Sub:	**SCHEME – IAT1** **Operating System Concepts**						
Date:	**13/03/2023**	**Duration:**	**90 min's**	**Max Marks:**	**50**	**Sem:**	**I**

<u>Note : Answer FIVE FULL Questions, choosing ONE full question from each Module</u>

PART I

1 What is the user point of view and system point of view services of the operating system?
User Services- 6marks
System services-4 marks

OR

2 Explain in detail System Calls. What are the various types?
Definition and diagram-4marks
Types- 4marks
Eg. 2 marks

PART II

3 What are the functions of an operating system?
Each function- 2 marks

OR

PART III

5 Explain various types of operating systems with advantages and disadvantages.
Each type-2 marks

OR

6 Explain the operations of the operating system with diagram.
Dual mode-4 marks
Diagram- 2 marks
Timer mode- 4 marks

PART IV

7 Explain process states with a neat diagram.
Each state-2 marks

OR

8 What is a PCB? Explain context switching in detail.
PCB-3 marks
Context switching – definition – 3marks, Diagram-4 marks

PART V

9 Explain Priority (non preemptive) and SJF (preemptive) scheduling with an example.
Each Explanation-3 marks, example- 2 marks

OR

10 Calculate the average waiting time, turn around time for i)SJF ii)Priority Scheduling and iii)Round Robin(tq=2ms) with the following set of process.

Process	**P1**	[illegible]	**P3**	**P4**	**P5**
Burst Time	10	[illegible]	2	1	5
Priority	3	[illegible]	3	4	5

SJF- 3 marks
Priority- 3 marks
RR-4 marks

SOLUTION

PART I

1 What is the user point of view and system point of view services of the operating system?

Operating System Services

- One set of operating-system services provides functions that are helpful to the user
- Communications – Processes may exchange information, on the same computer or between computers over a network.
- Communications may be via shared memory or through message passing (packets moved by the OS)
- Error detection – OS needs to be constantly aware of possible errors may occur in the CPU and memory hardware, in I/O devices, in user program
- For each type of error, OS should take the appropriate action to ensure correct and consistent computing.
- Debugging facilities can greatly enhance the user's and programmer's abilities to efficiently use the system.
- Another set of OS functions exists for ensuring the efficient operation of the system itself via resource sharing
- Resource allocation - When multiple users or multiple jobs running concurrently, resources must be allocated to each of them
- Many types of resources - Some (such as CPU cycles, main memory, and file storage) may have special allocation code, others (such as I/O devices) may have general request and release code
- Accounting - To keep track of which users use how much and what kinds of computer resources
- Protection and security - The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other.
- Protection involves ensuring that all access to system resources is controlled.
- Security of the system from outsiders requires user authentication, extends to defending external I/O devices from invalid access attempts.
- If a system is to be protected and secure, precautions must be instituted throughout it. A chain is only as strong as its weakest link.

OR

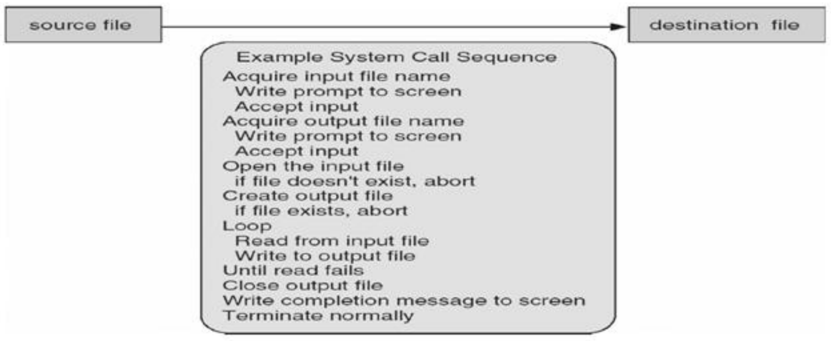
2 Explain in detail System Calls. What are the various types?

