Assignment 1

Batch : Sep21

Subject : OPERATING SYSTEM

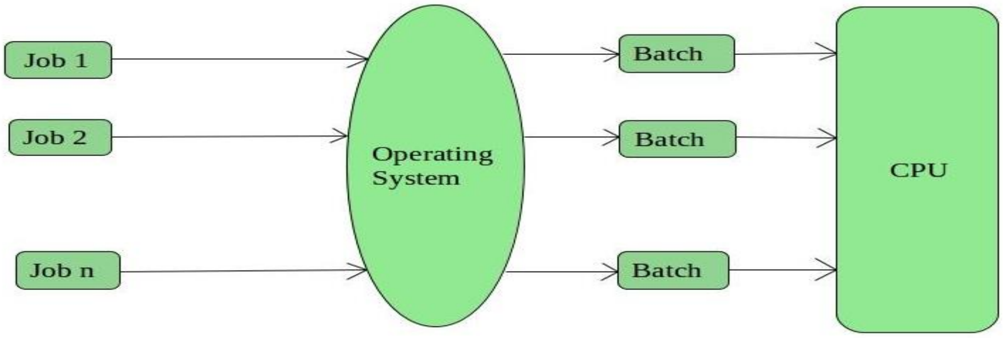
Submission Date: 10/10/2021

Q.1 Define the types of Operating System?

An Operating System performs all the basic tasks like managing files, processes, and memory. Thus operating system acts as the manager of all the resources, i.e. **resource manager**. Thus, the operating system becomes an interface between user and machine.

Types of Operating Systems: Some widely used operating systems are as follows-

1. **Batch Operating System** –
This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirement and group them into batches. It is the responsibility of the operator to sort jobs with similar needs.



Advantages of Batch Operating System:

- It is very difficult to guess or know the time required for any job to complete. Processors of the batch systems know how long the job would be when it is in queue
- Multiple users can share the batch systems
- The idle time for the batch system is very less
- It is easy to manage large work repeatedly in batch systems

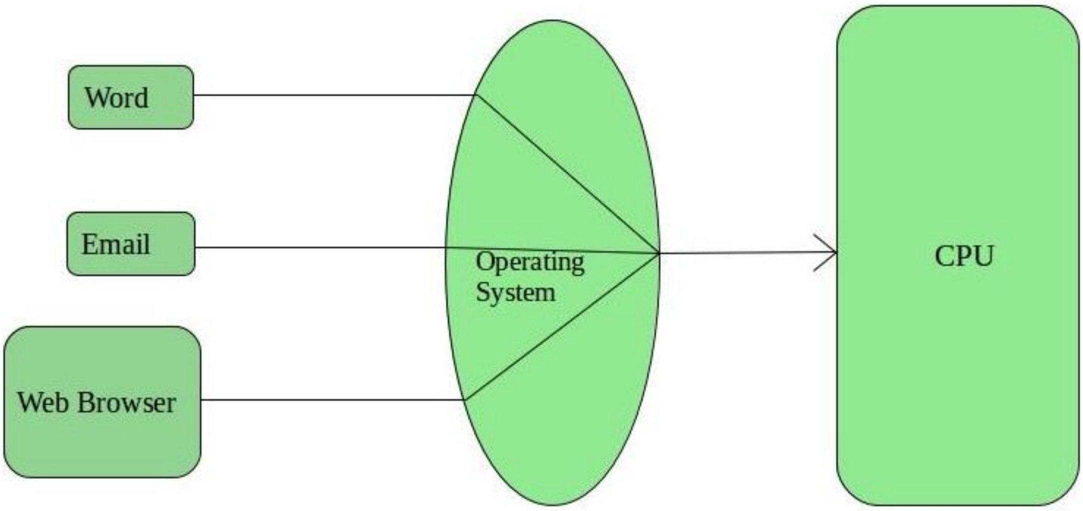
Disadvantages of Batch Operating System:

- The computer operators should be well known with batch systems
- Batch systems are hard to debug
- It is sometimes costly
- The other jobs will have to wait for an unknown time if any job fails

Examples of Batch based Operating System: Payroll System, Bank Statements, etc.

2. **Time-Sharing Operating Systems –**

Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of CPU as they use a single system. These systems are also known as Multitasking Systems. The task can be from a single user or different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.



Advantages of Time-Sharing OS:

- Each task gets an equal opportunity
- Fewer chances of duplication of software
- CPU idle time can be reduced

Disadvantages of Time-Sharing OS:

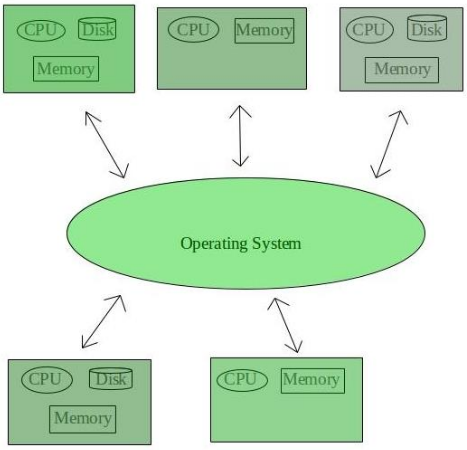
- Reliability problem
- One must have to take care of the security and integrity of user programs and data
- Data communication problem

Examples of Time-Sharing OSs are: Multics, Unix, etc.

3. **Distributed Operating System –**

These types of the operating system is a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, with a great pace. Various autonomous interconnected computers communicate with each other using a shared communication network.

Independent systems possess their own memory unit and CPU. These are referred to as **loosely coupled systems** or distributed systems. These system's processors differ in size and function. The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network.



Advantages of Distributed Operating System:

- Failure of one will not affect the other network communication, as all systems are independent from each other
- Electronic mail increases the data exchange speed
- Since resources are being shared, computation is highly fast and durable
- Load on host computer reduces
- These systems are easily scalable as many systems can be easily added to the network
- Delay in data processing reduces

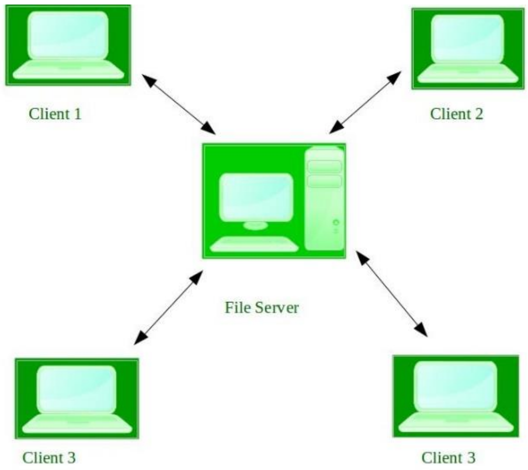
Disadvantages of Distributed Operating System:

- Failure of the main network will stop the entire communication
- To establish distributed systems the language which is used are not well defined yet
- These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet

Examples of Distributed Operating System are- LOCUS, etc.

4. Network Operating System –

These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These types of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network. One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections, etc. and that’s why these computers are popularly known as **tightly coupled systems**.



Advantages of Network Operating System:

- Highly stable centralized servers
- Security concerns are handled through servers
- New technologies and hardware up-gradation are easily integrated into the system
- Server access is possible remotely from different locations and types of systems

Disadvantages of Network Operating System:

- Servers are costly
- User has to depend on a central location for most operations
- Maintenance and updates are required regularly

Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD, etc.

