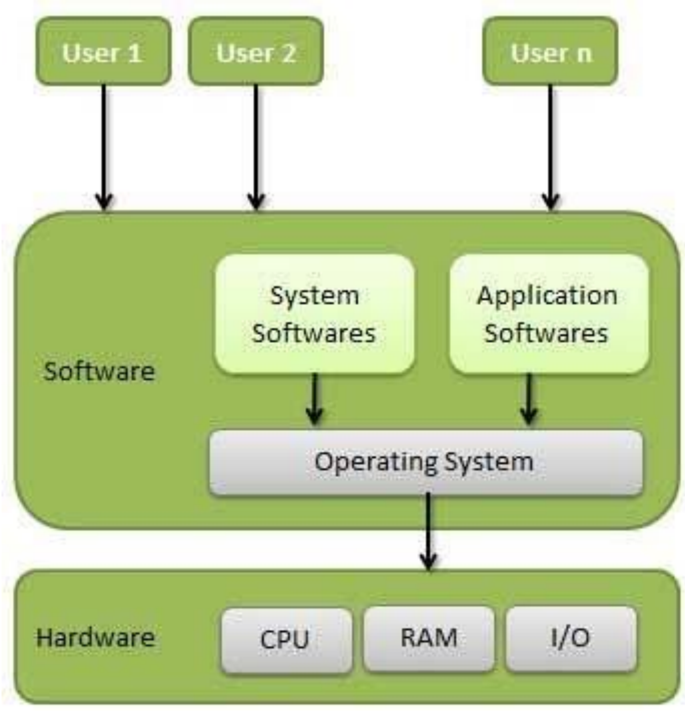
Q.1) DEFINE THE TYPES OF OPERATING SYSTEM ?

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

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

Definition

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



some of the important types of operating systems which are most commonly used.

Batch operating system

The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

The problems with Batch Systems are as follows −

- Lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.

- Difficult to provide the desired priority.

Time-sharing operating systems

Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing.

The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is that in case of Multiprogrammed batch systems, the objective is to maximize processor use, whereas in Time-Sharing Systems, the objective is to minimize response time.

Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response. For example, in a transaction processing, the processor executes each user program in a short burst or quantum of computation. That is, if **n** users are present, then each user can get a time quantum. When the user submits the command, the response time is in few seconds at most.

The operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

Advantages of Timesharing operating systems are as follows −

- Provides the advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

Disadvantages of Time-sharing operating systems are as follows −

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.

Distributed operating System

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

The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as **loosely coupled systems** or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers, and so on.

The advantages of distributed systems are as follows −

- With resource sharing facility, a user at one site may be able to use the resources available at another.
- Speedup the exchange of data with one another via electronic mail.

- If one site fails in a distributed system, the remaining sites can potentially continue operating.
- Better service to the customers.
- Reduction of the load on the host computer.
- Reduction of delays in data processing.

Network operating System

A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions. The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.

Examples of network operating systems include Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

The advantages of network operating systems are as follows −

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

The disadvantages of network operating systems are as follows −

- High cost of buying and running a server.
- Dependency on a central location for most operations.
- Regular maintenance and updates are required.

Real Time operating System

A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. The time taken by the system to respond to an input and display of required updated information is termed as the **response time**. So in this method, the response time is very less as compared to online processing.

Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application. A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail. For example, Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

There are two types of real-time operating systems.

Hard real-time systems

Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM. In these systems, virtual memory is almost never found.

Soft real-time systems

Soft real-time systems are less restrictive. A critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers, etc.

Q.2) EXPLAIN DHCP?

Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol (DHCP) is a network management protocol used to dynamically assign an IP address to nay device, or node, on a network so they can communicate using IP (Internet Protocol). DHCP automates and centrally manages these configurations. There is no need to manually assign IP addresses to new devices. Therefore, there is no requirement for any user configuration to connect to a DHCP based network.

DHCP can be implemented on local networks as well as large enterprise networks. DHCP is the default protocol used by the most routers and networking equipment. DHCP is also called RFC (Request for comments) 2131.

DHCP does the following:

- DHCP manages the provision of all the nodes or devices added or dropped from the network.
- DHCP maintains the unique IP address of the host using a DHCP server.
- It sends a request to the DHCP server whenever a client/node/device, which is configured to work with DHCP, connects to a network. The server acknowledges by providing an IP address to the client/node/device.

DHCP is also used to configure the proper subnet mask, default gateway and DNS server information on the node or device.

There are many versions of DCHP are available for use in IPV4 (Internet Protocol Version 4) and IPV6 (Internet Protocol Version 6).

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.

Components of DHCP

When working with DHCP, it is important to understand all of the components. Following are the list of components:

- **DHCP Server:** DHCP server is a networked device running the DCHP service that holds IP addresses and related configuration information. This is typically a server or a router but could be anything that acts as a host, such as an SD-WAN appliance.
- **DHCP client:** DHCP client is the endpoint that receives configuration information from a DHCP server. This can be any device like computer, laptop, IoT endpoint or anything else that requires connectivity to the network. Most of the devices are configured to receive DHCP information by default.
- **IP address pool:** IP address pool is the range of addresses that are available to DHCP clients. IP addresses are typically handed out sequentially from lowest to the highest.
- **Subnet:** Subnet is the partitioned segments of the IP networks. Subnet is used to keep networks manageable.

- **Lease:** Lease is the length of time for which a DHCP client holds the IP address information. When a lease expires, the client has to renew it.
- **DHCP relay:** A host or router that listens for client messages being broadcast on that network and then forwards them to a configured server. The server then sends responses back to the relay agent that passes them along to the client. DHCP relay can be used to centralize DHCP servers instead of having a server on each subnet.

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?

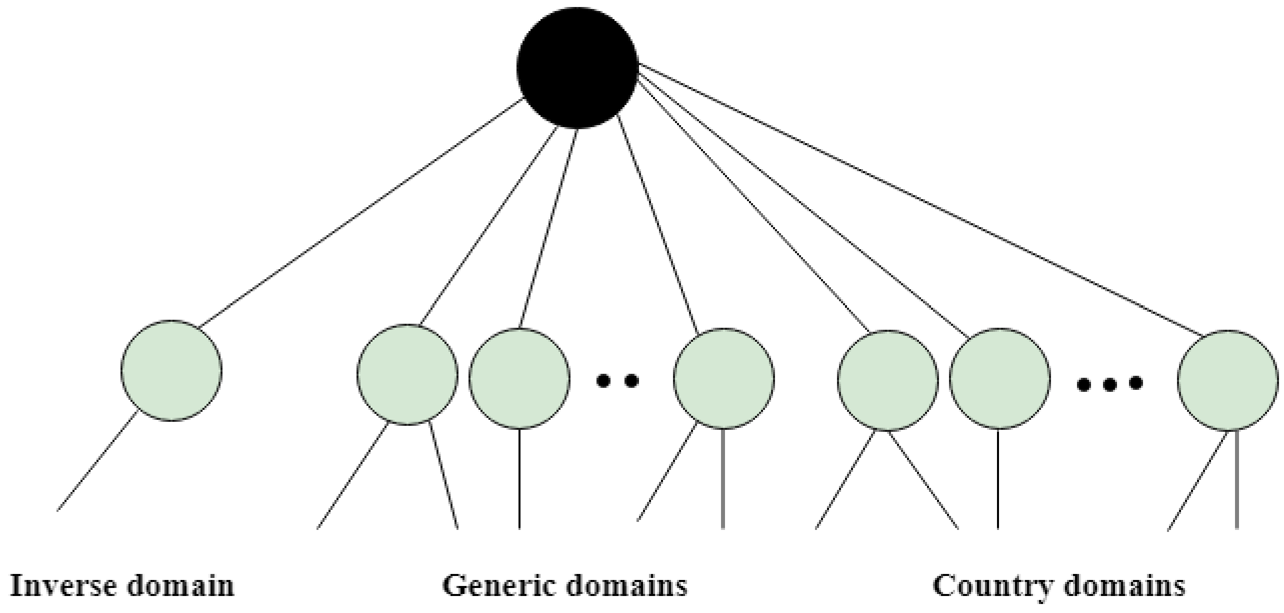
DNS

An application layer protocol defines how the application processes running on different systems, pass the messages to each other.

- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- DNS is required for the functioning of the internet.
- Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.
- DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.

- For example, suppose the FTP site at EduSoft had an IP address of 132.147.165.50, most people would reach this site by specifying ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.

DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.

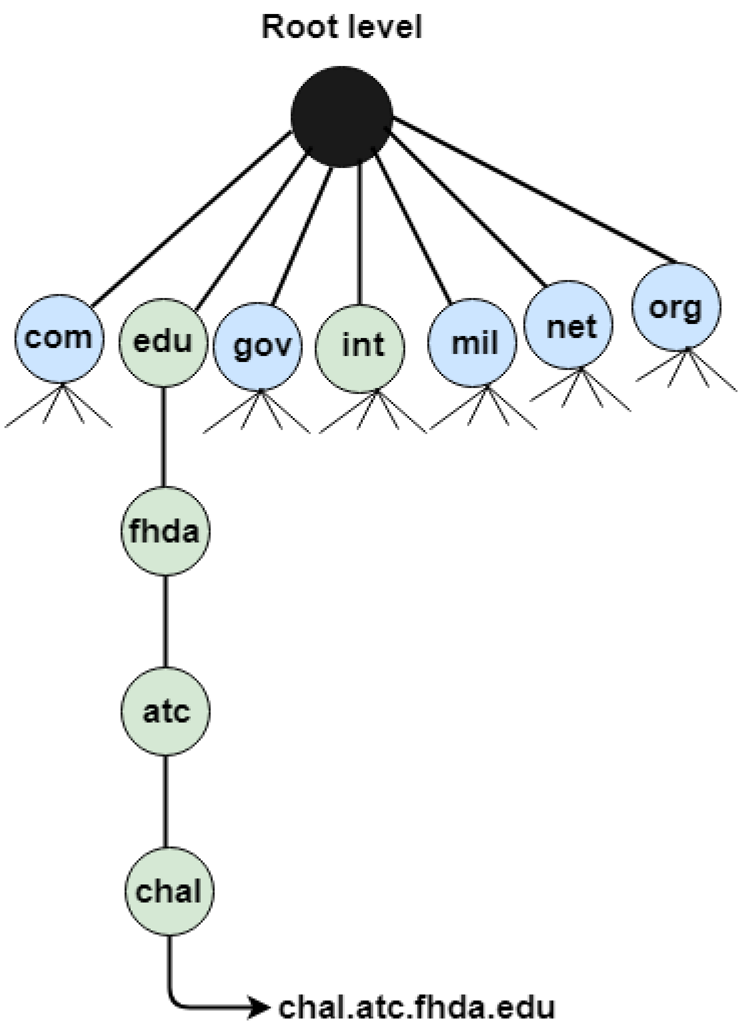


Generic Domains

- It defines the registered hosts according to their generic behavior.
- Each node in a tree defines the domain name, which is an index to the DNS database.
- It uses three-character labels, and these labels describe the organization type.

