Page:
Date:

Acharya Vishwas R.
181240116001
IT - 4th sem
Subject :- Operating System & Virtulization

INDEX

NO.	Name	Given Date	Submission Date	Sign	Grade
1.	Assignment-1	16/01/2020	22/01/2020	22/01	10
2.	Assignment-2	24/01/2020	3/2/2020	22/01	10
3.	Assignment-3	18/2/2020	25/2/2020	25/02	10
4.	Assignment-4	18/3/2020	30/3/2020		
5.	Assignment-5	18/4/2020	25/4/2020		
6.	Assignment-6	18/4/2020	25/4/2020		

Assignment - 1

Page:
Date:

1) What is Operating System? Explain Different types of Operating System.

Ans:- Operating System is an interface between user and hardware.

- In technical terms, Operating System is a collection of software to control coordinate and responsible for overall management of computer.

-:- Types of OS -:- (∵ OS = operating System)

1) Batch OS

→ User is not directly interact with computer each user prepare his job on offline device like punch card and submit to the computer operator to speed up processing similar types of jobs are combined together.

E.g.:- Bank Statement

2) Multiprogramming Os

→ Os perform multiple programs parallely on a single process.

CPU can be effeciently utilized more than one task or program or job or process can be done into main memory at same point of time.

Eg.:- CPU (windows, mac, ...)

3) Time Sharing Os

-> This type of Os is a technique which enables many people located at various locations to use particular computer system at same time.

-> Processor's time, which is shared among multiple users simultaneously is term as time sharing.

E.g.:- Multics

4) Distributed Os:

-> It means multiple central processor are used to serve multiple realtime application to multiple users.

E.g:- Online Examination System.

Page:
Date:

5) Network Os

-> It runs own server.

-> Network Os provides the server, the capability to manage data users, groups, security apps and other network function.

-> It is used to allow shared file and hardware excess among multiple computer in a network.

E.g :- Windows server 2003, 2008 & Sun Solaris.

6) Real time Os

-> A system which is designed to work on the basis of realtime app is known as realtime system.

(A) Hard realtime (Eg. :- Airlines)

(B) Soft realtime (Eg :- Ticket booking, etc)

Page:
Date:

2) What is system call? Explain the different categories of system call in detail.

Ans → When user wants to use any functionality of OS it is done with the system call.

- System call invoke kernel to perform task.

- System call is a special function used in kernel mode.

Kernel
RAM
Process

- There are various categories in system call.

1) File management
- Some common system calls are create, delete, read, write, reposition or close.
- Also, there is a need to determine the file attributes - get and set file attribute.

Page:
Date:

- Many times the OS provides an API to make these system calls.

2) Device Management
-> User programs request the device and when finished they release the device.
- Similar to files, we can read, write and reposition the device.

3) Information Management
- Some system calls exist purely for transferring information between the user program and the operating system.
- E.g. :- Time and Date

4) Process control
- A running program needs to be able to stop execution either normally or abnormally.
- when execution is stopped abnormally, often a dump of memory is taken and can be examined with a debugger

5) Communication
- There are two models of interprocess communication, the message-passing

Page:
Date:

model and the shared memory model.

- Message - passing uses a common mailbox to pass messages between processes.
- Shared memory use certain system calls to create and gain access to create and gain access to regions of memory owned by other processes.
  The two processes exchange information by reading and writing in the shared data.

6) Protection
- This type of system call provides protection of any data of any file so that no other people can misuse it.

E.g:- chmod ( ); [∵ change mode]

setfile security ( );

Page:
Date:

3) Explain the structure of operating System in detail.

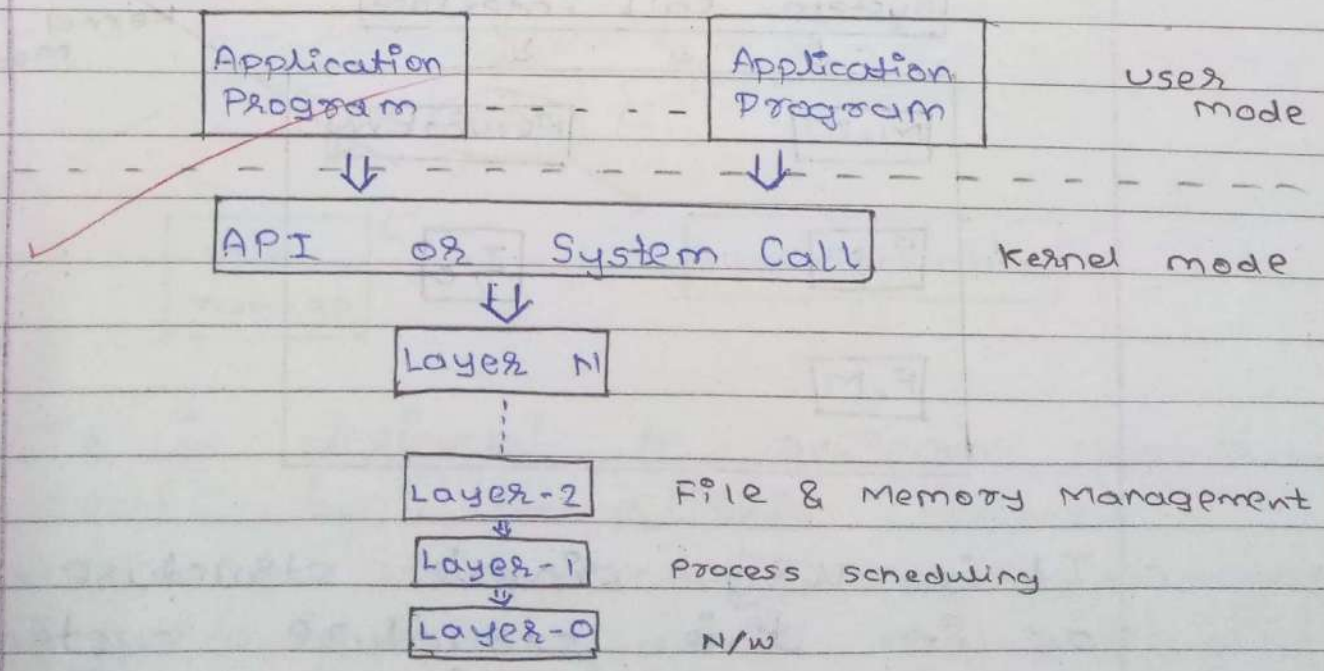
Ans: There are maily three types of structure of operating system.

1) Layered structure
2) Monolithic structure
3) Micro kernel structure

- Also one another is there which is Exokernel but it is under execution.

- Now, let discuss in detail.

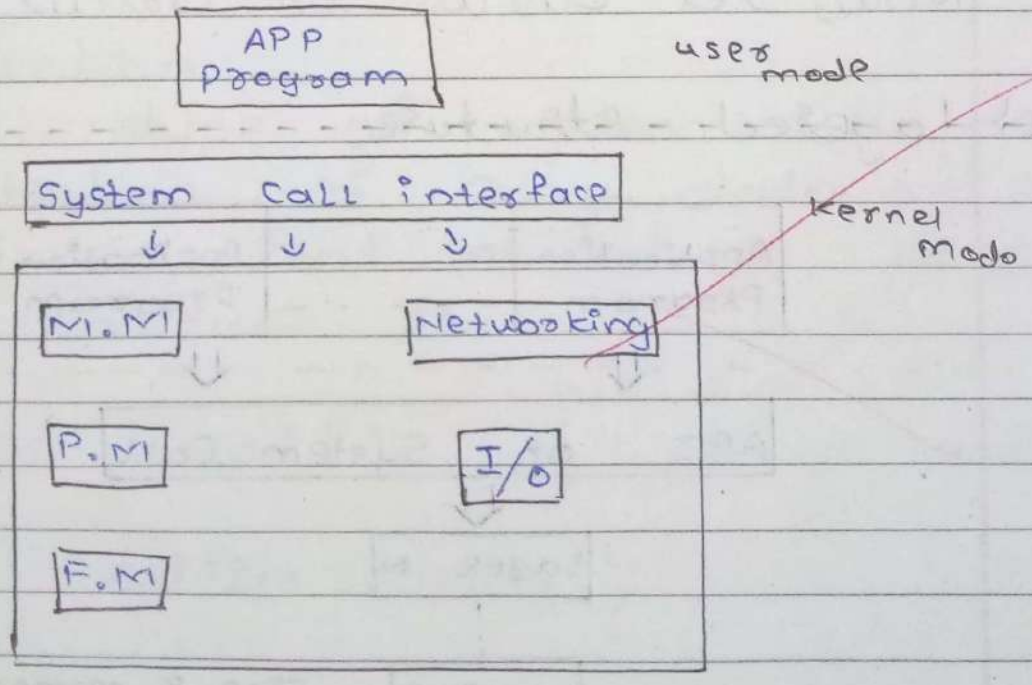
1) Layered structure



- In this approach the OS consist of several layer's where each layer has will defined functionalities
- Each layer can be designed, coded, & tested independently.
- The lowest layer - 0 deale will the hardware and top most layer end provides user interface and communicate takes place between adjecent layer only