5. Real-Time Operating System –

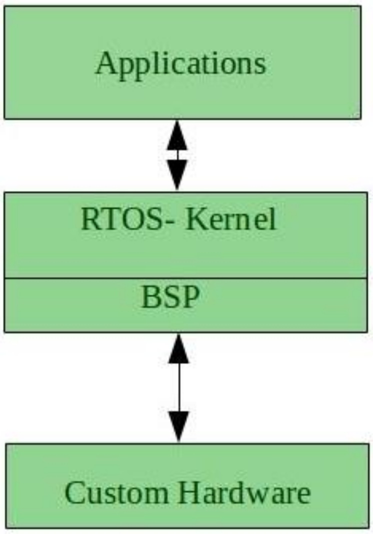
These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called **response time**.
Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

Two types of Real-Time Operating System which are as follows:

- **Hard Real-Time Systems:**

  These OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of any accident. Virtual memory is rarely found in these systems.
- **Soft Real-Time Systems:**

  These OSs are for applications where for time-constraint is less strict.



Advantages of RTOS:

- **Maximum Consumption:** Maximum utilization of devices and system, thus more output from all the resources
- **Task Shifting:** The time assigned for shifting tasks in these systems are very less. For example, in older systems, it takes about 10 microseconds in shifting one task to another, and in the latest systems, it takes 3 microseconds.
- **Focus on Application:** Focus on running applications and less importance to applications which are in the queue.
- Real-time **operating system in** the **embedded system:** Since the size of programs are small, RTOS can also be used in embedded systems like in transport and others.
- **Error Free:** These types of systems are error-free.
- **Memory Allocation:** Memory allocation is best managed in these types of systems.

Disadvantages of RTOS:

- **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on few applications to avoid errors.
- **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
- **Device driver and interrupt signals:** It needs specific device drivers and interrupts signals to respond earliest to interrupts.
- **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

Examples of Real-Time Operating Systems are: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

Q.2 Explain DHCP?

How DHCP works

DHCP runs at the application layer of the TCP/IP protocol stack to dynamically assign IP addresses to DHCP clients/nodes and to allocate TCP/IP configuration information to the DHCP clients. Information includes subnet mask information, default gateway, IP addresses and domain name system addresses.

DHCP is based on client-server protocol in which servers manage a pool of unique IP addresses, as well as information about client configuration parameters, and assign addresses out of those address pools.

The DHCP lease process works as follows:

- First of all, a client (network device) must be connected to the internet.
- DHCP clients request an IP address. Typically, client broadcasts a query for this information.
- DHCP server responds to the client request by providing IP server address and other configuration information. This configuration information also includes time period, called a lease, for which the allocation is valid.
- When refreshing an assignment, a DHCP clients request the same parameters, but the DHCP server may assign a new IP address. This is based on the policies set by the administrator.

Benefits of DHCP

There are following benefits of DHCP:

Centralized administration of IP configuration: DHCP IP configuration information can be stored in a single location and enables that administrator to centrally manage all IP address configuration information.

Dynamic host configuration: DHCP automates the host configuration process and eliminates the need to manually configure individual host. When TCP/IP (Transmission control protocol/Internet protocol) is first deployed or when IP infrastructure changes are required.

Seamless IP host configuration: The use of DHCP ensures that DHCP clients get accurate and timely IP configuration IP configuration parameter such as IP address, subnet mask, default gateway, IP address of DND server and so on without user intervention.

Flexibility and scalability: Using DHCP gives the administrator increased flexibility, allowing the administrator to move easily change IP configuration when the infrastructure changes.

Q.**3 Explain DNS?**

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).

DNS WORKING :-

The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 192.168.1.1). An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - like a street address is used to find a particular home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machine-friendly address necessary to locate the example.com webpage.

In order to understand the process behind the DNS resolution, it's important to learn about the different hardware components a DNS query must pass between. For the web browser, the DNS lookup occurs "behind the scenes" and requires no interaction from the user's computer apart from the initial request.

Q.4. Explain paging?

- In this technique, physical memory (i.e. user space of a main memory) is divided into fixed size of blocks reffered as frames, and process's logical memory space is divided into same size of blocks reffered as pages, whereas maximum size of page must be equal to size of frame, i.e. if e.g. size of frame = 4K, then maximum size of each page must be 4K, size of page may be less than 4K.

- As process is divided into pages, so when it is requesting for memory, pages of one process may gets loaded into the main memory at any free frames, and for a process memory gets allocated in a non-contiguos manner.

- As pages of a one process gets loaded randomly into the main memory, and in a system thousands processes are running at a time, to keep track on all the pages of each process, an OS maintains one table per process reffered as a page table in which information about all the pages of that process can be kept.

- There is no external fragmentation in paging.

- Internal fragmentation may exists in paging when the size of page is less than size of frame.

Virtual Memory Management: -

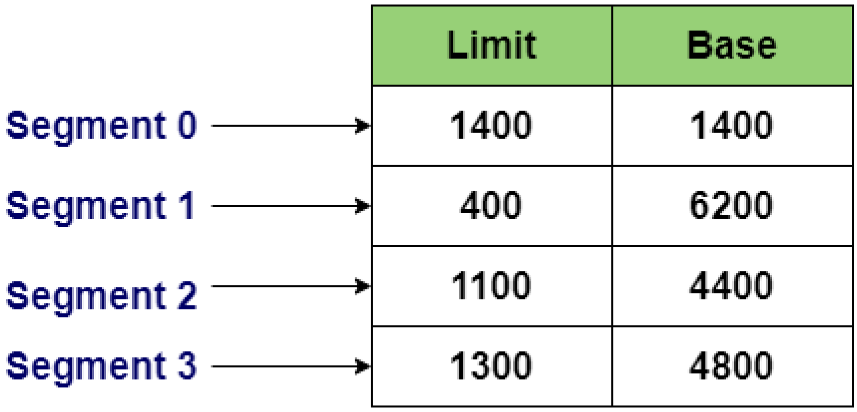
Demand Paging: any page of a process gets loaded into the main memory only after requesting by that process i.e. on demand and hence reffered as demand paging, page which is never requested never gets loaded into the main memory and hence it also called as pure demand paging.

- If a process is requesting for any page and if that page is not exists in the main memory, then it is referred as page fault.

Q.5 Explain segmentation?

Segmentation –

In this technique, process in its logical memory is divided into small size segments like stack segment, heap segment, data segment, bss segment, rodata segment, code segment etc..., and when process is requesting for memory it is not requesting memory contiguosly for whole process, memory gets allocated contiguosly only for small size segments, and segments of one process may get load into the memory randomly at any locations, i.e. for a process memory gets allocated in a non-contiguos manner, and then only an execution of a process can be completed. As segments of a one process gets loaded randomly, and in a system thousands processes are running at a time, hence to keep track on all the segments of each process, an OS maintains one table per process reffered as a segment table in which information about all the segments of that process can be kept. - Using segmentation an external fragmentation can be reduced but cannot be completely avoided



	Limit	Base
Segment 0	1400	1400
Segment 1	400	6200
Segment 2	1100	4400
Segment 3	1300	4800

Segment Table

Q6. Explain 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. 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.

Memory management Techniques:

The Memory management Techniques can be classified into following main categories:

- Contiguous memory management schemes

In a Contiguous memory management scheme, each program occupies a single contiguous block of storage locations, i.e., a set of memory locations with consecutive addresses.

