National Textile University

Department of Computer Science

Subject:

Operating System

Submitted to:

Nasir Mahmood

Submitted by:

Afia Maham

Reg number:

23-NTU-CS-1126

Assignment no:

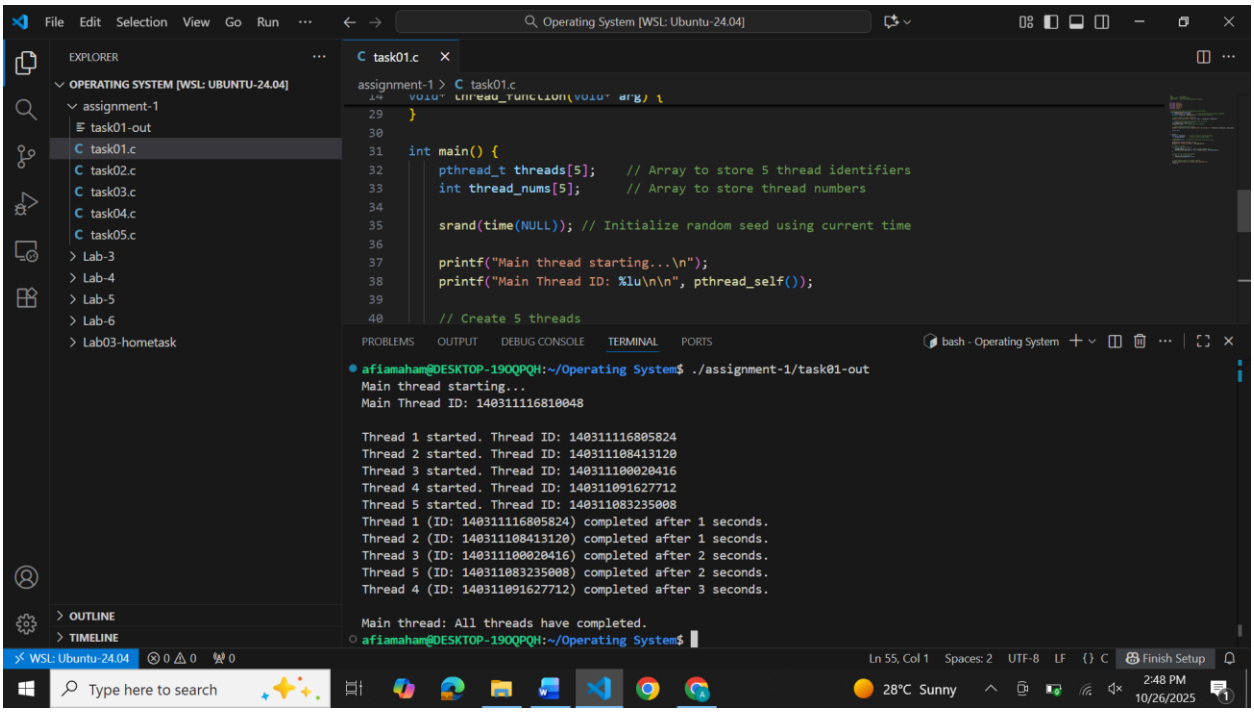
01

Semester:

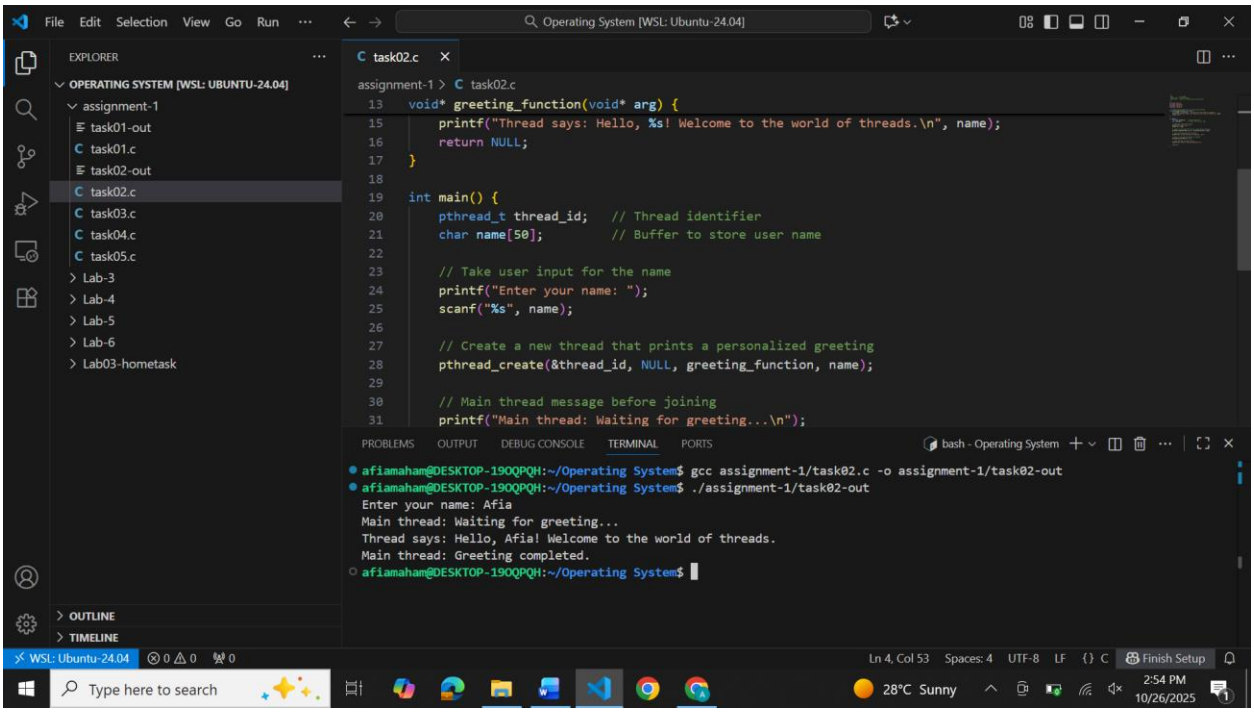
5th

Section-A: Programming Tasks

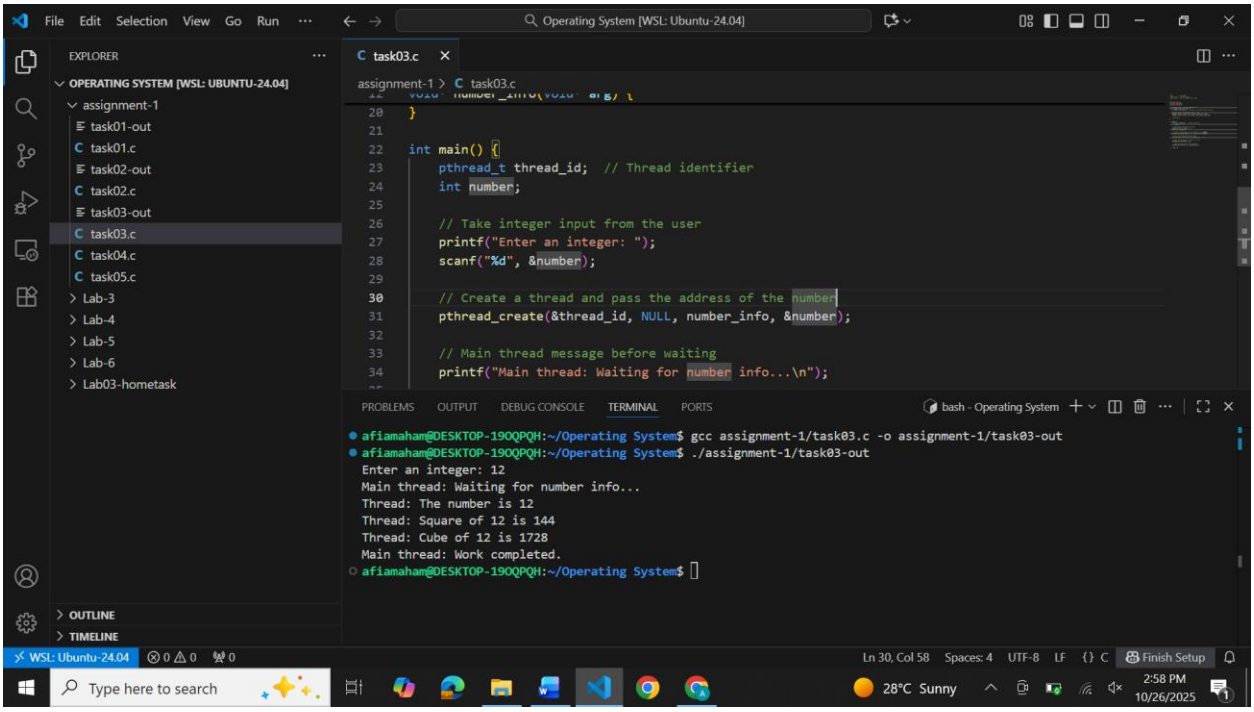
Task 1 - Thread Information Display



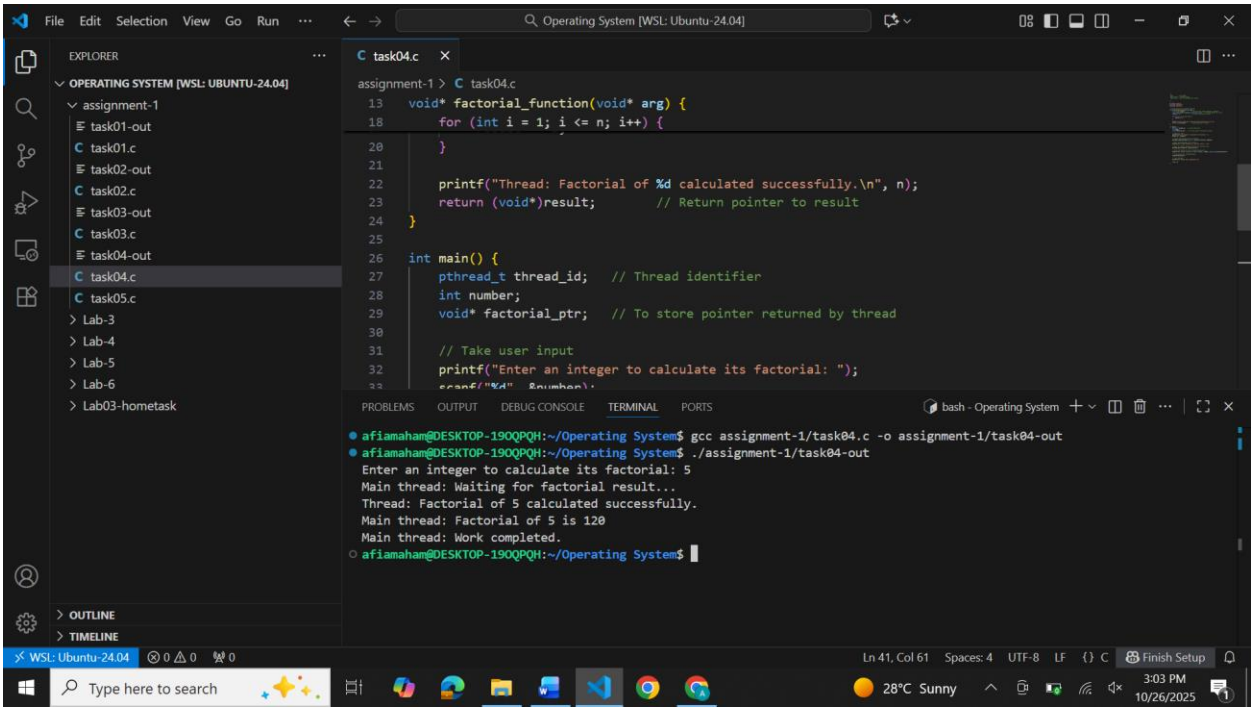
Task 2 - Personalized Greeting Thread



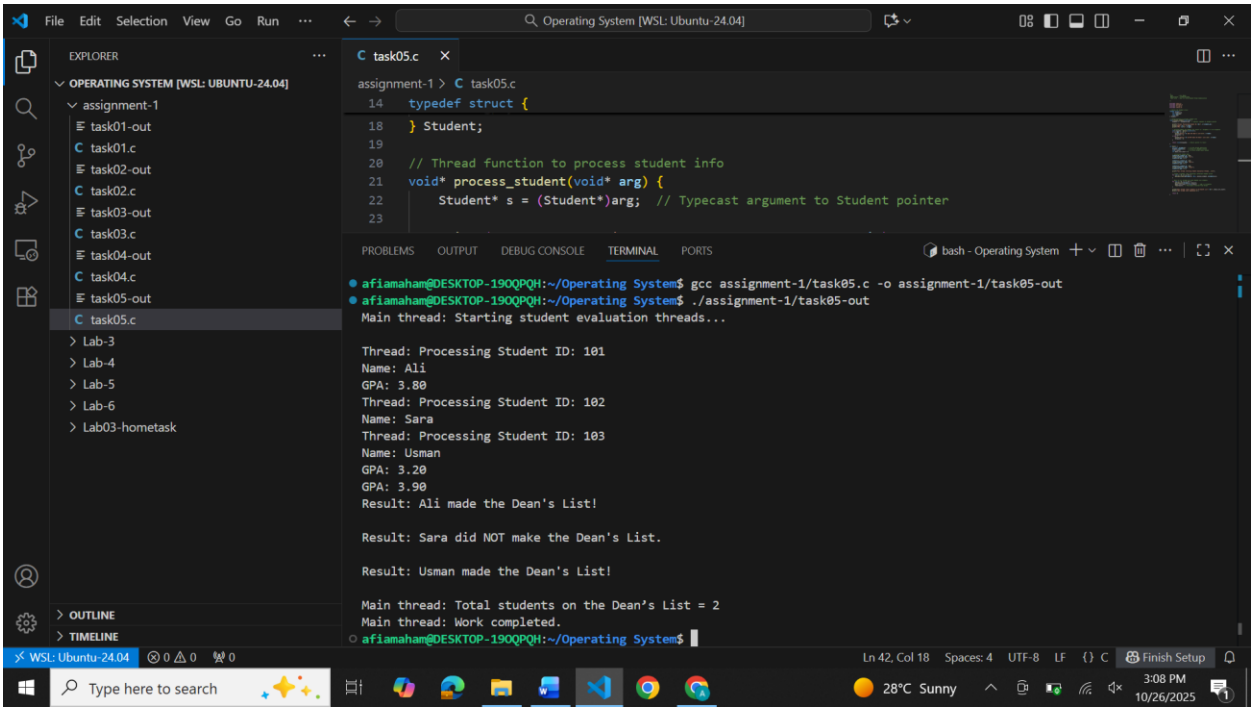
Task 3 - Number Info Thread



Task 4 - Thread Return Values



Task 5 - Struct-Based Thread Communication



Section-B: Short Questions

1. Define an Operating System in a single line.

An operating system (OS) is a program that acts as a bridge between user and computer hardware and manages all the other application programs on a computer.

2. What is the primary function of the CPU scheduler?

CPU scheduling is a process used by the operating system to decide which task or process will utilize the CPU next, help in performance optimization.

3. List any three states of a process.

Three main process states are:

- **Ready**
- **Running**
- **Waiting (Blocked)**

4. What is meant by a Process Control Block (PCB)?

A Process Control Block (PCB) is a data structure used by the operating system to store all the necessary information about a process such as process ID, state, registers, memory limits, and scheduling information.

5. Differentiate between a process and a program.

Program	Process
A **program** is a passive entity. It is a set of instructions that are stored on disk.	A **process** is an active entity. It is a program in execution.
It requires no CPU or memory while idle.	It consumes CPU time and memory while running.

6. What do you understand by context switching?

Context switching is the process where the CPU saves the current state of a process that has stopped running and loads the saved state of another process so that multiple processes can share the CPU efficiently.

7. Define CPU utilization and throughput.

- **CPU Utilization:** The amount of time the CPU is actively executing processes without being idle.
- **Throughput:** The number of processes completed by the system in specific time period.

8. What is the turnaround time of a process?

Turnaround time is the total time taken from process creation to its completion, i.e.,

Turnaround time = Completion time - Arrival time

9. How is waiting time calculated in process scheduling?

Waiting Time is the total time a process spends waiting in the ready queue, i.e.,

Waiting time = Turnaround time - burst time

10. Define response time in CPU scheduling.

It is the time when a process is created until it first gets the CPU for execution.

11. What is preemptive scheduling?

In preemptive scheduling, based on the priority the CPU can be taken away from a running process and given to another process.

12. What is non-preemptive scheduling?

In non-preemptive scheduling, once a process starts running, it cannot be blocked/stopped until it finishes or releases the CPU.