E.g :- Microsoft Windows NT

2) Monolithic structure



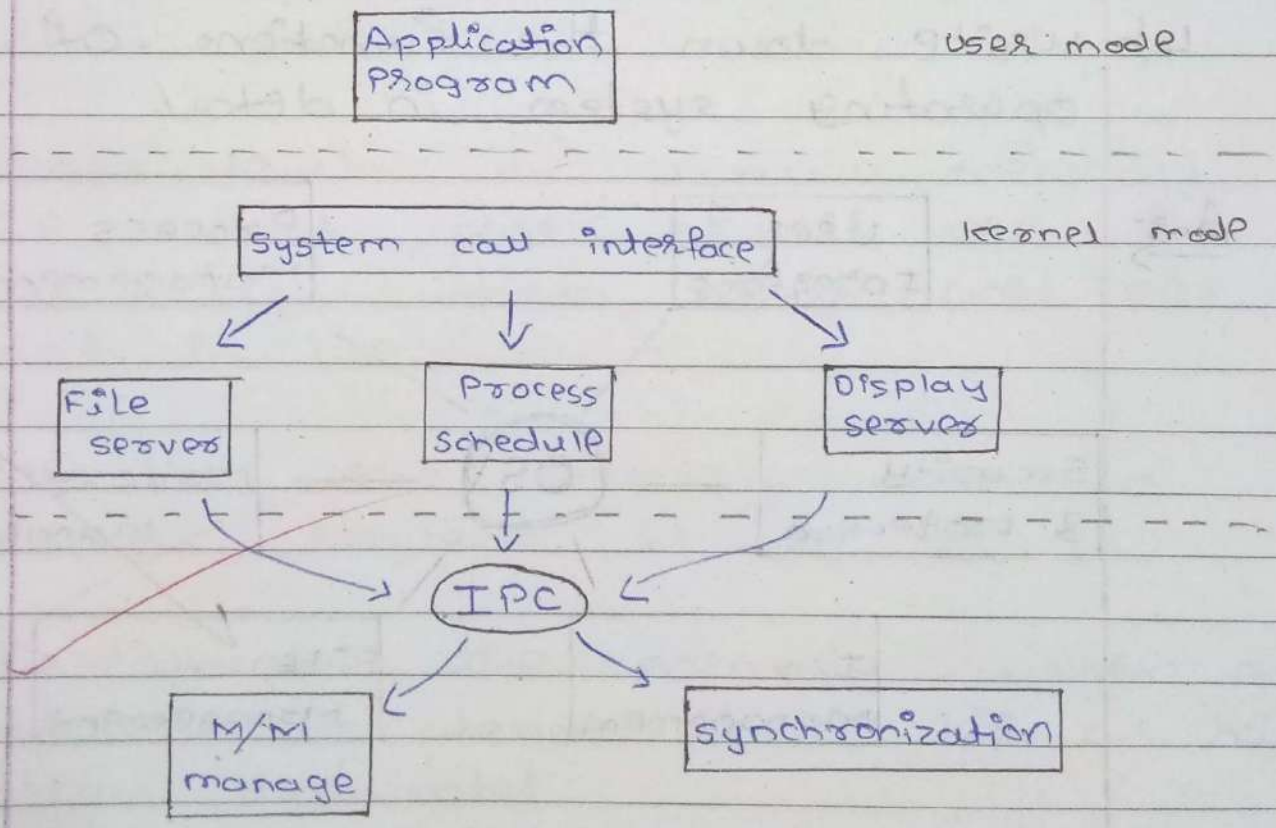
- It is very simple structure of OS in this structure system in very simple having a single CPU,

small size memory and few output device.

- In monolithic structure all the computer of OS are integrated under single unit.

E.g:- OS/360, Linux

3) Microkernel Structure



- It is designed to overcome problem test by monolithic kernel memory management, process management, Input/output ... were not included within kernel instead of each of

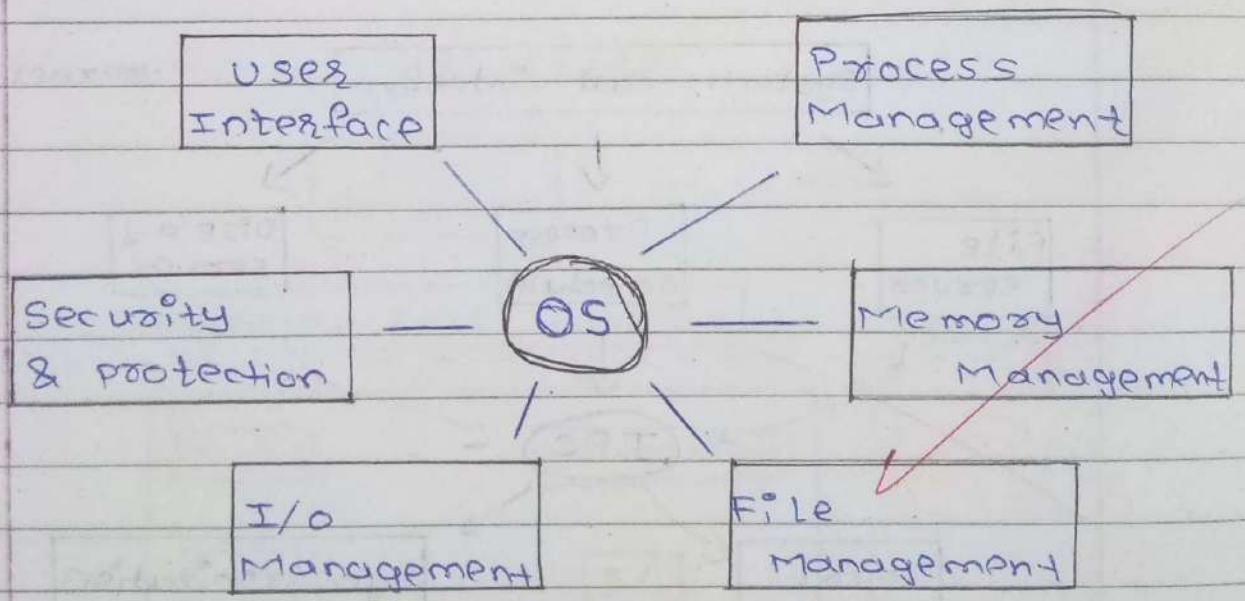
this functionality implimented through seperates server process each sorver occupies seperate spaces.

- User sends request to server for specific task and server sends back reply to user.

Eg :- Mach, AIX, OSX, MINIX

4) Write down the functions of operating system in detail.

Ans-



1) Process Management :

- keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller.

- Allocates the processor (CPU) to a process.

- De-allocates processor when a process is no longer required.

2) Memory Management:

- In multiprogramming, the OS decides which process will get memory when and how much.

- Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.

- Allocates the memory when a process requests it to do so.

- De-allocates the memory when a process no longer needs it or has been terminated.

3) File Management:

- Decides who gets the resources.

- Allocates the resources.

Page:
Date:

- De-allocates the resources.

- Keeps track of information, location, uses, status etc. The collective facilities are often known as file system.

4) I/O management:

- Keeps tracks of all devices connected to system.

- Designates a program responsible for every device known as the I/O controller.

- Decides which process get access to a certain device and for how long.

- Allocates device in an effective and efficient way.

- Deallocates devices when they are no longer required.

5) Security & Protection:

- OS uses password protection to

protect user data and similar other techniques.

- Also prevents unauthorized access to programs and user data

6) User Interface :

- Give's a Graphical and icon-based view to the user.

- Gets easier to do any program or use any functions.

=> Hence this all are the function's of operating system.

5) Explain the different services provided by the OS.

Ans: There are various types of services which are given below:

- Program execution
- I/O operations
- File System manipulation
- Communication
- Error Detection
- Resource Allocation

Page:
Date:

- Protection

-> let discuss in detail,

1) Program execution:
- loads a program into memory
- Executes the program
- Handles program's execution
- Provides a mechanism for process synchronization, communication and deadlock handling.

2) I/O operation:

- It read and write any file or device.
- Give access to required I/O device when required

3) File System manupilation

- Permission varies from read-only, read-write, denied and so on

- It provides an interface to create/delete files, directories and backup of file system.

Page:
Date:

4) Communication:

- Two processes often require data to be transferred between them

- Both processes can be on one computer or on different computers, but are connected through a computer network.

- Communication may be implemented by two methods, either by shared memory ~~ti~~ or by message passing.

5) Error handling:

- The OS constantly checks for possible errors.

- The OS takes an appropriate action to ensure correct and consistent computing.

6) Resource Management:

- The OS manages all kinds of resources using schedulers.

Page:
Date:

- CPU scheduling algorithms are used for better utilization of CPU.

7) Protection:

- The OS ensures that all access to system resources is controlled.

- The OS ensures the external I/O devices are protected from invalid access attempts.

- The OS provides authentication features for each user by means of passwords.

22/01

Assignment - 2

1) Explain PCB in detail.

Ans → PCB stands for Process Control Block, It is also known as Task control Block.

- PCB is a data structure that contains information of the process and related to it.
- List of processes are mentioned below:
  - Process identification
  - Priority Number
  - Program counter
  - Memory allocation
  - I/o status information
  - List of open files
  - Accounting information
  - Number of register
  - Process states.

- Explaination in detail

1) Process identification

:- Each process is uniquely identified by this process.

2) Priority Number

:- Os allocates priority to each process

Page:
Date:

3) Program Counter
- It indicates the address of next instruction to be executed.

4) Memory allocation.
- It contains the value of base register, limit register & page table depending on the memory system used by operating system.

5) I/O status information
- It maintains information about the open files and list of I/O devices allocated to the process.

6) List of open files.
- This are the different files that are associated with the process.

7) Accounting information
- It includes the amount of CPU used in real time & time limits also.

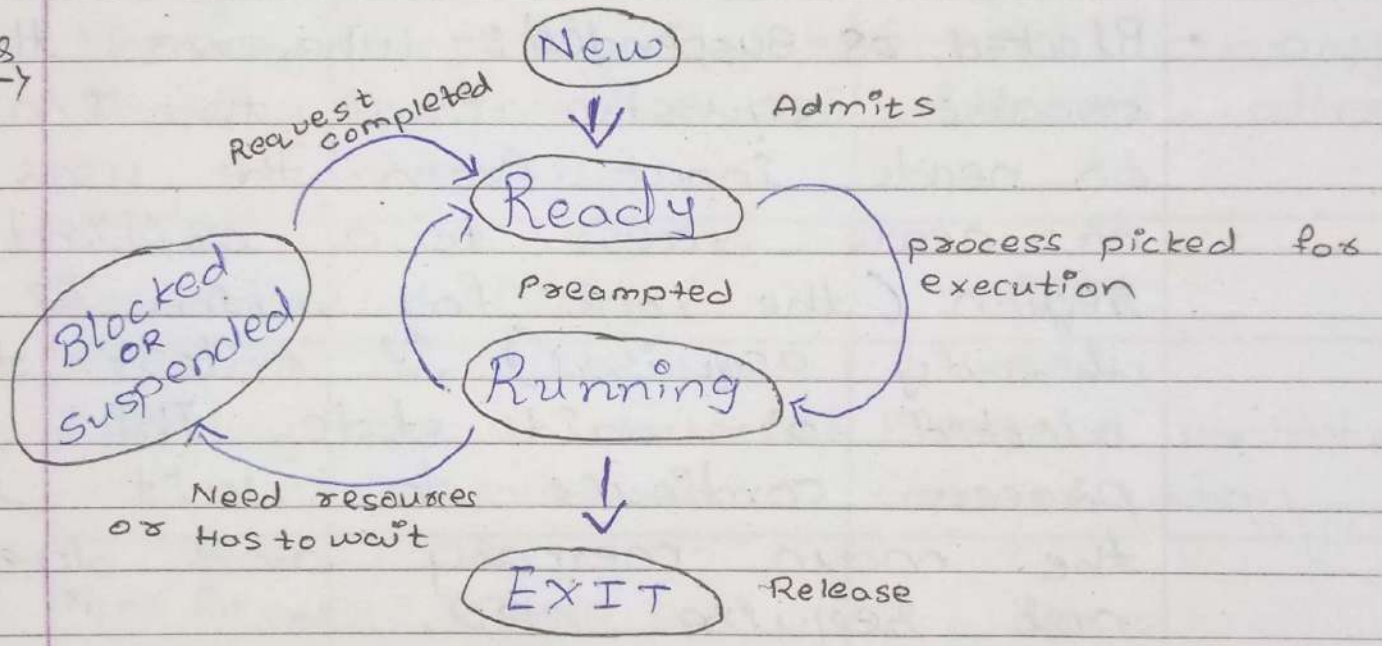
8) Number of Register
- This specifies the registers that are used by the process.

