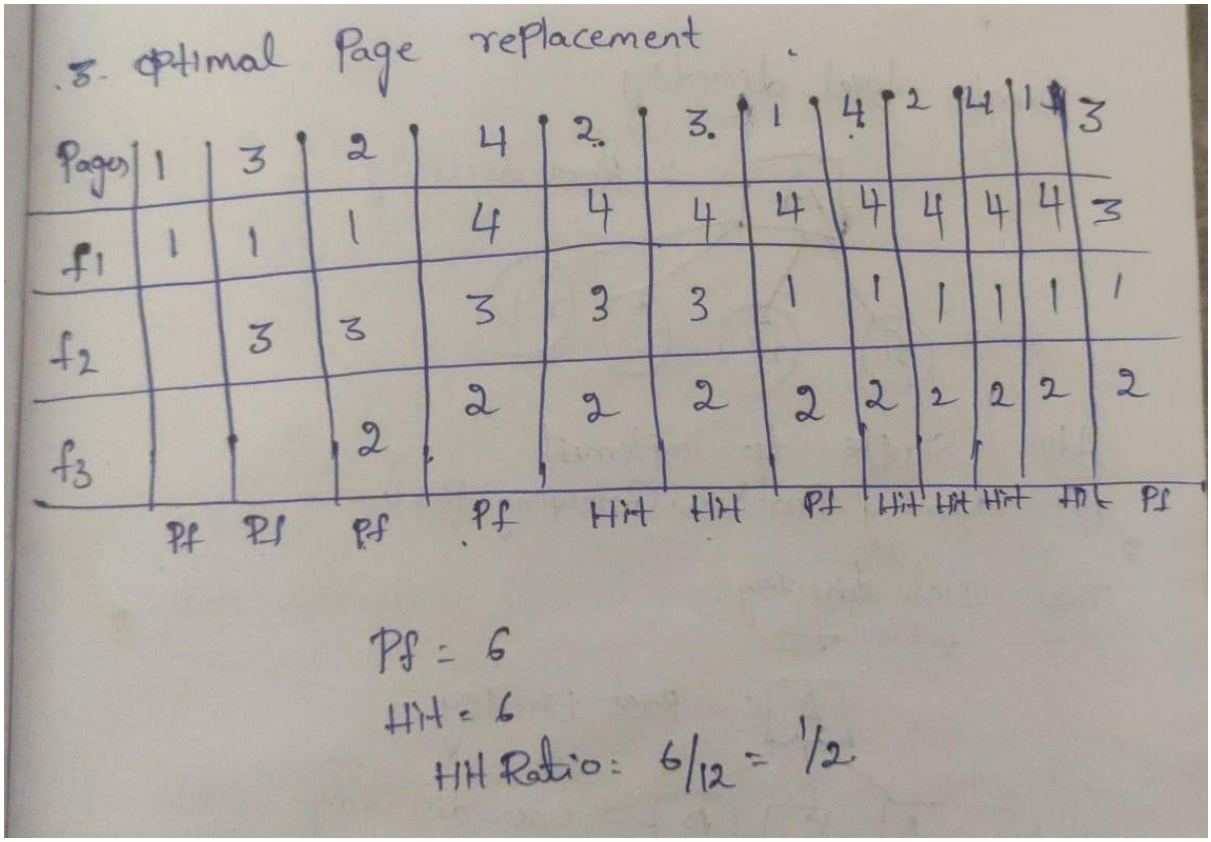
- **Least Recently Used (LRU)**

  This algorithm stands for "Least recent used" and this algorithm helps the Operating system to search those pages that are used over a short duration of timeframe.In this algorithm, when a page fault occurs, then the page that has not been used for the longest duration of time is replaced by the newly requested page.



- **Optimal Page replacement**

In this algorithm, pages are replaced which would not be used for the longest duration of time in the future.