Fixed Partitioning

The main memory is divided into several fixed-sized partitions in a fixed partition memory management scheme or static partitioning. These partitions can be of the same size or different sizes. Each partition can hold a single process. The number of partitions determines the degree of multiprogramming, i.e., the maximum number of processes in memory. These partitions are made at the time of system generation and remain fixed after that.

Advantages of Fixed Partitioning memory management schemes:

- Simple to implement.
- Easy to manage and design.

Disadvantages of Fixed Partitioning memory management schemes:

- This scheme suffers from internal fragmentation.
- The number of partitions is specified at the time of system generation.

Dynamic Partitioning

The dynamic partitioning was designed to overcome the problems of a fixed partitioning scheme. In a dynamic partitioning scheme, each process occupies only as much memory as they require when loaded for processing. Requested processes are allocated memory until the entire physical memory is exhausted or the remaining space is insufficient to hold the requesting process. In this scheme the partitions used are of variable size, and the number of partitions is not defined at the system generation time.

- Non-Contiguous memory management schemes

In a Non-Contiguous memory management scheme, the program is divided into different blocks and loaded at different portions of the memory that need not necessarily be adjacent to one another. This scheme can be classified depending upon the size of blocks and whether the blocks reside in the main memory or not.

What is paging?

Paging is a technique that eliminates the requirements of contiguous allocation of main memory. In this, the main memory is divided into fixed-size blocks of physical memory called frames. The size of a frame should be kept the same as that of a page to maximize the main memory and avoid external fragmentation.

Advantages of paging:

- Pages reduce external fragmentation.
- Simple to implement.
- Memory efficient.
- Due to the equal size of frames, swapping becomes very easy.
- It is used for faster access of data.

What is Segmentation?

Segmentation is a technique that eliminates the requirements of contiguous allocation of main memory. In this, the main memory is divided into variable-size blocks of physical memory called segments. It is based on the way the programmer follows to structure their programs. With segmented memory allocation, each job is divided into several segments of different sizes, one for each module. Functions, subroutines, stack, array, etc., are examples of such modules.

Q7.Explain the function of OS?

The operating system performs the following functions

Memory management

It refers to management of Primary Memory or Main Memory. Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. 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.

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.
- Allocates the processor (CPU) to a process.
- De-allocates processor when a process is no longer required.

Device Management

An Operating System manages device communication via their respective drivers. It 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.

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.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

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's and users − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

Q8.Explain a kernel? Its architecture and working?

The core part of the OS is called Kernel. The Kernel runs the user Instructions on System and provides secure approach to the hardware. There are lot of set of Programs, and assets are in small number, so the kernel takes decision that when program will run. This mechanism is called Organizing. The direct access of the hardware is complicated, as Because There are different type's hardware designs for single component. To overcome this problem, the kernels provide hardware abstraction (set of instruction common to certain type of devices) to hide complication from set of applications and give a standard pinpoint for different system. Thus the application programmers can easily develop programs without knowing how to deal with a specific device. The kernel only depends on the software drivers which transform the generic command into instruction codes for that specific device.

Q9.Explain a shell script?

A shell script is a list of commands in a computer program that is run by the UNIX shell which is a command line interpreter. The different operations performed by shell scripts are program execution, file manipulation and text printing.

There are two major types of shells in UNIX. These are:

- Bourne Shell

This is the default shell for version 7 UNIX. The character $ is the default prompt for the Bourne shell.

C Shell

This is a UNIX shell and a command processor that is run in a text window. The character % is the default prompt for the C shell.

Advantages of Shell Script
Some of the advantages of shell script are −

- The commands and syntax of the shell script are the same as that entered at the command line. Because of this, there is no need to switch to a completely different syntax.
- It is much faster to write a code in shell script than in other programming languages. This also means that the program is easier to create and files required can be selected easily.
- Shell script can also be used to provide linkage for already existing programs.

Disadvantages of Shell Script

Some of the disadvantages of shell script are −

- There may be errors in shell scripting that prove to be quite costly.
- The programs in shell script are quite slow while executing and a new process is required for every shell command executed.
- Different platforms in shell scripting may also have compatibility problems.

Q.10 Page fault in OS:-

A page fault occurs when a program attempts to access data or code that is in its address space, but is not currently located in the system RAM.

If any program tries to access piece of memory but which is not existed into physical memory, means **main memory**, then page fault will be occurred. The fault specifies the O/S that it must trace the all data in to **virtual memory management**, and after that moves it from secondary memory like as hard disk to **primary memory** of system.

Page Hit – If CPU tries to retrieve the needed page from main memory, and that page is existed in the main memory (RAM), then it is known as “PAGE HIT”.

Page Miss – If required page is not existed in the RAM then it is known as “PAGE MISS”.

Page Fault Rate – That rate, on which threads find the page fault in the memory, it is known as “PAGE FAULT RATE”. Page Fault Rate is measured in Per Second.

Page Fault Time – Time taken to fetch page from secondary memory + time taken to retrieve from RAM after loading the needed page, and that taken combine time is known as “PAGE FAULT TIME”.

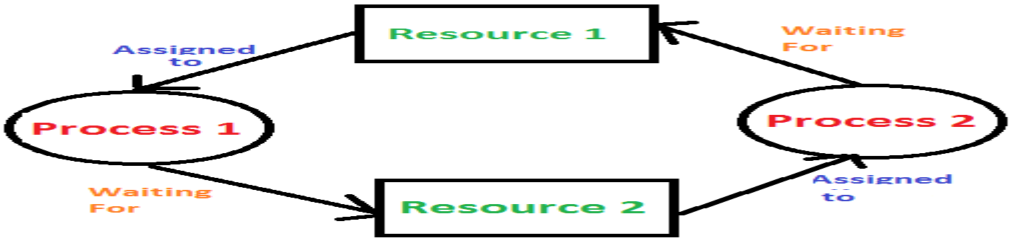
Hard Page Fault – If needed page is existed in the page file in the hard disk, then it is known as “HARD PAGE FAULT”.

Soft Page Fault – If need page is not presented in hard disk but it found somewhere else in memory, and then it is known as “SOFT PAGE FAULT”.

Minor Page Fault – If process requires any data, and that data existed in memory but the same time it is allotted to another process, then it is known as “MINOR PAGE FAULT”.

Q.11 Deadlock in os:

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. Consider an example when two trains are coming toward each other on the same track and there is only one track, none of the trains can move once they are in front of each other. A similar situation occurs in operating systems when there are two or more processes that hold some resources and wait for resources held by other(s).



Q.12 Necessary condition for deadlock

Deadlock can arise if the following four conditions hold simultaneously (Necessary Conditions)**:**

Mutual Exclusion: One or more than one resource are non-shareable (Only one process can use at a time)

Hold and Wait: A process is holding at least one resource and waiting for resources.

No Pre-emption: A resource cannot be taken from a process unless the process releases the resource.

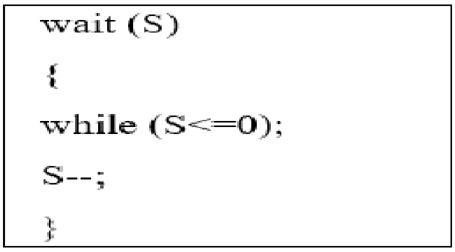
Circular Wait: A set of processes are waiting for each other in circular form.

Q.13 Semaphore:

Semaphore is defined as an integer variable which is used to solve the problem of the critical section in process synchronization. In semaphore, we use two types of atomic operations, and that operations are wait and signal.