Page:
Date:

9) Process States.
- This specifies the process state i.e. new, ready, waiting or terminated

2) Explain process state diagram in detail.

Ans →



- New :- In this step, the process is about to be created but not yet created, it is the program which is present in secondary memory that will be picked up by OS to create the process.

- Ready :- Here, the process is loaded into main memory and ready to run and is waiting to get

Page:
Date:

the CPU time for its execution.

- Running :- The process is chosen by CPU for execution and the instructions within the process are executed by any one of the available CPU cores.

- Blocked OR Suspended :- Whenever the process requests access to I/O or needs input from the user or needs access to a critical region (the lock for which is already acquired) it enters the blocked or wait state. The process continues to wait in the main memory and does not require CPU.
  Once the I/O operation is completed the process goes to the ready state.

- Exit :- Process is killed as well as PCB is deleted.

3b Difference between process and thread & Explain thread in detail.

Ans →

Basis for comparison	Process	Thread
Basic	Program in execution	Lightweight process or part of it.
Memory Sharing	Completely isolated & do not share memory	Shares memory with each other
Resource Consumption	More	LESS
Efficiency	Less in context of Communication	More in context of Communication
Time for creation	More	Less
switch time	More	Less
Uncertain termination	Results in loss of process	A thread can be reclaimed
Time for termination	More	Less

Page:
Date:

- THREAD :-

- A thread is a flow of execution through the process code, with its own program counter the keeps track of which instruction to execute next, system registers which hold its current working variables, and a stack which contains the execution history.

- A thread shares with its peer threads few information like code segment, data segment and open files.

- When one thread alters a code segment memory item, all other threads see that.

- A thread is also called a lightweight process.

- There are two types of threads

(1) User Level Thread (ULT)
(2) Kernel Level Thread (KLT)

Page:
Date:

* User Level Thread (ULT)

- ULT are created by runtime.
- ULT are small & Faster.
- In ULT whole work of thread management is taken care by application rather than kernal
- User threads are supported above the kernel and manage without kernel support.
- ULT are also called many to one mapping thread.

Eg:- POSIX thread and MACHC thread

* Kernel Level Thread (KLT)

- In KLT management is done by kernel
- OS directly support's the kernel Level thread
- KLT supports one to one thread mapping
- KLT are slower than ULT
- Threads are constructed and control by system call.

Eg:- Window 95/98/NT, Sun Solaris

4) Following table gives arrival time and expected run time of five processes.

Process	Burst time	Arrival Time
A	8	1
B	1	4
C	2	2
D	1	5
E	5	6

Ignore process switching overhead. Find average turnaround time for following scheduling algorithm.

1) Round Robin (quantum = 1 sec)

=) Ready queue

A	C	A	B	C	A	D	E	A	E	A	E	A	E	A	E	A	

Gantt chart

A	C	A	B	C	A	D	E	A	E	A	E	A	E	A	E	A	

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Process	C.T	T.A.T	W.T
A	18	17	9
B	5	1	0
C	6	4	2
D	8	3	2
E	17	11	6

Page:
Date:

- Avg. TAT $= \frac{17+1+4+3+11}{5} = 7.2$

2) Shortest Job First

(A) Preemptive.

=> Gantt chart

A	C	C	B	D	E	E	A

1 2 3 4 5 6 7 11 18

Process	W.T	T.A.T
A	9	17
B	0	1
C	0	2
D	0	1
E	0	5

- Average waiting time $= \frac{9}{5} = 1.8$

- Avg. T.A.T $= \frac{26}{5} = 5.2$

(B) Non-Preemptive Scheduling

=> Gantt chart

B	D	C	E	A

1 2 3 5 10 18

Process	W.T	T.A.T
A	9	17
B	-3	-2
C	1	3
D	-3	-2
E	-1	4

Avg. Waiting time $= \frac{10-7}{5} = \frac{3}{5} = 0.6$

Avg. TAT $= \frac{20}{5} = 4$

3) First Come First serve

=> Gantt chart

A	B	C	D	E

1 9 10 12 13 18

Process	W. T	T. A. T
A	0	8
B	5	6
C	8	10
D	7	6
E	7	7

Avg. WT $= \frac{5+8+7+7}{5} = \frac{27}{5} = 5.4$

Avg. TAT $= \frac{8+6+10+6+7}{5} = \frac{37}{5} = 7.4$

Page:
Date:

5) What is schedule ? Explain queuing diagram representation of process schedules with figure.

Ans → Scheduling is a function by which resource are assigned to process.

- Process scheduling is a activity of process manager that handels the removal of the running process from the CPU and select the another process on basis of some stragey

:- Scheduling Queue

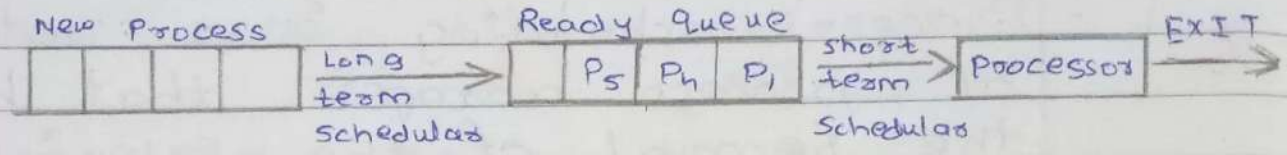
- It is queue of processor or I/O devices

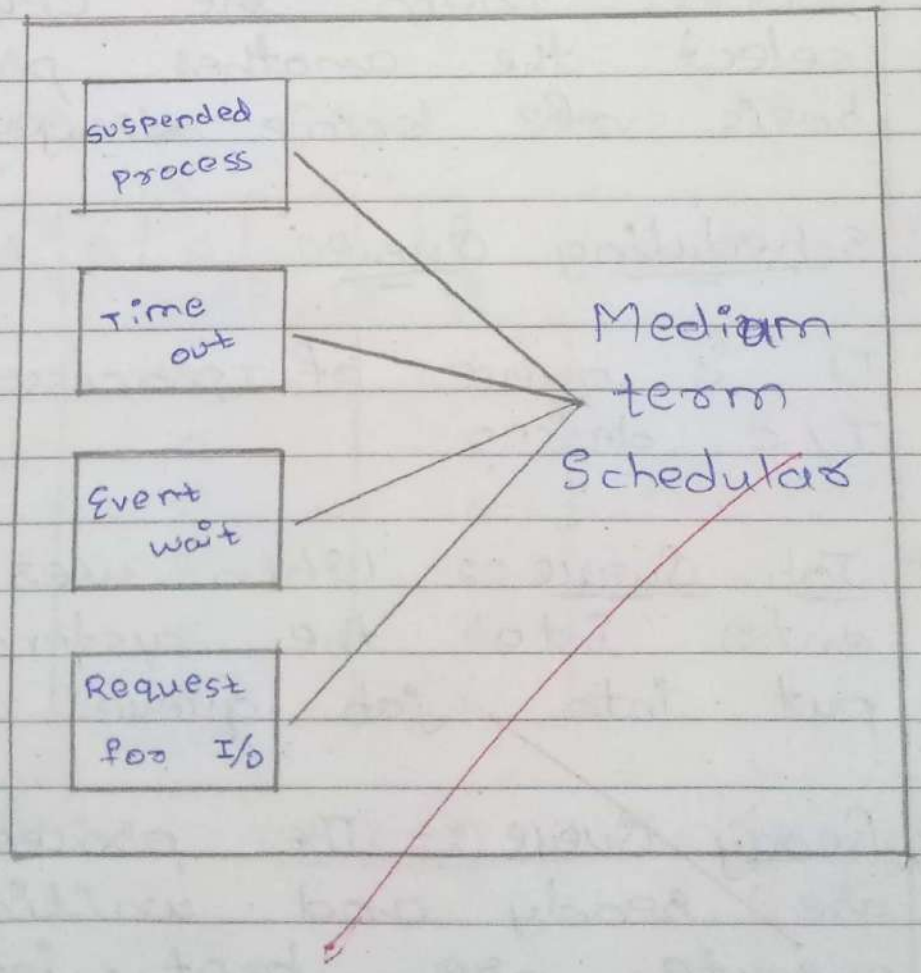
1) Job Queue :- when user process enter into the system then put into job queue.

2) Ready Queue :- The processes which are ready and waiting to execute are kept into ready queue.
Ready queue is stored into main memory.

3) Device queue : It is a queue for which multiple processes are waiting for a particular I/O device.

- Schedular





Page:
Date:

Q.4(2)(b)

Process	Burst time	A.T
A	8	1
B	1	4
C	2	2
D	1	5
E	5	6

Gantt chart

A	C	B	D	E

1 9 11 12 13 18

Process	W.T	T.A.T
A	0	8
B	7	8
C	7	9
D	7	8
E	7	12

Avg. W.T $= \frac{28}{5} = \boxed{5.6}$

Avg. T.A.T $= \frac{45}{5} = \boxed{9}$

Assignment 3

Page:
Date:

Q.1] Define the terms.

(1) Race Condition.
- It is an undesireable situation that occurs when a device or system attempts to perform two or more operations at the same time.

(2) Critical Section
- The part of program where the shared resource is accessed is called critical section or critical region.

(3) Mutual Exclusion
- Way of making sure that if one process is using a shared variable or file; the other process will be excluded (stopped) from doing the same thing.

(4) Monitor
- It is a collection of procedures, variables, and data structures that are all grouped together in a special kind of module or package.

(5) Signal.
- Signals are software interrupts sent to a program to indicate that an

Page:
Date:

important event has occurred.

Q.2] What is critical Section Problem and list the requirements to solve it. Write Peterson's Solution for the same.

=> The part of program where the shared resource is accessed is called critical section or critical region.

- Solution list

(1) Mutual Exclusion
- No two processes may be simultaneously inside the same critical section

(2) Bounded Waiting
- No process should have to wait forever to enter a critical section

(3) Progress
- No process running outside its critical region may block other processes

Page:
Date:

(4) Arbitary Speed.

- No assumption can be made about the relative speed of different processes (through all processes have a non-zero speed)

* Peterson's Solution

```
#define FALSE 0
#define TRUE 1
#define N 2
int turn;
int interested[N];
void enter region (int process)
{
  int other;
  other = 1- process;
  interested[process] = TRUE;
  turn = process;
  while (turn == process && interested[other==TRUE]
  }
  void leave region (int process)
  {
    interested[process] = FALSE;
  }
```

Page:
Date:

Q3] Explain the Semaphores.

