B S Shreesha
1JT21IS010

OPERATING SYSTEMS ASSIGNMENT-1

1a) What is Interprocess communication? Explain direct and indirect communication with respect to message passing system

Ans Interprocess Communication:- Processes executing may either be co-operative or independent.

The communication between 2 or more processes is known as the Interprocess communication.

- Independent processes:- Process that cannot affect other processes or be affected by other processes executing in the system.
- Cooperating processes:- Process that can affect other processes or be affected by other processes executing in the system

(i) Direct Communication:- The sender and receiver must explicitly know each other's name. The syntax for send() & receive() function are as follows

- send (P, message) - Send a message to process P
- receive (Q, message) - receive a message from process Q.

Disadvantages:- Any other/changes in the identifier of a process may have to change the identifier in the whole system where the messages are sent & received.

Types of addressing

- Symmetric addressing:- Both sender & receiver have to name each other
- Asymmetric addressing:- Only sender's name is mentioned but the receiving data can be from any system.

Properties:-

- A link is established automatically b/w every pair of processes that wants to communicate
- A link is associated with exactly one pair
- Between each pair, there exists only one link.

(ii) Indirect Communication: It uses shared mailboxes or ports. A mailbox or port is used to send and receive messages. It has a unique ID. Using this identifier messages are sent and received.

The send and receive functions are

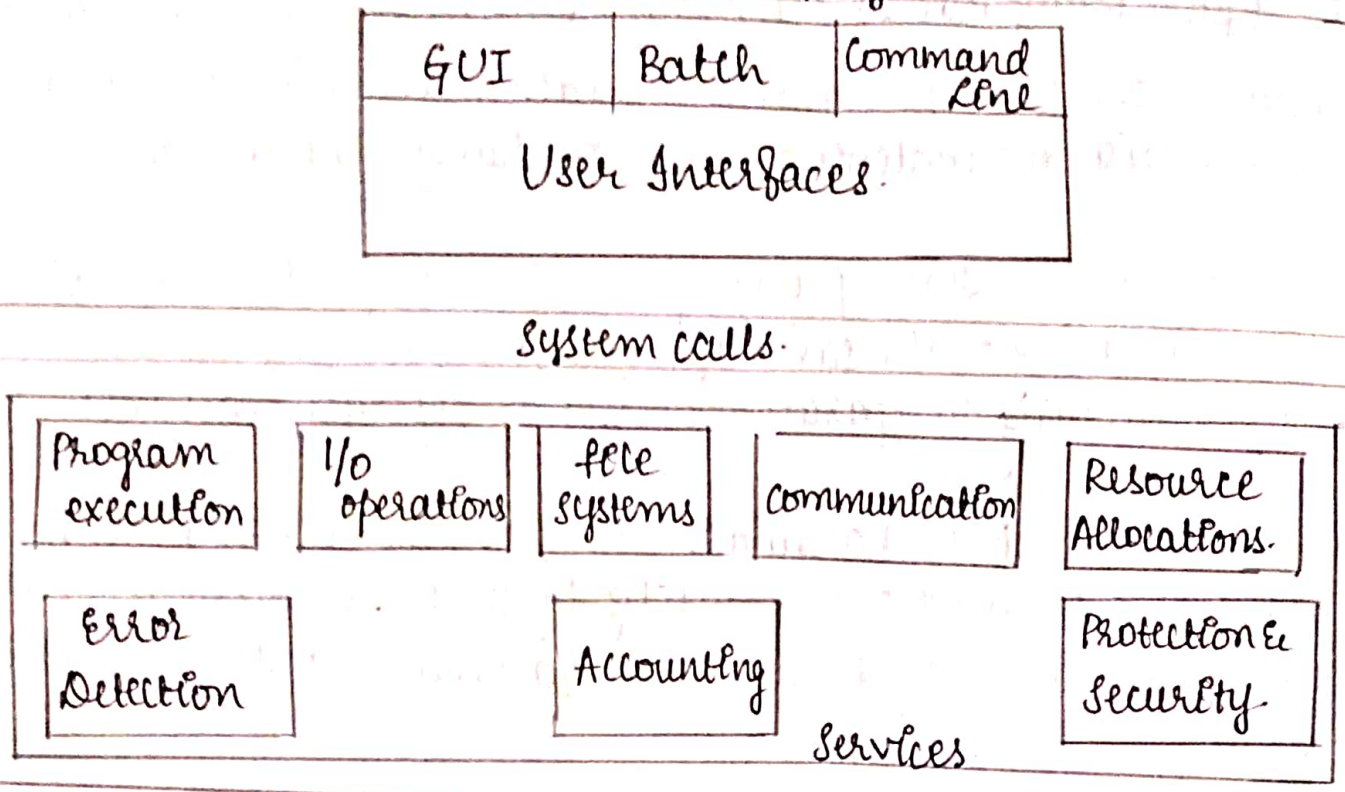
- send (A, message) - send message to mailbox A
- receive (A, message) - Receive message from mailbox A