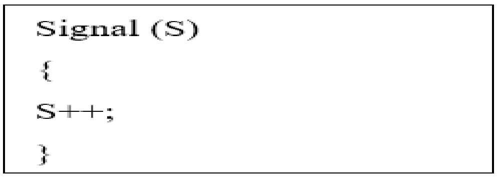
The definitions of wait and signal are as follows:

Wait: – In wait operation, the argument ‘S’ value is decrement by 1 if the value of the ‘S’ variable is positive. If the value of the argument variable ‘S’ is zero or negative, no operation is performed.



```
wait (S)
{
while (S<=0);
S--;
}
```

Signal: – In Signal atomic operation, the value of the argument variable ‘S’ is incremented.



```
Signal (S)
{
S++;
}
```

Characteristic of Semaphore
There are following characteristics of Semaphore:
Semaphore carries a non-negative integer value always.

1. We use semaphore mechanism to offer synchronization of tasks.
2. It is a low-level synchronization mechanism.

Types of Semaphores
We have following two types of Semaphores:

1. Counting Semaphores
2. Binary Semaphores
3. Counting Semaphores: – Counting Semaphore is defined as a semaphore that contains integer values, and these values have an unrestricted value domain. A counting semaphore is helpful to coordinate the resource access, which includes multiple instances.
4. Binary Semaphores: – Binary Semaphores are also called Mutex lock. There are two values of binary semaphores, which are 0 and 1. The value of binary semaphore is initialized to 1. We use binary semaphore to remove the problem of the critical section with numerous processes.

14. Explain a mutex.

Mutex (mutual exclusion) is a locking mechanism used to synchronize access to a resource. Only one task (can be a thread or process based on OS abstraction) can acquire the mutex. It means there is ownership associated with a mutex, and only the owner can release the lock (mutex).

15. Difference among kernel space and user space.

The kernel runs in kernel space, and normal programs run in user space. User space is basically a form of sand-boxing -- it restricts user programs so they can't mess with memory (and other resources) owned by other programs or by the OS kernel. This limits (but usually doesn't entirely eliminate) their ability to do bad things like crashing the machine.

The kernel is the core of the operating system. It normally has full access to all memory and machine hardware (and everything else on the machine). To keep the machine as stable as possible, you normally want only the most trusted, well-tested code to run in kernel mode/kernel space.

16. Write in brief the ping command.

Ping (packet inter-network groper)
Purpose: Sends an echo request to a network host.

- Determining the status of the network and various foreign hosts.
- Tracking and isolating hardware and software problems.
- Testing, measuring, and managing networks.

If the host is operational and on the network, it responds to the echo. Each echo request contains an Internet Protocol (IP) and ICMP header, followed by a ping PID and a timeval structure, and enough bytes to fill out the packet. The default is to continuously send echo requests until an Interrupt is received (Ctrl-C).

17. Explain UNIX?

UNIX is an operating system which was first developed in the 1960s, and has been under constant development ever since. By operating system, we mean the suite of programs which make the computer work. It is a stable, multi-user, multi-tasking system for servers, desktops and laptops.

Directory structure
All the files are grouped together in the directory structure. The file-system is arranged in a hierarchical structure, like an inverted tree. The top of the hierarchy is traditionally called **root** (written as a slash /)

File and Process
Everything in UNIX is either a file or a process. A process is an executing program identified by a unique PID (process identifier). A file is a collection of data. They are created by users using text editors, running compilers etc.

18. Explain grep.

You use the grep command within a Linux or Unix-based system to perform text searches for a defined criteria of words or strings. grep stands for **G**lobally search for a **R**egular **E**xpression and **P**rint it out.
Grep syntax

The following example shows the basic command structure:
grep 'string' filename(s)

This command searches for and returns any lines of text that contain the given criteria *string* in *filename(s)*.

Options:

You can add any of the following options individually or in combination to refine your search:

- -i: Prints lines with matching criteria while ignores casing (Upper/Lowecase).
- -l: Prints filenames only.
- -n: Prints lines with matching criteria and line numbers.
- -c: Prints count of lines with matching criteria.
- -v: Prints lines not matching criteria (inverse search).
- -w: Prints whole word matches.
- -A n: Prints *n* lines after matches.
- -B n: Prints *n* lines before matches.
- -C n: Prints *n* lines before and after matches.

19. Explain pipe?

Pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on. It can also be visualized as a temporary connection between two or more commands/ programs/ processes. A pipe is a form of redirection (transfer of standard output to some other destination) that is used in Linux operating systems to send the output command/program/process another command/program/process for further processing. Pipes are unidirectional i.e data flows from left to right through the pipeline.

Syntax :

```
command_1 | command_2 | command_3 | .... | command_N
```

Q.20 Difference among Thread & Process.

A process, in the simplest terms, is an executing program. One or more threads run in the context of the process. A thread is the basic unit to which the operating system allocates processor time. A thread can execute any part of the code, including parts currently being executed by another thread A process is a program under execution i.e an active program. A thread is a lightweight process that can be managed by a scheduler. Processes require more time for switching as they are more heavy. Threads require less time for context switching as they are lighter than processes.

Q.21 Explain a scheduling algorithm?

Definition: A Scheduling Algorithm is the algorithm which tells us how much CPU time we can allocate to the processes. These scheduling algorithms are either preemptive or non-preemptive. scheduling algorithm The method used to determine which of several processes, each of which can safely have a resource allocated to it, will actually be granted use of the resource. The algorithm may take into account the priority of the user associated with the process, the requirement to maintain high utilization of system resources, and deadlines for the job.

Q.22 Explain pre-emptive and non-preemptive scheduling?

Preemptive Scheduling:

In Preemptive Scheduling, the tasks are mostly assigned with priorities. Sometimes it is important to run a task with a higher priority before another lower priority task, even if the lower priority task is still running. The lower priority task holds for some time and resumes when the higher priority task finishes its execution.

Non-Preemptive Scheduling:

In this type of scheduling method, the CPU has been allocated to a specific process. The process that keeps the CPU busy will release the CPU either by switching context or terminating. It is the only method that can be used for various hardware platforms. That's because it doesn't need special hardware (for example, a timer) like preemptive scheduling.

Q.23.Define the different scheduling algorithms.

There are mainly six types of process scheduling algorithms

1.First Come First Serve (FCFS)-

First Come First Serve is the full form of FCFS. It is the easiest and most simple CPU scheduling algorithm. In this type of algorithm, the process which requests the CPU gets the CPU allocation first. This scheduling method can be managed with a FIFO queue.

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

SJF is a full form of (Shortest job first) is a scheduling algorithm in which the process with the shortest execution time should be selected for execution next. This scheduling method can be preemptive or non-preemptive. It significantly reduces the average waiting time for other processes awaiting execution.

3)Shortest Remaining Time

The full form of SRT is Shortest remaining time. It is also known as SJF preemptive scheduling. In this method, the process will be allocated to the task, which is closest to its completion. This method prevents a newer ready state process from holding the completion of an older process.

4)Priority Scheduling-

Priority scheduling is a method of scheduling processes based on priority. In this method, the scheduler selects the tasks to work as per the priority.

Priority scheduling also helps OS to involve priority assignments. The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis. Priority can be decided based on memory requirements, time requirements, etc.

5)Round Robin Scheduling-

Round robin is the oldest, simplest scheduling algorithm. The name of this algorithm comes from the round-robin principle, where each person gets an equal share of something in turn. It is mostly used for scheduling algorithms in multitasking. This algorithm method helps for starvation free execution of processes.