=> A semaphore is a variable that provides an abstraction for controlling the access of a shared resource by multiple processes in parallel programming environment.

- There are two types of semaphores:

(1) Binary Semaphores
- It can take only 2 values (0/1).
- It have 2 methods associated with it (up, down/lock, unlock).
- They are used to acquire locks.

(2) Counting semaphores.
- Counting semaphore can have possible values more than two

- We want function insert_item and remove_item such that the following hold:

(1) Mutually exclusive access to buffer:
At any time only one process should be executing

Page:
Date:

(2.) No buffer overflow: A process executes insert_item only when the buffer is not full.

(3.) No buffer underflow: A process executes remove_item only when the buffer is not empty.

(4.) No busy waiting.

(5.) No producer starvation: A process does not wait forever at insert_item() provided the buffer repeatedly becomes full.

(6) No consumer starvation: A process does not wait forever at remove_item() provided the buffer repeatedly becomes empty

=> Two operations on semaphores are defined.

(1) Down Operation.
- The down operation on a semaphore checks to see if the value is greater than 0.

- If so, it decrements the value and just continues.
- If the value is 0, the process is put to sleep without completing the down for the moment.

(2.) Up Operation.

- The up operation increments the value of the semaphore addressed
- If one or more processes were sleeping on that semaphore, unable to complete an earlier down operation, one of them is chosen by the system and is allowed to complete its down.
- The operation of incrementing the semaphore and waking up one process is also ~~a~~ indivisible.
- No process ever blocks doing an up, just as no process ever blocks doing a wakeup in the earlier model.

Q.4] Explain the Producer-Consumer Problem. Solve it using Semaphores. Write the Pseudocode for the same.

=> The problem is to make sure that the producer won't try to add data (information) into the buffer if it is

full and consumer won't try to remove data (information) from the an empty buffer.

- To solve this problem, we need two counting semaphores - Full and Empty.

- 'Full' keeps track of number of items in the buffer at any given time and 'Empty' keeps track of number of unoccupied slots.

Initialization of Semaphores -
mutex = 1
Full = 0
Thus full slots are 0
Empty = n

- Solution of producer

```
do {
    wait (empty);
    wait (mutex);
    signal (mutex);
    signal (full);
} while (true)
```

- Solution for Consumer

```
do {
        wait (full);
        wait (mutex);
        signal (mutex);
        signal (empty);
    } while (true)
```

=> Hence this is the required solution of Producer-Consumer Problem.

Q.5] Explain the IPC Problem known as Dinning Philosopher Problem.

=> In this problem 5 philosophers sitting at a round table doing 2 things eating and thinking.

- But while eating they are not thinking and while thinking they are not eating.
- Each philosopher has plates that is total of 5 plates.
- And there is a fork place between each pair of adjacent philosophers that is total of 5 forks.
- Each philosopher needs 2 forks to eat and each philosopher can only use the forks on his immediate

left and immediate right.

- <u>Solution</u>

```
#define N 5
#define LEFT (i+N-1) % 5
#define RIGHT (i+1) % 5
#define THINKING 0
#define HUNGRY 1
#define EATING 2
typedef int semaphore;
int state [N];
semaphore mutex = 1;
semaphores[N];
void philosopher (int i)
{
  void test (i)
  {
     if(state [i] == HUNGRY &&
        state [LEFT] != EATING &&
        state [RIGHT] != EATING)
        {
           state [i] = EATING;
           up (& s[i]);
        }
  }
  test(i);
  up(& mutex);
  down (& s[i]);
}
```

```
void take forks (int i)
{
    down (& mutex);
    state[i] = HUNGRY;
    test (i);
    up (& mutex);
    down (& s[i]);
}
void put forks (int i)
{
    down (& mutex);
    state [i] = THINKING;
    test (LEFT);
    test (RIGHT);
    up( & mutex);
}
void test (i)
{
    if (state [i] HUNGRY &&
        state [LEFT] != EATING &&
        state [RIGHT] != EATING)
    {
        state [i] = EATING;
        up (& s[i]);
    }
}
```

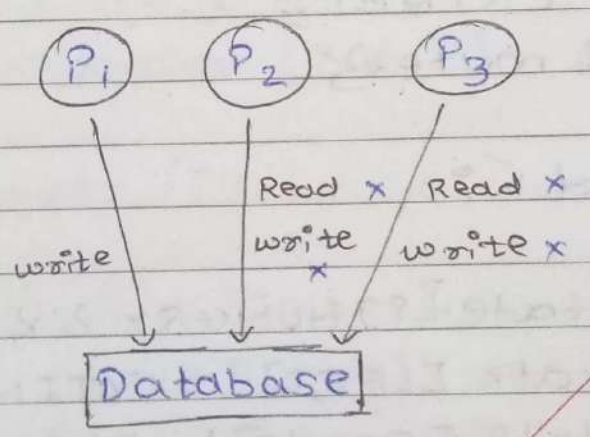
Page:
Date:

Q.6] What is semaphore? Explain the Readers-Writers Problem using semaphore

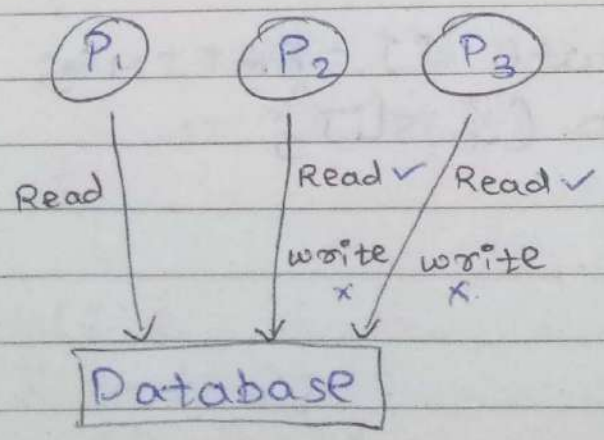
=> In the readers and writer problem, many competing processes are wishing to perform reading and writing operations in a database.

- It is acceptable to have multiple processes reading the database at the same time, but if one process is updating (writing) the database, no other processes may have access to the database not even readers

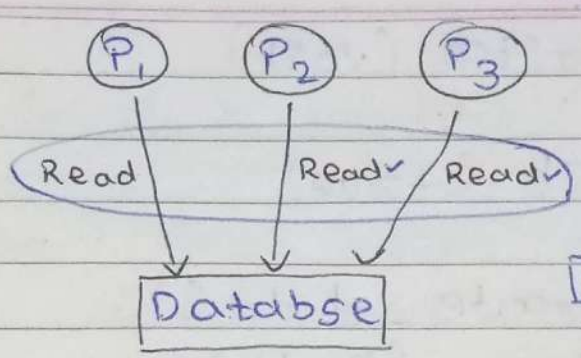
(1)



(2)



(3)



[3] Reader count

```
typedef int  semaphore;
semaphore  mutex = 1;
semaphore  db = 1;
int reader_count = 0;

void Reader(void)
{
   while (true) {
         down (&mutex);
         reader_count = reader_count + 1;
         if (reader_count == 1)
               down(&db)
               up (&mutex)
               read_database();
               down (&mutex);
               reader_count = reader_count - 1;
               if (reader_count == 0)
                     up(&db)
                     up(&mutex)
                     use_read_data();
                     }
}
```

Page:
Date:

```
void Writer (void)
{
    while (true)
    {
      create_data();
      down (&db);
      write_db();
      up (&db);
    }
}
```

25/02

Assignment - 4

Q. 1] Explain Deadlock recovery.

=> There are few steps of deadlock Recovery which are explain below:

(1.) Recovery through preemption.
- Some times resources are temporarily taken away from current process and allocated to other process.

(2.) Recovery through Rollback
- when a process in a system terminates, the system performs a roll back by undoing every operation related to terminated process
- Checkpointing a process means that its state is written to a file (PCB) so that it can be started later.
- Start the process after it is determined it can run again.

(3) Process Termination.
- Deadlock is removed by aborting a process. But aborting a process is not easy. All deadlock process are aborted.
- Circular wait can be eliminated

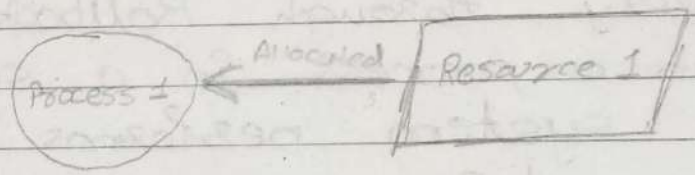
Page:
Date:

by aborting one by one.

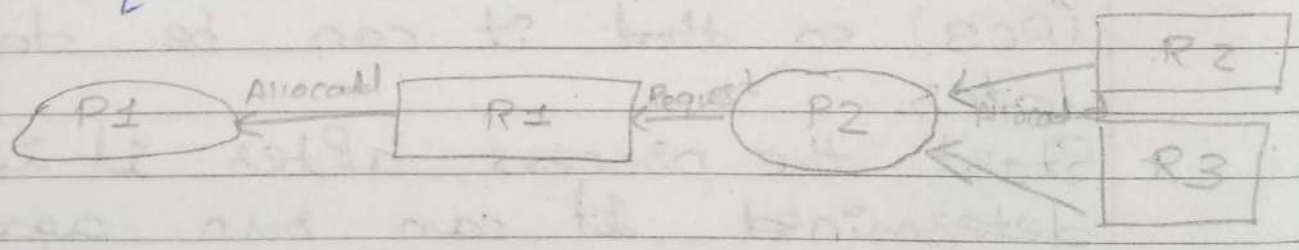
Q.2] What are the necessary conditions for deadlock? Explain each one.

=> The four condition's for deadlock is explain below,

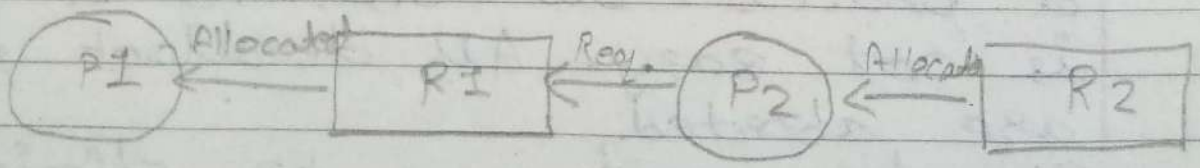
(1) Mutual Exclusion.
- Each resource is assigned to at most one process.