System Calls

- Programming interface to the services provided by the OS
- Typically written in a high-level language (C or C++)
- Mostly accessed by programs via a high-level Application Program Interface (API) rather than direct system call usenThree most common APIs are Win32 API for Windows, POSIX API for POSIX-based systems (including virtually all versions of UNIX, Linux, and Mac OS X), and JavaAPI for the Java virtual machine (JVM)

- Why use APIs rather than system calls?(Note that the system-call names used throughout this textare generic)

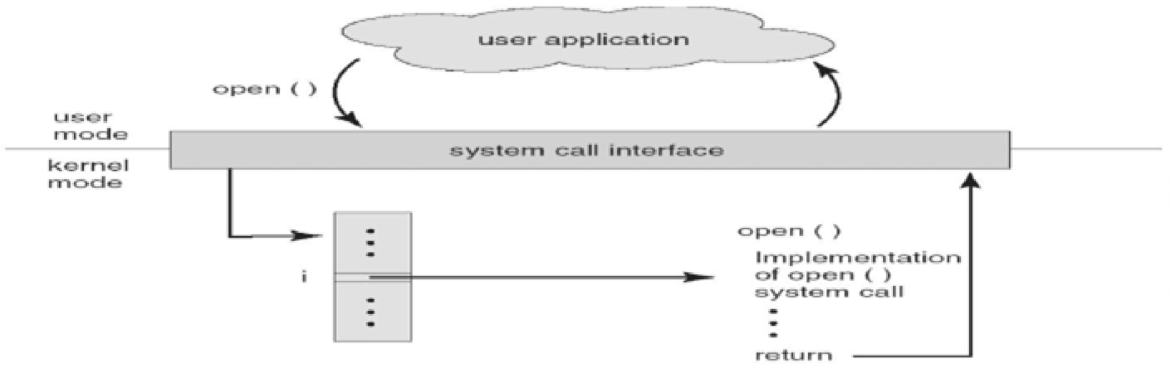
Example of System Calls



```
Example System Call Sequence
Acquire input file name
  Write prompt to screen
  Accept input
Acquire output file name
  Write prompt to screen
  Accept input
Open the input file
  if file doesn't exist, abort
Create output file
  if file exists, abort
Loop
  Read from input file
  Write to output file
Until read fails
Close output file
Write completion message to screen
Terminate normally
```

System Call Implementation

- Typically, a number associated with each system call
- System-call interface maintains a table indexed according to these numbers
- The system call interface invokes intended system call in OS kernel and returns status of thesystem call and any return values
- The caller need know nothing about how the system call is implemented
- Just needs to obey API and understand what OS will do as a result call
- Most details of OS interface hidden from programmer by API Managed by run-time supportlibrary (set of functions built into libraries included with compiler)



3 PART II

What are the functions of an operating system?

Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An Operating System does the following activities for memory management −

- Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
- In multiprogramming, the OS decides which process will get memory when and how much.
- 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.

Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**. An Operating System does the following activities for processor 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.

Device Management

An Operating System manages device communication via their respective drivers. It does the following activities for device management −

- Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

An Operating System does the following activities for file management −

- Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

I/O Management

- An I/O subsystem comprises of I/O devices and their corresponding driver software. Drivers hide the peculiarities of specific hardware devices from the users.
- An Operating System manages the communication between user and device drivers.
- I/O operation means read or write operation with any file or any specific I/O device.
- Operating system provides the access to the required I/O device when required.

OR

5 PART III
Explain various types of operating systems with advantages and disadvantages.

Batch operating system

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

The problems with Batch Systems are as follows −

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

Time-sharing operating systems

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

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

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

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

Advantages of Timesharing operating systems are as follows −

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

Disadvantages of Time-sharing operating systems are as follows −

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

Distributed operating System

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

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

The advantages of distributed systems are as follows −

- With resource sharing facility, a user at one site may be able to use the resources available at another.
- Speedup the exchange of data with one another via electronic mail.
- If one site fails in a distributed system, the remaining sites can potentially continue operating.
- Better service to the customers.
- Reduction of the load on the host computer.
- Reduction of delays in data processing.