6)Multilevel Queue Scheduling-

This algorithm separates the ready queue into various separate queues. In this method, processes are assigned to a queue based on a specific property of the process, like the process priority, size of the memory, etc.

However, this is not an independent scheduling OS algorithm as it needs to use other types of algorithms in order to schedule the jobs.

24. Explain booting process?

- Booting is the process of starting a computer.
- It can be initiated by hardware such as a button press or by a software command. After it is switched on, a CPU has no software in its main memory, so some processes must load software into memory before execution. This may be done by hardware or firmware in the CPU or by a separate processor in the computer system.
- Restarting a computer also is called rebooting, which can be "*hard*", e.g., after electrical power to the CPU is switched from off to on, or "*soft*", where the power is not cut.

Sequencing of Booting

- Booting is a start-up sequence that starts the operating system of a computer when it is turned on. A boot sequence is the initial set of operations that the computer performs when it is switched on. Every computer has a boot sequence.

Bootloader
Boot Device
Boot Sequence

- Boot loader
- A boot loader is a critical piece of software running on any system. Whenever a computing system is initially powered on, the first piece of code to be loaded and run is the boot loader. It **provides an interface for the user to load an operating system and applications**.

Boot Devices:

➢ The boot device is the device from which the operating system is loaded. A modern PC BIOS (Basic Input/Output System) supports booting from various devices. These include the local hard disk drive, optical drive, floppy drive, a network interface card, and a USB device. The BIOS will allow the user to configure a boot order. If the boot order is set to:

- CD Drive
- Hard Disk Drive
- Network

1) **Boot Sequence:** There is a standard boot sequence that all personal computers use. First, the CPU runs an instruction in memory for the BIOS. That instruction contains a jump instruction that transfers to the BIOS start-up program. This program runs a power-on self-test (POST) to check that devices the computer will rely on are functioning properly. Then, the BIOS goes through the configured boot sequence until it finds a bootable device. Once BIOS has found a bootable device, BIOS loads the bootsector and transfers execution to the boot sector.

25. Explain bias?

BIOS **stand for Basic Input Output System**. It is built-in software. It is the first software run by the computer when you turned on your computer system. This software is usually stored in Read Only Memory (ROM) and located on the motherboard. In modern computer systems, the BIOS contents are stored in a flash memory.

The main function of BIOS is to set up hardware and start an OS, and it contains generic code that is needed to control display screens, the keyboard, and other functions. The BIOS is built-in software that manages the hard drives and cannot live on one. It cannot reside in the RAM (Random Access Memory) as it is accessible before the computer system boots up.

Actually, it lives on the ROM of the computer system, and mainly it is located on EPROM (erasable programmable read-only memory) chip. Therefore, the CPU accesses the EPROM when you turn on the computer and provides control to the BIOS.

Users can perform hardware configuration

They can select boot drives

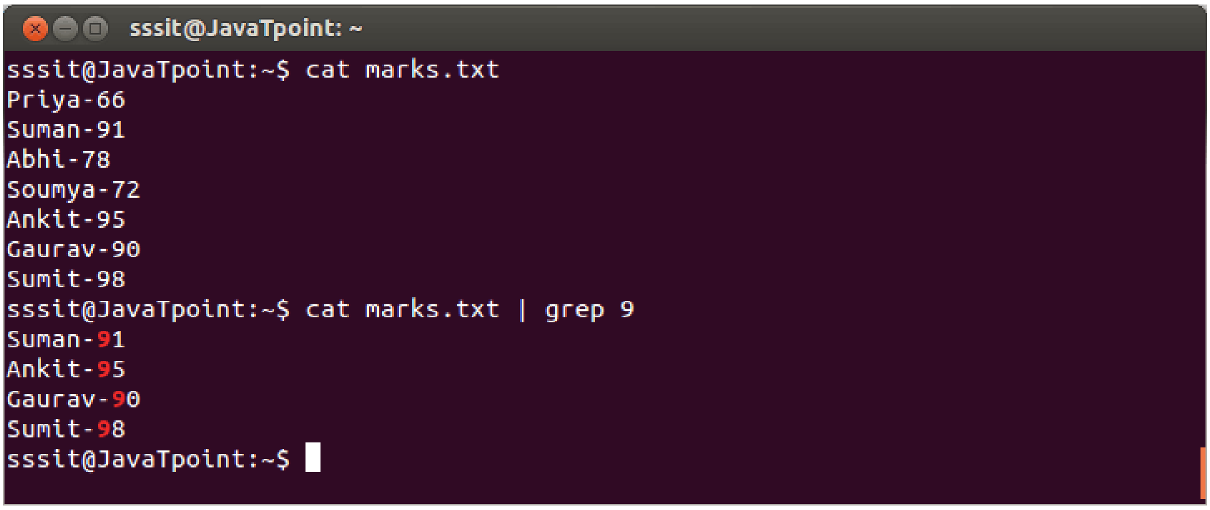
- They can set the system clock
- The BIOS allows users to enable and disable certain system components
- To BIOS user interface function, it provides set password prompts for secured access

Q.27 UNIX commands like touch, sed, grep.

grep

The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for globally search for regular expression and print out). **Syntax:**

grep [options] pattern [files]

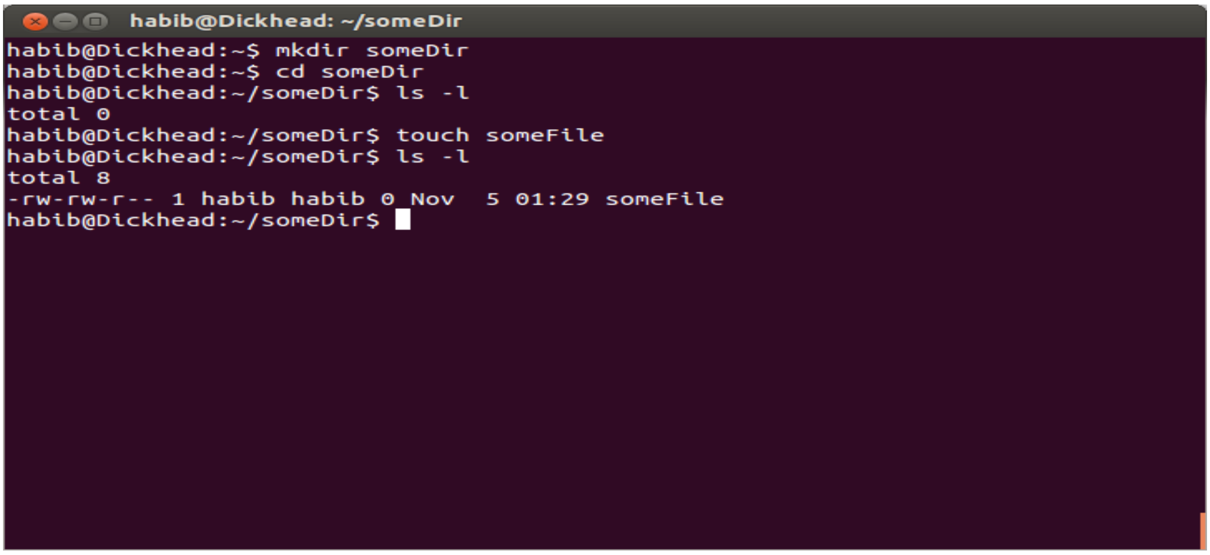


```
sssit@JavaTpoint:~$ cat marks.txt
Priya-66
Suman-91
Abhi-78
Soumya-72
Ankit-95
Gaurav-90
Sumit-98
sssit@JavaTpoint:~$ cat marks.txt | grep 9
Suman-91
Ankit-95
Gaurav-90
Sumit-98
sssit@JavaTpoint:~$
```

Touch

The *touch command* is a standard *command* used in *UNIX*/Linux operating system which is used to create, change and modify timestamps of a file.

