**1. What is an operating system?**

An operating system is a program that acts as an intermediary between the user and the computer hardware. The purpose of an OS is to provide a convenient environment in which user can execute programs in a convenient and efficient manner.

**2. What are the different operating systems?**

1.     Batched operating systems

2.     Multi-programmed operating systems

3.     timesharing operating systems

4.     Distributed operating systems

5.     Real-time operating systems

**3. What are the basic functions of an operating system?**

Operating system controls and coordinates the use of the hardware among the various applications programs for various uses. Operating system acts as resource allocator and manager. Also operating system is control program which controls the user programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control of I/O devices.

**4. What is kernel?**

Kernel is the core and essential part of computer operating system that provides basic services for all parts of OS.

**5. What is difference between micro kernel and macro kernel?**

Micro kernel is a kernel which run services those are minimal for operating system performance. In this kernel all other operations are performed by processor.

Macro Kernel is a combination of micro and monolithic kernel. In monolithic kernel all operating system code is in single executable image.

**6. What is dead lock?**

Deadlock is a situation or condition where the two processes are waiting for each other to complete so that they can start. This result both the processes to hang.

**7. What is a process?**

A program in execution is called a process.

Processes are of two types:  
1. Operating system processes  
2. User processes

**8. What are the states of a process?**

1. New  
2. Running  
3. Waiting  
4. Ready  
5. Terminated

**9. What is starvation and aging?**

Starvation is Resource management problem where a process does not get the resources it needs for a long time because the resources are being allocated to other processes.

Aging is a technique to avoid starvation in a scheduling system.

**10. What is semaphore?**

Semaphore is a variable, whose status reports common resource, Semaphore is of two types one is Binary semaphore and other is Counting semaphore.

**11. What is context switching?**

Transferring the control from one process to other process requires saving the state of the old process and loading the saved state for new process. This task is known as context switching.

**12. What is a thread?**

A thread is a program line under execution. Thread sometimes called a light-weight process, is a basic unit of CPU utilization; it comprises a thread id, a program counter, a register set, and a stack

**13. What is process synchronization?**

A situation, where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place, is called race condition. To guard against the race condition we need to ensure that only one process at a time can be manipulating the same data. The technique we use for this is called process synchronization.

**14. What is virtual memory?**

Virtual memory is hardware technique where the system appears to have more memory that it actually does. This is done by time-sharing, the physical memory and storage parts of the memory one disk when they are not actively being used.

**15. What is thrashing?**

It is a phenomenon in virtual memory schemes when the processor spends most of its time swapping pages, rather than executing instructions. This is due to an inordinate number of page faults.

**16. What is fragmentation? Tell about different types of fragmentation?**

When many of free blocks are too small to satisfy any request then fragmentation occurs. External fragmentation and internal fragmentation are two types of fragmentation. External Fragmentation happens when a dynamic memory allocation algorithm allocates some memory and a small piece is left over that cannot be effectively used.  Internal fragmentation is the space wasted inside of allocated memory blocks because of restriction on the allowed sizes of allocated blocks.

**17. What are necessary conditions for dead lock?**

1. Mutual exclusion (where at least one resource is non-sharable)

2. Hold and wait (where a process holds one resource and waits for other resource)

3. No preemption (where the resources can’t be preempted)

4. Circular wait (where p[i] is waiting for p[j] to release a resource. i= 1,2,…n

j=if (i!=n) then i+1

else 1 )

**18. What is cache memory?**

Cache memory is random access memory (RAM) that a computer microprocessor can access more quickly than it can access regular RAM. As the microprocessor processes data, it looks first in the cache memory and if it finds the data there (from a previous reading of data), it does not have to do the more time-consuming reading of data from larger memory.

**19. What is logical and physical addresses space?**

Logical address space is generated from CPU; it bound to a separate physical address space is central to proper memory management. Physical address space is seen by the memory unit. Logical address space is virtual address space. Both these address space will be same at compile time but differ at execution time.

**20. Differentiate between Complier and Interpreter?**

An interpreter reads one instruction at a time and carries out the actions implied by that instruction. It does not perform any translation. But a compiler translates the entire instructions

**21. What is Throughput, Turnaround time, waiting time and Response time?**

Throughput – number of processes that complete their execution per time unit

Turnaround time – amount of time to execute a particular process

Waiting time – amount of time a process has been waiting in the ready queue

Response time – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment)

**22. What is Memory-Management Unit (MMU)?**

Hardware device that maps virtual to physical address. In MMU scheme, the value in the relocation register is added to every address generated by a user process at the time it is sent to memory.

->The user program deals with logical addresses; it never sees the real physical addresses

**23. What is a Real-Time System?**

A real time process is a process that must respond to the events within a certain time period. A real time operating system is an operating system that can run real time processes successfully

**24. What is a trap and trapdoor?**

Trapdoor is a secret undocumented entry point into a program used to grant access without normal methods of access authentication. A trap is a software interrupt, usually the result of an error condition.

**25. When is a system in safe state?**

The set of dispatchable processes is in a safe state if there exists at least one temporal order in which all processes can be run to completion without resulting in a deadlock.

**26. Explain the concept of the Distributed systems?**

Distributed systems work in a network. They can share the network resources, communicate with each other.

**27. What is cache-coherency?**

In a multiprocessor system there exist several caches each may containing a copy of same variable A. Then a change in one cache should immediately be reflected in all other caches this process of maintaining the same value of a data in all the caches s called cache-coherency.

**28. What is a long term scheduler & short term scheduler ?**

Long term schedulers are the job schedulers that select processes from the job queue and load them into memory for execution. The short term schedulers are the CPU schedulers that select a process from the ready queue and allocate the CPU to one of them.