Network operating System

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

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

The advantages of network operating systems are as follows −

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

The disadvantages of network operating systems are as follows −

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

Real Time operating System

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

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

OR

6 Explain the operations of the operating system with diagram.

Operating-System Operations

1. modern operating systems are **interrupt driven.** If there are no processes to execute, no I/O devices to service, and no users to whom to respond, an operating system will sit quietly, waiting for something to happen. Events are almost always signaled by the occurrence of an interrupt or a trap
2. A **trap (or** an **exception)** is a software-generated interrupt caused either by an error or by a specific request from a user program
that an operating-system service is performed.
3. The interrupt-driven nature of an operating system defines that system's general structure. For each type of interrupt, separate segments of code in the operating system determine what action should be taken. An interrupt service routine is provided that is responsible for dealing with the interrupt.
4. The operating system and the users share the hardware and software resources of the computer system, we need to make sure that an error in a user program could cause problems only for the one program that was running. With sharing, many processes could be adversely affected by a bug in one program. For example, if a process gets stuck in an infinite loop, this loop could prevent the correct operation of many other processes.
5. Without protection against these sorts of errors, either the computer must execute only one process at a time or all output must be suspect.

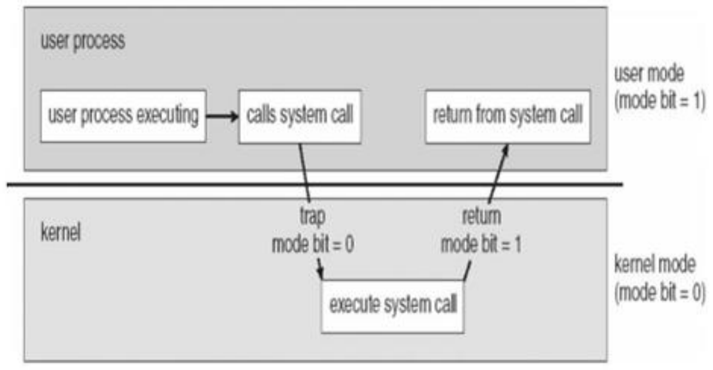
Dual-Mode Operation

Dual-mode operation allows OS to protect itself and other system components **User mode** and **kernel mode**

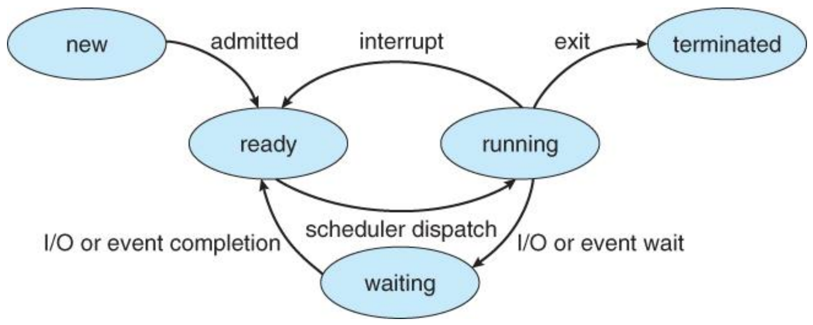
Mode bit provided by hardware Provides ability to distinguish when system is running user code or kernel code Some instructions designated as **privileged**, only executable in kernel mode System call changes mode to kernel, return from call resets it to user

Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources Set interrupt after specific period
- Operating system decrements counter
- When counter zero generate an interrupt
- Set up before scheduling process to regain control or terminate program that exceeds allotted time



7 PART IV
Explain process states with a neat diagram.



Process State

As a process executes, it changes **state.** The state of a process is defined in part by the current activity ofthat process. Each process may be in one of the following states:
• **New.** The process is being created.
• **Running.** Instructions are being executed.
• **Waiting.** The process is waiting for some event to occur (such as anI/O completion or reception of a signal).
• **Ready.** The process is waiting to be assigned to a processor.
• **Terminated.** The process has finished execution.

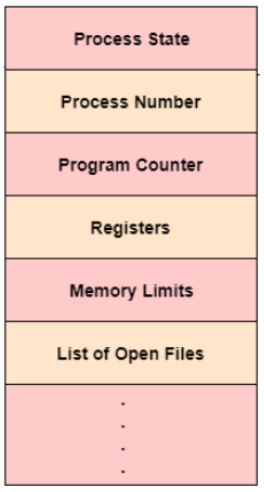
These names are arbitrary, and they vary across operating systems. Certain operating systems also more finely delineate process states. It is importantto realize that only one process can be *running* on any processor at any instant.

OR
8 What is a PCB? Explain context switching in detail.

Process Control Block

Each process is represented in the operating system by a **process control block (PCB)**—also called a *taskcontrol block.*

Process state. The state may be new, ready, running, and waiting, halted, and so on.