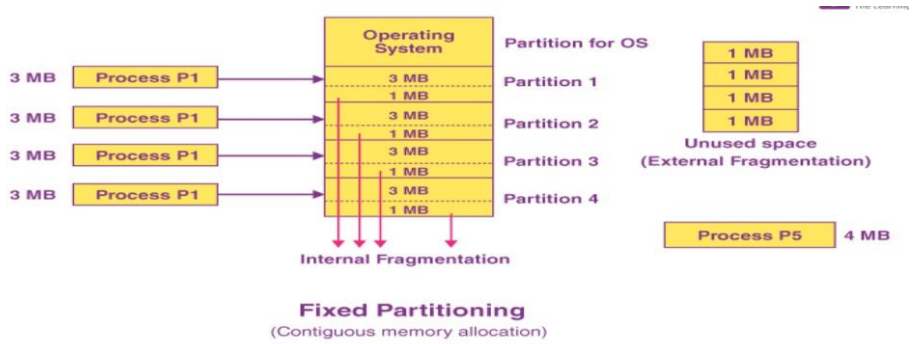
2. Explain about Contiguous Memory Allocation?

A. Contiguous memory allocation

- Fixed memory Partitions
- Variable memory Partitions

Fixed-Sized Partitions

Another name for this is static partitioning. In this case, the system gets dividedinto multiple fixed-sized partitions. In this type of scheme, every partition may consist of exactly one process. This very process limits the extent at which multiprogramming would occur, since the total number of partitions decides the total number of processes.

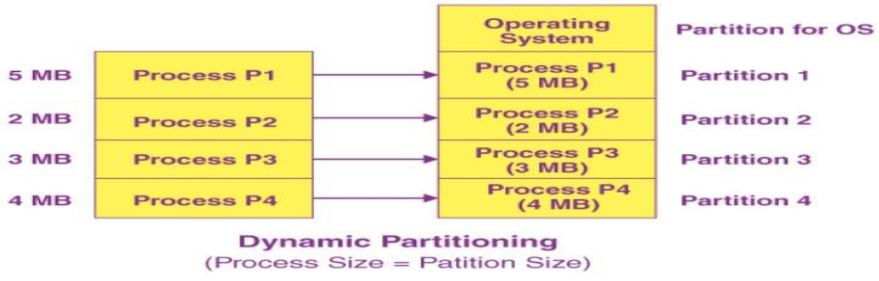


Variable-Sized Partitions

Dynamic partitioning is another name for this. The scheme allocation in this typeof partition is done dynamically. Here, the size of every partition isn't declared initially. Only once we know the process size, will we know the size of the

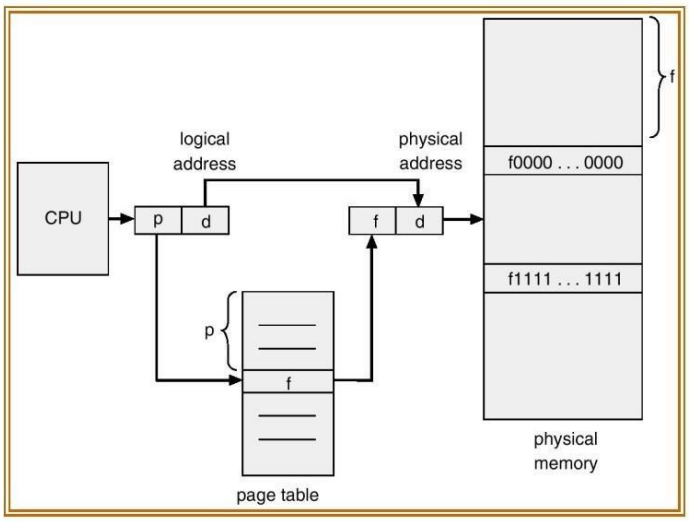
partitions. But in this case, the size of the process and the partition is equal; thus, ithelps in preventing internal fragmentation.