(2) Hold and Wait
- A process holding resources can request more resources

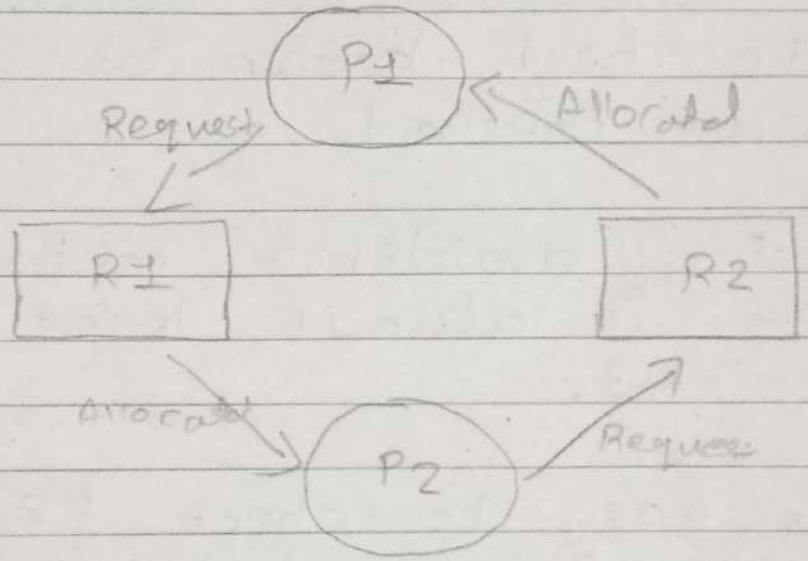


(3) No Preemption.
- Previously granted resources cannot be forcibly taken away.



(4) Circular Wait

- There must be a circular chain of 2 or more processes where each is waiting for a resource held by the next member of the chain.



Q.3] Explain the safety algorithm & resource request algorithm for Bankers algorithm.

=> Bankers's algorithm is the deadlock avoidence algorithm. The name is modeld after a banker who makes loads from a Root of capital and recivers payments that are returned to the real.

- Algorithm see that wheter it's granting leads to unsafe state. If it does the request is denied.

Page:
Date:

& if it do not then it is carried out.

- The chikstra proposed an algorithm to regular resource allocation to avoid deadlock. The banker's algorithm is the best known as the avoidence method.

- By using avoidence method the system is always kept in a safe stat.

- It is easy to check if a deadlock is created by granting a request deadlock avoidance uses the worst case analysis method to check for further deadlock.

- Safe state: There is at least one sequence of resource allocation in process that to not request in a deadlock.

- Deadlock algorithm is an unsafe state.

- Bankes algorithm uses following data structures

Weakness of Bankes's algorithm:

- It requires that there be a fixed number of resource to allocate.

- The algorithm require that users state their maximum need in advance.

- Number of user's must remain fixed

Q.4] Snapshot of the system is as per the below. (A, B, C) are 3 different resources

Process	Max			Allocation			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	0	1	0	0	1	1	5	2
P_1	1	7	5	1	0	0			
P_2	2	3	5	1	3	5			
P_3	0	6	5	0	6	3			

=> Firstly need matrix

Process	A	B	C
P_0	0	0	0
P_1	0	7	5
P_2	1	0	0
P_3	0	0	2

-> Process P_0 :

available = 1 5 2

need = 0 0 1

Available > need

so it will get allocated.

$$\begin{array}{r} 152 \\ 001 \\ \hline 153 \end{array}$$

=> Process P_1 :

available = 1 5 3

need = 0 7 5

need > available

So it will not get allocated

=> Process P_2 :

available = 1 5 3

need = 1 0 0

available > need

so it will get allocated

=> Process P_3 :

available = 2 8 8

need = 0 0 2

available > need

So it will get allocated

$$\left(\because \begin{array}{r} 153 \\ 135 \\ \hline 288 \end{array} \right)$$

=> Process P_1 :

available = 2 14 11

need = 0 7 8

available > need

so it will get allocated

$$\left(\because \begin{array}{r} 288 \\ 063 \\ \hline 2\,14\,11 \end{array} \right)$$

$$\begin{array}{r} 2 \quad 14 \quad 11 \\ 1 \quad 0 \quad 0 \\ \hline 3 \quad 14 \quad 11 \end{array}$$

Q.5] Explain memory partitioning with suitable diagrams.

=> Fixed Partitions

- Here memory is divided into fixed sized partitions.
- Size can be equal or unequal for different partitions.
- Generally unequal partitions are used for better utilizations.
- Each partition can accommodate exactly one process, means only single process can be placed in one partition.
- The partition boundaries are not movable.
- whenever any program needs to be loaded in memory, a free partition big enough to hold the programs is found. This partition will be allocated to that program or process.
- If there is no free partition available of required size, then the process needs to wait. Such process will be put in a queue.

Block size = 4mb	free = 3mb ←	Internal fragmentation
	P_1 = 1mB	
Block size = 8mb	P_2 = 7mb	
Block size = 8mb	P_3 = 7mb	
Block size = 16mb	P_4 = 14mb	

(Fixed size Partition)

Variable Partitioning

- It is a part of contiguous allocation technique. It is used to alleviate the problem faced by fixed partitioning
- In contrast with fixed partitioning, portions are not made before the execution or during system configure. Various features associated with variable partitioning.
- Initially RAM is empty and partitions are made during the run-time according to process's need instead of partitioning during system configure.
- The size of partition will be equal to incoming process.

- The partition size varies according to the need of the process so that the internal fragmentation can be avoided to ensure efficient utilisation of RAM.
- Number of partitions in RAM is not fixed and depends on the number of incoming process and Main memory's size.

operating system	
$P_1 = 2mb$	Block size = 2mb
$P_2 = 7mb$	Block size = 7mb
$P_3 = 1mb$	Block size = 1mb
$P_4 = 5mb$	Block size = 5mb
empty space of ram	

Q.6] What is paging? What is page table? Explain the conversion of virtual address to physical address in paging with example

=> Paging overcomes the problem of external fragmentation in OS by allowing non-contiguous memory allocation.

- There are two popular techniques used for non-contiguous memory

allocation : Paging & segmentation.

- Paging is a storage mechanism used to retrive process from the secondary storage into the main memory in the form of pages.
- 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.

PAGE TABLE :

- Page tabe is a data structure which translates virtual address into equivalent physical address.
- The virtual page number is used as an index into the page table to find the entry for that virtual page and from the page table physical page frame number is found.
- Thus the purpose of page table is to map virtual pages onto page frames.
- Page table consist of following fields
  - Page frame Number
  - Present / Absent bit
  - The protection bits
  - Modified bit

- Referenced bit
- Cashing disabled bit

* Conversion of Virtual Address to physical Address

- The virtual address is split into a virtual page number and an offset.
- With a 16-bit address and a 4KB page size, the upper 4 bits could specify one of the 11 virtual pages and the lower 12 bits would then specify the byte offset within the selected page.
- The virtual page number is used as an index into the Page table.
- If the present/absent bit is 0, it is page-fault; a trap to the operating system is caused to bring required page into main memory.
- If the present/absent bits is 1, required page is there with main memory and page frame number found in the page table is copied to the higher order bit of the output register along with the offset.
- Together page frame number and offset creates physical address.

- Physical address = Page frame Number + offset of virtual address

Q.7] Explain the Segmentation.

=> Segmentation is a memory management technique in which, the memory is divided into the ~~vela~~ variable size parts. Each parts is known as segment which can be allocated is a process.

- The details about each segment are stored in a table called as segment table is stored in one or many o s the segments

- Each segment is actually a different logical adress space as the programme

- When a process is to be executed, it is corresponding segmentation are loaded into non-continages memory through every segment is loaded into a continguas block, available memory.

- Segmentation memory management weeke very similar to ~~a~~ paging but here segments are of variable length where as in paging of fixed size.

Q.8 Which 3 are page replacement algorithm ? Discuss in terms of page fault.

=> The list of 3 page replacement algorithm are given below:

- First In First Out (FIFO)
- Optimal Page Replacement Algorithm
- Least Recently Used Algorithm

- Now, steps in handling a page Fault.

- Check the process control block table for reference which bit is set i.e. valid bit or invalid bit for memory access

- If invalid bit is set uses will terminate the process. If valid bit is loaded then page is loaded into main memory.

- Check for free frame for new page loading

- Start reading secondary storage device for desired page for allocated frame.

- Required page loaded into memory after completion of read operation.

- System restart the instruction that was interrupted by error, page is loaded into memory so process can use it.

Q.9] Explain concept of demand - paging in memory management.

=> Demand Paging System is similar to a paging system with swapping.

- A page swapper a page into memory only when it is needed.

- A process is a sequence of pages rather than one large configure address space

- In demand paging, instead as swapping the term paper is used

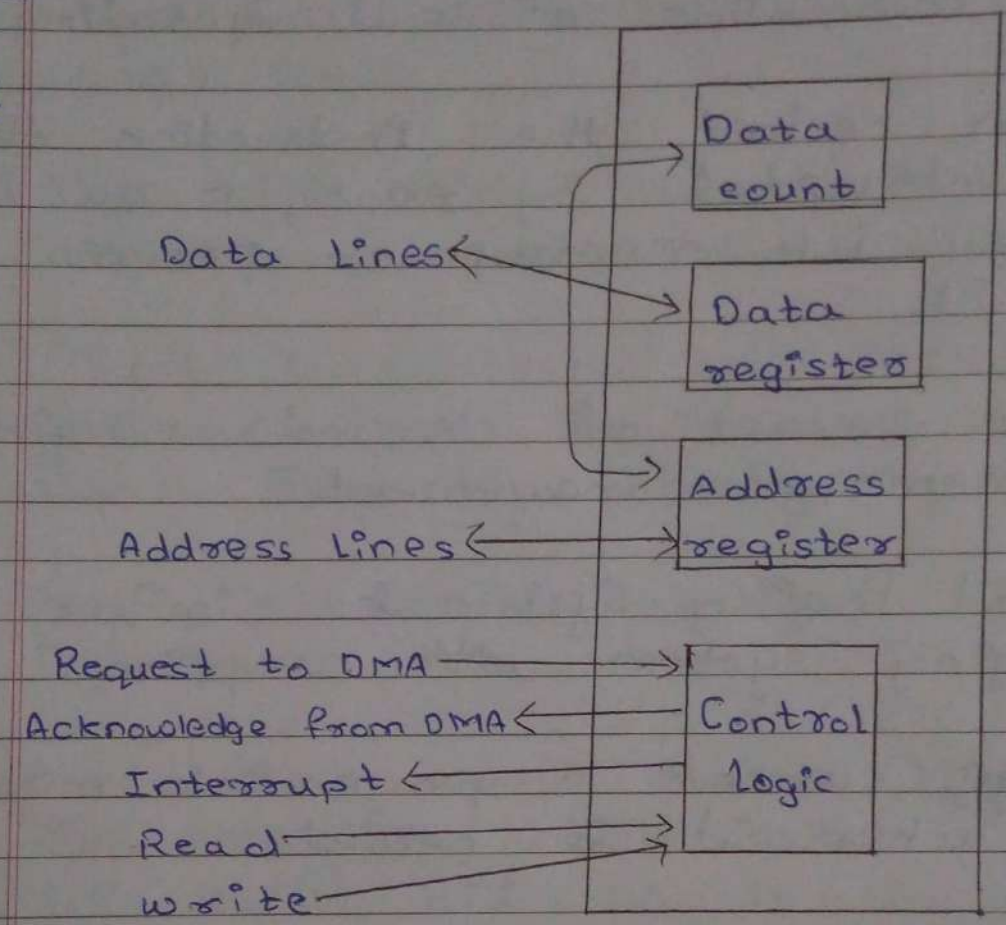
- Bringing a page into memory only when it is needed.