Label	Description
aero	Airlines and aerospace companies
biz	Businesses or firms
com	Commercial Organizations
coop	Cooperative business Organizations
edu	Educational institutions

gov	Government institutions
info	Information service providers
int	International Organizations
mil	Military groups
museum	Museum & other nonprofit organizations
name	Personal names
net	Network Support centers
org	Nonprofit Organizations
pro	Professional individual Organizations



Country Domain

The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.

Inverse Domain

The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name.

Working of DNS

- DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.

- Client requests contain a name which is converted into an IP address known as a forward DNS lookups while requests containing an IP address which is converted into a name known as reverse DNS lookups.
- DNS implements a distributed database to store the name of all the hosts available on the internet.
- If a client like a web browser sends a request containing a hostname, then a piece of software such as **DNS resolver** sends a request to the DNS server to obtain the IP address of a hostname. If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server. If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

Q.4) EXPLAIN PAGING?

Paging with Example

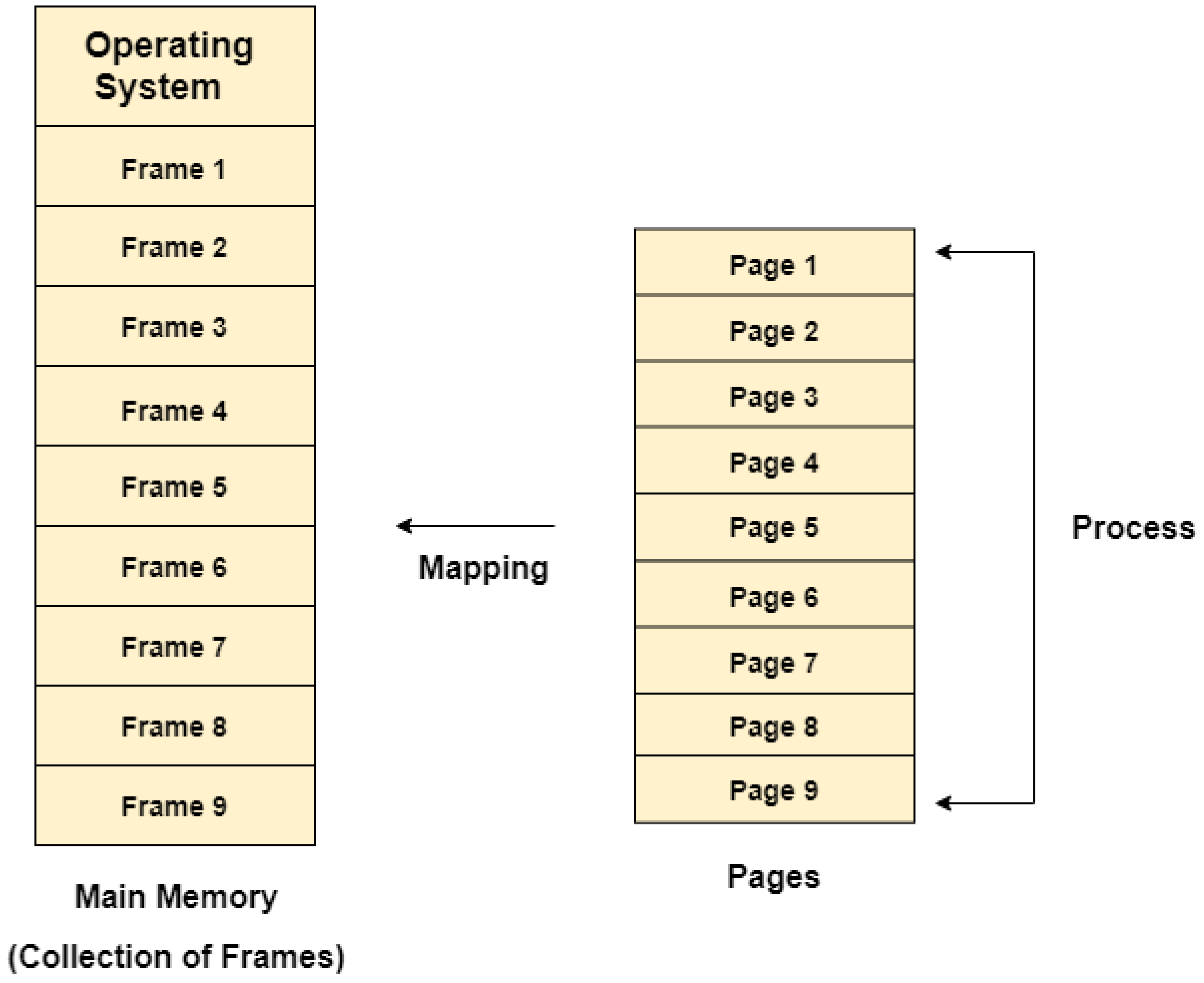
In Operating Systems, Paging is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of pages.

The main idea behind the paging is to divide each process in the form of pages. The main memory will also be divided in the form of frames.

One page of the process is to be stored in one of the frames of the memory. The pages can be stored at the different locations of the memory but the priority is always to find the contiguous frames or holes.

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

Different operating system defines different frame sizes. The sizes of each frame must be equal. Considering the fact that the pages are mapped to the frames in Paging, page size needs to be as same as frame size.



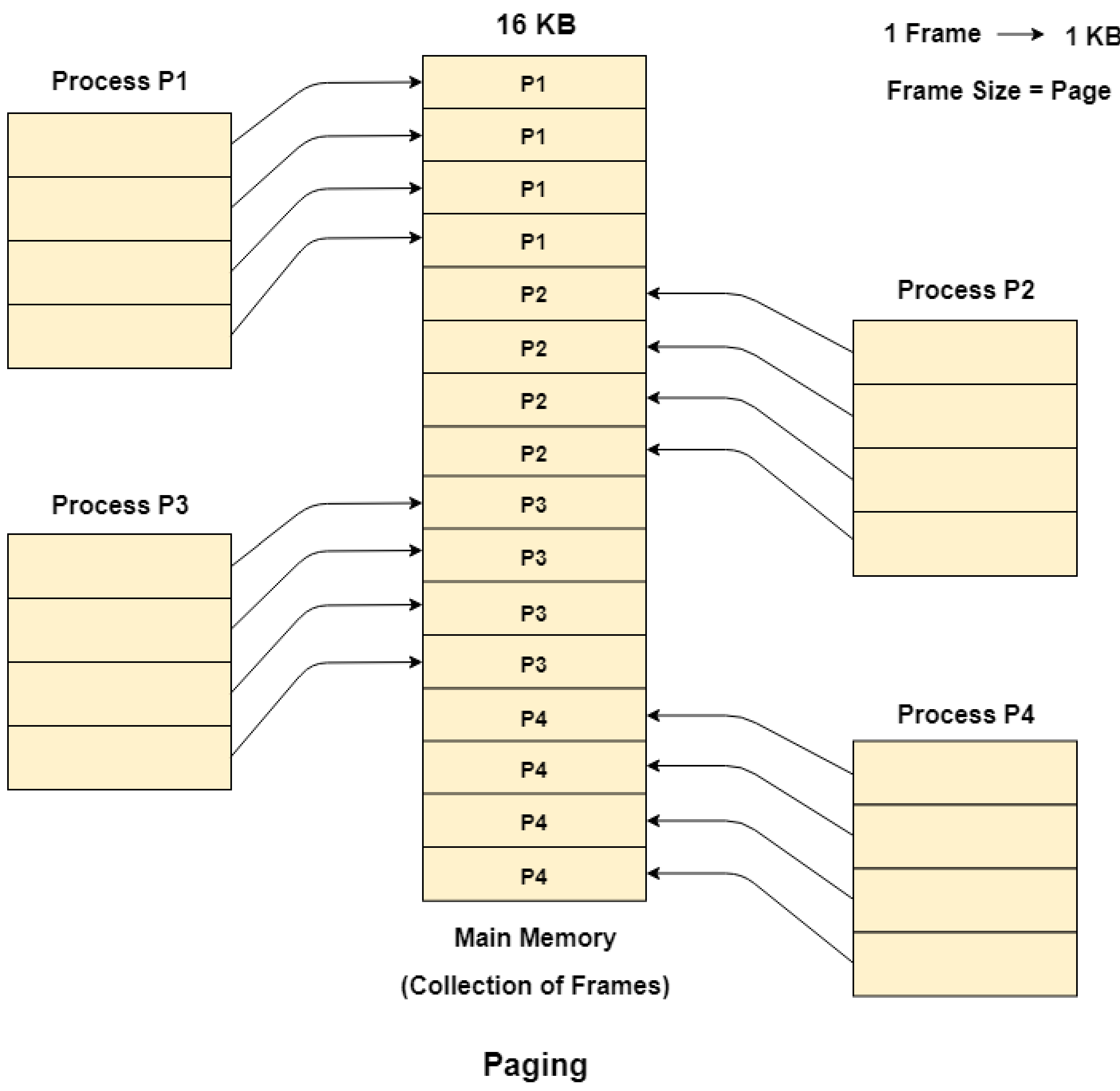
Example

Let us consider the main memory size 16 Kb and Frame size is 1 KB therefore the main memory will be divided into the collection of 16 frames of 1 KB each.

There are 4 processes in the system that is P1, P2, P3 and P4 of 4 KB each. Each process is divided into pages of 1 KB each so that one page can be stored in one frame.

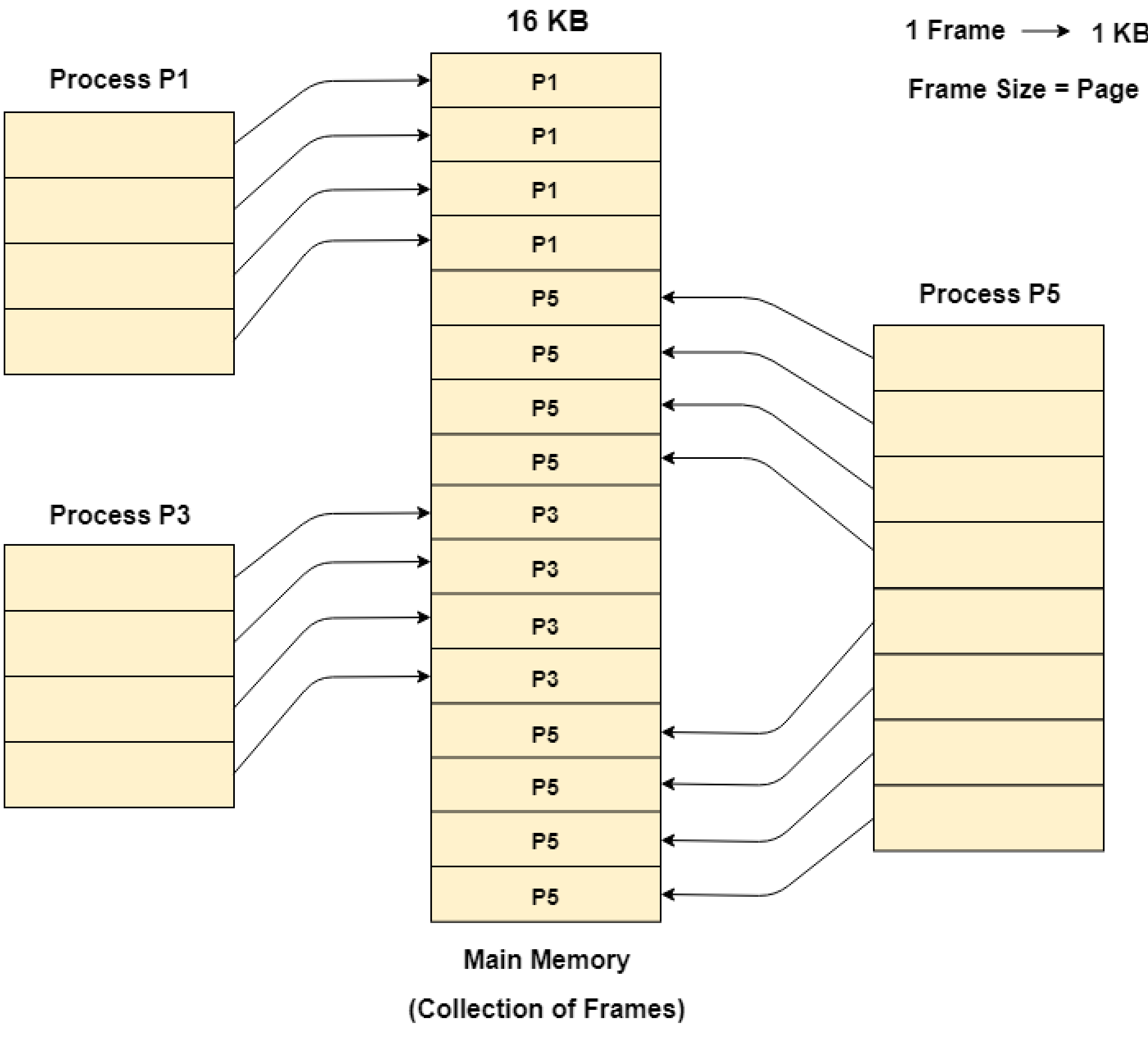
Initially, all the frames are empty therefore pages of the processes will get stored in the contiguous way.