On the other hand, when a process is smaller than its partition, some size of the partition gets wasted (internal fragmentation). It occurs in static partitioning, and dynamic partitioning solves this issue.

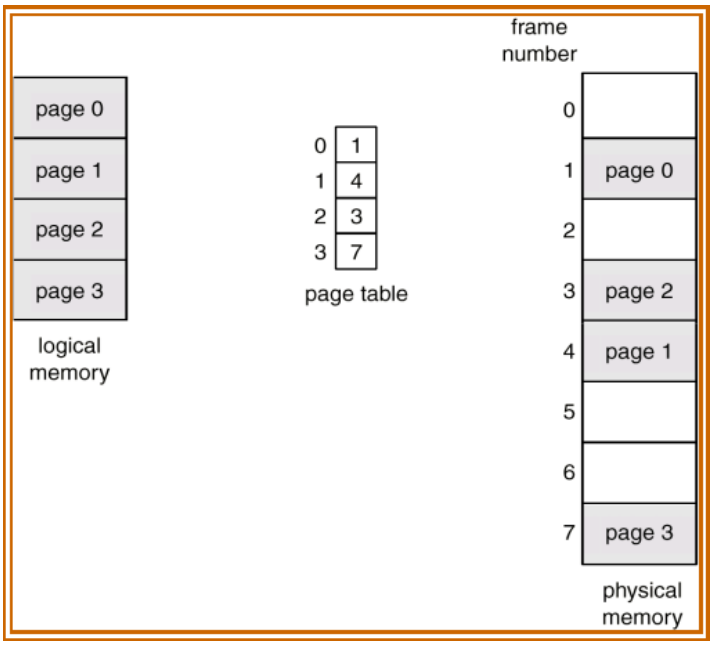


3.Explain about Paging in Memory Management?

- Paging is a memory management scheme that permits the physical address space of a process to be non contiguous.
- Divide physical memory into fixed-sized blocks called frames (size is power of 2, for example 512 bytes).
- Divide logical memory into blocks of same size called pages. When a process is to be executed, its pages are loaded into any available memory frames from the backing store. The backing store is divided into fixed sized blocks that are of the same size as the memory frames.
- The hardware support for paging is illustrated in below figure.
- Every address generated by the CPU is divided into two parts: a page number (p) and a page offset (d). The page number is used as an index into a page table. The page table contains the base address of each page in physical memory. This base address is combined with the page offset to define the physical memory address that is sent to the memory unit.



- The paging model of memory is shown in below figure. The page size is defined by the hardware. The size of a page is typically of a power of 2, varying between 512 bytes and 16 MB per page, depending on the computer architecture. The selection of a power of 2 as a page size makes the translation of a logical address into a page number and page offset particularly easy. If the size of logical address is 2^m, and a page size is 2^n addressing units, then the high order m-n bits of a logical address designate the page number, and the n low order bits designate the page offset.



- Keep track of all free frames.
- To run a program of size n pages, need to find n free frames and load program.
- Set up a page table to translate logical to physical addresses.

  Internal fragmentation may occur. Let us take an example. Suppose a program needs 32 KB memory for allocation. The whole program is divided into smaller units assuming 4 KB and is assigned some address. The address consists of two parts such as:
- A large number in higher order positions and
- Displacement or offset in the lower order bits.

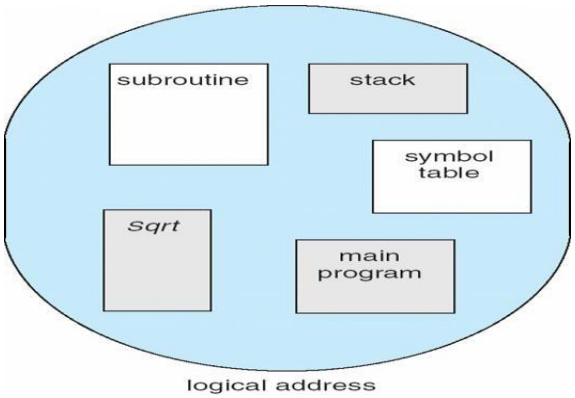
Short Answer Question

1.Explain about Segmentation?