13. State any two advantages of the Round Robin scheduling algorithm.

Every process gets equal CPU time resulting in maintaining fairness.
Provides good response time for interactive systems.

14. Mention one major drawback of the Shortest Job First (SJF) algorithm.

It may cause **starvation** for longer processes if short processes keep arriving.

15. Define CPU idle time.

CPU idle time is the total time the CPU remains **unused or waiting** when no process is ready for execution.

16. State two common goals of CPU scheduling algorithms.

Maximize CPU utilization.
Minimize waiting and turnaround time.

17. List two possible reasons for process termination.

Process completes its execution.
Process is terminated by the system or user.

18. Explain the purpose of the wait() and exit() system calls.

exit() - used by a process to terminate itself.
wait() - used by a parent process to wait for its child process to finish execution.

19. Differentiate between shared memory and message-passing models of inter-process communication.

Shared Memory	**Message Passing**
Processes share a common memory region to exchange data.	Processes communicate by sending and receiving messages.
Faster but require synchronization.	Slower but safer and easier to implement.

20. Differentiate between a thread and a process.

Thread	Process
A smaller unit of execution within a process.	An independent program in execution.
Threads share memory and resources of their process.	Processes have separate memory and resources.

21. Define multithreading.

Execution of multiple threads concurrently within a single process by CPU or program is called multithreading.

22. Explain the difference between a CPU-bound process and an I/O-bound process.

A **CPU-bound** process spends most of its time performing computations.
An **I/O-bound** process spends most of its time waiting for input/output operations.

23. What are the main responsibilities of the dispatcher?

The dispatcher is responsible for:

- Giving CPU control to the selected process.
- Performing context switching.
- Switching to user mode.
- Jumping to the correct program location to resume execution.

24. Define starvation and aging in process scheduling.

Starvation: A process waits for long time because other processes keep getting CPU time.
Aging: Gradual increase in a process's priority to prevent starvation.

25. What is a time quantum (or time slice)?

It is the fixed amount of time each process is allowed to run after which the CPU is given to another process in Round Robin scheduling.

26. What happens when the time quantum is too large or too small?

Too large: System behaves like First-Come, First-Served.
Too small: Too many context switches, increasing overhead.

27. Define the turnaround ratio (TR/TS).

Turnaround Ratio = **Turnaround Time / Service Time**,
used to measure how efficiently a process is completed relative to its execution time.

28. What is the purpose of a ready queue?

The ready queue holds all processes that are in main memory and ready to execute but are waiting for the allocation of CPU.

29. Differentiate between a CPU burst and an I/O burst.

CPU Burst	**I/O Burst**
Time during which a process is executed on the CPU.	Time during which a process waits for I/O operations.
Involves computation.	Involves data transfer or waiting.

30. Which scheduling algorithm is starvation-free, and why?

Round Robin (RR) is starvation-free because every process gets an equal amount of CPU time in a cyclic order, which ensures no process waits for longer period of time.

31. Outline the main steps involved in process creation in UNIX.

1. **fork()** - Creates a child process (duplicate of parent).
2. **exec()** - Replaces the child's memory with a new program.
3. **wait()** - Parent waits for the child to finish execution.
4. **exit()** - Child terminates and returns status to the parent.

32. Define zombie and orphan processes.

- **Zombie process:** A process that has finished execution but still has an entry in the process table (waiting for parent to read its exit status).
- **Orphan process:** A child process whose parent has terminated before it and it is then adopted by the init (or systemd) process.

33. Differentiate between Priority Scheduling and Shortest Job First (SJF).

Priority Scheduling	Shortest Job First (SJF)
Processes are selected based on assigned **priority values**.	Processes are selected based on **shortest CPU burst time**.
Can cause starvation of low-priority processes.	Can cause starvation of longer jobs.

34. Define context switch time and explain why it is considered overhead.

Context switch time is the time required by the CPU in which it save the state of one process and load the state of another.
It is **overhead** because no useful work is done during this period hence zero productivity, the CPU only manages switching.

35. List and briefly describe the three levels of schedulers in an Operating System.

1. **Long-Term Scheduler (Job Scheduler):** Decides which processes are admitted to the ready queue.
2. **Short-Term Scheduler (CPU Scheduler):** Selects which ready process will run next.
3. **Medium-Term Scheduler:** Temporarily removes or swaps out processes from memory to improve multitasking.

36. Differentiate between User Mode and Kernel Mode in an Operating System.

User Mode	Kernel Mode
Runs user applications with limited privileges.	Runs OS code with full hardware access.
Cannot directly access hardware or I/O devices.	Can execute all system instructions and manage hardware.

Section-C: Technical / Analytical Questions

Q1. Describe the complete life cycle of a process with a neat diagram showing transitions between New, Ready, Running, Waiting, and Terminated states.

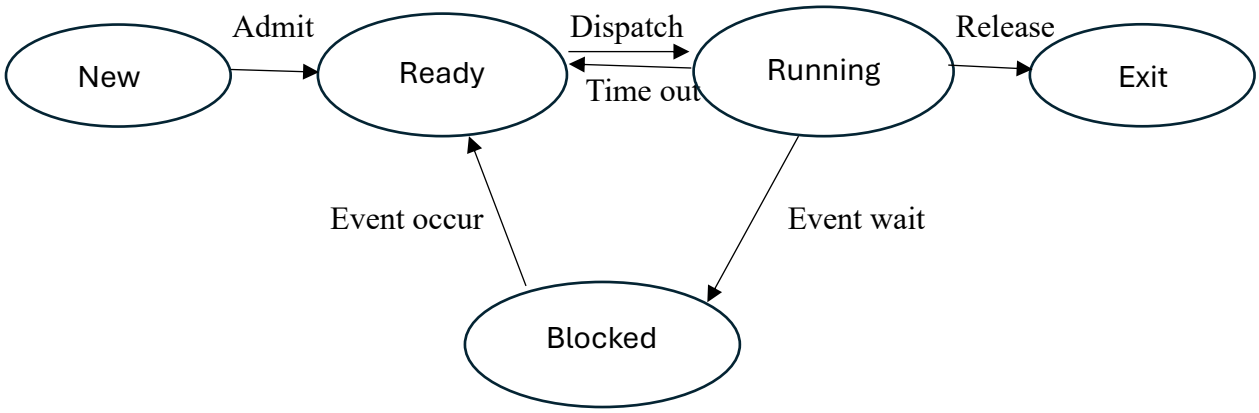
Process Life Cycle Explanation:

During the execution, a **process** passes through several states. These transitions are managed by the Operating System (OS) to ensure efficient CPU utilization and process management.

The **five main states** are:

1. **New:**
    - The process is being created.
    - OS allocates resources like memory and PCB (Process Control Block).
2. **Ready:**
    - The process is loaded in main memory and waiting for allocation of CPU.
    - Multiple processes can be in this state inside the **ready queue**.
3. **Running:**
    - The process is currently being executed by the CPU.
    - Only one process (per CPU core) can be in the running state at a time.
4. **Waiting (Blocked):**
    - The process cannot continue until some event occurs (like I/O completion).
    - It releases the CPU and waits for the required resource.
5. **Terminated (Exit):**
    - The process has finished execution or was killed by the OS.
    - All resources are released, and the PCB is deleted.

State Transitions Diagram:



Explanation of Transitions:

- **New to Ready:**
    - Triggered by: Process creation completed
    - Description: The operating system admits the newly created process to the ready queue.
- **Ready to Running:**
    - Triggered by: CPU scheduler dispatch
    - Description: The process is selected by the CPU scheduler and assigned to the CPU for execution.
- **Running to Waiting:**
    - Triggered by: I/O or event request
    - Description: The process requires some I/O operation or waits for an event, so it moves to the waiting (blocked) state.
- **Waiting to Ready:**
    - Triggered by: I/O completion
    - Description: The I/O operation or event is completed, and the process becomes ready to resume execution.
- **Running to Ready:**
    - Triggered by: Preemption (time quantum expires)
    - Description: The CPU is taken away from the running process to give other processes a chance to execute.
- **Running to Terminated:**
    - Triggered by: Normal exit or error
    - Description: The process completes its execution or encounters an error, and all allocated resources are released.

Q2. Write a short note on context switch overhead and describe what information must be saved and restored.

Context Switch:

A **context switch** occurs when the CPU switches from one process that is currently being executed to another.
During this switch, the operating system **saves the state** of the currently executing process and **load the state** of the next process to be executed.

Context Switch Overhead:

- **Definition:**
  Context switch overhead is the **time and CPU resources wasted** during the process of saving and restoring process states, no work is done by CPU resulting in zero productivity.
- **Reason for Overhead:**
  The CPU is busy managing the transition between processes instead of executing user instructions.
  Frequent context switching reduces system efficiency and increases response time.

Information Saved and Restored During Context Switch:

When switching from one process to another, the operating system saves the following information in the **Process Control Block (PCB)** of the current process and restores it for the next process:

- **Program Counter (PC):** Indicates the next instruction to execute.
- **CPU Registers:** Store temporary data and execution states.
- **Stack Pointer:** Points to the top of the process's stack.
- **Memory Management Information:** Includes base and limit registers, page tables, or segment tables.
- **Process State:** Indicates whether the process is ready, waiting, or running.
- **Accounting Information:** Such as CPU usage, process priority, and scheduling details.

Q3. List and explain the components of a Process Control Block (PCB)

Definition:

A **Process Control Block (PCB)** is a data structure maintained by the **Operating System (OS)** for every active process.
It **stores all essential information** about a process so that it can be resumed correctly after being suspended by the CPU.

Main Components of a PCB:

1. **Process ID (PID):**
   - A unique identifier assigned to each process.
   - Used by the OS to distinguish between multiple processes.
2. **Process State:**

- Indicates the current status of the process (New, Ready, Running, Waiting, or Terminated).

3. **Program Counter (PC):**
    - Holds the address of the **next instruction** the process will be executed.
    - Ensures that when the process resumes, it continues from the correct location.
4. **CPU Registers:**
    - Store temporary data, counters, and intermediate results.
    - Saved during a context switch and restored when the process resumes.
5. **Memory Management Information:**
    - Includes base and limit registers, page tables, or segment tables.
    - Helps the OS manage and protect the process's memory space.
6. **Accounting Information:**
    - Contains details like CPU usage, process priority, time limits, and job or user IDs.
    - Used for scheduling and performance monitoring.
7. **I/O Status Information:**
    - Lists I/O devices allocated to the process, open file descriptors, and pending I/O requests.
    - Ensures correct management of input/output operations.

Q4. Differentiate between Long-Term, Medium-Term, and Short-Term Schedulers with Examples

Schedulers are special system programs in the Operating System that determine which processes should be executed next after termination of the one process.
They manage process transitions between different states (new, ready, waiting, running).

Types of Schedulers and Their Differences:

Feature	**Long-Term Scheduler (Job Scheduler)**	**Medium-Term Scheduler (Swapper)**	**Short-Term Scheduler (CPU Scheduler)**
Purpose	Controls how many processes are admitted into the system.	Control transitioning of processes between main memory and secondary storage.	Decides which ready process gets the CPU next.