Frames, pages and the mapping between the two is shown in the image below.



Let us consider that, P2 and P4 are moved to waiting state after some time. Now, 8 frames become empty and therefore other pages can be loaded in that empty place. The process P5 of size 8 KB (8 pages) is waiting inside the ready queue.

Given the fact that, we have 8 non contiguous frames available in the memory and paging provides the flexibility of storing the process at the different places. Therefore, we can load the pages of process P5 in the place of P2 and P4.



Paging

Memory Management Unit

The purpose of Memory Management Unit (MMU) is to convert the logical address into the physical address. The logical address is the address generated by the CPU for every page while the physical address is the actual address of the frame where each page will be stored.

When a page is to be accessed by the CPU by using the logical address, the operating system needs to obtain the physical address to access that page physically.

The logical address has two parts.

1. Page Number
2. Offset

Memory management unit of OS needs to convert the page number to the frame number.

Example

Considering the above image, let's say that the CPU demands 10th word of 4th page of process P3. Since the page number 4 of process P1 gets stored at frame number 9 therefore the 10th word of 9th frame will be returned as the physical address.

Q.5 Explain Segmentation?

--> Segmentation is a memory management technique in which the memory is divided into the variable size parts. Each part is known as a segment which can be allocated to a process.The details about each segment are stored in a table called a segment table. Segment table is stored in one or many of the segments.

Segment table contains mainly two information about segment-

Base: It is the base address of the segment

Limit: It is the length of the segment.

Paging divides all the processes into the form of pages regardless of the fact that a process can have some relative parts of functions which need to be loaded in the same page.It may divide the same function into different pages and those pages may or may not be loaded at the same time into the memory. It decreases the efficiency of the system.

Hence,it is better to have segmentation which divides the process into the segments.Each segment contains the same type of functions such as the main function can be included in one segment and the other functions can be included in the other segments.

Q.6 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.

Swapping,paging,segmentation,contiguous memory allocation,Non-contiguous memory allocation,etc all these operations are examples of Memory Management by OS.

Q.7 Explain the function of OS?

--> 1. Input/output Management: What output will come from the input given by the user, the operating system runs this program. This management involves coordinating various input and output devices.

2. Memory Management: The operating system handles the responsibility of storing any data, system programs, and user programs in memory. This function of the operating system is called memory management.

3. File Management: The operating system is helpful in making changes in the stored files and in replacing them. It also plays an important role in transferring various files to a device.

4. Processor Management: The processor is the execution of a program that accomplishes the specified work in that program. It can be defined as an execution unit where a program runs.

5. Job Priority: The work of job priority is creation and promotion. It determines what action should be done first in a computer system.

6. Scheduling of resources and jobs: The operating system prepares the list of tasks to be performed for the device of the computer system.The scheduling programs of the operating system determine the order in which tasks are completed. It performs these tasks based on the priority of performing the tasks given by the user.

7. Security: Computer security is a very important aspect of any operating system. The reliability of an operating system is determined by how much better security it provides us. Modern operating systems use a firewall for security. A firewall is a security system that monitors every activity happening in the computer and blocks that activity in case of any threat.

Q.8 Explain a kernel? Its architecture and working?

--> Kernel is a computer program that is a core or heart of an operating system.It has full control over everything in the system.Each operation of hardware and software is managed and administrated by the kernel.

It is the central component of computer operating systems.It acts as a bridge between applications and data processing done at the hardware level.Not everybody can access kernel data because it is loaded into protected memory of the operating system reserved for only the specified purpose,known as the kernel space.

It is the computer program that first loaded on start-up the system after the bootloader.Once it is loaded, it manages the remaining start-ups. It also manages memory, peripheral, and I/O requests from software.It manages other tasks also such as task management, and disk management.Kernel runs in the background memory until the computer is shut down because it takes care of the various operations. Certain processes need to communicate with the operating system and to do that, they must first send the request to the intermediary between them which is the kernel. These requests are called as system calls which invoke the kernel so that the user’s desired operations are performed.

A shell script is a computer program designed to be run by the Unix shell, a command-line interpreter. The various dialects of shell scripts are considered to be scripting languages. Typical operations performed by shell scripts include file manipulation, program execution, and printing text. A script which sets up the environment, runs the program, and does any necessary cleanup, logging, etc. is called a wrapper. Shell is an environment in which we can run our commands, programs, and shell scripts. In Unix, there are two major types of shells −

Bourne shell − If you are using a Bourne-type shell, the $ character is the default prompt.

C shell − If you are using a C-type shell, the % character is the default prompt.

The Bourne Shell has the following subcategories −Bourne shell (sh)

Korn shell (ksh)

Bourne Again shell (bash)

POSIX shell (sh)

The different C-type shells follow −C shell (csh)

TENEX/TOPS C shell (tcsh)

The original Unix shell was written in the mid-1970s by Stephen R. Bourne while he was at the AT&T Bell Labs in New Jersey. Bourne shell was the first shell to appear on Unix systems, thus it is referred to as "the shell". Bourne shell is usually installed as /bin/sh on most versions of Unix. For this reason, it is the shell of choice for writing scripts that can be used on different versions of Unix.

Page fault occurs when any program tries to access the data or the code that is in the address space of the program, but that data is not currently located in the RAM of the system. So basically when the page referenced by the CPU is not found in the main memory then the situation is termed as Page Fault. Whenever any page fault occurs, then the required page has to be fetched from the secondary memory into the main memory. In case if the required page is not loaded into the memory, then a page fault trap arises

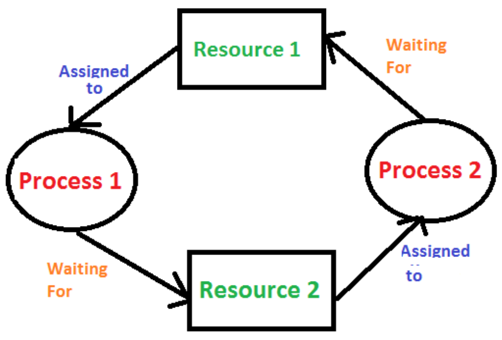
The page fault mainly generates an exception, which is used to notify the operating system that it must have to retrieve the "pages" from the virtual memory in order to continue the execution. Once all the data is moved into the physical memory the program continues its execution normally. The Page fault process takes place in the background and thus goes unnoticed by the user.

A process in operating system uses resources in the following way.

1) Requests a resource 2) Use the resource 3) Releases the resource

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). For example, in the below diagram, Process 1 is holding Resource 1 and waiting for resource 2 which is acquired by process 2, and process 2 is waiting for resource 1.



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

1. Mutual Exclusion- By this condition,

- There must exist at least one resource in the system which can be used by only one process at a time.
- If there exists no such resource, then deadlock will never occur.
- Printer is an example of a resource that can be used by only one process at a time.

2. Hold and Wait- By this condition,

There must exist a process which holds some resource and waits for another resource held by some other process.

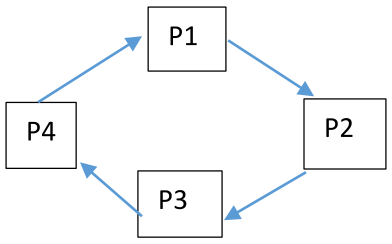
3. No Preemption-By this condition,

- Once the resource has been allocated to the process, it cannot be preempted.
- It means resource cannot be snatched forcefully from one process and given to the other process.
- The process must release the resource voluntarily by itself.

4. Circular Wait- By this condition,

All the processes must wait for the resource in a Cyclic manner where the last process waits for the resource held by the first process.

Here,



Process P1 waits for a resource held by process P2.
Process P2 waits for a resource held by process P3.
Process P3 waits for a resource held by process P4.
Process P4 waits for a resource held by process P1

Q 13. What is semaphore ?

