

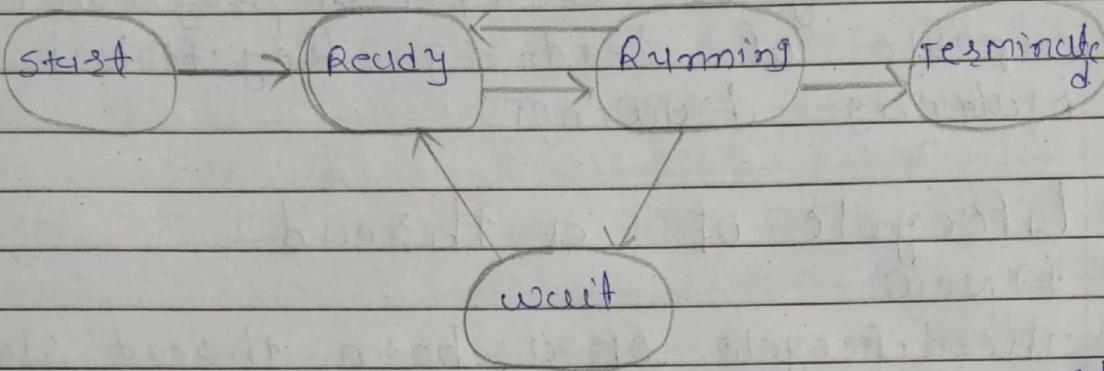
- switching the CPU to another process requires performing a state save of the current process and a state restore of a different process.
 - This task is known as a context switch.
 - Context switch time is pure overhead because the system does no useful work while switching.
2. What is OS? Lists out its features
- An operating system is in most general sense is software that allows users to run other application on a computing device.
 - operating system is a interface between user and hardware.
 - A system application which user can run their application and manage hardware on it.

- 1) security Management
- 2) process Management
- 3) storage and memory Management.
- 4) Disk Management
- 5) i/o Operations
- 6) device Management.

Q. 1 C

1 explain process state transition diagram in brief.

→ process state diagram



→ When a process executes it passes through different states.

→ These stages may differ in difference operating systems and the names of these states are also not standardized.

→ In general a process can have one of the following five states at a time.

1) Start :- This is initial state when a process is first started / created.

2) Ready :- This is process waiting to be assigned to a processor.

3) Running :- Once the process has been assigned to a processor by the OS scheduled.

4) Waiting :- Process moves into the waiting state if needs to wait for a resource.

5) Terminated :- Once the process finishes its execution or it is terminated by the operating system.

2) Explain multithreading.

- A thread is a path which is followed during program's execution.
- Majority of programs written now a days run as a single thread.
- Let's say for example a program is not capable of reading keystrokes while making drawings.

* Lifecycle of a thread.

- 1) New.
- The lifecycle of a born thread starts in this state if remains in this state till a program starts.

2) Runnable

- A thread becomes runnable after it starts it is considered to be executing the task given to it.

3) Waiting

- While waiting for another thread to perform a task the currently running thread goes into the waiting state.

4) Terminated (Dead)

- A thread enters into this state after completing its task.

Q. 1 D

1 Explain RR with suitable example

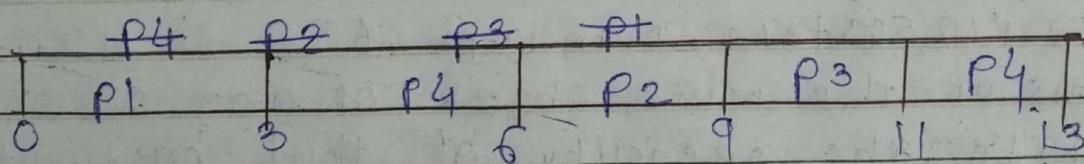
- Round robin is the preemptive process scheduling algorithm.
- Each process is provided a fix time to execute it is called a quantum.
- One process is executed for a given time period if it is preempted and other processes execute for a given time period.