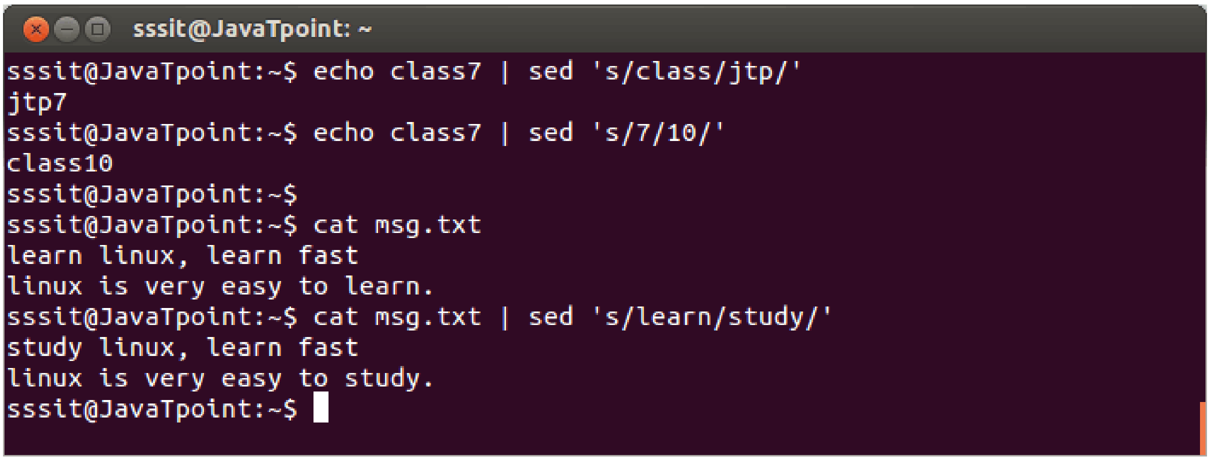
touch {options} {file}



```
habib@Dickhead:~$ mkdir someDir
habib@Dickhead:~$ cd someDir
habib@Dickhead:~/someDir$ ls -l
total 0
habib@Dickhead:~/someDir$ touch someFile
habib@Dickhead:~/someDir$ ls -l
total 8
-rw-rw-r-- 1 habib habib 0 Nov  5 01:29 someFile
habib@Dickhead:~/someDir$ 
```

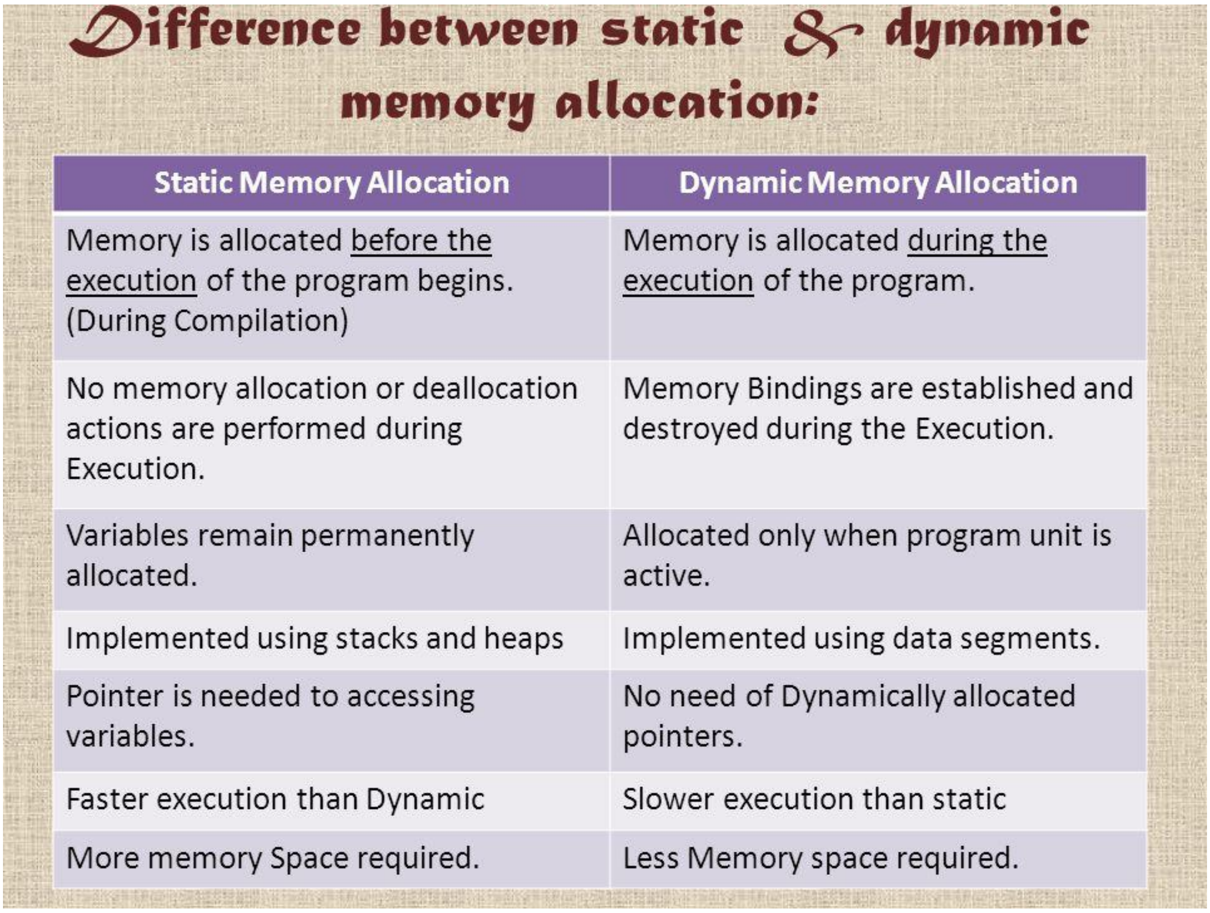
Sed

SED command in UNIX is stands for stream editor and it can perform lot's of function on file like, searching, find and replace, insertion or ...



```
sssit@JavaTpoint:~$ echo class7 | sed 's/class/jtp/'
jtp7
sssit@JavaTpoint:~$ echo class7 | sed 's/7/10/'
class10
sssit@JavaTpoint:~$
sssit@JavaTpoint:~$ cat msg.txt
learn linux, learn fast
linux is very easy to learn.
sssit@JavaTpoint:~$ cat msg.txt | sed 's/learn/study/'
study linux, learn fast
linux is very easy to study.
sssit@JavaTpoint:~$ 
```

Q.26 Explain the difference among static memory allocation and dynamic memory allocation



Difference between static & dynamic memory allocation:

Static Memory Allocation	Dynamic Memory Allocation
Memory is allocated before the execution of the program begins. (During Compilation)	Memory is allocated during the execution of the program.
No memory allocation or deallocation actions are performed during Execution.	Memory Bindings are established and destroyed during the Execution.
Variables remain permanently allocated.	Allocated only when program unit is active.
Implemented using stacks and heaps	Implemented using data segments.
Pointer is needed to accessing variables.	No need of Dynamically allocated pointers.
Faster execution than Dynamic	Slower execution than static
More memory Space required.	Less Memory space required.