Answser: Semaphore was proposed by Dijkstra in 1965 which is a very significant technique to manage concurrent processes by using a simple integer value, which is known as a semaphore. Semaphore is simply a variable that is non-negative and shared between threads. This variable is used to solve the critical section problem and to achieve process synchronization in the multiprocessing environment.
Semaphores are of two types:

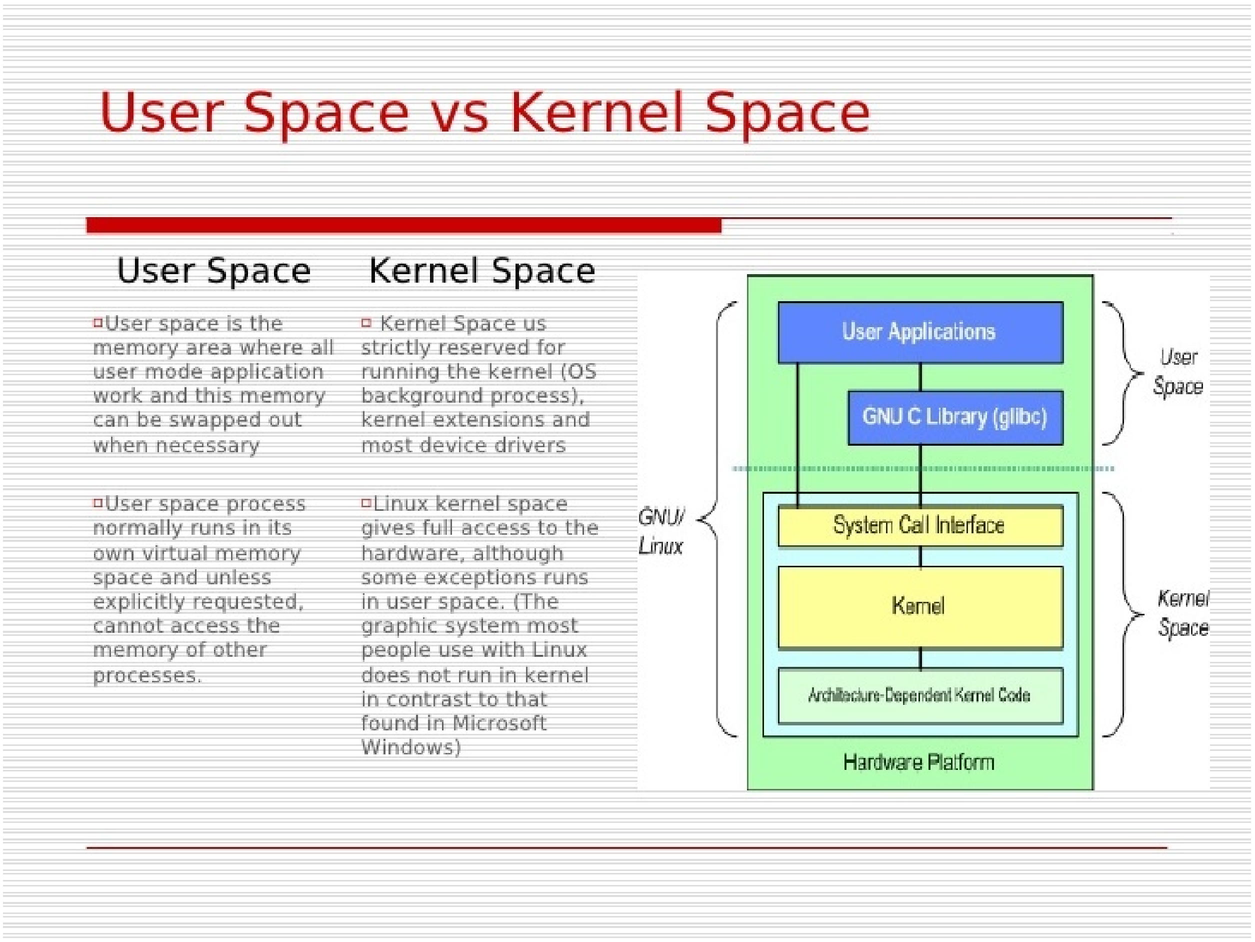
1. **Binary Semaphore –**
   This is also known as mutex lock. It can have only two values – 0 and 1. Its value is initialized to 1. It is used to implement the solution of critical section problems with multiple processes.
2. **Counting Semaphore –**
   Its value can range over an unrestricted domain. It is used to control access to a resource that has multiple instances.

Q14.What is mutex?

Answer: A mutex 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).

Q15.Difference among kernel space and user space?

Answer:



User Space vs Kernel Space

User Space

- User space is the memory area where all user mode application work and this memory can be swapped out when necessary
- User space process normally runs in its own virtual memory space and unless explicitly requested, cannot access the memory of other processes.

Kernel Space

- Kernel Space us strictly reserved for running the kernel (OS background process), kernel extensions and most device drivers
- Linux kernel space gives full access to the hardware, although some exceptions runs in user space. (The graphic system most people use with Linux does not run in kernel in contrast to that found in Microsoft Windows)

Q 16. What is ping command?

Answer: Ping is simple way to find if you are connected to Internet or not. We can also ping any particular computer to find if the computer is connected to the network or not. Press Ctrl+c to stop the loop.

Working:

The Ping utility uses the echo request, and echo reply messages within the Internet Control Message Protocol (ICMP), an integral part of any IP network. When a ping command is issued, an echo request packet is sent to the address specified. When the remote host receives the echo request, it responds with an echo reply packet.
By default, the ping command sends several echo requests, typically four or five. The result of each echo request is displayed, showing whether the request received a successful response, how many bytes were received in response, the Time to Live (TTL), and how long the response took to receive, along with statistics about packet loss and round trip times.

Syntax:

```
Ping 172.168.9.13
```

Exentions:

-t	Pings the specified host until stopped. To stop - type Control-C
-a	Resolve adresses to hostnames
-n	Number of echo requests to send
-l	Send buffer size
-f	Set *Don't Fragmet flag* in packet (IPv4-only)
-i	Set *Time To Live*
-v	Set *Type of Service* (Setting has been deprecated)
-r	Record route for count hops (IPv4-only)
-s	Timestamp for count hops (IPv4-only)
-j	Loose source route along host-list (IPv4-only)
-k	Strict source route along host-list (IPv4-only)
-w	Timeout in milliseconds to wait for each reply
-R	Use routing header to test reverse route also (IPv6-only, deprecated per RFC 5095)
-S	Source address to use
-c	Routing compartment identifier
-p	Ping a Hyper-V Network Virtualization provider address
-4	Force using IPv4
-6	Force using Ipv6

17. Explain unix.

What is 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

Types of UNIX

There are many different versions of UNIX, although they share common similarities. The most popular varieties of UNIX are Sun Solaris, GNU/Linux, and MacOS

The UNIX operating system

The UNIX operating system is made up of three parts; the kernel, the shell and the programs.

- The kernel

The kernel of UNIX is the hub of the operating system: it allocates time and memory to programs and handles the filestore and communications in response to system calls. As an illustration of the way that the shell and the kernel work together, suppose a user types **rm myfile** (which has the effect of removing the file **myfile**). The shell searches the filestore for the file containing the program **rm**, and then requests the kernel, through system calls, to execute the program **rm** on **myfile**

- The shell

The shell acts as an interface between the user and the kernel. When a user logs in, the login program checks the username and password, and then starts another program called the shell. The shell is a command line interpreter (CLI). It interprets the commands the user types in and arranges for them to be carried out. The commands are themselves programs: when they terminate, the shell gives the user another prompt (% on our systems).

- Files and processes

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?

Grep is an acronym that stands for Global Regular Expression Print.

Grep is an essential Linux and Unix command. It is used to search text and strings in a given file. In other words, grep command searches the given file for lines containing a match to the given strings or words. When it finds a match, it prints the line with the result. The grep command is handy when searching through large log files.

- **How to use grep**

1. We can use fgrep/grep to find all the lines of a file that contain a particular word. For example, to list all the lines of a file named address.txt in the current directory that contain the word “California”,

```
grep California address.txt
```

Please note that the above command also returns lines where “California” is part of other words, such as “Californication” or “Californian”.

2. To Find Whole Words Only

the `-w` option with the grep/fgrep command to get only lines where “California” is included as a whole word:

```
fgrep -w California address.txt
```

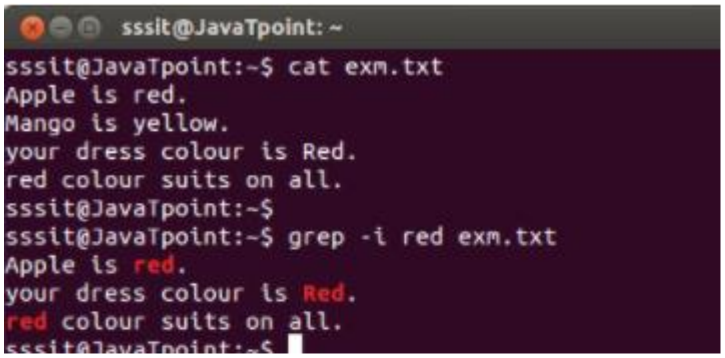
3. How to use grep to search 2 different words

```
$ egrep -w 'word1|word2' /path/to/file
```

4. To Ignore Case in Grep Searches

As grep commands are case sensitive, one of the most useful operators for grep searches is **-i.** Instead of printing lowercase results only, the terminal displays both uppercase and lowercase results.

```
grep -i red exm.txt
```

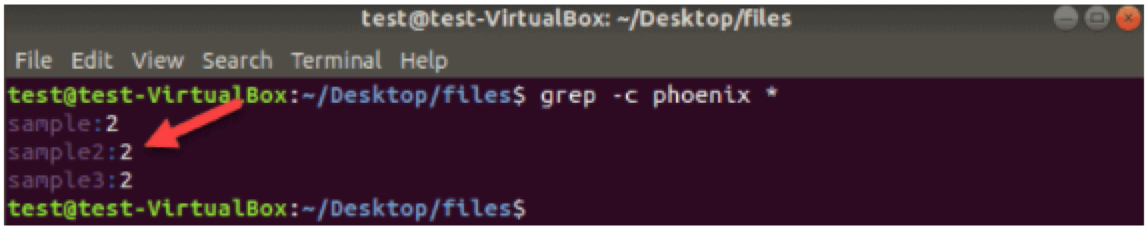


```
sssit@JavaTpoint:~$ cat exm.txt
Apple is red.
Mango is yellow.
your dress colour is Red.
red colour suits on all.
sssit@JavaTpoint:~$
sssit@JavaTpoint:~$ grep -i red exm.txt
Apple is red.
your dress colour is Red.
red colour suits on all.
sssit@JavaTpoint:~$
```

5. How can I count line when words has been matched

The grep can report the number of times that the pattern has been matched for each file using -c (count) option:

$ grep -c phoenix /path/to/file



```
test@test-VirtualBox:~/Desktop/files$ grep -c phoenix *
sample:2
sample2:2
sample3:2
test@test-VirtualBox:~/Desktop/files$
```

6. Inverse grep Search

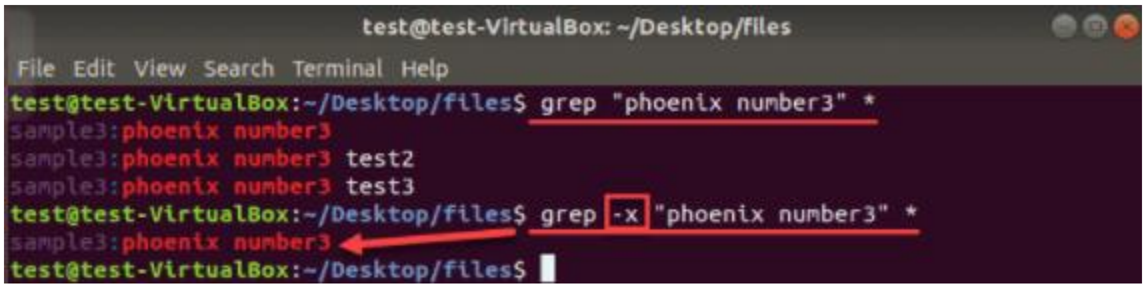
You can use grep to print all lines that **do not** match a specific pattern of characters. To invert the search, append **-v** to a grep command.

grep -v phoenix sample

7. To Show Lines That Exactly Match a Search String

The grep command prints entire lines when it finds a match in a file. To print only those lines that completely match the search string, add the ***-x*** option.

grep -x “phoenix number3”



```
test@test-VirtualBox:~/Desktop/files$ grep "phoenix number3" *
sample3:phoenix number3
sample3:phoenix number3 test2
sample3:phoenix number3 test3
test@test-VirtualBox:~/Desktop/files$ grep -x "phoenix number3" *
sample3:phoenix number3
test@test-VirtualBox:~/Desktop/files$
```

The output shows only the lines with the exact match. If there are any other words or characters in the same line, the grep does not include it in the search results.