Properties:-

- A link is established between a pair of processes only if they have a shared mailbox
- A link may be associated with more than 2 processes
- Between each communicating process, there maybe any number of links
- A mailbox can be owned by the OS. It must take steps to
  - Create a new mailbox
  - Send & receive from mailbox
  - Delete mailboxes.

1b) Explain the services provided by OS for the user & efficient operation of system

Ans



- User Interfaces:- Means by which user can issue commands to the system. Depending on the OS, there can be a command-line interface or a batch command systems.

  CLI - Commands are given to the system

Batch interface - commands & directives to control these commands are put in a file, then the file is executed.

- Program Execution:- The OS must be able to load a program into RAM, run the program and terminate the program either normally or abnormally.

- I/O operations:- The OS is responsible for transferring data to & from the I/O devices including keyboards, terminals, printers & files

A

- File System Manipulation:- Programs need to read & write files or directories. The services required to create or delete files, search for a file, list the contents of a file or change the permissions

- Communications:- Inter-process communication, IPC, either blw processes running on the same processes/processor or between processes running on separate processes or seperate machines

- Error Detection:- Both hardware and software errors must be detected and handled appropriately by the OS. Errors may occur in the CPU and memory hardware, in I/O devices, and in the user program

- Resource Allocation:- Resources like CPU cycles, main memory, storage space and I/O devices must be allocated to multiple users and multiple jobs at the same time

- Accounting:- There are services in OS to keep track of system activity and resource usage, either for billing purposes or for statistical record keeping that can be used to optimize future performance

- Protection & Security:- The owners of information in multiuser or networked computer system may want to control the use of that information. One process should be/should not interfere with other or with OS. Protection involves ensuring that all access to system resources is controlled

2b) What are System calls? Explain different types of system calls.

Ans A system call is a method for a computer program to request a service from the kernel of the operating system on which it is running

- System calls provides an interface to the services of the OS. These are generally written in C or C++
- The below figure illustrates the sequence of system calls required to copy a file content from one file to another

Types of System calls.

These maybe categorised into 6 major categories

ⓐ Process Control - end, abort, load, execute, create, terminate processes
ⓑ File Management - Create, delete files, open, close, read, write
ⓒ Device Management - request, release devices, read, write, reposition
ⓓ Information Management - get time & date, set time & date
ⓔ Communications - Create, delete communication connection, send/receive
ⓕ Protection.

ⓐ Process Control:- Process control system calls include end, abort, load, execute, create process, terminate process

- Process must be created, launched, monitored, paused, resumed & eventually stopped
- When one process pauses or stops, then another must be launched or resumed
- After creating the new processing/process, that the process (parent) may have to wait, or wait for an event to occur. The process sends back a signal when the event has occured

(b) File Management :- The file management functions of OS are
- File management systems calls include create file, delete file, open close, read, write reposition, get file attributes and set file attributes
- After creating a file, the file is opened. Data is read or written to a file.
- The file pointer may be/need to be repositted to a point
- These operations may also be supported for directories as well as ordinary files.

(c) Device management :- System calls include request device, release device, read, write, reposition, get/set device attributes & logically attach/detach devices
- When a process needs a resource, a request for the resource is done. Then the control is granted to the process. If already attached, then requesting process has to wait.
- In multiprogramming systems, after a process uses the device, it has to be returned back to the OS. so that another process can use the device
- Devices may be physical/virtual/abstract

(d) Information maintainence :- System calls includes calls to get/set the time, data, system data and process, file or device attributes
- These are used to transfer information between user & the OS. Information like current time & date, no. of current users etc are passed from the OS to the user

(e) Communication:- System calls create/delete communication connection, send/receive messages, transfer status information & attach/detach remote devices

- Message passing is simpler and easier, and is generally appropriate for smaller amounts of data. It is easy to implement, but there are system calls for each read & write processes
- Shared memory is faster and is generally the better approach where large amounts of data are to be shared.