Q.28 Explain a process and process table? Define different states of process?

What is process:-

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

While creating a process the operating system performs several operations. To identify the processes, it assigns a process identification number (PID) to each process. As the operating system supports multi-programming, it needs to keep track of all the processes. For this task, the process control block (PCB) is used to track the process's execution status. Each block of memory contains information about the process state, program counter, stack pointer, status of opened files, scheduling algorithms, etc. All these information is required and must be saved when the process is switched from one state to another. When the process makes a transition from one state to another, the operating system must update information in the process's PCB

Process Table:-

The **process table** is a data structure maintained by the operating system to facilitate context switching and scheduling, and other activities discussed later. Each entry in the table, often called a **context block**, contains information about a process such as process name and state (discussed below), priority (discussed below), registers, and a semaphore it may be waiting on (discussed later). The exact contents of a context block depend on the operating system. For instance, if the OS supports paging, then the context block contains an entry to the page table.

Different States of process:-

Processes in the operating system can be in any of the following states:

- NEW- The process is being created.
- READY- The process is waiting to be assigned to a processor.
- RUNNING- Instructions are being executed.
- WAITING- The process is waiting for some event to occur(such as an I/O completion or reception of a signal).
- TERMINATED- The process has finished execution.

Q.29 Define the benefits of multithreaded programming?

- Improved throughput. Many concurrent compute operations and I/O requests within a single process.
- Simultaneous and fully symmetric use of multiple processors for computation and I/O
- Superior application responsiveness. If a request can be launched on its own thread, applications do not freeze or show the "hourglass". An entire application will not block, or otherwise wait, pending the completion of another request.
- Improved server responsiveness. Large or complex requests or slow clients don't block other requests for service. The overall throughput of the server is much greater.
- Minimized system resource usage. Threads impose minimal impact on system resources.
- Program structure simplification. Threads can be used to simplify the structure of complex applications, such as server-class and multimedia applications. Simple routines can be written for each activity, making complex programs easier to design and code, and more adaptive to a wide variation in user demands.
- Better communication. Thread synchronization functions can be used to provide enhanced process-to-process communication. In addition, sharing large amounts of data through separate threads of execution within the same address space provides extremely high-bandwidth, low-latency communication between separate tasks within an application

Q.30 Explain Thrashing?

Thrashing is a condition or a situation when the system is spending a major portion of its time in servicing the page faults, but the actual processing done is very negligible.

Locality Model –
A locality is a set of pages that are actively used together. The locality model states that as a process executes, it moves from one locality to another. A program is generally composed of several different localities which may overlap.

For example when a function is called, it defines a new locality where memory references are made to the instructions of the function call, it's local and global variables, etc. Similarly, when the function is exited, the process leaves this locality.

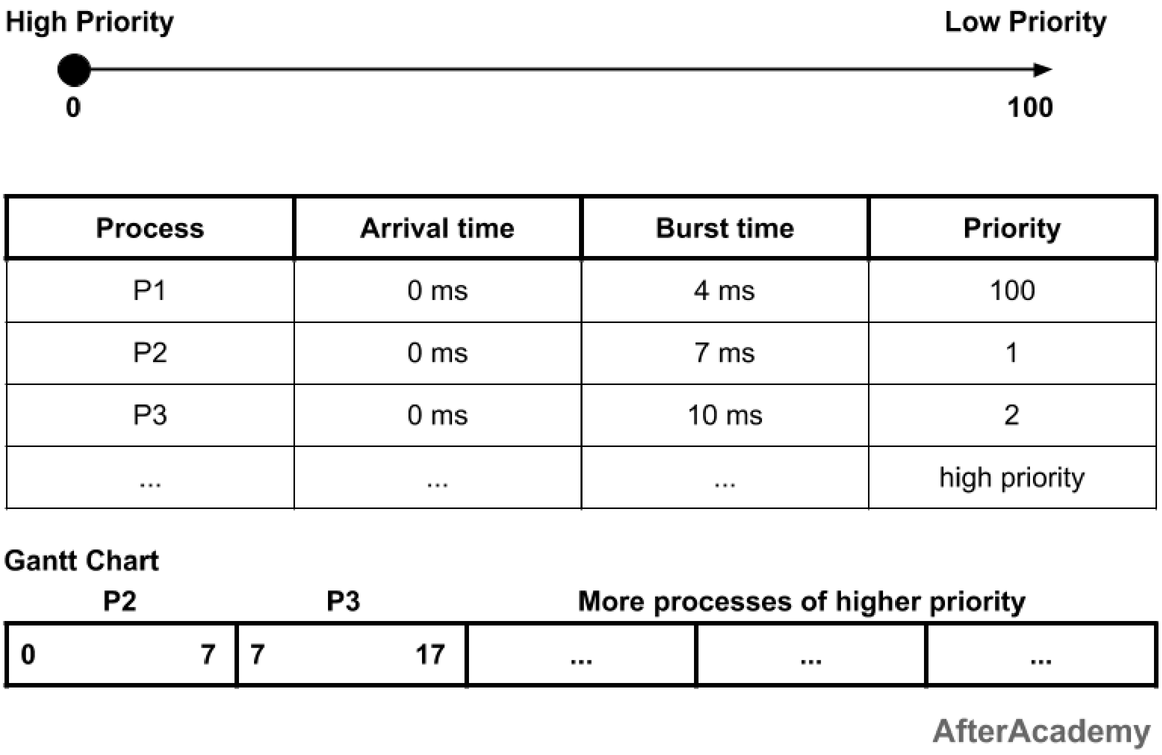
Q.31 Explain Belady's Anomaly?

Prerequisite – Page Replacement Algorithms
In Operating System, process data is loaded in fixed-sized chunks and each chunk is referred to as a page. The processor loads these pages in the fixed-sized chunks of memory called frames. Typically the size of each page is always equal to the frame size.
A page fault occurs when a page is not found in the memory and needs to be loaded from the disk. If a page fault occurs and all memory frames have been already allocated, then replacement of a page in memory is required on the request of a new page. This is referred to as demand-paging. The choice of which page to replace is specified by page replacement algorithms. The commonly used page replacement algorithms are FIFO, LRU, optimal page replacement algorithms, etc.

Q.32 Explain starvation and aging?

Starvation

If you closely look at the concept of Priority scheduling, then you might have noticed one thing. What if the priority of some process is very low and the higher priority processes keep on coming and the CPU is allocated to that higher priority processes and the low priority process keeps on waiting for its turn. Let's have an example:



Process	Arrival time	Burst time	Priority
P1	0 ms	4 ms	100
P2	0 ms	7 ms	1
P3	0 ms	10 ms	2
...	...	...	high priority

Gantt Chart

P2	P3	More processes of higher priority		
0 7	7 17	...	...	...

AfterAcademy

In the above example, the process P2 is having the highest priority and the process P1 is having the lowest priority. In general, we have a number of processes that are in the ready state for its execution. So, as time passes, if only that processes are coming in the CPU that are having a higher priority than the process P1, then the process P1 will keep on waiting for its turn for CPU allocation and it will never get CPU because all the other processes are having higher priority than P1. This is called Starvation.