Example

Process ID.	Arrival Time	Burst Time
P ₁	0	5
P ₂	2	3
P ₃	3	2
P ₄	1	3

Step 1

→ Grant burst



Step 2

PID	ARRIVAL TIME	BURST TIME	COMPLETION TIME	FURNISH DEADLINE	WAITING TIME
P ₁	0	5	13	13	8
P ₂	2	3	9	7	4
P ₃	3	2	11	8	6
P ₄	1	3	6	5	2

Step 3

w Avg. Turn Around Time = Total Turn Around Time / No. of processes.

$$(13 + 7 + 8) / 4 = 33 / 4 = 8.25$$

Waiting time = Total waiting time / No of processes
 $(8 + 4 + 6 + 2) / 4 = 20 / 4 = 5$

Step 4

\rightarrow Throughput = total burst time / No. of processes
 $(\frac{5}{3} + \frac{3}{4} + \frac{2}{8} + \frac{3}{8}) / 4 = 33 / 4 = 3.25$

Step 5

w CPU Utilization = total estimated time / * 100.

$$13 / 13 * 100 = 100\%$$

2 Explain function of OS?

w 1) I/O System Management

\rightarrow one of the main objects of any OS is to hide the specificity of that hardware device from the user

2) secondary storage management

\rightarrow system have several of storage which include primary storage, secondary storage and cache storage

8) Security

- security module protects the data and information of computer system against Malware threat and unauthorized access.

4) Command interpretation

- This module is interpreting commands given by the controling system resources to process that commands.

5) Networking

- A distributed system is a group of processes which do not share memory hardware devices.
- The processes communicate with one another through the network.

6) Job accounting

- Keeping track of time & resource used by various job and users.

7) Communication Management

- Condition and assignment of compilers, interpreters and other software resource of the various users of the computer systems.

Q. 2 A

1) What is fragmentation?

- In a computer storage system as processes size reduced and removed from memory the free memory space is broken into small pieces.

2) Full form of PCB

- Printed circuit Board.

3) Memory allocation is the process of

- Allocate memory to processes on demand.

4) Full form of SMT

- Simultaneous Multithreading.

Q. 2 B

1) Explain deadlock?

- A deadlock is a situation in which two computer programs sharing the same resource are effectively preventing each other from accessing the resource resulting in both programs failing to function.

→ The earliest computer operating system run only one program at a time.

- All of the resources of the system were available to this one program.

→ A 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.

Explanation.

2) Explain difference between paging and segmentation.

→ 1) Paging.

→ Paging is a method or technique which is used for non contiguous memory allocation.

→ It is fixed size partitioning theme.

→ In paging both main memory and secondary size divided into equal fixed size partitions.

→ The partitions of the secondary memory called unit and main memory called unit are known as pages and frames respectively.

2) Segmentation

→ Segmentation is another non contiguous memory allocation scheme like paging.

→ Like paging in segmentation the process is not divided indiscriminately into mounted size fixed size pages.

→ It is a variable size partitioning theme.

→ The partitions of secondary memory called units are known as segments.

Q.2 C

I Explain physical memory and virtual memory

1) Physical memory

- physical memory is the actual real memory used in RAM.
- physical memory is the only memory that is directly accessible to the CPU.
- the data that is operated will also be stored in Physical memory in uniform manner.

2) Virtual Memory

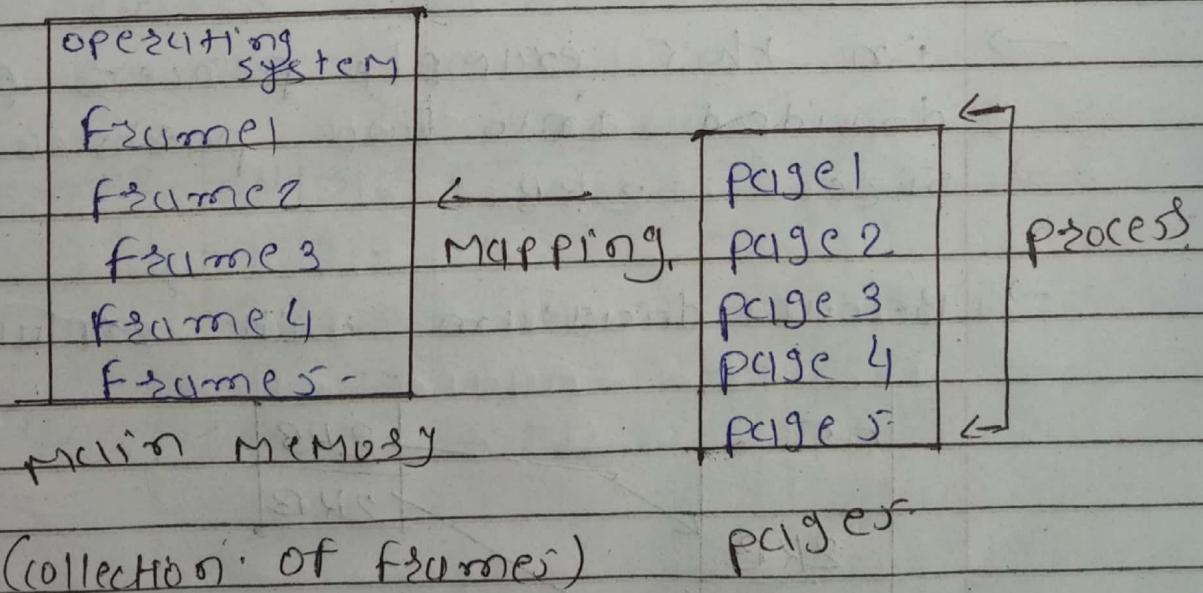
- Virtual memory as the name suggests is not real.
- Virtual memory is one classification of memory which was created by using the hard disk for simulating addition of RAM the addressable space available for the user.
- Virtual addresses are mapped into real addresses.
- The OS uses Virtual memory as a memory management technique.