8. To Display the Number of Lines Before or After a Search String

Sometimes you need more content in search results to decide what is most relevant.

Use the following operators to add the desired lines before, after a match, or both:

- Use ***-A*** and a number of lines to display after a match: grep -A 3 phoenix sample - this command prints three lines after the match.
- Use ***-B*** and a number of lines to display before a match: grep -B 2 phoenix sample - this command prints two lines before the match.
- Use ***-C*** and a number of lines to display before **and** after the match: grep -C 2 phoenix sample - this command prints two lines before and after the match.

19.Explain pipe?

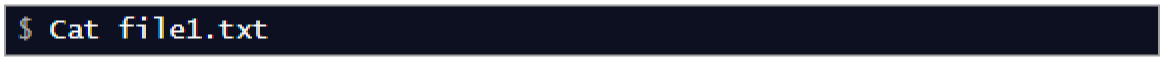
Pipes help combine two or more commands and are used as input/output concepts in a command. In the Linux operating system, we use more than one pipe in command so that the output of one command before a pipe acts as input for the other command after the pipe.

Syntax

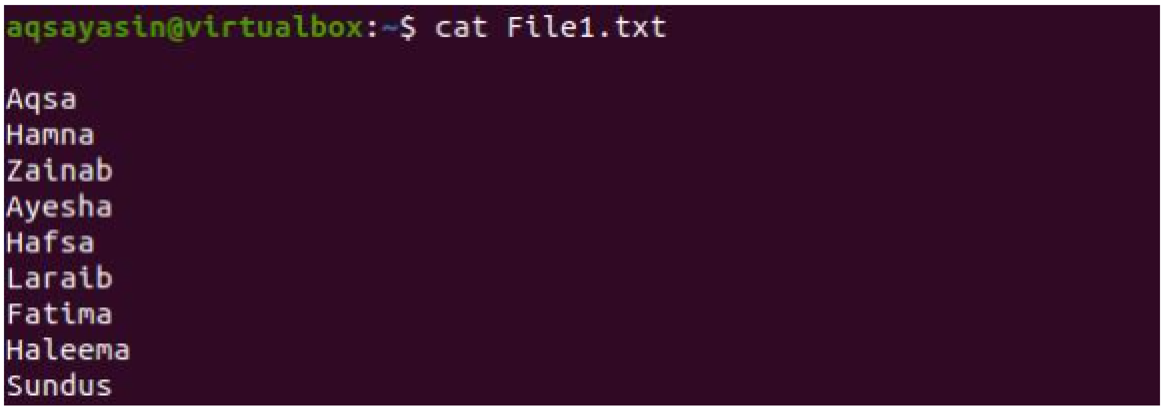
Command 1 | command 2 | command 3 | ……

Sort the list using pipes

1. Suppose we have a file named file1.txt having the names of the students. We have used the cat command to fetch the record of that file.

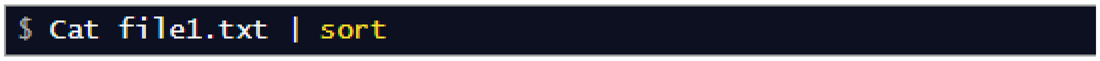


```
$ Cat file1.txt
```

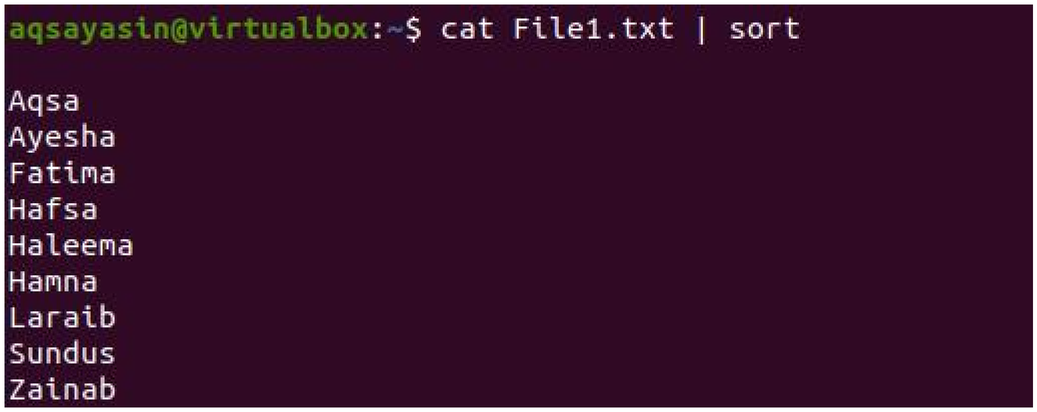


```
aqsayasin@virtualbox:~$ cat File1.txt

Aqsa
Hamna
Zainab
Ayesha
Hafsa
Laraib
Fatima
Haleema
Sundus
```

The data present in this file is unordered. So, to sort the data, we need to follow a piece of code here



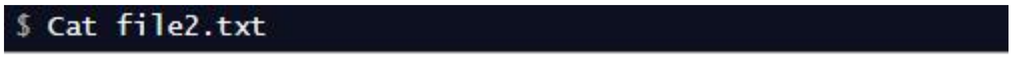
```
$ Cat file1.txt | sort
```



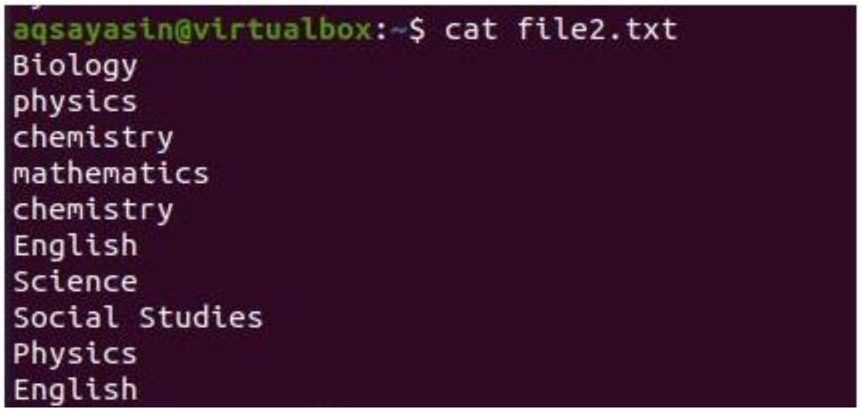
```
aqsayasin@virtualbox:~$ cat File1.txt | sort

Aqsa
Ayesha
Fatima
Hafsa
Haleema
Hamna
Laraib
Sundus
Zainab
```

Through the respective output, you can see that students' names are arranged alphabetically in a sequence from a to z.

2. Let's consider a file named file2.txt having the names of subjects in it. The same command is used for fetching data.

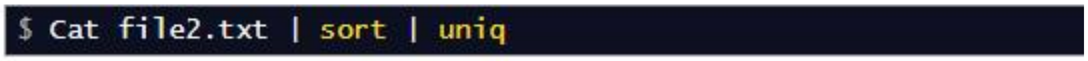


```
$ Cat file2.txt
```

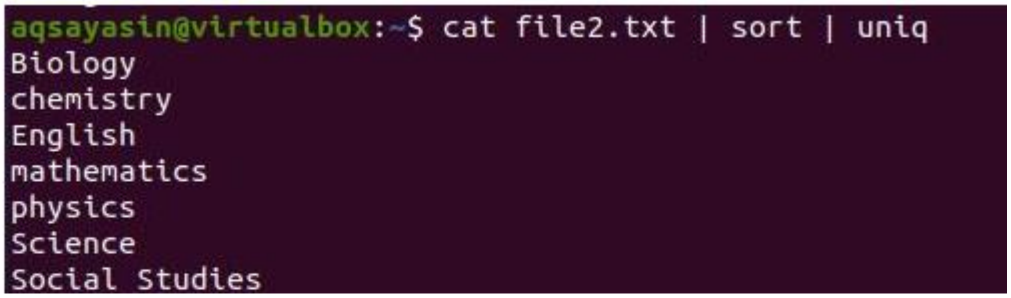


```
aqsayasin@virtualbox:~$ cat file2.txt
Biology
physics
chemistry
mathematics
chemistry
English
Science
Social Studies
Physics
English
```

Now we will use the command to remove all the words that are duplicated in the file



```
$ Cat file2.txt | sort | uniq
```

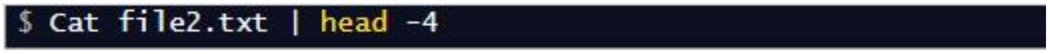


```
aqsayasin@virtualbox:~$ cat file2.txt | sort | uniq
Biology
chemistry
English
mathematics
physics
Science
Social Studies
```

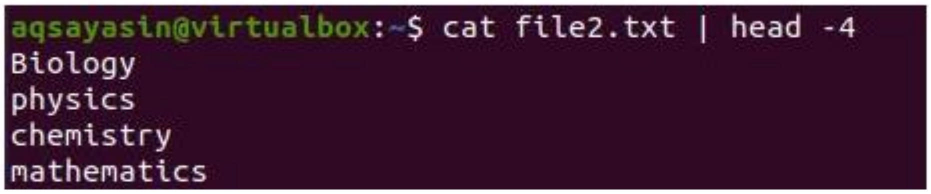
The output shows that the elements are organized and arranged alphabetically. At the same time, all the words that were duplicated are removed

3. **Display file data of a corresponding range**

in this example, we have declared the range up to 4. So the data will be from the first 4 lines of the file. Consider the same file file2.txt as we have taken an example above.



```
$ Cat file2.txt | head -4
```



```
aqsayasin@virtualbox:~$ cat file2.txt | head -4
Biology
physics
chemistry
mathematics
```

Similar to head, we can also use the tail option. This will limit the output to the last lines according to the range given.

20.Difference among thread and process.

What is Process?

A process is an instance of a program that is being executed. When we run a program, it does not execute directly. It takes some time to follow all the steps required to execute the program, and following these execution steps is known as a process.

A process can create other processes to perform multiple tasks at a time; the created processes are known as **clone or child process**, and the main process is known as the **parent process**. Each process contains its own memory space and does not share it with the other processes. It is known as the active entity. A typical process remains in the below form in memory.

What is Thread?

A thread is the subset of a process and is also known as the lightweight process. A process can have more than one thread, and these threads are managed independently by the scheduler. All the threads within one process are interrelated to each other. Threads have some common information, such as **data segment, code segment, files, etc.,** that is shared to their peer threads. But contains its own registers, stack, and counter.

Comparison Basis	Process	Thread
Definition	A process is a program under execution i.e an active program.	A thread is a lightweight process that can be managed independently by a scheduler.
Context switching time	Processes require more time for context switching as they are more heavy.	Threads require less time for context switching as they are lighter than processes.
Memory Sharing	Processes are totally independent and don't share memory.	A thread may share some memory with its peer threads.
Communication	Communication between processes requires more time than between threads.	Communication between threads requires less time than between processes .
Blocked	If a process gets blocked, remaining processes can continue execution.	If a user level thread gets blocked, all of its peer threads also get blocked.
Resource Consumption	Processes require more resources than threads.	Threads generally need less resources than processes.