A. Segmentation is a memory-management scheme that supports user view of memory. A program is a collection of segments. A segment is a logical unit such asmain program, procedure, function, method, object, local variables, global variables, common block, stack, symbol table and arrays.

In the segmentation the logical memory can be divided into fixed sized blocks these blocks are called as segments. Segmentation is a memory management functionality that supports the user view of memory. In this, the logical memory is divided into segments and each segment consists of segments number and offset

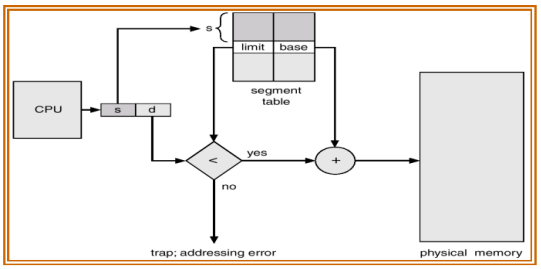
For example the segments in the user view of memory (logical address)isrepresented as follows.



The segment table is used to mapping between the logical memory and physicalmemory the segment table consists of two columns

a) base – contains the starting physical address where the segments reside inmemory.

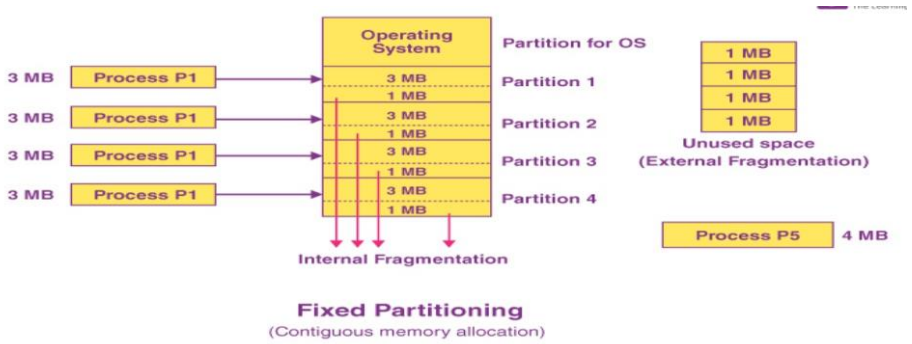
b) Length-Specifies the length of the segment the mapping is represented as follows



2.Write about Fixed and Variable Partitions?

A. Fixed-Sized Partitions

Another name for this is static partitioning. In this case, the system gets dividedinto multiple fixed-sized partitions. In this type of scheme, every partition may consist of exactly one process. This very process limits the extent at which multiprogramming would occur, since the total number of partitions decides the total number of processes.

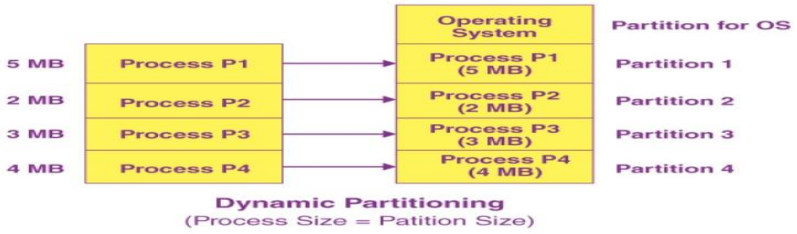


Variable-Sized Partitions

Dynamic partitioning is another name for this. The scheme allocation in this typeof partition is done dynamically. Here, the size of every partition isn't declared initially. Only once we know the process size, will we know the size of the

partitions. But in this case, the size of the process and the partition is equal; thus, ithelps in preventing internal fragmentation.