**29. Explain the meaning of mutex.**

Mutex is the short form for ‘Mutual Exclusion object’. A mutex allows multiple threads for sharing the same resource. The resource can be file. A mutex with a unique name is created at the time of starting a program. A mutex must be locked from other threads, when any thread that needs the resource. When the data is no longer used / needed, the mutex is set to unlock.

**30. What is cycle stealing?**

We encounter cycle stealing in the context of Direct Memory Access (DMA). Either the DMA controller can use the data bus when the CPU does not need it, or it may force the CPU to temporarily suspend operation. The latter technique is called cycle stealing. Note that cycle stealing can be done only at specific break points in an instruction cycle.

**31. What is Marshalling?**

The process of packaging and sending interface method parameters across thread or process boundaries.

**32. What is a daemon?**

Daemon is a program that runs in the background without user’s interaction. A daemon runs in a multitasking operating system like UNIX. A daemon is initiated and controlled by special programs known as ‘processes’.

**33. What is pre-emptive and non-preemptive scheduling?**

Preemptive scheduling: The preemptive scheduling is prioritized. The highest priority process should always be the process that is currently utilized.

Non-Preemptive scheduling: When a process enters the state of running, the state of that process is not deleted from the scheduler until it finishes its service time.

**34. What is busy waiting?**

The repeated execution of a loop of code while waiting for an event to occur is called busy-waiting. The CPU is not engaged in any real productive activity during this period, and the process does not progress toward completion.

**35. What is page cannibalizing?**

Page swapping or page replacements are called page cannibalizing.

**36. What is SMP?**

To achieve maximum efficiency and reliability a mode of operation known as symmetric multiprocessing is used. In essence, with SMP any process or threads can be assigned to any processor.

**37. What is process migration?**

It is the transfer of sufficient amount of the state of process from one machine to the target machine.

**38. Difference between Primary storage and secondary storage?**

Primary memory is the main memory (Hard disk, RAM) where the operating system resides.

Secondary memory can be external devices like CD, floppy magnetic discs etc. secondary storage cannot be directly accessed by the CPU and is also external memory storage.

**39. Define compactions.**

Compaction is a process in which the free space is collected in a large memory chunk to make some space available for processes.

**40. What are residence monitors?**

Early operating systems were called residence monitors.

**41. What is dual-mode operation?**

In order to protect the operating systems and the system programs from the malfunctioning programs the two mode operations were evolved  
System mode  
User mode.

**42. What is a device queue?**

A list of processes waiting for a particular I/O device is called device queue.

**43. What are the different types of Real-Time Scheduling?**

Hard real-time systems required to complete a critical task within a guaranteed amount of time.  
Soft real-time computing requires that critical processes receive priority over less fortunate ones.

**44. What is relative path and absolute path?**

Absolute path-- Exact path from root directory.  
Relative path-- Relative to the current path.

**45. What are the disadvantages of context switching?**

Time taken for switching from one process to other is pure over head. Because the system does no useful work while switching. So one of the solutions is to go for threading when ever possible.

**46. What is a data register and address register?**

Data registers - can be assigned to a variety of functions by the programmer. They can be used with any machine instruction that performs operations on data.  
Address registers - contain main memory addresses of data and instructions or they contain a portion of the address that is used in the calculation of the complete addresses.

**47. What is DRAM?**

Dynamic Ram stores the data in the form of Capacitance, and Static RAM stores the data in Voltages.

**48. What are local and global page replacements?**

Local replacement means that an incoming page is brought in only to the relevant process' address space. Global replacement policy allows any page frame from any process to be replaced. The latter is applicable to variable partitions model only.

**49. Explain the concept of the batched operating systems?**

In batched operating system the users gives their jobs to the operator who sorts the programs according to their requirements and executes them. This is time consuming but makes the CPU busy all the time.

**50. What is SCSI?**

SCSI - Small computer systems interface is a type of interface used for computer components such as hard drives, optical drives, scanners and tape drives. It is a competing technology to standard IDE (Integrated Drive Electronics).

**51.When is a system in safe state?**

The set of dispatchable processes is in a safe state if there exists at least one temporal order in which all processes can be run to completion without resulting in a deadlock.

**52. What is cycle stealing?**

We encounter cycle stealing in the context of Direct Memory Access (DMA). Either the DMA controller can use the data bus when the CPU does not need it, or it may force the CPU to temporarily suspend operation. The latter technique is called cycle stealing. Note that cycle stealing can be done only at specific break points in an instruction cycle.

**53. What is an idle thread?**

The special thread a dispatcher will execute when no ready thread is found.

**54. What is FtDisk?**

It is a fault tolerance disk driver for Windows NT.

**55.What is  Dispatcher?**

Dispatcher module gives control of the CPU to the process selected by the short-term scheduler; this involves: Switching context, Switching to user mode, Jumping to the proper location in the user program to restart that program, dispatch latency – time it takes for the dispatcher to stop one process and start another running.

**56. When does the condition 'rendezvous' arise?**

In message passing, it is the condition in which, both, the sender and receiver are blocked until the message is delivered.

**57. What is process spawning?**

When the OS at the explicit request of another process creates a process, this action is called process spawning

**58. What are the reasons for process suspension?**  
1) swapping  
2) interactive user request  
3) timing  
4) parent process request

**59. What are the sub-components of I/O manager in Windows NT?**  
1) Network redirector/ Server  
2) Cache manager.  
3) File systems  
4) Network driver  
5) Device driver

**60. What is a drawback of MVT?**  
1) ability to support multiple processors  
2) virtual storage  
3) source level debugging

**File Commands**

**File Administration**

**ls** [option(s)] [file(s)]

If you run **ls** without any additional parameters, the program will list the contents of the current directory in short form.