- When a page is refrenced and that page has no memory then get the page from disk and re-execute statement.

Assignment-5

Q.1] Explain block diagram of DMA.

Ans



- DMA stands for "Direct Memory Access" and is method of transferring data from the computer's RAM to another part of the computer without processing it using the CPU.

- While most data that is input or output from your computer is processed by the CPU, some data does not require processing, or can be processed by another device

- In these situation, DMA can save processing time and is a more

efficient way to move data from the computer's memory to other devices.

- In order for devices to use direct memory access, they must be assigned to a DMA channel.

- Each type of port on a computer has a set of DMA channels that can be assigned to each connected device.

- For example, a PCI controller and a hard drive controller each have their own set of DMA channels

Q.2] Explain RAID with all levels.

Ans RAID stands for Redundant Array of Independent Disks.

- RAID is a data storage virtualization technology that combines multiple physical disk drive components int a logical unit for the purposes of data redundancy, performance improvement, large storage capacity or all.

- Data is distributed across the drives in one of several ways, referred to as RAID levels, depending on

the required level of redundancy and performance.

- All RAID have the property that the data are distributer over drives, to allow parallel operation.

* RAID 0

- It is easy to implement, no parity calculation overhead is involved
- provides no redundancy or error detection and after certain amount of drives, performance does not increase significantly.
- It requires minimum 2 drives to implement.

* RAID 1

- It is most expensive implementation and requires twice as much storage space.
- In case of drive fails, data do not have to be rebuild, they just have to be copied to the replacement drive.
- Read performance can be improved and some fgs data can be read in parallel.
- write performance suffers.

* RAID 2

- It stripes data at bit level (rather than block level) with ~~ded~~ dedicated Hamming code parity.
- It uses Ecc [Error correcting code] to monited correction of information on disk.
- A parity disk is then used to ~~reconn~~ reconstruct corrupted or lost data
- In this RAID-2 each of 7 drives needs synchronized in terms of arm partition & rotational position & then it would be possible to write 7-bit Hamming coded word over the seven drives.

* RAID 3

- This techniques uses striping & dedicates one drive for storing parity information
- Here single parity bit is computed for each data word & written to a parity drive
- The embeded Ecc information is used to detect errors
- As in RAID 2 the drives must be exactly synchronized.

* RAID 4

- This level uses large stripes which means you can read records from any single drives.
- They donot require synchronization of drives
- It creates heavy load on parity drive.

* RAID 5

- It is based on block level striping with parity
- The parity information is stripped across each drive.
- As with RAID-4, there is a heavy load in the parity drive, it may become bottleneck.

~~* RAID~~

Q.3] Explain various types of I/O buffering

Ans I/O buffering is a mechanism that improves the throughout of input & output operations
- The various I/O buffering techniques are as follows.

(i) Single Buffering: Os assigns a buffer in main memory of an I/O request

- Block oriented
- Input transfers made to buffer.
- Block moved to user space when needed.
- Another block is moved into the buffer.
- User process can process one block of data while next block is read in.
- stream oriented
- uses a line at time
- Output to the terminal is one line at ~~a~~ buffering

(ii) Double buffering:- Uses two system buffers instead of one.

- A process can transfer data to or from one buffer while the operating system empties or fills the other buffer.
- It is also called as buffer swapping

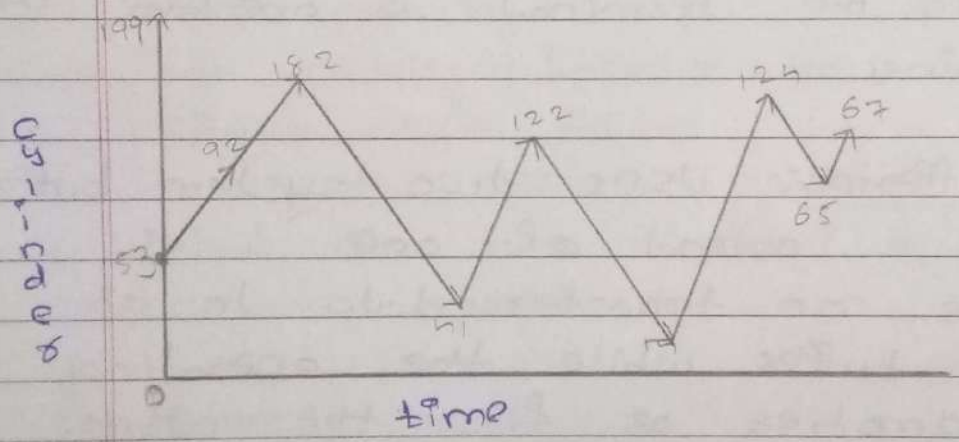
(iii) Circular Buffering:- In this more then two buffer are used.

- Each individual buffer is one unit in a circular buffer.
- Used when I/O operation must keep up with process.
- In this, producer can not pass the consumer because it would overwrite buffers before they had consuming.

Q.4) Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The head is initially at cylinder number 53. The cylinders are numbered from 0 to 199. Make the disk scheduling with FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK algorithm and find the seek time for all.

Soln: 98, 183, 41, 122, 14, 124, 65, 67

(i) FCFS:-



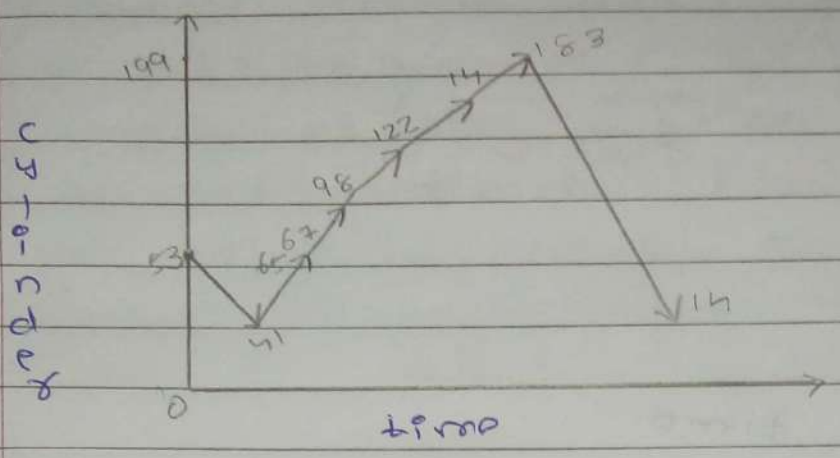
no. of tracked traversed = |53-98| + |98-183| + |183-41| + |41-122| + |122-14| + |14-124| + |124-65| + |65-67|

= 65 + 85 + 142 + 81 + 108 + 110 + 59 + 2

= 632

Avg. seek time = $\frac{692}{8}$ = $\boxed{79}$

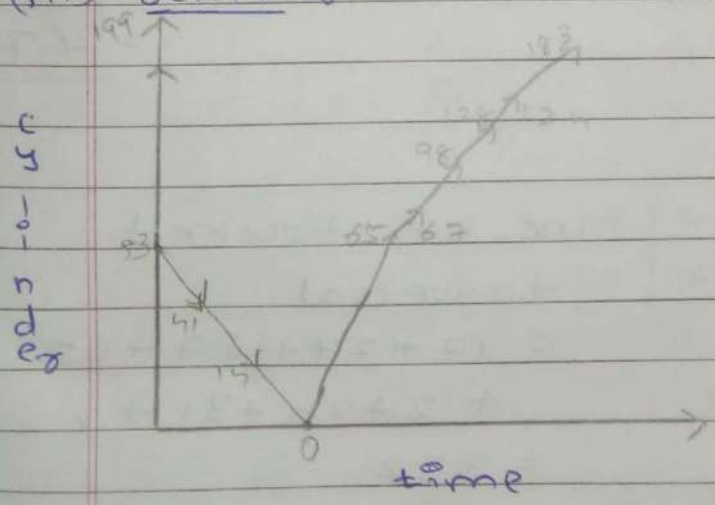
(ii) SSTF :-



no. of tracked traversal = 12 + 23 + 2 + 31 + 24 + 2 + 59 + 175
= 327

Avg. seek time = $\frac{328}{8}$ = $\boxed{41}$

(iii) SCAN :-



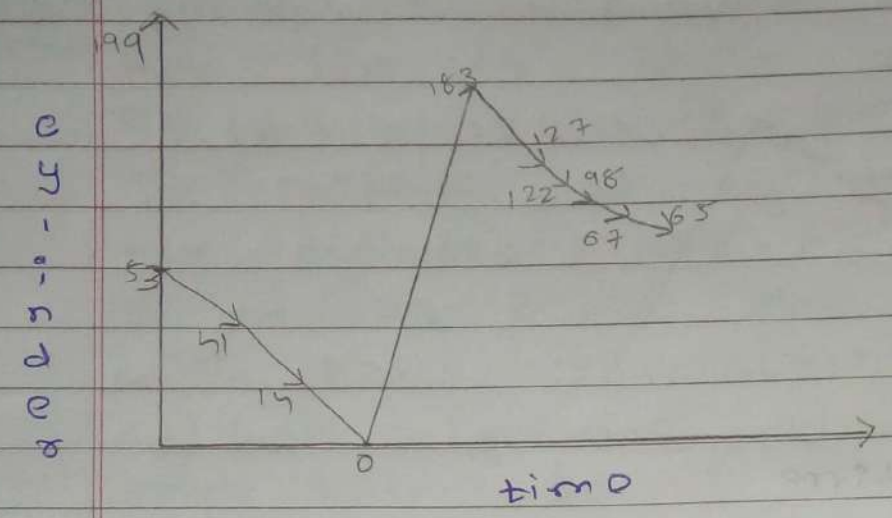
no. of tracked traversal
= 12 + 27 + 14 + 65 + 24 + 31 + 24 + 2 + 59
= 236