On the other hand, when a process is smaller than its partition, some size of the partition gets wasted (internal fragmentation). It occurs in static partitioning, and dynamic partitioning solves this issue.



Dynamic Partitioning
(Process Size = Patition Size)

3.Discuss about Paging?

A.Refer question essay question 3

UNIT-5

1. What is file ? Explain about File Operations in detail?

A. A file is a collection of logically related data that is recorded on the secondary storage in the form of sequence of operations.

Create operation:

This operation is used to create a file in the file system. It is the most widely used operation performed on the file system. To create a new file of a particular type the associated application program calls the file system.

Open operation:

The user wants to open a file, it provides a file name to open the particular file inthe file system.

Write operation:

This operation is used to write the information into a file. Data are written to the file at the current position. The system must keep a write pointer to know the location in the file where the next write is to take place. The write pointer must be updated whenever a write occurs.

Read operation:

This operation reads the contents from a file. A Read pointer is maintained by the OS, pointing to the position up to which the data has been read.

Re-position or Seek operation:

The seek system call re-positions the file pointers from the current position to a specific place in the file i.e. forward or backward depending upon the user'srequirement. This operation is generally performed with those file management systems that support direct access files.

Delete operation:

Deleting the file will not only delete all the data stored inside the file it is also used so that disk space occupied by it is freed. In order to delete the specified file the directory is searched. When the directory entry is located, all the associated file space and the directory entry is released.

Truncate operation:

Truncating is simply deleting the file except deleting attributes. The file is not completely deleted although the information stored inside the file gets replaced.

Close operation:

When the processing of the file is complete, it should be closed so that all the changes made permanent and all the resources occupied should be released. On closing it deallocates all the internal descriptors

that were created when the file wasopened.

Append operation:

This operation adds data to the end of the file.

Rename operation:

This operation is used to rename the existing file.

2. Explain about Directory Structure?

A. A Directory is the collection of the correlated files on the disk. In simple words, a directory is like a container which contains file and folder. In a directory, we can store the complete file attributes or some attributes of the file.

There are various types of information which are stored in a directory:

1. Name
2. Type
3. Location
4. Size
5. Position
6. Protection
7. Usage

Name: - Name is the name of the directory, which is visible to the user.

Type: - Type of a directory means what type of directory is present such as single-level directory, two-level directory, tree-structured directory, and Acyclic graph directory.

Location: - Location is the location of the device where the header of a file is located.

Size: - Size means number of words/blocks/bytes in the file.

Position: - Position means the position of the next-read pointer and the next-write pointer.

Protection: - Protection means access control on the read/write/delete/execute.

Usage: - Usage means the time of creation, modification, and access, etc.

Operations on Directory

The various types of operations on the directory are:

- Creating
- Deleting
- Searching
- List a directory
- Renaming

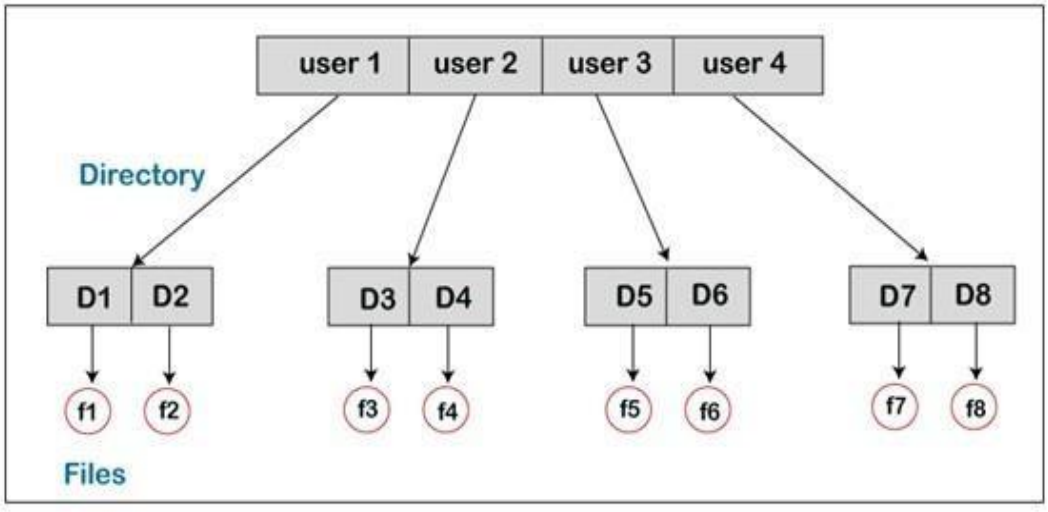
Types of Directory Structure

1. Single-Level Directory
2. Two-Level Directory
3. Tree-Structured Directory
4. Acyclic Graph Directory

Single-Level Directory: - Single-Level Directory is the easiest directory structure. There is only one directory in a single-level directory, and that directory is called a root directory. In a single-level directory, all the files arepresent in one directory that makes it easy to understand. In this, under the root directory, the user cannot create the subdirectories.

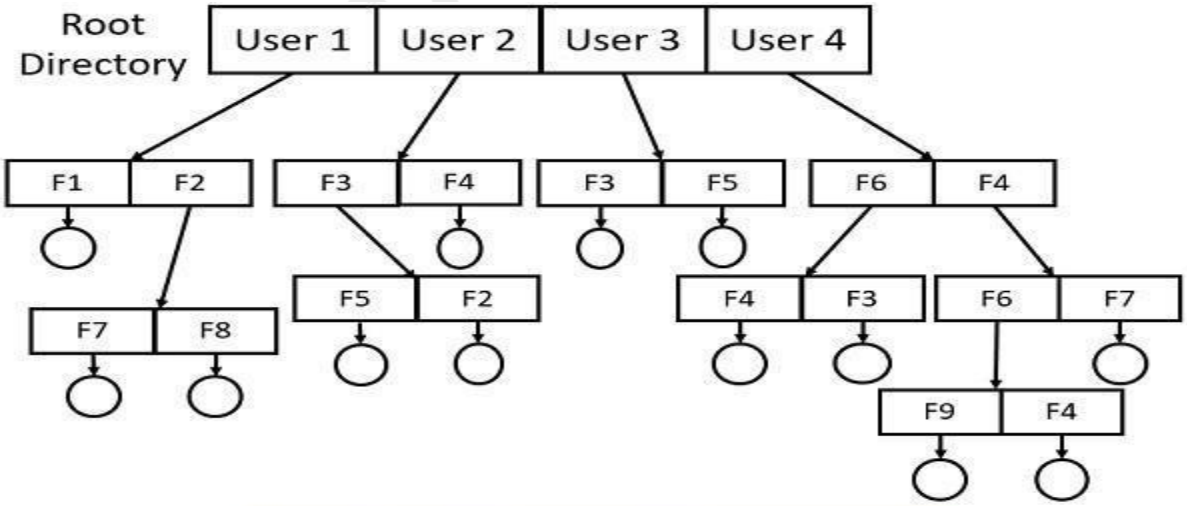
Two-Level Directory

Two-Level Directory is another type of directory structure. In this, it is possible to create an individual directory for each of the users. There is one master node in the two-level directory that include an individual directory for every user. At the second level of the directory, there is a different directory present for each of the users. Without permission, no user can enter into the other user's directory.