-l

detailed list

-a

displays hidden files

**cp** [option(s)] sourcefile targetfile

Copies sourcefile to targetfile.

-i

Waits for confirmation, if necessary, before an existing targetfile is overwritten

-r

Copies recursively (includes subdirectories)

**mv** [option(s)] sourcefile targetfile

Copies sourcefile to targetfile then deletes the original sourcefile.

-b

Creates a backup copy of the sourcefile before moving

-i

Waits for confirmation, if necessary, before an existing targetfile is overwritten

**rm** [option(s)] file(s)

Removes the specified files from the file system. Directories are not removed by **rm** unless the option -r is used.

-r

Deletes any existing subdirectories

-i

Waits for confirmation before deleting each file.

**ln** [option(s)] sourcefile targetfile

Creates an internal [*link*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#link) from the sourcefile to the targetfile, under a different name. Normally, such a link points directly to the sourcefile on one and the same file system. However, if **ln** is executed with the -s option, it creates a symbolic link that only points to the directory where the sourcefile is located, thus enabling linking across file systems.

-s

Creates a symbolic link

**cd** [options(s)] [directory]

Changes the current directory. **cd** without any parameters changes to the user's home directory.

**mkdir** [option(s)] directoryname

Creates a new directory.

**rmdir** [option(s)] directoryname

Deletes the specified directory, provided it is already empty.

**chown** [option(s)] username.group file(s)

Transfers the ownership of a file to the user with the specified user name.

-R

Changes files and directories in all subdirectories.

**chgrp** [option(s)] groupname file(s)

Transfers the group ownership of a given file to the group with the specified group name. The file owner can only change group ownership if a member of both the existing and the new group.

**chmod** [options] mode file(s)

Changes the access permissions.

The mode parameter has three parts: group, access, and access type. group accepts the following characters:

u

user

g

group

o

others

For access, access is granted by the + symbol and denied by the - symbol.

The access type is controlled by the following options:

r

read

w

write

x

eXecute — executing files or changing to the directory.

s

Set uid bit — the application or program is started as if it were started by the owner of the file.

**gzip** [parameters] file(s)

This program compresses the contents of files, using complex mathematical algorithms. Files compressed in this way are given the extension .gz and need to be uncompressed before they can be used. To compress several files or even entire directories, use the **tar** command.

-d

decompresses the packed gzip files so they return to their original size and can be processed normally (like the command **gunzip**).

**tar** options archive file(s)

The **tar** puts one file or (usually) several files into an archive. Compression is optional.

**tar** is a quite complex command with a number of options available. The most frequently used options are:

-f

Writes the output to a file and not to the screen as is usually the case

-c

Creates a new tar archive

-r

Adds files to an existing archive

-t

Outputs the contents of an archive

-u

Adds files, but only if they are newer than the files already contained in the archive

-x

Unpacks files from an archive (*extraction*)

-z

Packs the resulting archive with **gzip**

-j

Compresses the resulting archive with **bzip2**

-v

Lists files processed

The archive files created by **tar** end with .tar. If the tar archive was also compressed using **gzip**, the ending is .tgz or .tar.gz. If it was compressed using **bzip2**, .tar.bz2.

Application examples can be found in Section [“Archives and Data Compression”](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/ch24s02.html#sec:bash.tar).

**locate** pattern(s)

The locate command can find in which directory a specified file is located. If desired, use [*wild cards*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#wildcard) to specify file names. The program is very speedy, as it uses a database specifically created for the purpose (rather than searching through the entire file system). This very fact, however, also results in a major drawback: locate is unable to find any files created after the latest update of its database.

The database can be generated by root with **updatedb**.

**updatedb** [options(s)]

This command performs an update of the database used by **locate**. To include files in all existing directories, run the program as root. It also makes sense to place it in the background by appending an ampersand (&), so you can immediately continue working on the same command line (**updatedb &**).

**find** [option(s)]

The **find** command allows you to search for a file in a given directory. The first argument specifies the directory in which to start the search. The option **-name** must be followed by a search string, which may also include [*wild cards*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#wildcard). Unlike **locate**, which uses a database, **find** scans the actual directory.

**Commands to Access File Contents**

**cat** [option(s)] file(s)

The **cat** command displays the contents of a file, printing the entire contents to the screen without interruption.

-n

Numbers the output on the left margin

**less** [option(s)] file(s)

This command can be used to browse the contents of the specified file. Scroll half a screen page up or down with **PgUp** and **PgDn** or a full screen page down with **Space**. Jump to the beginning or end of a file using **Home** and **End**. Press **Q** to exit the program.

**grep** [option(s)] searchstring filenames

The grep command finds a specific searchstring in the specified file(s). If the search string is found, the command displays the line in which the searchstring was found along with the file name.

-i

Ignores case

-l

Only displays the names of the respective files, but not the text lines

-n

Additionally displays the numbers of the lines in which it found a hit

-l

Only lists the files in which searchstring does not occur

**diff** [option(s)] file1 file2

The **diff** command compares the contents of any two files. The output produced by the program lists all lines that do not match.

This is frequently used by programmers who need only send their program alterations and not the entire source code.

-q

Only reports *whether* the two given files differ

**File Systems**

**mount** [option(s)] [<device>] mountpoint

This command can be used to mount any data media, such as hard disks, CD-ROM drives, and other drives, to a directory of the Linux file system.