Process Control Block (PCB)

Program counter-The counter indicates the address of the next instruction to be executed for thisprocess.
• CPU **registers-** The registers vary in number and type, depending on the computer architecture. They include accumulators, index registers, stack pointers, and general-purpose registers, plus any condition- code information.

CPU-scheduling information- This information includes a process priority, pointers to scheduling queues,and any other scheduling parameters.

Memory-management information- This information may include such information as the value of the base and limit registers, the page tables, or the segment tables, depending on the memory system used by the operating system

Accounting information-This information includes the amount of CPU and real time used, time limits, account members, job or process numbers, and so on.

I/O status information-This information includes the list of I/O devices allocated to the process, a list of open files, and so on.

9 PART V
Explain Priority (non preemptive) and SJF (preemptive) scheduling with an example.

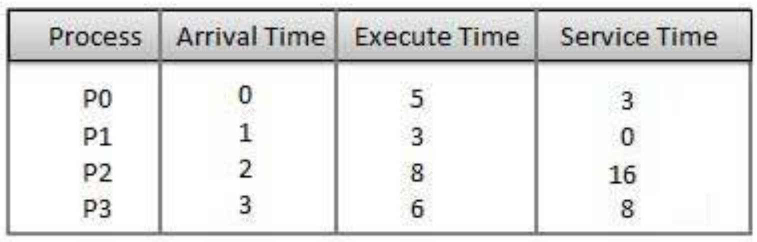
Shortest Job First (SJF)

- This is also known as **shortest job first**, or SJF
- This is a non-preemptive, pre-emptive scheduling algorithm.

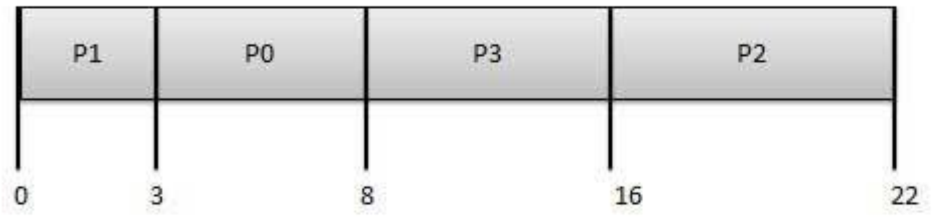
- Best approach to minimize waiting time.
- Easy to implement in Batch systems where required CPU time is known in advance.
- Impossible to implement in interactive systems where required CPU time is not known.
- The processer should know in advance how much time process will take.

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

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



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



Waiting time of each process is as follows −

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

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

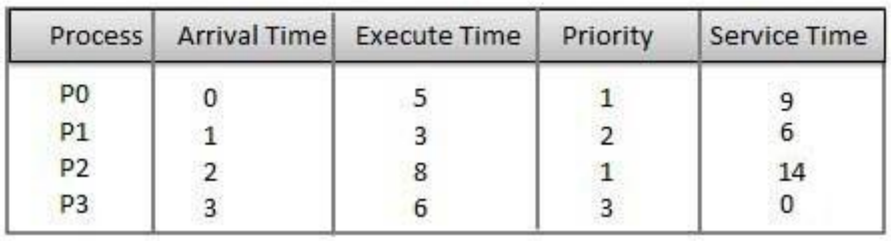
Priority Based Scheduling

- Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems.

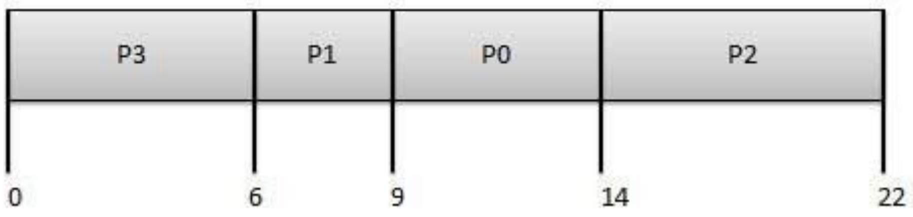
- Each process is assigned a priority. Process with highest priority is to be executed first and so on.
- Processes with same priority are executed on first come first served basis.
- Priority can be decided based on memory requirements, time requirements or any other resource requirement.