Function	Selects processes from the job pool on disk and loads them into main memory.	Suspends and resumes processes to control the degree of multiprogramming.	Select one of the ready processes to execute.
Frequency of Execution	Infrequent (runs occasionally).	Moderate frequency.	Very frequent (milliseconds).
Speed Requirement	Slow	Medium	Very fast
Process State Change	Moves process from **New to Ready**.	Moves process from **Ready to Suspend and vice versa**.	Moves process from **Ready to Running**.
Example	In batch systems, selects which jobs to load for execution.	Temporarily swaps out a background process during high CPU load.	In Round Robin scheduling, selects the next process in the ready queue.

Q5. Explain CPU Scheduling Criteria (Utilization, Throughput, Turnaround, Waiting, and Response) and Their Optimization Goals

CPU scheduling criteria are performance **measures** used to evaluate and compare the efficiency of different CPU scheduling algorithms.
These criteria help determine how well the system manages CPU time and process execution.

1. CPU Utilization

- **Meaning:**
  The percentage of time the CPU remains busy executing processes.
- **Formula:**

$$\text{CPU utilization} = (\text{Busy time}/\text{Total time}) * 100$$

- **Goal:**
  Maximize CPU utilization (keep CPU as busy as possible).

2. Throughput

- **Meaning:**
  The number of processes completed per unit of time.
- **Example:** If 10 processes finish in 5 seconds, throughput = 2 processes/second.
- **Goal:**
  Maximize throughput (finish more processes in less time).

3. Turnaround Time

- **Meaning:**
  Total time taken from process submission to its completion.

- **Formula:**

  Turnaround Time = completion time - arrival time

- **Goal:**
  Minimize turnaround time (complete processes quickly).

4. Waiting Time

- **Meaning:**
  The total time a process spends waiting in the ready queue before getting CPU.
- **Formula:**

  Waiting time = turnaround time - burst time

- **Goal:**
  Minimize waiting time (reduce idle waiting in ready queue).

5. Response Time

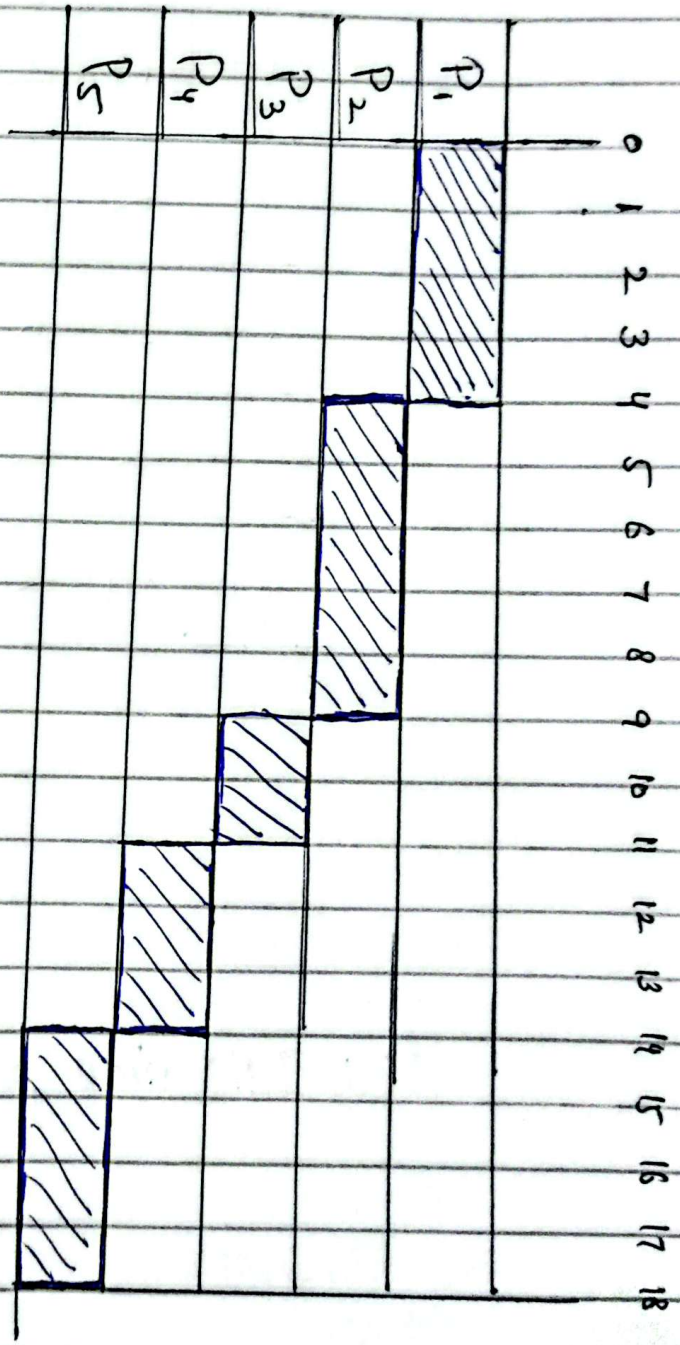
- **Meaning:**
  The time from process submission until the first response (first CPU allocation).
- **Goal:**
  Minimize response time (improve system responsiveness, especially in interactive systems).