-r

mount read-only

-t filesystem

Specifies the file system. The most common are ext2 for Linux hard disks, msdos for MS-DOS media, vfat for the Windows file system, and iso9660 for CDs.

For hard disks not defined in the file /etc/fstab, the device type must also be specified. In this case, only root can mount. If the file system should also be mounted by other users, enter the option user in the appropriate line in the /etc/fstab file (separated by commas) and save this change. Further information is available in mount.

**umount** [option(s)] mountpoint

This command unmounts a mounted drive from the file system. To prevent data loss, run this command before taking a removable data medium from its drive. Normally, only root is allowed to run the commands **mount** and **umount**. To enable other users to run these commands, edit the /etc/fstab file to specify the option user for the respective drive.

**System Commands**

**System Information**

**df** [option(s)] [directory]

The **df** (disk free) command, when used without any options, displays information about the total disk space, the disk space currently in use, and the free space on all the mounted drives. If a directory is specified, the information is limited to the drive on which that directory is located.

-H

shows the number of occupied blocks in gigabytes, megabytes, or kilobytes — in human-readable format

-t

Type of file system (ext2, nfs, etc.)

**du** [option(s)] [path]

This command, when executed without any parameters, shows the total disk space occupied by files and subdirectories in the current directory.

-a

Displays the size of each individual file

-h

Output in human-readable form

-s

Displays only the calculated total size

**free** [option(s)]

The command **free** displays information about RAM and swap space usage, showing the total and the used amount in both categories.

-b

Output in bytes

-k

Output in kilobytes

-m

Output in megabytes

**date** [option(s)]

This simple program displays the current system time. If run as root, it can also be used to change the system time. Details about the program are available in date.

**Processes**

**top** [options(s)]

top provides a quick overview of the currently running [*processes*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#process). Press **H** to access a page that briefly explains the main options to customize the program.

**ps** [option(s)] [process ID]

If run without any options, this command displays a table of all *your own* programs or processes — those you started. The options for this command are not preceded by hyphen.

aux

Displays a detailed list of all processes, independent of the owner.

**kill** [option(s)] process ID

Unfortunately, sometimes a program cannot be terminated in the normal way. However, in most cases, you should still be able to stop such a runaway program by executing the **kill** command, specifying the respective process ID (see **top** and **ps**).

**kill** sends a *TERM* signal that instructs the program to shut itself down. If this does not help, the following parameter can be used:

-9

Sends a *KILL* signal instead of a *TERM* signal, with which the process really is *annihilated* by the operating system. This brings the specific processes to an end in almost all cases.

**killall** [option(s)] processname

This command is similar to **kill**, but uses the process name (instead of the process ID) as an argument, causing all processes with that name to be killed.

**Network**

**ping** [option(s)] host name|IP address

The ping command is the standard tool for testing the basic functionality of TCP/IP networks. It sends a small data packet to the destination host, requesting an immediate reply. If this works, ping displays a message to that effect, which indicates that the network link is basically functioning.

-c

number Determines the total number of packages to send and ends after they have been dispatched. By default, there is no limitation set.

-f

*flood ping*: sends as many data packages as possible. A popular means, reserved to root, to test networks.

-i

value Specifies the interval between two data packages in seconds. Default: one second

**nslookup**

The Domain Name System resolves domain names to IP addresses. With this tool, send queries to information servers (DNS servers).

**telnet** [option(s)] host name or IP address

Telnet is actually an Internet protocol that enables you to work on remote hosts across a network. telnet is also the name of a Linux program that uses this protocol to enable operations on remote computers.

***Warning***

Do not use telnet over a network on which third parties can *eavesdrop.* Particularly on the Internet, use encrypted transfer methods, such as **ssh**, to avoid the risk of malicious misuse of a password (see the man page for **ssh**).

**Miscellaneous**

**passwd** [option(s)] [username]

Users may change their own passwords at any time using this command. Furthermore, the administrator root can use the command to change the password of any user on the system.

**su** [option(s)] [username]

The **su** command makes it possible to log in under a different user name from a running session. When using the command without specifying a user name, you will be prompted for the root password. Specify a user name and the corresponding password to use the environment of the respective user. The password is not required from [*root*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#root), as root is authorized to assume the identity of any user.

**halt** [option(s)]

To avoid loss of data, you should always use this program to shut down your system.

**reboot** [option(s)]

Does the same as **halt** with the difference that the system performs an immediate reboot.

**clear**

This command cleans up the visible area of the console. It has no options.

**OPERATING SYSTEM :-**

Operating System can be defined as an interface between user and the hardware. It provides an environment to the user so that, the user can perform its task in convenient and efficient way.

**Process Management Introduction**

There may exist more than one process in the system which may require the same resource at the same time. Therefore, the operating system has to manage all the processes and the resources in a convenient and efficient way.

The operating system is responsible for the following activities in connection with Process Management

1. Scheduling processes and threads on the CPUs.
2. Creating and deleting both user and system processes.
3. Suspending and resuming processes.
4. Providing mechanisms for process synchronization.
5. Providing mechanisms for process communication.

# Attributes of a process

he Attributes of the process are used by the Operating System to create the process control block (PCB) for each of them.

### 1. Process ID

When a process is created, a unique id is assigned to the process which is used for unique identification of the process in the system.

### 2. Program counter

A program counter stores the address of the last instruction of the process on which the process was suspended. The CPU uses this address when the execution of this process is resumed.

### 3. Process State

The Process, from its creation to the completion, goes through various states which are new, ready, running and waiting. We will discuss about them later in detail.

### 4. Priority

Every process has its own priority. The process with the highest priority among the processes gets the CPU first. This is also stored on the process control block.