Tree-Structured Directory

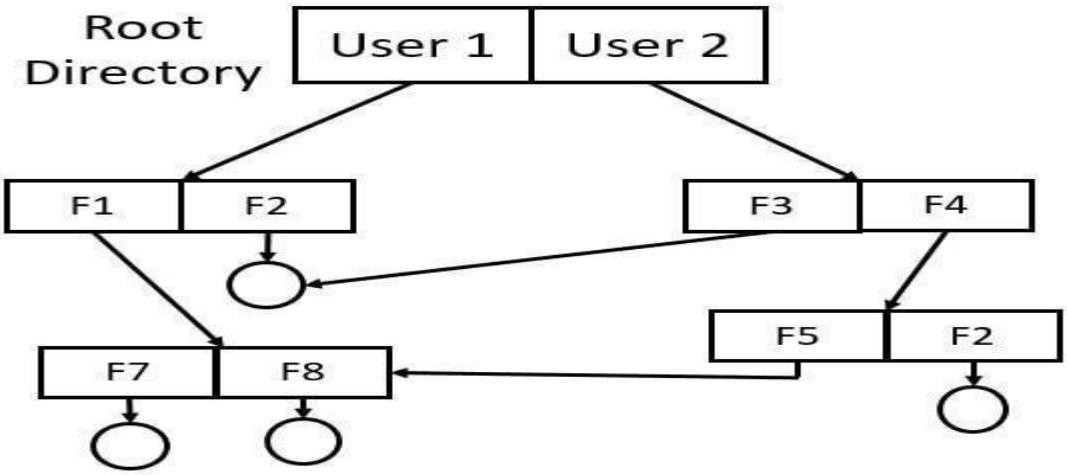
In a tree directory structure, except root directory, every directory or file has only **one parent directory**. So, there is a total separation between the users which provide complete naming freedom. Here, if a user wishes to access another users file, it has to go through two or more directories.



Tree Directory Structure

Acyclic-Graph Directory Structure

This problem can be solved by the **acyclic-graph** directory structure. As thisdirectory structure allows a directory or a file to have many parentdirectories. So, a shared file in a directory can be pointed by the other userdirectories who have access to that shared file using the links.



Acyclic Graph Directory Structure

SHORT ANSWER QUESTIONS:

1. Illustrate about File Allocation Methods?

A. File Allocation Methods in OS File allocation methods are basically ways in which any file is stored in the memory block of the system. There are three typesfor file allocation methods.

- Contiguous Allocation.
- Linked Allocation.
- Indexed Allocation.

1. *Contiguous Allocation*

- Each file is stored in a contiguous block of space on the disk.

- Advantages: Efficient disk access and allocation.

- Disadvantages: External fragmentation, difficult to allocate space for files of varying sizes.

2. *Linked Allocation*

- Each file is broken into fixed-size blocks, and each block is stored in a different location on the disk.

- A linked list is used to keep track of the blocks.

- Advantages: Efficient use of disk space, no external fragmentation.

- Disadvantages: Slow disk access, difficult to implement.

3. *Indexed Allocation*

- A separate block, called an index block, is used to store the location of each file block.

- Advantages: Fast disk access, efficient use of disk space.

- Disadvantages: Requires additional space for the index block.

2. Discuss about File Concept?

A. A file is a collection of related data or information that is stored on a computer's storage device.file system is a way of organizing and storing files on a computer's storage device. It provides a hierarchical structure for storing files, including directories, subdirectories, and files.

Characteristics of a File:

Name: - Name is the name of the directory, which is visible to the user.

Type: - Type of a directory means what type of directory is present such as single-level directory, two-level directory, tree-structured directory, and Acyclic graph directory.

Location: - Location is the location of the device where the header of a file is located.

Size: - Size means number of words/blocks/bytes in the file.

Position: - Position means the position of the next-read pointer and the next-write pointer.

Protection: - Protection means access control on the read/write/delete/execute.

Usage: - Usage means the time of creation, modification, and access, etc.

File Operations:

1. Create: Create a new file.

2. Delete: Delete an existing file.

3. Read: Read the contents of a file.

4. Write: Write data to a file.

5. Copy: Copy a file from one location to another.

6. Move: Move a file from one location to another.