(f) Protection:- Provides mechanisms for controlling which users/processes that have access to which system resources
- System calls allow the access mechanisms to be adjusted as needed, & for non privileged users to be granted elevated access permissions under carefully controlled temporary circumstances.

3a) What are interrupts? How is an interrupt handled by the OS.

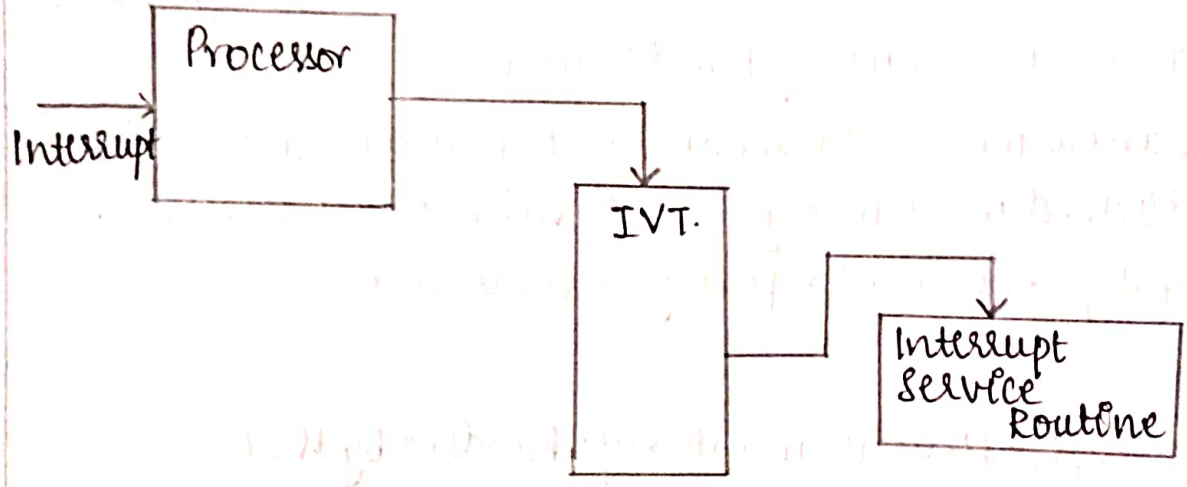
Ans An interrupt is an event that alters the sequence in which the processor executes the instruction. An SVC interrupts the program being executed and passes control to the supervisor so that it can perform the service

Interrupt Handling

- The occurance of an event is usually signalled by an interrupt. The interrupt can be either from the hardware/software. Hardware may trigger an interrupt any time by sending a signal to the CPU. Software triggers an interrupt by executing a special operation called system call

- When the CPU is Interrupted, it stops what it is doing and immediately transfers execution to a fixed location. The fixed location contains the starting address where the service routine for the Interrupt is located. After the execution of the Interrupt service routine, the CPU resumes the Interrupted computation.

- Interrupts are an important part of the computer architecture. Each computer design has its own Interrupt mechanism, but several function are common. The Interrupt must transfer control to the appropriate Interrupt service routine



Stored at a fixed location.

3b) What is Pre-emptive & Non-pre-emptive scheduling? Explain with examples

Ans Pre-emptive scheduling.

- This is driven by the idea of prioritized computation
- Processes that are runnable maybe temporarily suspended
- Disadvantages:- • Incurs a cost associated with access to shared data
  • Affects the design of the OS kernel

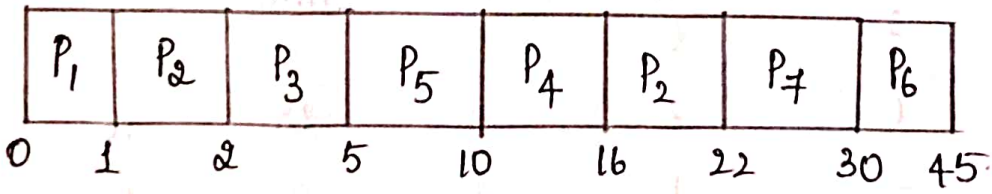
Non Pre-emptive scheduling

- Once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either
  - by terminating or
  - by switching to the waiting state

Example for pre emptive scheduling

Process ID	Priority	Arrival Time	Burst Time
P_1	2	0	1
P_2	6	1	7
P_3	3	2	3
P_4	5	3	6
P_5	4	4	5
P_6	10	5	15
P_7	9	15	8