2) write short note on paging.

- paging is a method of technique which is used for non contiguous memory allocation.
- it is fixed size partitioning theme.
- in paging both main memory and secondary memory are divided into equal fixed size partitions.

page :- A fixed length contiguous block of virtual memory residing on disk.

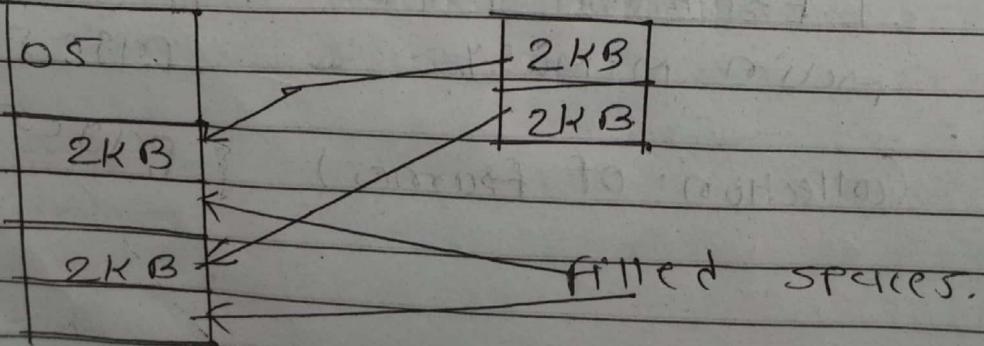
frame :- A fixed length contiguous block located in RAM whose sizing is identical to pages.



Q. 2 D

1) Explain non contiguous memory allocation

- In the non contiguous memory allocation process will circumscribe the memory space but it is not a one place it is at the different location according to the process requirement.
- This technique of non contiguous memory allocation reduce the wastage of memory which leads to internal and external fragmentation.
- This utilizes all the free memory space which is created by 4 different processes.
- in this example process P can be divided into two parts of equal size — 2 KB
- Below diagram will explain in better way



u) These are two types of nonconfigurable memory allocation

1) paging (Q, 2, C) (2)

2) Segmentation (Q)

2) Segmentation.

- The process known as segmentation is a virtual process that executes address spaces of various sizes in a computer system called segments.
- each segments is a different distinct address space that directly corresponds to process objects.
- when a process executes, Segmentation assigns selected data into segments for faster processing.

Process table				Memory
segment	SN	size	Memory Address	05
1	1	400	100	100
	2	200	500	200
	3	100	600	300
2	N	X	NM	400
				500
				600
				700
				800
3				NM
N				

2 Explain virtual memory using Paging
→ Q.2.C (2)

Q.3 A

- 1) Unix was developed By.
→ Ken Thompson and Dennis Ritchie.
- 2) which is the extension for shell script file?
→ .sh.
- 3) The Bourne shell written by?
→ Stephen Bourne.
- 4) BASH stand for
→ Bourne - Again Shell.

Q. 3 B

- 1) Explain shell
- The shell is the utility that processes your requests
 - When you type in a command at your terminal the shell interprets the command and calls the program that you want
 - c shell Bourne shell and Korn shell are the most famous shells which are available with most of the Unix variants.
 - A Unix shell is a command-line interpreter of shell that provides a command-line user interface for Unix-like operating system. Type
 - 1) The Bourne shell
 - 2) The c shell
 - 3) The Korn shell
 - 4) The Bourne Again shell

2) Explain following commands.

- Cut

- ls

- Cat

- This command is generally used to create a file see the content of the file and append data to a file.