Avg. seek time = $\boxed{29.5}$

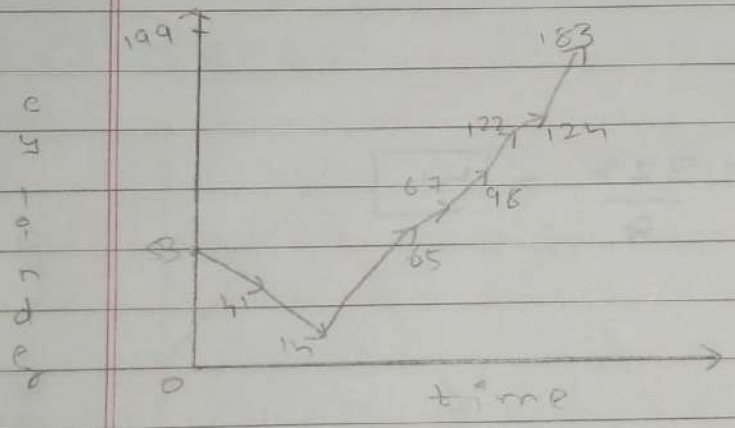
(iv) C-SCAN :-

no. of tracked traversal
= 12 + 27 + 14 + 183 + 59 + 2 + 24 + 31 + 2
= 354

Avg seek time = $\boxed{44.25}$



(v) <u>Look</u> :-



no. of tracked traversal

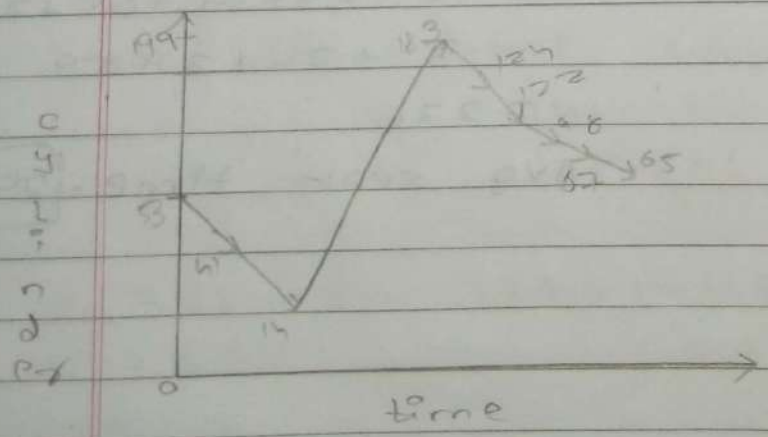
$= 12 + 27 + 51 + 2 + 31 + 24 + 2 + 65$

$= 214$

Avg. seek time $= \frac{214}{8}$

$= \boxed{26.75}$

(vi) <u>C-Look</u> :-



no. of tracked traversal

$= 12 + 27 + 169 + 65 + 2 + 24 + 31 + 2$

$= 332$

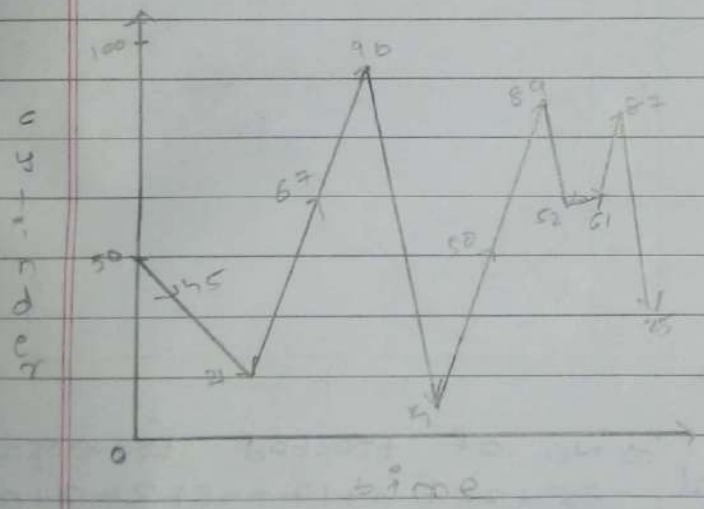
Avg. seek time

$= \frac{332}{8} = 41.5$

Q.5] Consider the following disk request sequence for a disk with 100 tracks 45, 21, 67, 90, 4, 50, 89, 52, 61, 87, 25 Head point starting at 50 & moving in left direction. Find the no. of head movements in cylinders using FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK.

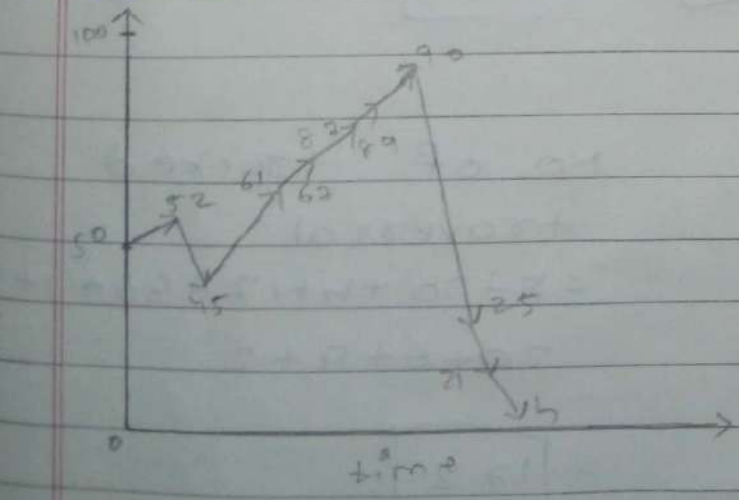
Sol^n 45, 21, 67, 90, 4, 50, 89, 52, 61, 87, 25
intially at 50.

(i) FCFS :-



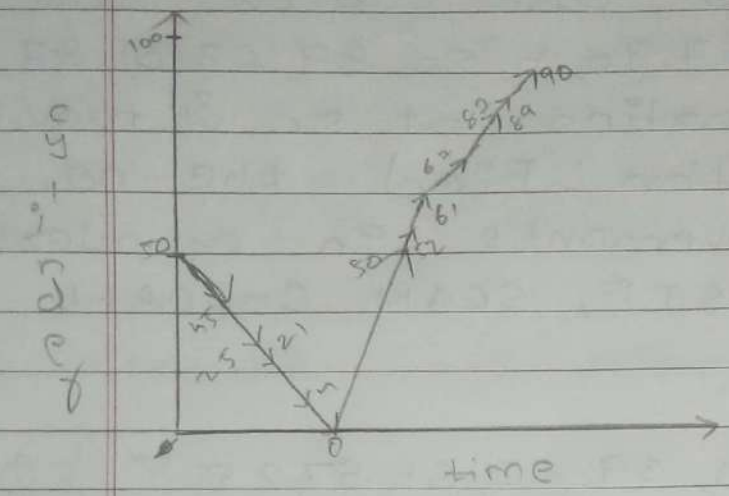
No. of tracked traversal
= 5 + 24 + 46 + 23 + 36 + 46 + 49 + 37 + 9 + 26 + 62
= $\boxed{413}$

(ii) SSTF :-



No. of tracked traversed
= 2 + 0 + 7 + 16 + 2 + 20 + 22 + 23 + 65 + 4 + 17
= $\boxed{178}$

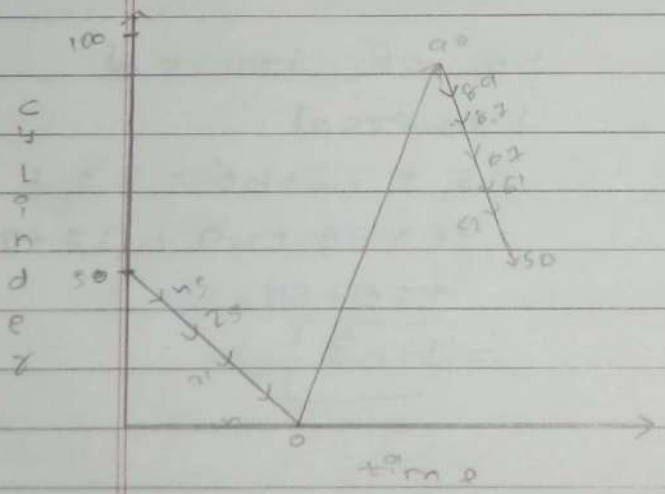
(iii) S-CAN :-



No. of tracked traversel = 5 + 20 + 4 + 17 + 4 + 50 + 2 + 9 + 6 + 20 + 2 + 1

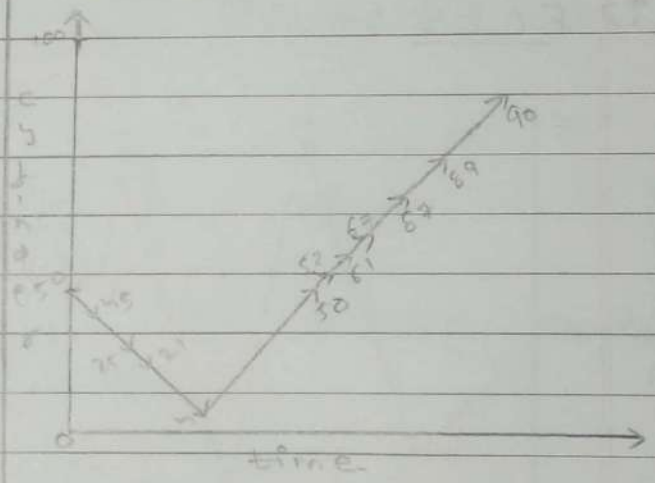
= 140

(iv) C-SCAN :-



⇒ No. of tracked traversal = 5 + 20 + 4 + 17 + 4 + 90 + 3 + 20 + 6 + 9 + 2 = 181

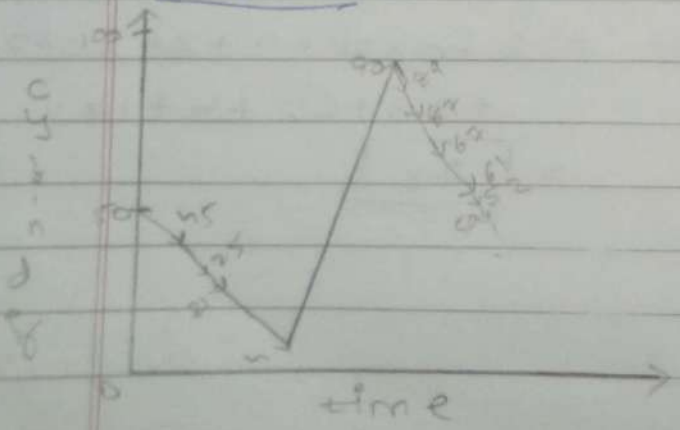
(v) Look :-



⇒ No. of tracked traversal = 5 + 20 + 4 + 17 + 46 + 2 + 9 + 6 + 26 + 2 + 1

= 132

(vi) C-Look :-



No. of tracked traversal = 5 + 20 + 4 + 17 + 86 + 1 + 2 + 20 + 6 + 9 + 2

= 172

~~Q.6] Consider the following disk request sequence for a disk with 100 tracks 45, 21, 67, 90, 4, 50, 89, 52, 61, 87, 25. Head pointer starting at 50 and moving in left direction. Find the number of head movements in cylinders using FCFS, SSTF, SCAN, C-SCAN~~