Comparison Basis	**Process**	**Thread**
Dependency	Individual processes are independent of each other.	Threads are parts of a process and so are dependent.
Data and Code sharing	Processes have independent data and code segments.	A thread shares the data segment, code segment, files etc. with its peer threads.
Treatment by OS	All the different processes are treated separately by the operating system.	All user level peer threads are treated as a single task by the operating system.
Time for creation	Processes require more time for creation.	Threads require less time for creation.
Time for termination	Processes require more time for termination.	Threads require less time for termination.

21. Explain a scheduling algorithm?

A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms. There are six popular process scheduling algorithms which we are going to discuss in this chapter −

- First-Come, First-Served (FCFS) Scheduling
- Shortest-Job-Next (SJN) Scheduling
- Priority Scheduling
- Shortest Remaining Time
- Round Robin(RR) Scheduling
- Multiple-Level Queues Scheduling

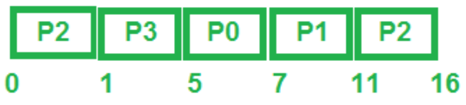
These algorithms are either **non-preemptive or preemptive**. Non-preemptive algorithms are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time, whereas the preemptive scheduling is based on priority where a scheduler may preempt a low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

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

1. Preemptive Scheduling:

Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to ready state. The resources (mainly CPU cycles) are allocated to the process for a limited amount of time and then taken away, and the process is again placed back in the ready queue if that process still has CPU burst time remaining. That process stays in the ready queue till it gets its next chance to execute.
Algorithms based on preemptive scheduling are: Round Robin (RR),Shortest Remaining Time First (SRTF), Priority (preemptive version), etc.

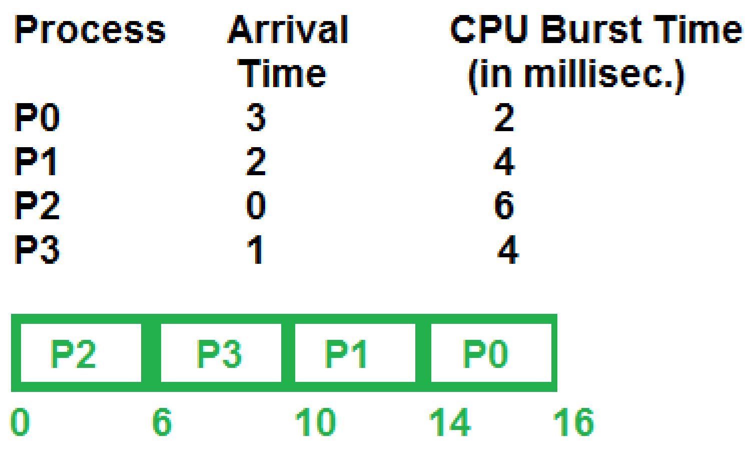
Process	Arrival Time	CPU Burst Time (in millisec.)
P0	3	2
P1	2	4
P2	0	6
P3	1	4



Preemptive Scheduling

2. Non-Preemptive Scheduling:
Non-preemptive Scheduling is used when a process terminates, or a process switches from running to the waiting state. In this scheduling, once the resources (CPU cycles) are allocated to a process, the process holds the CPU till it gets terminated or reaches a waiting state. In the case of non-preemptive scheduling does not interrupt a process running CPU in the middle of the execution. Instead, it waits till the process completes its CPU burst time, and then it can allocate the CPU to another process.

Algorithms based on non-preemptive scheduling are: Shortest Job First (SJF basically non preemptive) and Priority (non preemptive version), etc.



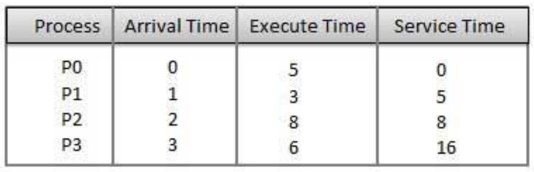
Process	Arrival Time	CPU Burst Time (in millisec.)
P0	3	2
P1	2	4
P2	0	6
P3	1	4

Non-Preemtive Scheduling

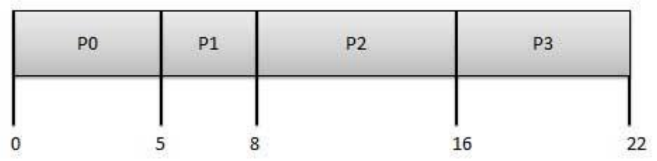
23. Define the different scheduling algorithms

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.



Process	Arrival Time	Execute Time	Service Time
P0	0	5	0
P1	1	3	5
P2	2	8	8
P3	3	6	16



Wait time of each process is as follows −

Process	Wait Time : Service Time - Arrival Time
P0	0 - 0 = 0
P1	5 - 1 = 4
P2	8 - 2 = 6
P3	16 - 3 = 13

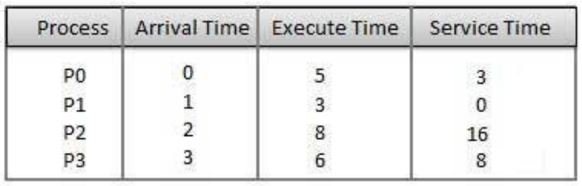
Average Wait Time: (0+4+6+13) / 4 = 5.75

Shortest Job Next (SJN)

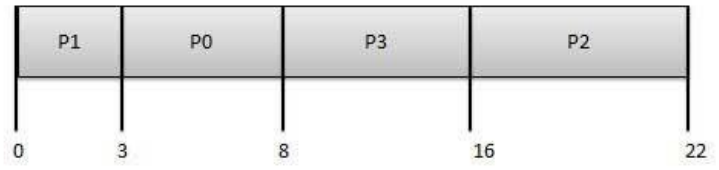
- This is also known as **shortest job first**, or SJF
- This is a non-preemptive, pre-emptive scheduling algorithm.
- Best approach to minimize waiting time.
- Easy to implement in Batch systems where required CPU time is known in advance.
- Impossible to implement in interactive systems where required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

Given: Table of processes, and their Arrival time, Execution time

Process	Arrival Time	Execution Time	Service Time
P0	0	5	0
P1	1	3	5
P2	2	8	14
P3	3	6	8



Process	Arrival Time	Execute Time	Service Time
P0	0	5	3
P1	1	3	0
P2	2	8	16
P3	3	6	8



Waiting time of each process is as follows −

Process	Waiting Time
P0	0 - 0 = 0
P1	5 - 1 = 4
P2	14 - 2 = 12
P3	8 - 3 = 5

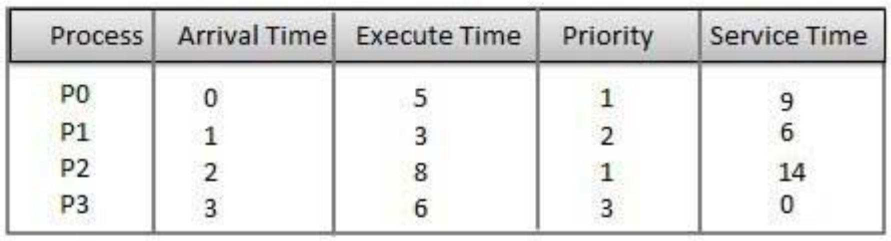
Average Wait Time: (0 + 4 + 12 + 5)/4 = 21 / 4 = 5.25

Priority Based Scheduling

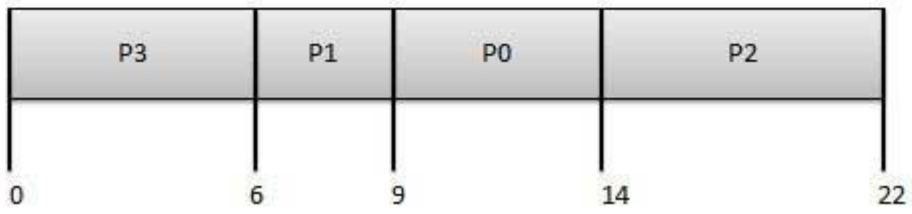
- Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems.
- Each process is assigned a priority. Process with highest priority is to be executed first and so on.
- Processes with same priority are executed on first come first served basis.
- Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Given: Table of processes, and their Arrival time, Execution time, and priority. Here we are considering 1 is the lowest priority.

Process	**Arrival Time**	**Execution Time**	**Priority**	**Service Time**
P0	0	5	1	0
P1	1	3	2	11
P2	2	8	1	14
P3	3	6	3	5



Process	Arrival Time	Execute Time	Priority	Service Time
P0	0	5	1	9
P1	1	3	2	6
P2	2	8	1	14
P3	3	6	3	0



Waiting time of each process is as follows −

Process	Waiting Time
P0	0 - 0 = 0
P1	11 - 1 = 10
P2	14 - 2 = 12
P3	5 - 3 = 2

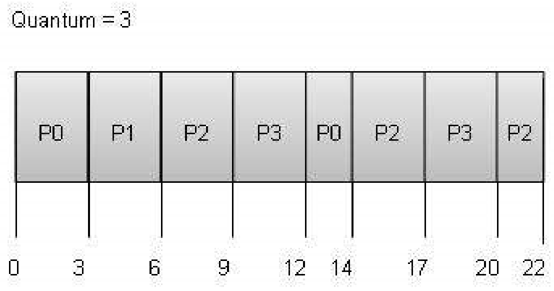
Average Wait Time: (0 + 10 + 12 + 2)/4 = 24 / 4 = 6

Shortest Remaining Time

- Shortest remaining time (SRT) is the preemptive version of the SJN algorithm.
- The processor is allocated to the job closest to completion but it can be preempted by a newer ready job with shorter time to completion.
- Impossible to implement in interactive systems where required CPU time is not known.
- It is often used in batch environments where short jobs need to give preference.

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.



Wait time of each process is as follows −

Process	Wait Time : Service Time - Arrival Time
P0	(0 - 0) + (12 - 3) = 9
P1	(3 - 1) = 2
P2	(6 - 2) + (14 - 9) + (20 - 17) = 12
P3	(9 - 3) + (17 - 12) = 11

Average Wait Time: (9+2+12+11) / 4 = 8.5

Multiple-Level Queues Scheduling

Multiple-level queues are not an independent scheduling algorithm. They make use of other existing algorithms to group and schedule jobs with common characteristics.

- Multiple queues are maintained for processes with common characteristics.
- Each queue can have its own scheduling algorithms.
- Priorities are assigned to each queue.

For example, CPU-bound jobs can be scheduled in one queue and all I/O-bound jobs in another queue. The Process Scheduler then alternately selects jobs from each queue and assigns them to the CPU based on the algorithm assigned to the queue

24. Explain booting process?

When our computer is switched on, It can be started by hardware such as a button press, or by software command, a computer's central processing unit (CPU) has no software in its main memory, there is some process which must load software into main memory before it can be executed.