Syntax executing a file \$ cat > filename

Q. 3 B

1 Explain shell

- The shell is the utility that processes your requests
- When you type in a command at your terminal the shell interprets the command and calls the program that you want
- C shell Bourne shell and Korn shell are the most famous shells which are available with most of the Unix variants.
- A Unix shell is a command-line interpreter of shell that provides a command-line user interface for Unix-like operating system. Type
 - 1) The Bourne shell
 - 2) The c shell
 - 3) The korn shell
 - 4) The Bourne Again shell

2) Explain following commands.

- Cat

- ls

- Cat

- This command is generally used to create a file see the content of the file and append data to a file.

Syntax executing a file \$ cat > filename

Syntax 2 :- Append data to a file

ex \$ cat → filename

IS

→ is it lists the directories and files in the current directory

Syntax \$ ls

ex

\$ ls -l

output :- detailed list of files and directories.

OPTION

- -l : long listing of file showing with seven attributes.
- -m : it displays the user group id with long list format.
- a : it is used for displaying hidden files.
- R : it displays folders and subfolders.

Q. 3 C

1) Explain comparison command in Linux

i) cmp

- This command compares two files byte by byte that means the first byte of first file with the first byte of the second file.

Syntax

`$cmp file1 file2`
 file1 file2 differs ; byte 13 line 1

2) Comm

- > This command is useful to compare two sorted files line by line

Syntax

`$comm [option] file1
file2`

- 1 lines unique to file1
- 2 lines unique to file2
- 3 lines that appear in both files

3) diff

- > The command finds difference between two files.

Syntax

- q treat all files as text and compare them line by line
- b ignore changes in amount of white space.
- i ignore change in case

- 4) tr :- This command is useful for changing the case from upper to lower.

`tr [option] 'set1' 'set2' $cat file`
`tr "[a-z]" "[A-Z]"`

- d delete characters in the first set
- s replaces repeated characters

5) sed :- sed is a stream editor used for modifying the files in unix.

Syntax

\$ sed 's/unix/linux/' file.txt

2) Explain any three process related commands.

-1) ps

on every unix-like os the process status command displays information about active processes.

Syntax

ps

Ex

\$ ps

2) nice

- nice command affects process scheduling
- A process with a lower nice value is given higher priority.

Syntax

nice <command>

Ex

3 nice

3) kill

- On unix like os the kill command sends a signal to a process.

Syntax

kill pid

Ex

\$ kill -9

- 4) cat and batch.
- 5) cron and crontab
- 6) wait
- 7) sleep

Q.3 D

- 1) Explain types of file in Unix.

The unix file system contains several different types of files.

① Ordinary files.

- used to store your information such as some text you have written.
- Always located within / under a directory file.
- Do not contain other files.

② Directories.

- Branching points in the hierarchical tree
- used to organize group of files.

- May contain ordinary files or other directories.

3) SPECIAL FILES

- used to represent a real physical device such as a printer, tape drive or terminal used for input / output operations.

2) EXPLAIN UNIX ARCHITECTURE ?

-

→ The Unix operating system is a set of programs that act as a link between the computer and the user.

→ Unix was originally developed in 1969 by a group of AT&T employees Ken Thompson, Dennis Ritchie at Bell labs.

→ user communicate with the kernel through a program known as the shell.

1) Kernel.

- The kernel is the heart of the OS.
- it interacts with the hardware and most of the tasks like memory management, task scheduling and file management.

2) shell

- The shell is the utility that processes your requests.
- When you type in a command at your terminal that shell interprets the command and calls the program that you want.
- C shell Bourne shell and korn shell are the most famous shells which are available with most of the Unix variants.

3) Command and utilities.

- These use various command and utilities which you can make use of in your day to day activities.
- CP MV CAT and grep etc. etc.
Few examples of commands and utilities.
- All the commands come along with various options.

4) Files and Directories

- All the data of Unix is organized into files.
- All files are then organized into directories.
- These directories are further organized into a tree-like structure called the filesystem.

Q. 4 A

1) How to define variable?

- type the name you want and set its value using the equals sign (=).

2)

2) gives the name of some shell keyword

- echo if trap
read else wait
set fi eval

3) what is system variable?

- These variables are used to configure the environment during the boot sequence of after user login so they are also variable

4) Explain test command

- evaluates the expression, parameters and if the expression value is true returns a zero exit value.

Q.4 B

1 Explain positional parameters.

- A positional parameter is a variable within a shell program.
- its value is set from an argument specified on the command line that invokes the program.
- positional parameters are numbered and are referred to with a preceding "\$": \$1, \$2, \$3, and so on.