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

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



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



Waiting time of each process is as follows −

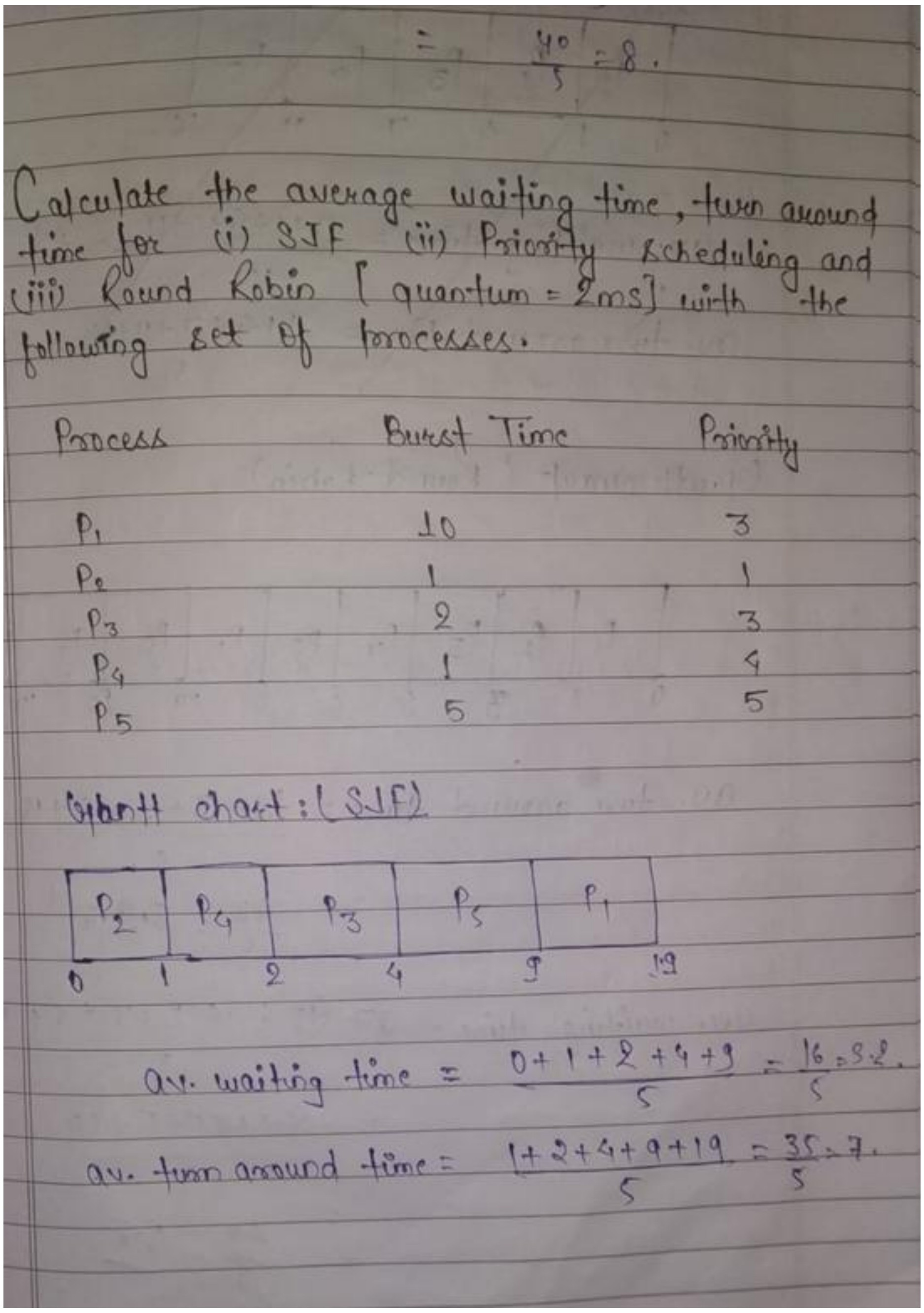
Process	Waiting Time
P0	0 - 0 = 0
P1	11 - 1 = 10
P2	14 - 2 = 12
P3	5 - 3 = 2

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

OR

10 Calculate the average waiting time, turn around time for i)SJF ii)Priority Scheduling and iii)Round Robin(tq=2ms) with the following set of process.