### 5. General Purpose Registers

Every process has its own set of registers which are used to hold the data which is generated during the execution of the process.

### 6. List of open files

During the Execution, Every process uses some files which need to be present in the main memory. OS also maintains a list of open files in the PCB.

### 7. List of open devices

OS also maintain the list of all open devices which are used during the execution of the process.

# Process States

# **1.New**

# **2.Ready**

# **3.Running**

# **4.Wait**

# **5.Terminate**

# **6. Suspend ready**

# **A process in the ready state, which is moved to secondary memory from the main memory due to lack of the resources (mainly primary memory) is called in the suspend ready state.**

# **7. Suspend wait**

# **Instead of removing the process from the ready queue, it's better to remove the blocked process which is waiting for some resources in the main memory.**

# Operations on the Process

1. Creation
2. Scheduling
3. Execution
4. Deleting / Killing

# Process Schedulers

### 1. Long term scheduler

Long term scheduler is also known as job scheduler. It chooses the processes from the pool (secondary memory) and keeps them in the ready queue maintained in the primary memory.

### 2. Short term scheduler

Short term scheduler is also known as CPU scheduler. It selects one of the Jobs from the ready queue and dispatch to the CPU for the execution.

### 3. Medium term scheduler

Medium term scheduler takes care of the swapped out processes.If the running state processes needs some IO time for the completion then there is a need to change its state from running to waiting.

# Process Queues

### 1. Job Queue

In starting, all the processes get stored in the job queue. It is maintained in the secondary memory. The long term scheduler (Job scheduler) picks some of the jobs and put them in the primary memory.

### 2. Ready Queue

Ready queue is maintained in primary memory. The short term scheduler picks the job from the ready queue and dispatch to the CPU for the execution.

### 3. Waiting Queue

When the process needs some IO operation in order to complete its execution, OS changes the state of the process from running to waiting. The context (PCB) associated with the process gets stored on the waiting queue which will be used by the Processor when the process finishes the IO.

# Various Times related to the Process

### 1. Arrival Time

The time at which the process enters into the ready queue is called the arrival time.

### 2. Burst Time

The total amount of time required by the CPU to execute the whole process is called the Burst Time. This does not include the waiting time.

### 3. Completion Time

The Time at which the process enters into the completion state or the time at which the process completes its execution, is called completion time.

### 4. Turnaround time

The total amount of time spent by the process from its arrival to its completion, is called Turnaround time.

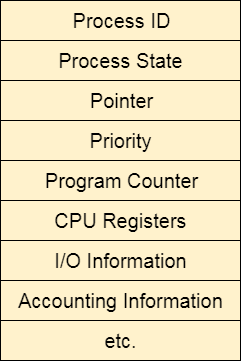
### 5. Waiting Time

The Total amount of time for which the process waits for the CPU to be assigned is called waiting time.

### 6. Response Time

The difference between the arrival time and the time at which the process first gets the CPU is called Response Time.

**What is saved in the Process Control Block?**



**The Purpose of a Scheduling algorithm**

1. Maximum CPU utilization
2. Fare allocation of CPU
3. Maximum throughput
4. Minimum turnaround time
5. Minimum waiting time
6. Minimum response time

There are the following algorithms which can be used to schedule the jobs.

### 1. First Come First Serve

It is the simplest algorithm to implement. The process with the minimal arrival time will get the CPU first. The lesser the arrival time, the sooner will the process gets the CPU. It is the non-preemptive type of scheduling.

### 2. Round Robin

In the Round Robin scheduling algorithm, the OS defines a time quantum (slice). All the processes will get executed in the cyclic way. Each of the process will get the CPU for a small amount of time (called time quantum) and then get back to the ready queue to wait for its next turn. It is a preemptive type of scheduling.

### 3. Shortest Job First

The job with the shortest burst time will get the CPU first. The lesser the burst time, the sooner will the process get the CPU. It is the non-preemptive type of scheduling.

### 4. Shortest remaining time first

It is the preemptive form of SJF. In this algorithm, the OS schedules the Job according to the remaining time of the execution.

### 5. Priority based scheduling

In this algorithm, the priority will be assigned to each of the processes. The higher the priority, the sooner will the process get the CPU. If the priority of the two processes is same then they will be scheduled according to their arrival time.

### 6. Highest Response Ratio Next

In this scheduling Algorithm, the process with highest response ratio will be scheduled next. This reduces the starvation in the system.

# Convoy Effect in FCFS

FCFS may suffer from the **convoy effect** if the burst time of the first job is the highest among all.

# Highest Response Ratio Next (HRRN) Scheduling

Highest Response Ratio Next (HRNN) is one of the most optimal scheduling algorithms. This is a non-preemptive algorithm in which, the scheduling is done on the basis of an extra parameter called Response Ratio. A Response Ratio is calculated for each of the available jobs and the Job with the highest response ratio is given priority over the others.

Response Ratio is calculated by the given formula.

1. Response Ratio = (W+S)/S

**Where,**

1. W → Waiting Time
2. S → Service Time or Burst Time

**Memory Management**

**Fixed Partition -** > Continuous Memory Allocation (partition size declared initially)

**Dynamic Partition** -> No Continuous Memory Allocation (partition size not declared initially)

**Compaction :-**

**Problem ->**Process stored in different places in the memory(External Fragmentation)

compaction minimize the probability of external fragmentation. In compaction, all the free partitions are made contiguous and all the loaded partitions are brought together.

# Partitioning Algorithms

**1. First Fit Algorithm**

First Fit algorithm scans the linked list and whenever it finds the first big enough hole to store a process, it stops scanning and load the process into that hole.