Aging:

Aging is a technique to avoid starvation in a scheduling system. It works by adding an aging factor to the priority of each request. The aging factor must increase the requests priority as time passes and must ensure that a request will eventually be the highest priority request

Q.33 Explain a trap and trapdoor?

Ans: Trapdoor is a secret undocumented entry point into a program used to grant access without normal methods of access authentication. It is a type of security breach where the designer of a program or a system leaves a hole in software that only he is capable of using A trap is a software interrupt, usually the result of an error condition, and is also a non mask able interrupt and has highest priority

Q.34 Explain a daemon?

A daemon (also known as background processes) is a Linux or UNIX program that runs in the background. Almost all daemons have names that end with the letter "d". For example, httpd the daemon that handles the Apache server, or, sshd which handles SSH remote access connections. Linux often start daemons at boot time. Shell scripts stored in /etc/init.d directory are used to start and stop daemons.

Q.35 which application software's executed on OS?

An application program (app or application for short) is a computer program designed to carry out a specific task other than one relating to the operation of the computer itself typically to be used by end-users. Word processors, media players, and accounting software are examples. The collective noun refers to all applications collectively. The other principal classifications of software are system software, relating to the operation of the computer, and utility software ("utilities").

Applications may be bundled with the computer and its system software or published separately and may be coded as proprietary, open-source, or projects. The term "app" often refers to applications for mobile devices such as phones.

- Program execution
- I/O operations
- File System manipulation
- Communication

- Error Detection
- Resource Allocation
- Protection

Q.36 Define daemon objects and thread objects?

Daemon thread is a low priority thread that runs in background to perform tasks such as garbage collection.

- They can not prevent the JVM from exiting when all the user threads finish their execution.
- JVM terminates itself when all user threads finish their execution
- If JVM finds running daemon thread, it terminates the thread and after that shutdown itself. JVM does not care whether Daemon thread is running or not.
- It is an utmost low priority thread.

Q.37 Give commands for finding process ID

PID refers to process ID, which is commonly used by most operating system kernels, such as Linux, Unix, Mac OS and Windows. This is a unique ID that is automatically assigned to each process when it is created. The default maximum value of PIDs is 32768 on 32-bit machine. And you can set the value higher on 64-bit systems up to 2^22. This can be verified by running the following command:
cat /proc/sys/kernel/pid_max

We can find the PID of processes running on the system using following commands:
1.**pidof**: pidof – find the process ID of a running program.
The pidof command is used to find the process ID of the running program. It prints those IDs into the standard output.
2.**pgrep**: pgrep – look up or signal processes based on name and other attributes.
The pgrep command looks at the processes currently running, and lists the process IDs that match the selection criteria.
3.**ps**: ps – report a snapshot of the current processes.
The ps command displays information about a selection of the active processes which includes the process ID (pid=PID), terminal associated with the process (tname=TTY), cumulated CPU time in [DD-]hh:mm:ss format (time=TIME), and executable name (ucmd=CMD).
4. **pstree**: pstree – display a tree of processes.
The pstree command shows running processes as a tree-like format which is very convenient way to display the process hierarchy and makes the output more visually appealing. If a user name is specified in the pstree command, then it shows all the processes owned by the respective user. pstree visually merges identical branches by putting them in square brackets and prefixing them with the repetition count.
5**. ss**: ss is used to dump socket statistics.
The ss command is used to dump socket statistics. It allows showing information similar to netstat. It can display more TCP and state information than other tools. It can display stats for all kind of sockets such as PACKET, TCP, UDP, DCCP, RAW, Unix domain, etc.
6. **netstat**: netstat is displays a list of open sockets.
The netstat command is used to print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. By default, netstat displays a list of open sockets. If you don’t specify any address families, then the active sockets of all configured address families will be printed. This program is obsolete. Replacement for netstat is ss.

7. **lsof**: lsof – list open files.
The lsof command is used to list open files. The Linux lsof command lists information about files that are open by processes running on the system.
8. **fuser**: fuser – list process IDs of all processes that have one or more files open
The fuser utility shall write to standard output, the process IDs of processes running on the local system that have one or more named files open.
9. **systemctl**: systemctl – Control the systemd system and service manager
The systemctl command is used to control the **systemd** service manager. This is a replacement for the old **SysVinit** system management, and most of the modern Linux operating systems have been moved to the systemd.
10. **top: top** command is used to show the Linux processes. It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.

38. How to edit, rename and move file in Linux?

The **mv** command is used to rename or move files from one location to another. The syntax for the mv command is as follows:
mv source destination
example: to rename the file file1.txt as file2.txt
mv file1.txt file2.txt

39. Give 5 commands in Linux with explanation

1. **pwd Command**
The pwd command is used to display the location of the current working directory.
Syntax: **pwd**
2.touch Command
The touch command is used to create empty files. We can create multiple empty files by executing it once.
Syntax:
touch **<file** name**>**
touch **<file1 name> <file2 name>** ....
3. tac Command
The tac command is the reverse of cat command, as its name specified. It displays the file content in reverse order (from the last line).
Syntax: **tac <file name>**
4. grep Command
The grep is the most powerful and used filter in a Linux system. The 'grep' stands for "**global regular expression print**." It is useful for searching the content from a file. Generally, it is used with the pipe.
Syntax: **grep <searchWord>**
Example: cat file_name | grep search_word
5. **sort Command**
The sort command is used to sort files in alphabetical order.
Syntax: **sort <file name>**

40. Which are deadlock handling situations?

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. It is an undesirable state of the system.

Deadlock can arise due to following conditions:
1.Mutual Exclusion: A resource can be used by only one process at a time. If another process requests for that resource, then the requesting process must be delayed until the resource has been released.

Hold and Wait: Some processes must be holding some resources in non-shareable mode and at the same time must be waiting to acquire some more resources, which are currently held by other processes in non-shareable mode.
2.No Pre-emption: Resources granted to a process can be released back to the system only as a result of voluntary action of that process, after the process has completed its task.
A resource cannot be taken from a process unless the process releases the resource.
3.Circular Wait: Deadlocked processes are involved in a circular chain such that each process holds one or more resources being requested by the next process in the chain.

Methods for handling deadlock
There are three ways to handle deadlock
1.Deadlock prevention or avoidance: The strategy of deadlock prevention is to design the system in such a way that the possibility of deadlock is excluded. We use Banker's algorithm in order to avoid deadlock.
2.Deadlock Avoidance: This approach allows the three necessary conditions of deadlock but makes judicious choices to assure that deadlock point is never reached. It allows more concurrency than avoidance detection. A decision is made dynamically whether the current resource allocation request will, if granted, potentially lead to deadlock. It requires the knowledge of future process requests.

Advantages:
- Not necessary to pre-empt and rollback processes
- Less restrictive than deadlock prevention

Disadvantages:
- Future resource requirements must be known in advance
- Processes can be blocked for long periods
- Exists fixed number of resources for allocation

3.Deadlock detection
Deadlock detection is used by employing and algorithm that tracks the circular waiting and killing one or more processes so that deadlock is removed. The system state is examined periodically to determine if a set of processes is deadlocked. A deadlock is resolved by aborting and restarting a process, relinquishing all the resources that the process held.
- This technique does not limit resources access or restrict process action.
- Requested resources are granted to processes whenever possible.
- It never delays the process initiation and facilitates online handling.
- The disadvantage is the inherent pre-emption losses.