Section - D

CPU Scheduling Calculations

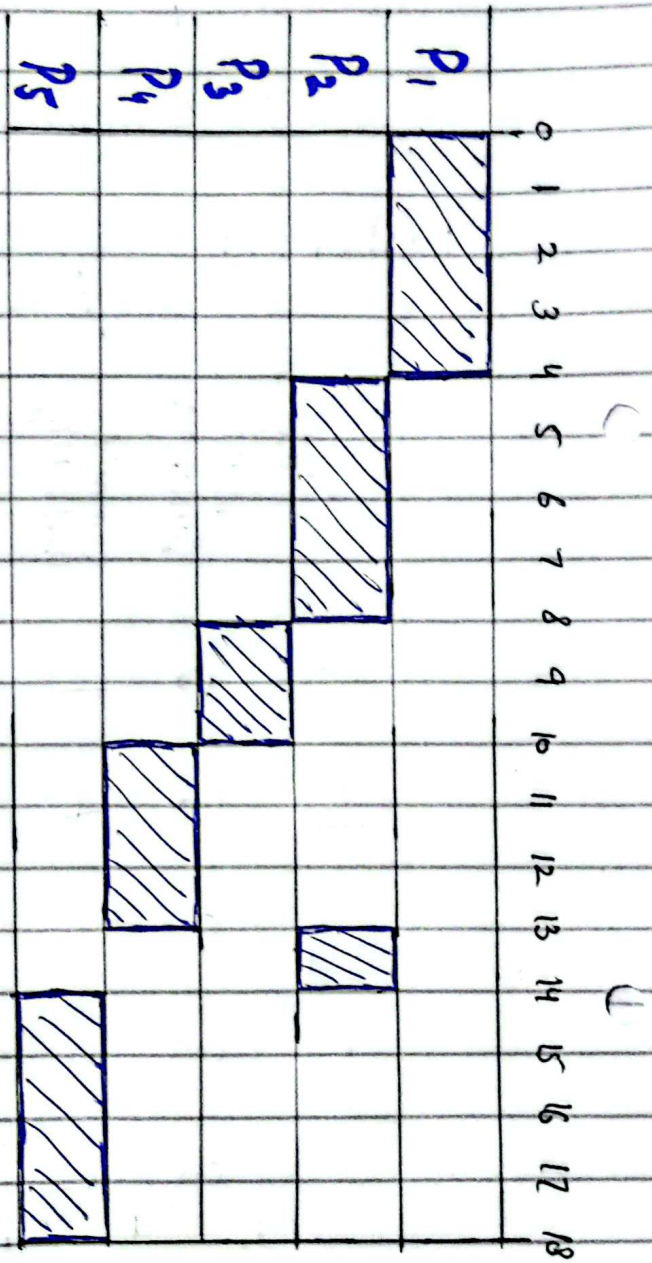
PART - A (FCFS)

	Turnaround time	Waiting time	TR/TS
P_1	4	0	1
P_2	7	2	1.4
P_3	7	5	3.5
P_4	8	5	2.66
P_5	9	5	2.25



CPU idle time = 0

Round Robin (Q=4)



	Turnaround Time	Waiting Time	TR/TS
P_1	4	0	1
P_2	12	7	2.4
P_3	6	4	3
P_4	7	4	2.33
P_5	9	5	2.25

CPU Idle Time = 0

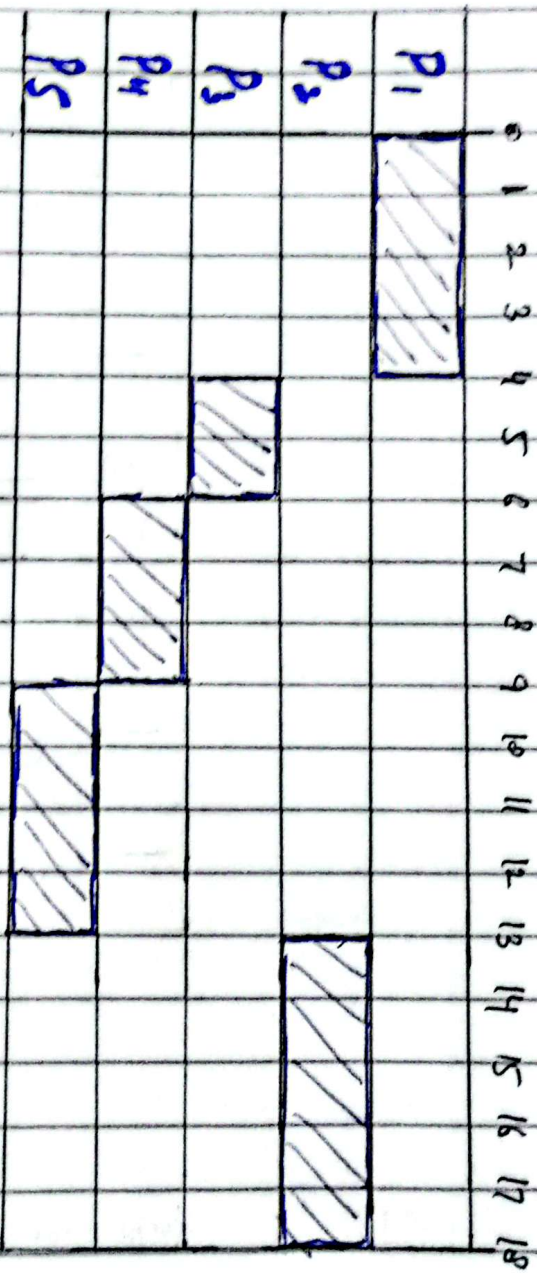
SJF

	Turnaround time	Waiting time	TR/TS
P_1	4	0	1
P_2	16	11	3.2
P_3	2	0	1
P_4	3	0	1
P_5	4	0	1