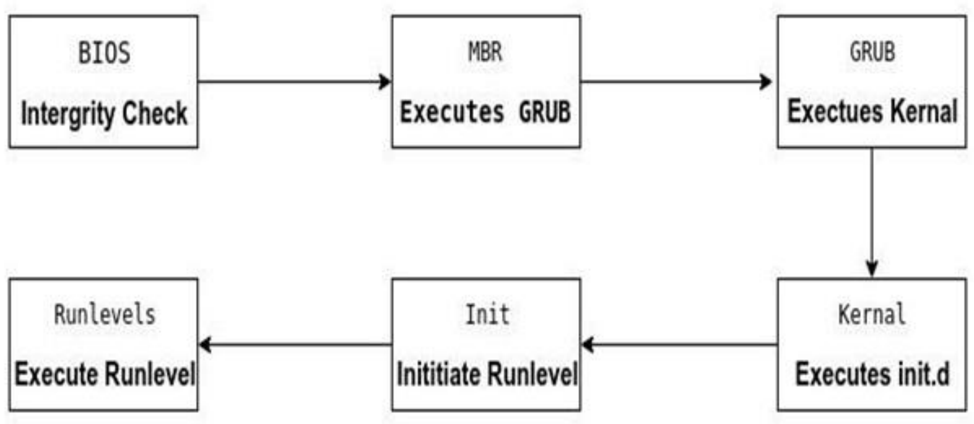
After the computer is turned on, a specific program needs to load in the computer's main memory which is Operating System.

In the **Booting process**, System will check all the hardware's and Software's those are installed or attached with the system and all the Files those are required for running a system, also loads into memory.

ROM also reads the information stored in these files.

At the Time of **Booting** all Instructions will be read required to start the system.

OS holds the following processes at the time of **booting**:



Categories of Booting Process

There are two **categories of Booting process**,

1. Hard (Cold) Booting
2. Soft (Warm) Booing

1) Hard (Cold) Booting

Restart a computer is referred to as rebooting, It can be "hard" or "cold", e.g. after power to the CPU is switched from off to on. In this category of **booting** the computer starts from a completely dead state.

For instance, When we press the Power Button, the system starts with its initial state. It reads all the information that is stored in the Read-Only Memory (ROM) and automatically the Operating System will be loaded into the system's main memory.

2) Soft (Warm) Booing

In "soft" or "warm" **booting**, the power is not cut. In some systems, a soft boot may optionally clear RAM to zero.

POST (Power On Self Test)

- The Power On Self Test happens each time after turn on your computer.
- It is a diagnostic testing sequence for all the hardware devices. It checks hardware device availability. It sends a signal to all the hardware devices and they send an acknowledgment back to it. If an acknowledgment is sent by that device then that device is working, if not then it will be removed from the system.
- If there is any error then a beep like sound generates or some error messages displays on the monitor. These beeps are referred to as POST beep codes.

Master Boot Record

- The Master Boot Record (MBR) is the information which is in the first sector of any hard disk that indicates how and where an operating system is located so that it can be boot (loaded) into the computer's main memory or random access memory.
- The MBR is also sometimes referred to as the "partition sector" or the "master partition table". It has only four primary partitions. We can create more partitions by setting the fourth partition as the extended partition and also we can create sub-partitions (or logical drives) within it.
- It holds information about GRUB (or LILO in old systems).
- Its size is less than 512 bytes. It has three components
    1. primary boot loader information in 1st 446 bytes.
    2. partition table information in next 64 bytes.
    3. MBR validation checks in the last 2 bytes.

init

This is the last step of the **booting process**. It decides the run level by looking at the / etc / inittab file.

- The initial state of the operating system decides by the run level.
- Following are the run levels of Operating System:
  Level
  0 - System Halt
  1 - Single user mode
  2 - Multiuser, without NFS
  3 - full multiuser mode

4 - unused
5 - Full multiuser mode with network and X display manager(X11)
6 - Reboot

- We would set the default run level to either 3 or 5.
- We can execute **'grep initdefault / etc/ inittab'** on your system to identify the default run level.
- Init uses run levels to load all appropriate program.

The step after is to start up various daemons that support networking and other services. X server daemon manages display, keyboard, and mouse. You can see a Graphical Interface and a login screen is displayed during X server daemon is started.

Failure during boot

If the computer cannot boot, we will get a boot failure error. This error indicates that the computer is not passing POST or a device in the computer, such as the hard drive or memory, has failed.

You may also hear a beep code to identify which hardware is failing during the POST.

An error message or blue screen may show on the screen as operating system files cannot be loaded, due to not being found or being corrupt.

25. Explain bias?

==>>to computer systems that systematically and unfairly discriminate against certain individuals or groups of individuals in favour of others.

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

==>> In static memory allocation whenever the program executes it fixes the size that the program is going to take, and it can't be changed further. So, the exact memory requirements must be known before. Allocation and deallocation of memory will be done by the compiler automatically. When everything is done at compile time (or) before run time, it is called static memory allocation.. In Dynamic memory allocation size initialization and allocation are done by the programmer. It is managed and served with pointers that point to the newly allocated memory space in an area which we call the heap. Heap memory is unorganized and it is treated as a resource when you require the use of it if not release it. When everything is done during run time or execution time it is known as Dynamic memory allocation.

For more-https://youtu.be/7icNeEZ8PDo

27. UNIX commands like touch, sed, grep.

==>> 1. touch command: It is used to create a file without any content. The file created using touch command is empty. This command can be used when the user doesn't have data to store at the time of file creation.

2.Sed Command: 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 deletion. Though most common use of SED command in UNIX is for substitution or for find and replace.

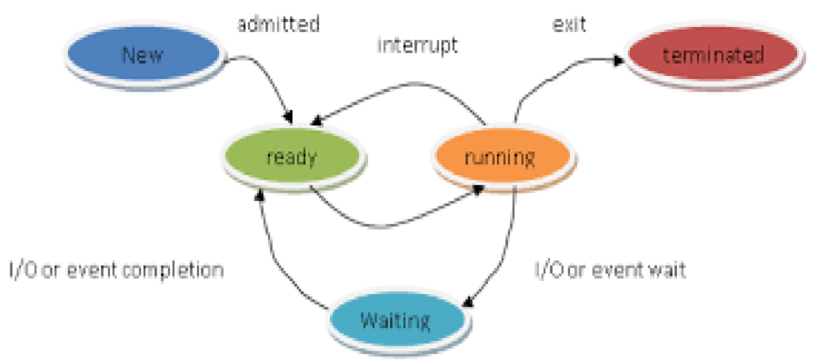
3.grep Command: The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for globally search for regular expression and print out).

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

Process-a process is the instance of a computer program that is being executed by one or many threads.

Process table-The process table is a data structure maintained by the operating system to facilitate context switching and scheduling, and other activities.

States of Process:-



1.New: When PCB get created in Karnal space.

2.Ready: When Process is in main memory and waiting for CPU time.

3.Running: CPU Executing any process.

4.Waiting: Process requesting any i/o devices.

5.Terminated: Upon exit, Process goes into terminated state & PCB get destroyed.

29) Define the benefits of multithreaded programming?

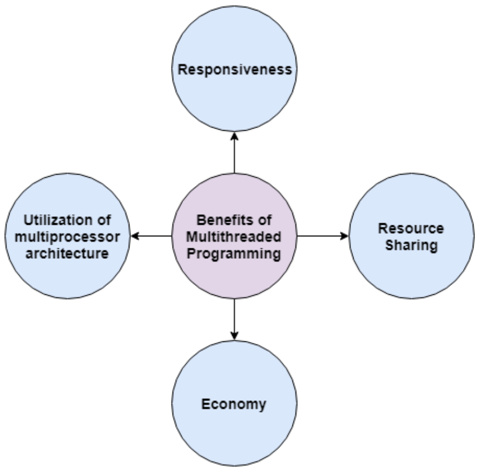
Multithreading is a process of executing multiple threads simultaneously. A thread is a lightweight sub-process, the smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking.

However, we use multithreading because threads use a shared memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.

Java Multithreading is mostly used in games, animation, etc.

Multithreading allows the execution of multiple parts of a program at the same time. These parts are known as threads and are lightweight processes available within the process. So multithreading leads to maximum utilization of the CPU by multitasking.

Some of the benefits of multithreaded programming are given as follows −


Responsiveness
Utilization of multiprocessor architecture
Benefits of Multithreaded Programming
Resource Sharing
Economy

- **Resource Sharing**

  All the threads of a process share its resources such as memory, data, files etc. A single application can have different threads within the same address space using resource sharing.

- **Responsiveness**

  Program responsiveness allows a program to run even if part of it is blocked using multithreading. This can also be done if the process is performing a lengthy operation. For example - A web browser with multithreading can use one thread for user contact and another for image loading at the same time.

- **Utilization of Multiprocessor Architecture**

  In a multiprocessor architecture, each thread can run on a different processor in parallel using multithreading. This increases concurrency of the system. This is in direct contrast to a single processor system, where only one process or thread can run on a processor at a time.

- **Economy**

  It is more economical to use threads as they share the process resources. Comparatively, it is more expensive and time-consuming to create processes as they require more memory and resources. The overhead for process creation and management is much higher than thread creation and management.

30) What is Thrash?

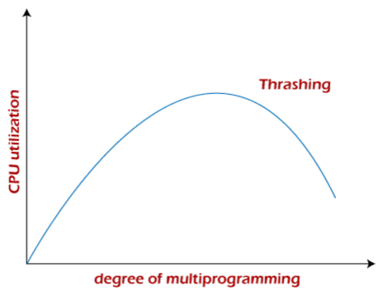
In computer science, ***thrash*** is the poor performance of a virtual memory (or paging) system when the same pages are being loaded repeatedly due to a lack of main memory to keep them in memory. Depending on the configuration and algorithm, the actual throughput of a system can degrade by multiple orders of magnitude.

In computer science, ***thrashing*** occurs when a computer's virtual memory resources are overused, leading to a constant state of paging and page faults, inhibiting most application-level processing. It causes the performance of the computer to degrade or collapse. The situation can continue indefinitely until the user closes some running applications or the active processes free up additional virtual memory resources.

To know more clearly about thrashing, first, we need to know about page fault and swapping.

- **Page fault:** We know every program is divided into some pages. A page fault occurs when a program attempts to access data or code in its address space but is not currently located in the system RAM.
- **Swapping:** Whenever a page fault happens, the operating system will try to fetch that page from secondary memory and try to swap it with one of the pages in RAM. This process is called swapping.

Thrashing is when the page fault and swapping happens very frequently at a higher rate, and then the operating system has to spend more time swapping these pages. This state in the operating system is known as thrashing. Because of thrashing, the CPU utilization is going to be reduced or negligible.



The basic concept involved is that if a process is allocated too few frames, then there will be too many and too frequent page faults. As a result, no valuable work would be done by the CPU, and the CPU utilization would fall drastically.The long-term scheduler would then try to improve the CPU utilization by loading some more processes into the memory, thereby increasing the degree of multiprogramming. Unfortunately, this would result in a further decrease in the CPU utilization, triggering a chained reaction of higher page faults followed by an increase in the degree of multiprogramming, called thrashing.