**2. Next Fit Algorithm**

Next Fit algorithm is similar to First Fit algorithm except the fact that, Next fit scans the linked list from the node where it previously allocated a hole.

**3. Best Fit Algorithm**

The Best Fit algorithm tries to find out the smallest hole possible in the list that can accommodate the size requirement of the process.

**4. Worst Fit Algorithm**

The worst fit algorithm scans the entire list every time and tries to find out the biggest hole in the list which can fulfill the requirement of the process.

**5. Quick Fit Algorithm**

The quick fit algorithm suggestsmaintaining the different lists of frequently used sizes.

## **Need for Paging**

To remove external fragmentation . Compaction made system inefficient

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.

# Belady'sAnomaly

Increase in page fault increases the number of frames

# Segmentation

In Operating Systems, Segmentation is a memory management technique in which, the memory is divided into the variable size parts. Each part is known as segment which can be allocated to a process.

The details about each segment are stored in a table called as segment table. Segment table is stored in one (or many) of the segments.

Segment table contains mainly two information about segment:

1. Base: It is the base address of the segment
2. Limit: It is the length of the segment.

# Segmented Paging

Pure segmentation is not very popular and not being used in many of the operating systems. However, Segmentation can be combined with Paging to get the best features out of both the techniques.

In Segmented Paging, the main memory is divided into variable size segments which are further divided into fixed size pages.

1. Pages are smaller than segments.
2. Each Segment has a page table which means every program has multiple page tables.
3. The logical address is represented as Segment Number (base address), Page number and page offset.

## Translation look aside buffer (TLB)

A Translation look aside buffer can be defined as a memory cache which can be used to reduce the time taken to access the page table again and again.

**1) What is Unix?**

UNIX is a portable operating system that is designed for efficient multitasking and multi-user functions. Since it is a portable operating system, it can run on different hardware platforms. It is written in C language. It was developed by Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna.

### 2) List the distributions of UNIX.

UNIX has many distributions including Solaris UNIX, AIX, HP UNIX and BSD and many more.

### 3) List some features of UNIX.

UNIX includes the following features:

**UNIX supports the multiuser system:** In UNIX it is possible that many users can use the system with their separate workspace and logins i.e.it has full support for the multiuser environment.

**UNIX supports the multitasking environment:** In UNIX many apps can run at a single instance of time this is also known as a multitasking environment.

### What are the core concepts of UNIX

The core concepts of UNIX are given below.

* **Kernel-** The kernel is also known as the heart of the operating system. Its fundamental role is to interact with the hardware and also monitor major processes like memory management, file management, and task scheduling.
* **Shell-** It is also called command prompt, it connects the user to the operating system, whatever is typed by the user is translated into the language understood by the command prompt, and then the corresponding actions are performed.
* **Commands and Utilities-** Many built-in commands help the user perform day to day activities.mv,cat,cp,and grep etc. Some of the examples
* **Directories-** Every bit of data is stored in files, and these files are stored in directories, these directories combine to form a tree-like structure.

### 5) What is a UNIX shell?

The UNIX shell is a program which is used as an interface between the user and the UNIX operating system. It is not a part of the kernel, but it can communicate directly with the server.

### 6) What is filter?

A filter is a program that takes input from standard inputs and performs some operation on that input to produce a result as standard output.

### 7) What are the devices represented in UNIX?

All devices in UNIX are represented by particular files that are located in /dev directory.

### 8) Is there any method to erase all files in the current directory, along with its all sub-directories, by using only one command?

Yes, you should use "rm-r\*" command for this purpose.

Here, the "rm" command is used for deleting files, the -r option will erase directories and subdirectories with their internal data and \* is used for selecting all entries.

### 9) What is necessary before running a shell script from the terminal?

You must make the shell script executable by using the UNIX "chmod" command.

### 10) How to terminate a shell script if statement?

A shell script if statement can be terminated by using "fi."

**11) Write down some common shells with their indicators?**

* sh - Bourne shell
* csh - C Shell
* bash - Bourne Again Shell
* tcsh - enhanced C Shell
* zsh - Z Shell
* ksh - Korn Shell

**12) What are the main features of Korn Shell?**

* Arrays
* Job control
* Command Aliasing
* String manipulation ability
* Built-in integer arithmetic

### 13) What is the difference between cat command and more command?

The cat command is used to display the file contents on the terminal, whereas more command is used like a pager which displays the screen page by page If the file is large and you have to scroll off the screen before you view it.

### 14) Which command is used to restrict incoming messages?

The "mesg" command is used to restrict incoming messages.

### 15) Which command is used to kill the last background job?

The "kill $!" Command is used to kill the last background job.

### 16) Which data structure is used to maintain the file identification?

The "inode" data structure is used to maintain the file identification. Each file has a separate inode and a unique inode number.

### 17) What a pipe?

A pipe is used to join two or more commands by using a pipe "I" character. The output of the first command is propagated to the second command through the pipe.

### 18) What are the links and symbolic links in a UNIX file system?

A link is a second name for a file. Links are used to assign more than one name to a file, but cannot be used to designate a directory more than one name or link filenames on different computers.

Symbolic links are the files that only contain the name of another file. The operations on the symbolic link are directed to the file pointed by it. Both the limitations of connections are eliminated in symbolic links.

**19) Explain system bootup in UNIX.**

System bootup is the first thing that takes place when the power button is pressed in UNIX. Whenever the power button is pressed, BIOS is fired up and checks if all the hardware connected to the system are working correctly, after being successful the system asks the user to provide authentication.