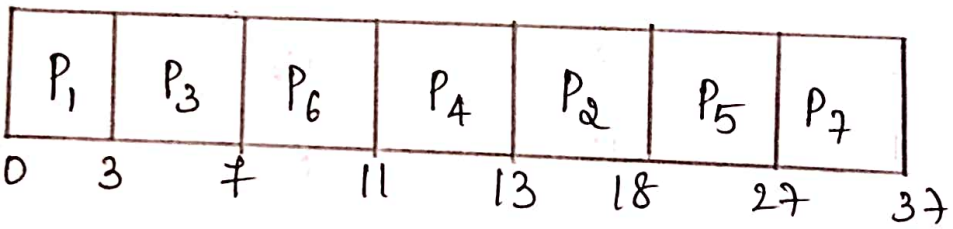
Gantt chart



Process ID	Priority	Arrival time	Burst time	Completion time	T-A-T	Waiting time
P_1	2	0	1	1	1	0
P_2	6	1	7	22	21	14
P_3	3	2	3	5	3	0
P_4	5	3	6	16	13	7
P_5	4	4	5	10	6	1
P_6	10	5	15	45	40	25
P_7	9	6	8	30	24	16

Avg. waiting time = $(0+14+0+7+1+25+16)/7 = 63/7$
$= 9$ units

Non preemptive for the same example with change in priority



Process ID	Priority	Arrival time	Burst time	Completion time	T-A-T	waiting time
P_1	2	0	3	3	3	0
P_2	6	2	5	18	16	11
P_3	3	1	4	7	6	2
P_4	5	4	2	13	9	7
P_5	7	6	9	27	21	12
P_6	4	5	4	11	6	2
P_7	10	7	10	37	30	18

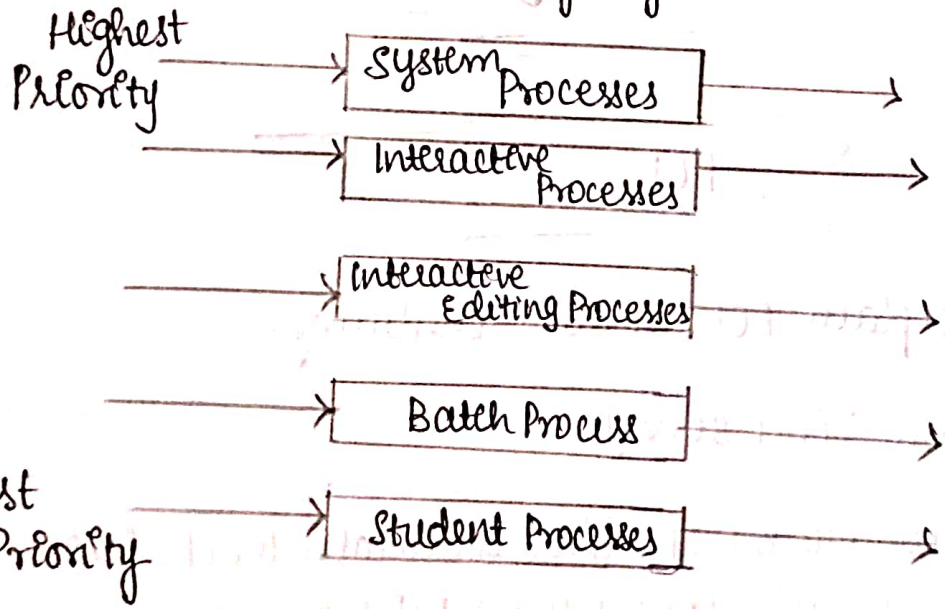
Avg waiting time = (0+11+2+7+12+2+18)/7 = 52/7

= 7.42 units

4a) With diagram, explain multilevel queue & multilevel feedback queue scheduling algorithms