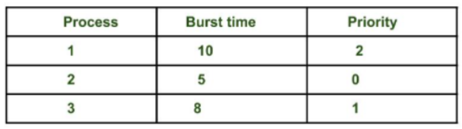
31) Belady’s Anomaly

The rate of page faults varies directly with the number of frames allocated to the individual process. The increase in the number of frames considerably decreases the number of page faults. However, sometimes reverse action occurs when the increased number of frames results in increased page faults. This exception is known as the **Belady’s Anomaly**. The occurrence of this exception depends on the **page replacement algorithm,** which governs the **demand paging** process. The page replacement algorithms in which **Belady’s Anomaly** occurs the most includes:

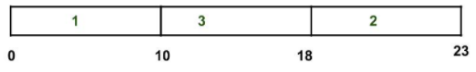
1. First In First Out (FIFO)
2. Second Chance Algorithm
3. Random Page Replacement Algorithm

32) Explain starvation and aging?

Starvation or indefinite blocking is phenomenon associated with the Priority scheduling algorithms, in which a process ready to run for CPU can wait indefinitely because of low priority. In heavily loaded computer system, a steady stream of higher-priority processes can prevent a low-priority process from ever getting the CPU.



Process	Burst time	Priority
1	10	2
2	5	0
3	8	1



As we see in the above example process having higher priority than other processes getting CPU earlier. We can think of a scenario in which only one process is having very low-priority (for example 127) and we are giving other process with high-priority, this can lead indefinitely waiting for the process for CPU which is having low-priority, this leads to **Starvation**

Solution to Starvation : Aging
Aging is a technique of gradually increasing the priority of processes that wait in the system for a long time.For example, if priority range from 127(low) to 0(high), we could increase the priority of a waiting process by 1 Every 15 minutes. Eventually even a process with an initial priority of 127 would take no more than 32 hours for priority 127 process to age to a priority-0 process.

Q. 33 Explain a trap and trapdoor?

A trap is a software interrupt, usually the result of an
error condition, and is also a non maskable interrupt and has highest priority

a hardware interrupt that standard interrupt-masking techniques in the system cannot ignore

Trojans allow confidential or personal information to be compromised by system creating a backdoor on your computer that gives unauthorized users access to your system.

A trap door is kind of a secret entry point into a program that allows anyone gain access to any system without going through the usual security access procedures. Other definition of trap door is it is a method of bypassing normal authentication methods. Therefore it is also known as back door.

Q. 34 Explain a daemon?
Daemon process **:** They are system-related background processes that often run with the permissions of root and services requests from other processes, they most of the time run in the background and wait for processes it can work along with for ex print daemon.

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

Application Software**:**
Application Software is one of the type of software which runs or executes as per user request. High level languages such as java, c, c++ etc are used to develop the application software. Application software is a specific purpose software which is intended to perform some task grouped together. Without an operating system application software can not be installed. It’s examples are Photoshop, VLC media player, Mozilla Firefox, Opera, Google chrome etc.

Q.36 Define daemon objects and thread objects?
Daemon threads is a low priority thread that provide supports to user threads. These threads can be user defined and system defined as well.
Daemon thread is a low priority thread that runs in background to perform tasks such as garbage collection.

Daemon threads are low priority threads which always run in background and user threads are high priority threads which always run in foreground. User Thread or Non-Daemon are designed to do specific or complex task where as daemon threads are used to perform supporting tasks.

Thread Objects

Each thread is associated with an instance of the class `Thread`. There are two basic strategies for using `Thread` objects to create a concurrent application.

- To directly control thread creation and management, simply instantiate `Thread` each time the application needs to initiate an asynchronous task.
- To abstract thread management from the rest of your application, pass the application's tasks to an *executor*.

37. Give commands for finding process ID.

Ans : 1> Firstly we should know about process ID i.e. PID

PID:PID refers to process ID, which is commonly used by most operating system kernels, such as Linux, Unix, MacOS and Windows.

2> A process is a running instance of a program.

3> You can find the PID of processes running on the system using the below nine command.

pidof: pidof – find the process ID of a running program.

pgrep: pgre – look up or signal processes based on name and other attributes.

ps: ps – report a snapshot of the current processes.

pstree: pstree – display a tree of processes.

ss: ss is used to dump socket statistics.

netstat: netstat is displays a list of open sockets.

lsof: lsof – list open files.

fuser: fuser – list process IDs of all processes that have one or more files open

systemctl: systemctl – Control the systemd system and service manager

==

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

Ans : 1> Rename files on Linux

To rename a file in Linux you use the mv command. The command accepts two or more arguments.

For renaming files, only two arguments are needed, which are the source file and the target file.

The mv command will take the source file specified and rename it to the target file.

```
mv old-filename new-filename
```

To rename a file named student1 to student10, for example, you would run the following command.

```
mv student1 student10
```

Provided the file target is the same directory, all file attributes will remain, including permissions.

2> Moving a file on Linux

To move a file to another location we use the same method as renaming a file, except the file path should be different.

```
mv source-file /new/path
```

For example, to move a file from /home/student1/lab-work.log to /var/labs/student1/lab-work.log, you would run the following command.

```
mv /home/student1/lab-work.log /var/labs/student1/lab-work.log
```

3> Moving and Renaming files on Linux

A file can be renamed during a move process using the mv command. You simply give the target path a different name. When mv moves the file, it will be given a new name.

```
mv old-filename /new/path/new-filename
```

For example, to move a file named student1.txt to /var/students and rename it to class1-student1.txt, you would run the following command.

```
mv student1.txt /var/students/class1-student1.txt
```

4> Moving Multiple files on Linux

The mv command accepts multiple source files, which means we can move two or more files at the same time.

When executing the mv command, each file listed will be considered a source with the last path being the exception. The last path will be treated as the target.

```
mv source-file-1 source-file-2 target-path
```

For example, to move student1.txt and student2.txt to /var/students, you would run the following command.

```
mv student1.txt student2.txt /var/students
```

==

39. Give 5 commands in Linux with explanation

Ans :1> cat command

The cat command is simple, and one of the most frequently used commands that are used in a variety of ways.

It is short for ‘concatenate’ and permits users to create files, redirect output, list the contents of a file, and even concatenate multiple files.

2> echo command

The echo command is leveraged to print a string of text passed as an argument to the terminal window.

It comes as the built-in command and is frequently used get output status text in shell scripts.

3> exit command

The exit is the most basic command of all. All it does is exit the shell in which it is active, close the terminal and even logs out of an SSH remote session.

Coders can use the exit command with or without parameters.

4> whoami command

The whoami command is simple and mostly comes handy for amateur Linux users. It tells the user with a username they are logged in as.

More so, coders can use the command to know if anyone has logged into an unnamed Linux terminal.

It is straightforward to execute and is one of the few commands in Linux that perform a single task

5> history command

The history command is a way to view the commands that a user has inputted previously on the command line.

The default limit is set to display only the latest five hundred commands.

It is an easy way to repeat the commands just by inputting ! (exclamation) the point with the number of the command that the user wants to repeat from the list

==

40. Which are deadlock handling

Ans :1> Deadlock:

Deadlock is a state of a database system having two or more transactions,

when each transaction is waiting for a data item that is being locked by some other transaction. A deadlock can be indicated by a cycle in the wait-for-graph.

This is a directed graph in which the vertices denote transactions and the edges denote waits for data items.

2> Deadlock Handling in Centralized Systems

There are three classical approaches for deadlock handling, namely −

A. Deadlock prevention.

B. Deadlock avoidance.

C. Deadlock detection and removal.

All of the three approaches can be incorporated in both a centralized and a distributed database system

A. Deadlock Prevention:

The deadlock prevention approach does not allow any transaction to acquire locks that will lead to deadlocks.

The convention is that when more than one transactions request for locking the same data item, only one of them is granted the lock.

One of the most popular deadlock prevention methods is pre-acquisition of all the locks.

In this method, a transaction acquires all the locks before starting to execute and retains the locks for the entire duration of transaction.

If another transaction needs any of the already acquired locks, it has to wait until all the locks it needs are available.

Using this approach, the system is prevented from being deadlocked since none of the waiting transactions are holding any lock.

B. Deadlock Avoidance

The deadlock avoidance approach handles deadlocks before they occur. It analyzes the transactions and the locks to determine whether or not waiting leads to a deadlock.

The method can be briefly stated as follows. Transactions start executing and request data items that they need to lock.

The lock manager checks whether the lock is available. If it is available, the lock manager allocates the data item and the transaction acquires the lock.

However, if the item is locked by some other transaction in incompatible mode,

the lock manager runs an algorithm to test whether keeping the transaction in waiting state will cause a deadlock or not.

Accordingly, the algorithm decides whether the transaction can wait or one of the transactions should be aborted.

There are two algorithms for this purpose, namely wait-die and wound-wait.

Let us assume that there are two transactions, T1 and T2, where T1 tries to lock a data item which is already locked by T2. The algorithms are as follows −

Wait-Die − If T1 is older than T2, T1 is allowed to wait. Otherwise, if T1 is younger than T2, T1 is aborted and later restarted.

Wound-Wait − If T1 is older than T2, T2 is aborted and later restarted. Otherwise, if T1 is younger than T2, T1 is allowed to wait.

C. Deadlock Detection and Removal

The deadlock detection and removal approach runs a deadlock detection algorithm periodically and removes deadlock in case there is one.

It does not check for deadlock when a transaction places a request for a lock. When a transaction requests a lock, the lock manager checks whether it is available.

If it is available, the transaction is allowed to lock the data item; otherwise the transaction is allowed to wait.

Since there are no precautions while granting lock requests, some of the transactions may be deadlocked.

To detect deadlocks, the lock manager periodically checks if the wait-forgraph has cycles.

If the system is deadlocked, the lock manager chooses a victim transaction from each cycle.

The victim is aborted and rolled back; and then restarted later. Some of the methods used for victim selection are −

Choose the youngest transaction.

Choose the transaction with fewest data items.

Choose the transaction that has performed least number of updates.

Choose the transaction having least restart overhead.

Choose the transaction which is common to two or more cycles.

This approach is primarily suited for systems having transactions low and where fast response to lock requests is needed.

3> Deadlock Handling in Distributed Systems :

Transaction processing in a distributed database system is also distributed, i.e. the same transaction may be processing at more than one site.

The two main deadlock handling concerns in a distributed database system that are not present in a centralized system are transaction location and transaction control.

Once these concerns are addressed, deadlocks are handled through any of deadlock prevention, deadlock avoidance or deadlock detection and removal.

Transaction Location :

Transactions in a distributed database system are processed in multiple sites and use data items in multiple sites.

The amount of data processing is not uniformly distributed among these sites. The time period of processing also varies.

Thus the same transaction may be active at some sites and inactive at others.

When two conflicting transactions are located in a site, it may happen that one of them is in inactive state.

This condition does not arise in a centralized system. This concern is called transaction location issue.

This concern may be addressed by Daisy Chain model. In this model, a transaction carries certain details when it moves from one site to another.

Some of the details are the list of tables required, the list of sites required, the list of visited tables and sites,

the list of tables and sites that are yet to be visited and the list of acquired locks with types. After a transaction terminates by either commit or abort,

the information should be sent to all the concerned sites.

Transaction Control :

Transaction control is concerned with designating and controlling the sites required for processing a transaction in a distributed database system.

There are many options regarding the choice of where to process the transaction and how to designate the center of control, like −

One server may be selected as the center of control.

The center of control may travel from one server to another.

The responsibility of controlling may be shared by a number of servers.