**20) How to change the password in UNIX operating system?**

To change the password in UNIX operating system :

1. Type in the command passwd.
2. You will get a screen which prompts to enter your default(current) password, type your current password.
3. if the current password is verified, then the terminal will prompt you to enter the new password.
4. Enter the new password twice, and your password will be updated.

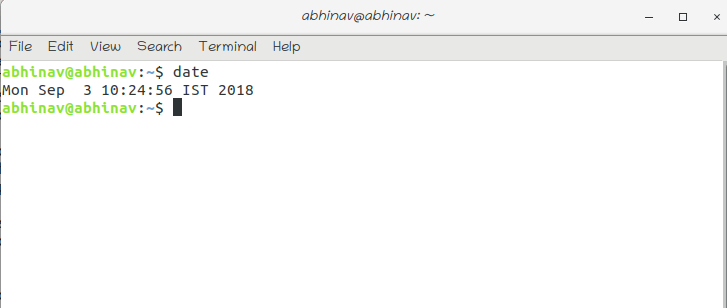
**21) How to list directories in UNIX?**

Command ls can be used to list directories in command prompt. Also, we can also use a variety of ls commands like:

|  |  |
| --- | --- |
| ls -a | In Linux, hidden files start with. (dot) Symbol and they are not visible in the regular directory. The (ls -a) command will enlist the whole list of the current directory including the hidden files. |
| ls -l | It will show the list in a long list format. |
| ls -lh | This command will show you the file sizes in human readable format. Size of the file is tough to read when displayed regarding a byte. The (ls -lh)command will give you the data regarding Mb, Gb, Tb, etc. |
| ls -lhS | If you want to display your files in descending order (highest at the top) according to their size, then you can use (ls -lhS) command. |
| ls -l - -block-size=[SIZE] | It is used to display the files in a specific size format. Here, in [SIZE] you can assign size according to your requirement. |
| ls -d \*/ | It is used to display only sub directories. |
| ls -g or ls -lG | With this, you can exclude column of group information and owner. |
| ls -n | It is used to print group ID and owner ID instead of their names. |
| ls --color=[Value] | This command is used to print list as colored or discolored. |
| ls -li | This command prints the index number if the file in the first column. |
| ls -p | It is used to identify the directory easily by marking the directories with a slash (/) line sign. |
| ls -R | It will display the content of the sub-directories also. |
| ls -lX | It will group the files with the same extensions together in the list. |
| ls -lt | It will sort the list by displaying a recently modified file at the top. |
| ls ~ | It gives the contents of the home directory. |
| ls ../ | It gives the contents of the parent directory. |
| ls --version | It checks the version of ls command. |

**22) How to check the date in UNIX?**

To display the date in UNIX use the **date** command in command prompt.



**23) How to log out in UNIX?**

To log out of UNIX type the **logout** command in the command prompt.

**24) How to perform a system shutdown in UNIX?**

To perform system shutdown in UNIX, you can use the following commands:

* halt
* init 0
* init 6
* power off
* reboot
* shutdown

**25) How many types of files are there in UNIX?**

There are three kinds of files in UNIX:

* Ordinary files: An ordinary file is the one which contains data, text or program instructions.
* Directories: These include both ordinary files and special files.
* Special Files: These are the files which provide unique access to hardware such as hard drives, CD-Rom Drives e.t.c.

**26) What are hidden files in UNIX?**

Hidden files in UNIX are the files which have a .(dot) before the file name. These files do not show up in the traditional file manager.

Common examples of hidden files are:

* .profile
* .kshrc
* .rhosts
* .cshrc

**27) What is the difference between a single dot and double dot in UNIX?**

**.(Single dot)**-represents the current directory

**..(Double dot)**-represents the parent directory.

**28) How to create files in UNIX?**

Creating files in UNIX is simple. The User needs to use the vi editor to create new files.

Type **vi filename** in command prompt to create new files. We can also use the **touch** command to create a zero byte file.

**29) How to display the contents of a file?**

The user can use the cat command followed by the filename to display the command of a file. This command should be entered in the command prompt. The syntax of the command is shown below.

**$ cat filename**

Where the cat is the command to view contents of the file specified by the filename. Also if you want the line number to be displayed along with the content, you can use cat command with option -b.

**30) How to calculate the number of words in a file?**

To count the number of words in a file, Use the following command.

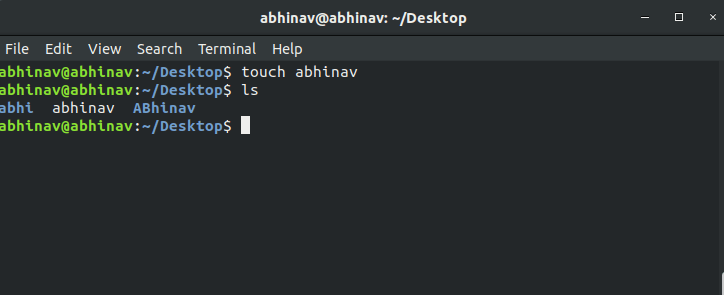
**$ wc filename**

Where wc is the command to count the number of words in the file specified by **filename**.

**31) How to create a blank file in UNIX?**

Blank files can be created by using the touch command, the syntax for the touch command is as follows:

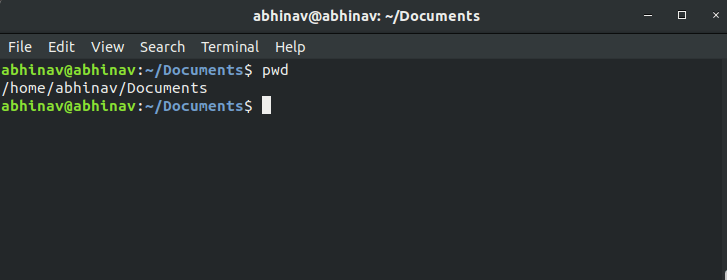
**$ touch filename**



**32) How to know the present working directory in UNIX?**

To know the present working directory, Run the following command on the terminal.