Ans Multilevel Queue Scheduling

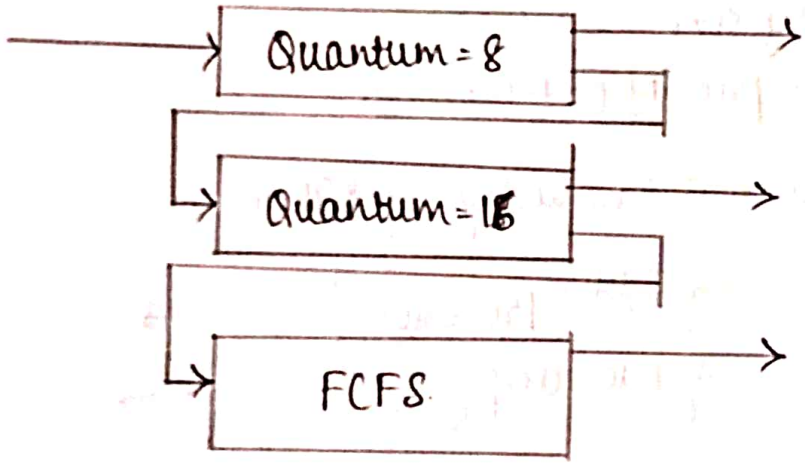
- Useful for situations in which processes are easily classified into different groups
- The ready queue is partitioned into several queues
- The processes are permanently assigned to one queued on some property
  - Memory size
  - Process priority (or) Process type.
- Each queue has its own scheduling algorithm.



- There must be scheduling among the queues which is commonly implemented as fixed priority preemptive scheduling

Multilevel Feedback Queue Scheduling

- A process may move between queues
- The basic idea: separate processes according to the features of their CPU bursts
- The following are the parameters
  - The number of queues
  - The scheduling algorithm for each queue
  - The method used to determine the process to higher priority
  - The method used to determine when to demote the process to lower priority
  - The method used to determine which queue a process will enter when that process needs service



4 b) With example explain FCFS & SJF scheduling.

Ans FCFS [First Come First Serve]

- The process that requests the CPU first is allocated the CPU first
- The implementation is easily done using a FIFO Queue
- Procedure
  - When a process enters the ready queue, its PCB is linked to one tail of the queue
  - When the CPU is free, the CPU is allocated the process at the queue's head. The running process is then removed from the queue

Advantage

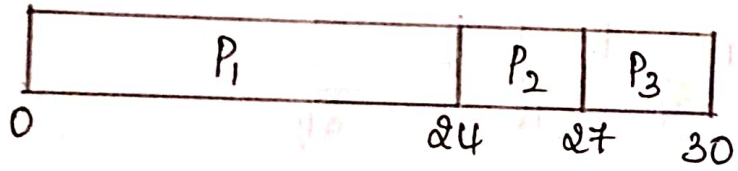
- Code is simple & easy to understand

Disadvantage

- Convoy effect
- Non preemptive
- Not good for time-sharing systems

Example:-

Process	Burst time
P_1	24
P_2	3
P_3	3



Waiting time:- $P_1 = 0$, $P_2 = 24$, $P_3 = 27$

Avg:- $(0+24+27)/3 = 17ms$

SJF Scheduling

SJF [Shortest Job First].

- The CPU is assigned to the process that the smallest(process) next CPU burst
- If 2 processes have the same length CPU burst, FCFS is used to break the tie

Advantage

- It is optimal i.e. It gives minimum average waiting time

Disadvantage

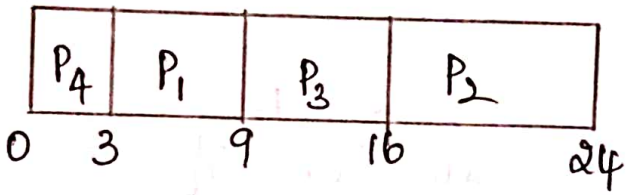
Determining the length of the next CPU burst

This maybe either @ non preemptive (or) (b) Preemptive

Non Preemptive :- The current process is allowed to finish its CPU burst

Example :-

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3



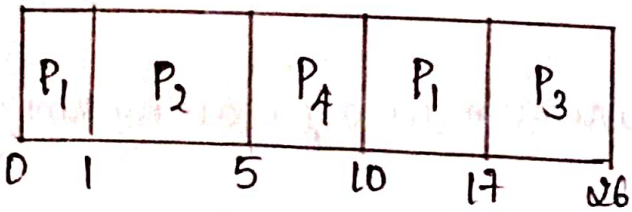
Waiting time :- $P_1 = 3$, $P_2 = 16$, $P_3 = 9$ $P_4 = 0$

Avg :- $(3+16+9+0)/4 = 7$.