Process	P1	P2	P3	P4	P5
Burst Time	10	1	2	1	5
Priority	3	1	3	4	5



$= \frac{40}{5} = 8.$

Calculate the average waiting time, turn around time for (i) SJF (ii) Priority Scheduling and (iii) Round Robin [quantum = 2ms] with the following set of processes.

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	5

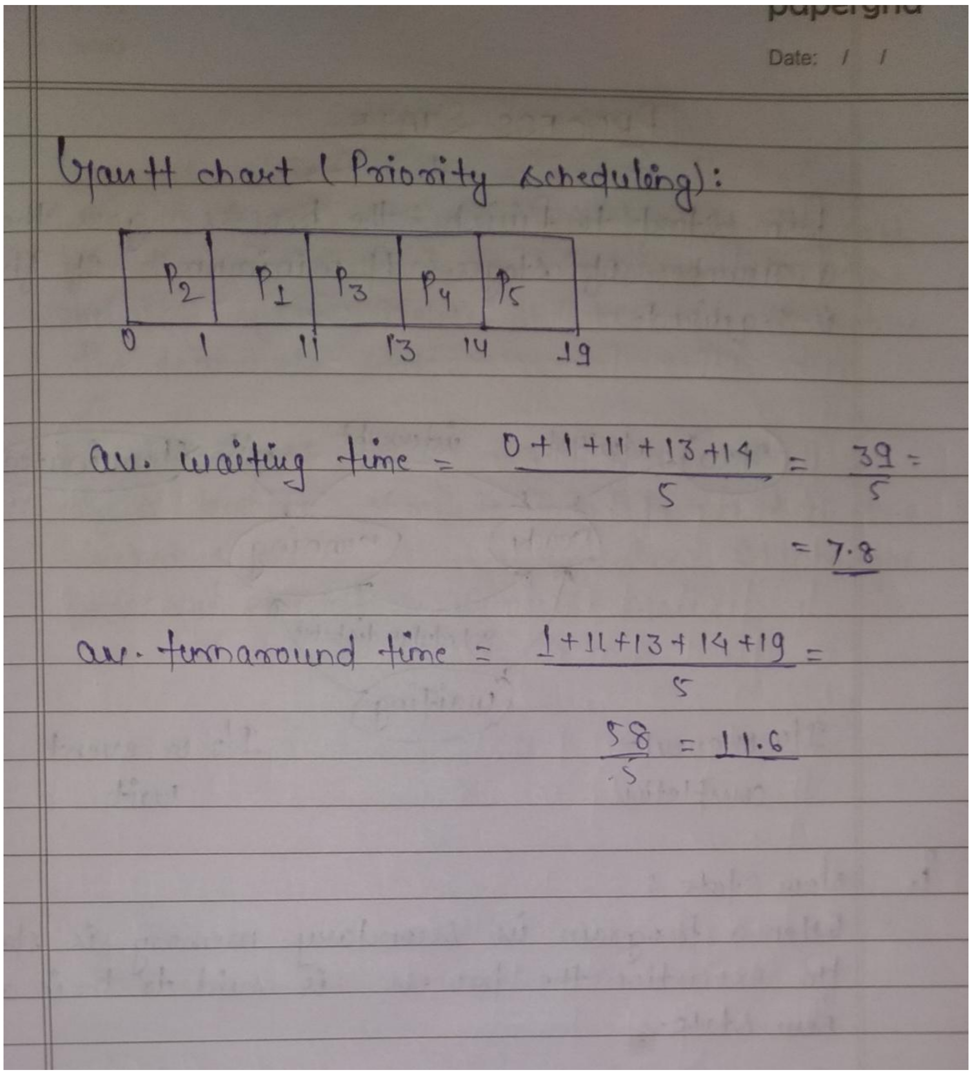
Gantt chart : (SJF)

P_2	P_4	P_3	P_5	P_1

0 1 2 4 9 19

av. waiting time $= \frac{0+1+2+4+9}{5} = \frac{16}{5} = 3.2.$

av. turn around time $= \frac{1+2+4+9+19}{5} = \frac{35}{5} = 7.$



Date: / /

Gantt chart (Priority scheduling):