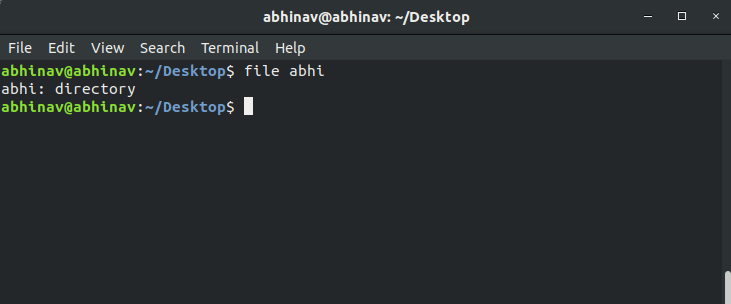
**$ pwd**



**33) How to know the information about a file?**

To fetch the information about a file, use the following command.

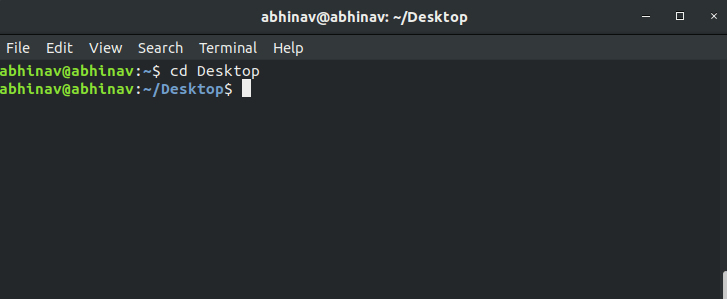
**$ file filename**



**34) How to change the directory in UNIX?**

To change the directory, you can use the cd command in the terminal window. It changes the current directory to the specified directory.

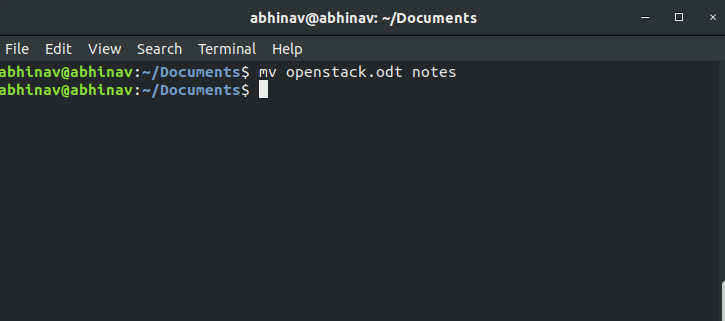
**$ cd directory-name**



### 35) How to move files from one directory to other in UNIX?

In UNIX, mv command is used to move the file from one directory to some other directory.

**$ mv <file-name> <destination path>**

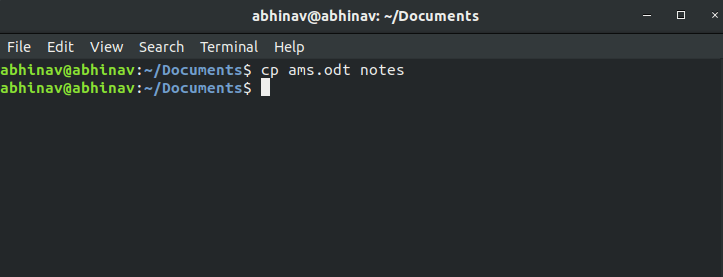


### 36) How to copy files from one directory to other in UNIX?

In UNIX, cp command is used to copy a file from one directory to some other directory. The syntax of the cp command is given below.

**$ cp -r source filename destination file name.**

The -r is used to copy all the content of a directory including sub-directories recursively.

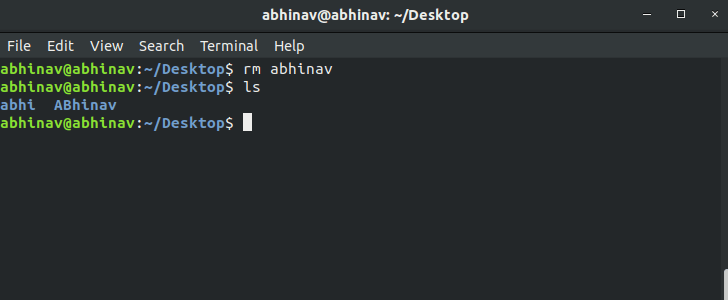


### 37) How to remove files in UNIX?

To remove files, you can use the rm command. The syntax of the rm command is given below.

**$ rm <filename>**

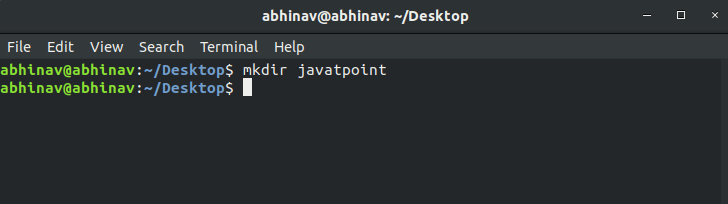
**we can use -r with the rm command to delete all the sub-directories recursively.**



### 38) How to make a new directory in UNIX?

To make a new directory, you can use the mkdir command.

**$ mkdir <directory-name>**



### 39) How to remove the directory in UNIX?

To remove the directory, you can use the rmdir command. To use this command, use the following syntax.

**$ rmdir filename.**