Q.6] Consider a reference string: 4, 7, 6, 1, 7, 6, 1, 2, 7, 2. The number of frames in the memory is 3. Find out the number of page faults respective to: Optimal Page Replacement Algorithm, FIFO Page Replacement Algorithm and LRU Page Replacement Algorithm

Sol^n: 4, 7, 6, 1, 7, 6, 12, 7, 2

(i) Optimal Page Replacement Algorithm

f_3			6	6	6	6	~~6~~	2	2						
f_2		7	7	7	7	7	7	7	7						
f_1	4	4	~~4~~	1	1	1	1	1	1						
	PF	PF	Hit	PF	PF	PF	Hit	PF	PF						

No. of page Fault = 7
No. of page Hit = 2

(ii) FIFO Page Replacement Algorithm

f_3			6	6	6	6	6	~~6~~	7	7
f_2		7	7	7	7	7	~~7~~	2	2	2
f_1	4	4	~~4~~	1	1	1	1	1	1	1
	PF	PF	Hit	PF	Hit	Hit	Hit	PF	PF	Hit

no. of PF = 5
no. of PHit = 5

(iii) LRU Page Replacement Algorithm

f_3			6	6	6	6	6	6	6	6	
f_2		7	7	7	7	7	7	7	7	7	
f_1	4	4	~~4~~	1	1	1	~~1~~	2	2	2	
	PF	PF	PF	PF	Hit	Hit	Hit	PF	Hit	Hit	

no. of Page Fault = 5
no. of Page Hit = 5

Q.7) The reference String is given as 0 1 5 3 0 1 4 0 1 5 3 4. number of frames in memory are 3. Find the number of page fault to analyze belady's anamoly.

Sol^n 0, 1, 5, 0, 1, 4, 0, 1, 5, 3, 4
Frames = 3
Belady's anamoly (In FIFO page replacement algorithm)

f_3			5	5	~~5~~	1	1	1	1	1	3	3					
f_2		1	1	~~1~~	0	0	0	0	0	5	5	5					
f_1	0	0	~~0~~	3	3	3	4	4	4	4	4	4					
	PF	PF	PF	PF	PF	PF	PF	Hit	Hit	PF	PF	Hit					

no. of Page Fault = 9
no. of Page Hit = 3

Assignment-6

Page No.:
Date:

Q.1] What is Virtualization? Explain virtual machines.

Ans Virtualization is a technology that helps us to install different operating System on hardware.

- "In computing virtualization is a broad term that refers to the abstraction of computer resources".

- It hides the physical characteristics of computing resources from their users, their applications or end users.

- This includes making a single physical resource appear to function as multiple virtual resources.

- It can also include multiple physical resources appear as a single virtual resource

• Single Physical Resource

- server, an OS, an application

• Multiple Physical Resource

- Storage devices

Page No.:
Date:

Virtual Machines

- It is a software program or OS that not only exhibits the behaviour of a separate computer, but is also capable of performing tasks such as running applications and programs like a seperate computer
- The end used has the same experience on a virtual machine as they would have on dedicated hardware
- It abstracts the hardware of our personal computer such as CPU, disk drives, memory, NIC (Network Interface Card) etc, into many different execution environments as per our requirements, giving us a feel that each execution environment is a single computer
- For example, Virtual Box.

Q2 What is hypervisors? Explain types of it.

Ans Hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time.

Page No.:
Date:

- A hypervisor is sometimes also called virtual machine manager (VMM)

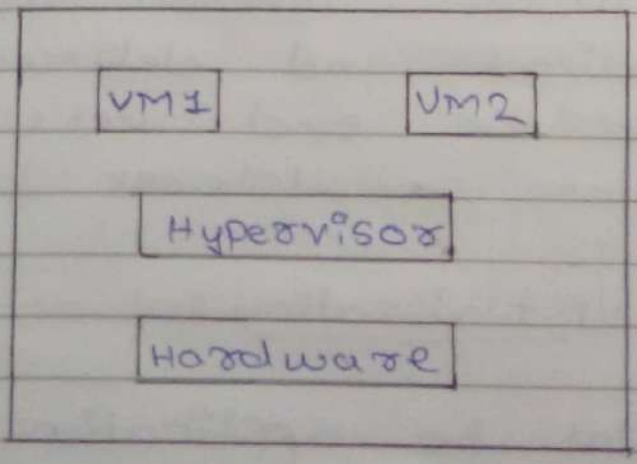
- It is a form of virtualization software used in cloud hosting to divide and allocate the resources on various pieces of hardware

- The program which provide partitioning, isolation or abstraction is called virtualization hypervisor

- It has two types:

• Type-1

- It run directly on underlying host system.
- And is also known as "Native Hypervisor" or "Bare metal hypervisor"
- It has direct access to hardware resources and it does not require any base server OS.
- Eg., VMware ESXi, Citrix XenServer and Microsoft Hyper-V hypervisor

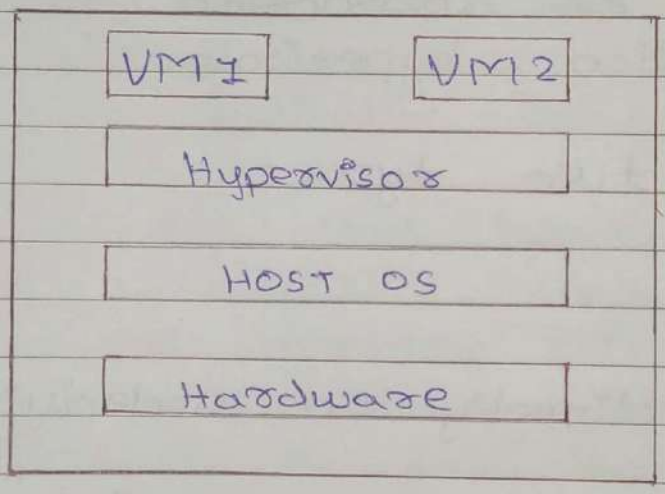


-: Figure

Page No.:
Date:

- Type - 2

- A Host OS runs on underlying host system. and it is also known as "Hosted Hypervisor".
- Basically a software installed on an OS. Hypervisor asks operating system to make hardware calls.
- E.g., VM Player or Parallel's Desktop.



-: Figure

Q3) Explain different types of Virtualization.

Ans Types of virtualization are given below:

1) Application Virtualization :

- This is a process where applications are virtualized and delivered from a server to the end user's device, such as laptops, smartphones & tablets.

2) Desktop Virtualization :

- It is similar to application virtualization

Page No.:
Date:

- It separates the desktop environment as a "Virtual desktop infrastructure" (VDI).
- One of the biggest advantages of desktop virtualization is that users are able to access all their personal files and applications on any PC.

3) Hardware Virtualization

- This virtualization is made possible by hypervisor. The hypervisor creates virtual versions of computer and operating systems and consolidates them into one large physical server.

4) Storage Virtualization

- This type of virtualization is very easy and cost-effective to implement, since it involves compiling your physical hard drives into a single cluster.
- It is handy when it comes to planning for disaster recovery, since the data stored on your virtual storage can be replicated and transferred to another location.

Page No.:
Date:

Q4 Explain the implementation levels of virtualization.

Ans The implementation levels of virtualization are given below:

1) Instruction Set Architecture Level:
At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine.

- For example, MIPS binary code can run on an x86-based host ~~can run on an~~ machine with the help of ISA emulation.

2) Hardware Abstraction Level: It is performed right on top of the bare hardware.
- On the one hand, this approach generates a virtual hardware environment for a VM.
- On the other hand, the process manages the underlying hardware through virtualization.
- The idea is to virtualize a computer's resources, such as its processors, memory, and I/O devices.

3) Operating System Level: This refers to an abstraction layer between traditional OS and user applications

Page No.:
Date:

- OS-level virtualization creates isolated containers on a ~~sig~~ single physical server and the OS instances to utilize the hardware and software in data centers.

4) Library Support Level : Most applications use API's exported by user-level libraries rather than using lengthy system calls by the OS. ~~Since~~
- Since most systems provide well-documented APIs, such as interface becomes another candidate for virtualization.
- Virtualization with library interfaces is possible by controlling the communication link between applications and rest of a system through API hooks.

5) User-Application Level : Virtualization at the application level virtualizes an application as a VM.
- On a traditional OS, an application often runs as a process.
- Therefore, application level virtualization is also known as process-level virtualization

Q.5 Explain Java Virtual Machine.

Ans Java Virtual Machine is a virtual machine that enables a computer to run java programs as well as programs written in other languages and

Page No.:
Date:

compiled to java Bytecode
- Jvm is virtual because, it provides a machine interface that does not depend on the operating system and machine hardware architecture
- Jvm interprets the byte code into the machine code.
- Jvm itself is platform dependent, but Java is Not.

Q6) Explain Microsoft's hyper-V virtual machine.

Ans Hyper-V is Microsoft's hardware virtualization product. It lets you create and run a software version of a computer, called a virtual machine.

- Each virtual machine acts like a complete computer, running an operating system and programs.

- when you need computing resources, virtual machines give you more flexibility help save time and money, and are a more efficient way to use hardware than just running one operating system on physical hardware.

- Hyper-V runs each virtual machine in its own isolated space, which means you can run more than one virtual machine on the same hardware

Page No.:
Date:

at the same time.
- You might want to do this to avoid problems such as a crash affecting the other workloads, or to give different people, groups or servies access to different systems.

Q.7] Explain android virtual machine.

Ans The Dalvik Virtual Machine (DVM) is an android virtual machine optimized for mobile devices.
- It optimizes the virtual machine for memory, battery life & performance
- Dalvik's a name of town in Iceland. The Dalvik VM was written by Dan Bornstein.
- The Dex compiler converts the class files into the .dex file that run on the Dalvik VM.
- Multiple class files are converted into one dex file.
- Dalvik is discontinued process virtual machine (VM) in android OS that executes applications written for android.
- Dalvik virtual Machine (DVM) is a register-based virtual machine & specially designed & optimined for android to ensure that a devices can run multiple instances efficiently.
- It depends on the layer linux kernel for threading & low level memory management.