P_2	P_1	P_3	P_4	P_5

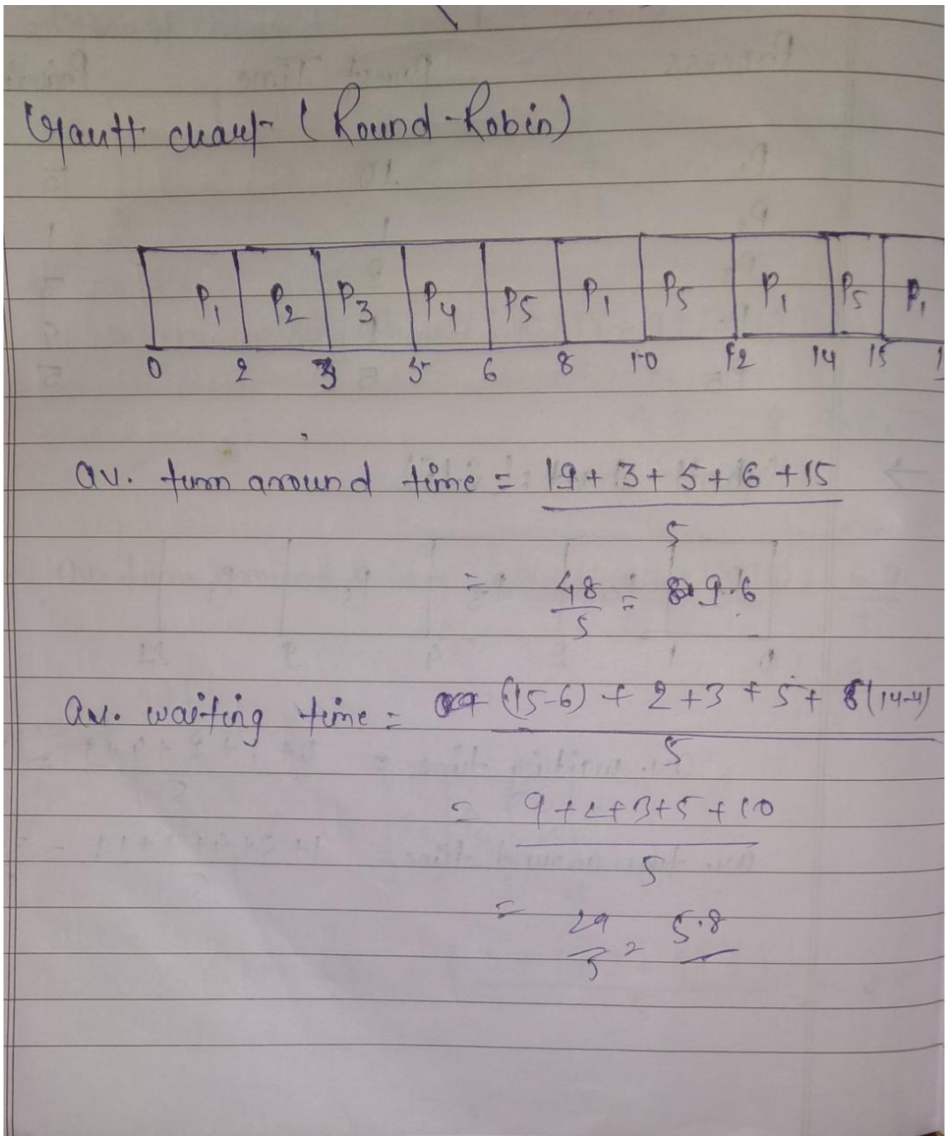
0 1 11 13 14 19

av. waiting time $= \frac{0+1+11+13+14}{5} = \frac{39}{5} =$

$= \underline{7.8}$

av. turnaround time $= \frac{1+11+13+14+19}{5} =$

$\frac{58}{5} = \underline{11.6}$



Gantt chart (Round-Robin)

P1	P2	P3	P4	P5	P1	P5	P1	P5	P1

0 2 3 5 6 8 10 12 14 15 1

Av. turn around time = $\frac{19+3+5+6+15}{5}$

$= \frac{48}{5} = 9.6$

Av. waiting time = $\frac{(15-6)+2+3+5+8(14-4)}{5}$

$= \frac{9+2+3+5+10}{5}$

$= \frac{29}{5} = 5.8$