\$0 The filename of the current script

\$# The number of arguments supplied to a script

\$" All the arguments are double quoted. If a script receive two arguments \$" is equivalent to \$1 \$2.

\$@ All the arguments are individually double quoted.

2 Explain Looping structure in unix.

- A looping is a series of commands that will continue to repeat over again until condition is met.
- In each case a block of code is executed repeatedly until a loop exit condition is satisfied.

1) while.

Syntax

while / condition :

do

Series of code

done.

- The bash while loop is a control flow statement that allows code or commands to be executed repeatedly based on given condition.

2) until

3) for

for variable in 1,2,3,4 - n

do

Series of code

done.

- The bash for loops allows be repeatedly executed the code to given no of iteration.

Q. 4 C

1) Explain decision statement in unix

- While writing a shell script may be a situation when you need to adopt one path out of the given two paths

- So you need to make use of conditional statement that allows your program to make correct decisions and perform the right actions.

- Unix shell supports conditional statements which are used to perform different action based on different conditions.
 - Unix shell provides decision making using if then else and case structures.
- if then fi
- If given condition is true then command is executed else command is not executed.

Syntax

```
if [condition]
then
    execute command if condition true
else
    execute some if condition false
fi
```

2) if _ elif _ else _ fi

- The if elif else fi stat is the one level conditional form of control stat that allows shell to make correct decision out of several cond.

3) The case esac statement

- Unix shell supports case esac stat which handles exactly this situation.
- There is only one form of case esac stat.

2) Explain nano editor!

→

GNU nano is a friendly and convenient text editor like vi and emacs.

- it offers many other extra features like word searching, replacing, jump to a line or column, filename, tab completion, auto indentation, etc.
- nano is a clone of the pico text editor. nano is not pre-installed in all distros but ubuntu has it.

Syntax

nano

- look at the above snapshot. This is the default nano screen on pressing command: nano.
- At the top GNU nano version is shown at the left and in the middle filename is shown being edited.
- At the end of the screen keyboard cmd are given.
- command written as $\text{ctrl} + \text{g}$ means press ctrl key and command $\text{M}-\text{R}$. means press $\text{alt} + \text{z}$.

- There is no use of uppercase letters in any of the keyboard command in nano editor.
- you can use lowercase letters with ctrl and alt keys.

Q. 4 D

1) Explain vi editors in detail.

- No matter what work you do with the unix system you will eventually write some c program or shell script.
- For all this you must learn to use an editor. An unix provides a very versatile one vi editor. vi editor is a screen editor.
- where a portion of the file is displayed on the terminal screen, and the cursor can be moved around the screen to indicate where you want to make changes.
- you can select which part of the file you want to have displayed.

* invoking vi

- it will put filename into a buffer and display the file on the screen.
- if filename does not exist, vi will create it.

* Mode is vi editor.



1) command mode

→ This is default mode of vi editor.

→ This mode is used to give some commands for navigation, edition and copy or cut.

2) Input or insert mode

→ This mode is used to enter data in vi editor.

→ This mode will be selected from

Command mode pressing `I` or `i` or `a`.

3) execute mode

→ This mode is used to save or quit from vi editor.

→ whatever the changes user has done in file using vi editor if user want to save

→ The vi editor this mode will supports user to quit from the vi editor.

* Switching mode in VI

→ While you start vi editor at that time you will be at vi mode that will ready to accept defined cmd on that particular key but not any input.

→ If you want to input text into the file you will have to go to input mode for that you will have to press `i`.

* Cursor Movement in Vi editor.

- whatever the command you are giving that will work on Vi mode only.
- The right arrow will cause the cursor to move one position to the right.

h. left arrow

j. down arrow

k. up arrow

l. right arrow

* Screen control command.

- whatever the command you are giving that will work on Vi mode only.

C + z + f ⇒ This will move the user forward.

C + z + b ⇒ This will move the user backward.

g ⇒ This will move the cursor to end of file.

num ⇒ This will bring the cursor.

2. write a shell script to display a digital clock.

- Unix has the inbuilt functionality to show the current time.

→ usually Linux shows the current time in general format rather than word and with this may some user can disagree.

→ if we direct run the time command in Linux or Unix then will get output like following.

Command: date +%T

Output: 11:02:08 PM IST

* Example

→ Current Time: 11:02:03:25 AM IST

Time in word: Two hours Three minutes
Twenty Five second AM

* Script code Approach

→

To write the script for display the time in the word we require some simple Linux inbuilt commands like date, echo.