Preemptive :- If the new process has a shorter next CPU burst, then the executing process is preempted

Example :-

Process	Burst time	Arrival time
P_1	8	0
P_2	4	1
P_3	9	2
P_4	5	3



Avg waiting time:- ((10-1)+(1-1)+(17-2)+(5-3))/4

= 26/4

= 6.5

5a) Briefly explain different criteria for process scheduling

Ans CPU utilisation:- We need to keep the CPU as busy as possible. CPU utilisation can range from 0 to 100%. In real system, it should range from 40% to 90%.

Throughput :- If the CPU is busy executing processes, then work is being done. One measure of work is number of processes that are completed per unit time called throughput

Turnaround time:- This is an important criterion which tells how long it takes to execute that process. The interval from time of submission to time of completion is the turnaround time. It is the sum of the periods spent waiting to get into the memory

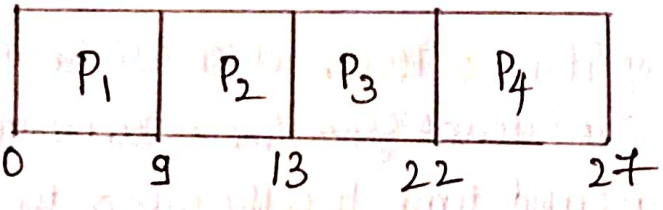
Waiting time:- Waiting time is the sum of the periods spent waiting in the ready queue. The CPU scheduling algorithm does not affect the amount of time that a process executes. It affects the process waiting in the ready queue

Response time :- The time from the submission of a request to the first response that is produced is called as the response time. It is time that the process takes to start responses.

5b) Calculate the average waiting time & average turnaround time by drawing Gantt chart using FCFS, and priority algorithms. Lower priority nor represents higher priority

Process ID	Arrival time	Burst time	Priority
P_1	0	9	3
P_2	1	4	2
P_3	2	9	1
P_4	3	5	4.

FCFS



Exec - Arri
$P_2 = 13 - 1 = 12$
$P_3 = 22 - 2 = 20$
$P_4 = 27 - 3 = 24$

Completion time:- $P_1 = 9$, $P_2 = 13$, $P_3 = 22$, $P_4 = 27$

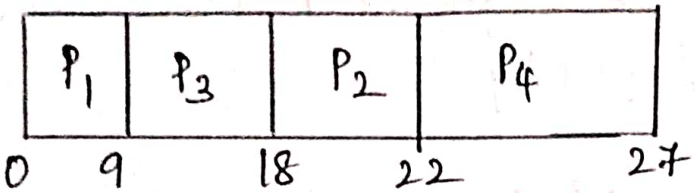
Turnaroundtime:- $P_1 = 9$, $P_2 = 12$, $P_3 = 20$, $P_4 = 24$

waiting (T.A.T - Burst) $P_1 = 0$, $P_2 = 8$ $P_3 = 11$ $P_4 = 19$.
(9-9) (12-4) (20-9) (24-5)

Avg waiting Time $\Rightarrow 16.25 \Rightarrow$ T.A.T avg $= \frac{9+12+20+24}{4} = 16.25$
Avg turnaround time

Avg waiting time $= \frac{0+8+11+19}{4} = \underline{\underline{9.5}}$

Priority



$P_1 = 9-0 = 9$
$P_2 = 22-1 = 21$
$P_3 = 18-2 = 16$
$P_4 = 27-3 = 16$

Completing time:- $P_1 = 9$, $P_2 = 22$, $P_3 = 18$, $P_4 = 27$

Turnaround time:- $P_1 = 9$ $P_2 = 21$, $P_3 = 16$ $P_4 = 24$

Waiting time :- $P_1 = 0$ (9-9), $P_2 = 17$ (21-4), $P_3 = 7$ (16-9), $P_4 = 19$ (24-5)

Avg. turnaround time = $70/4 = 17.5 = \left[\frac{9+21+16+24}{4}\right]$

Avg waiting time :- $\frac{0+17+7+19}{4} = \frac{43}{4} = 10.75$

2a) Write about 3 types of schedulers

Ans Schedulers are software which selects an available program to be assigned to the CPU.

- Long term scheduler / Job Scheduler : selects jobs from the job pool & loads them into the memory. If more processes are submitted, than that can be executed simultaneously / immediately. Such processes are in secondary memory

- Short term scheduler / CPU scheduler
  Selects job from memory and assigns the CPU to it. It must select the new process for CPU frequently

- Medium term scheduler :- Selects the process in the ready queue & reintroduced into the memory

Processes can be defined as either
- I/O bound process - Spends time on I/O than computation
- CPU bound process - Spends more time on computation than I/O.