CPU idle time = 0

SRTF

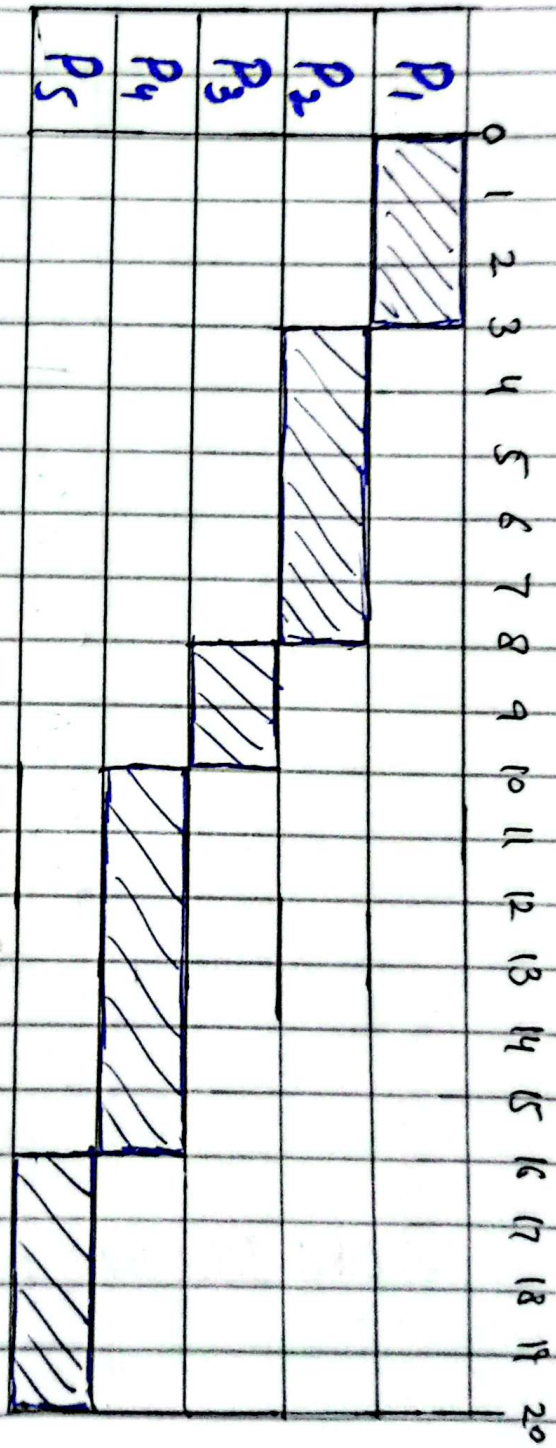
	Turnaround time	Waiting time	TA/TS
P_1	4	0	1
P_2	16	11	3.2
P_3	2	0	1
P_4	3	0	1
P_5	4	0	1



CPU Idle time = 0

PART-B. FCFS

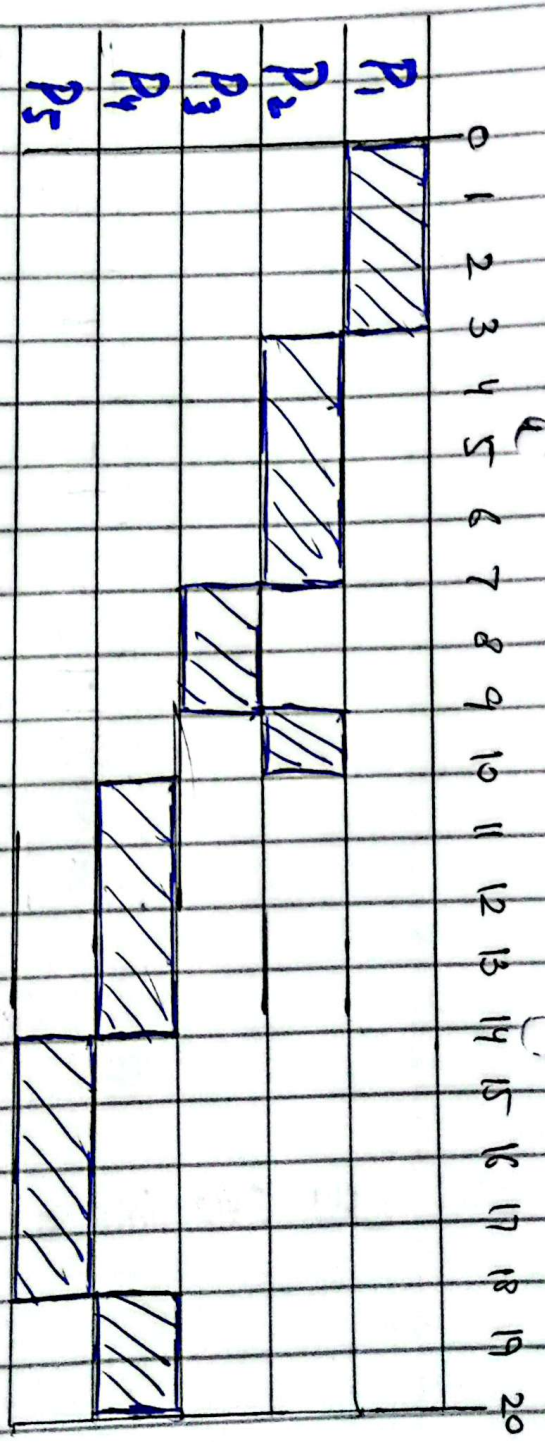
	Turnaround time	Waiting time	TR/TS
P_1	3	0	1
P_2	7	2	1.4
P_3	7	5	3.5
P_4	7	1	1.166
P_5	10	8	2.5



CPU Idle time = 0

Round Robin (Q=4)

	Turnaround time	Waiting time	TR/TS
P_1	3	0	1
P_2	9	4	1.8
P_3	6	4	3
P_4	11	5	1.833
P_5	8	4	2



CPU Idle time = 0

SJF

	Turnaround time	Waiting time	TT/TS
P_1	3	0	1
P_2	9	4	1.8
P_3	2	0	1
P_4	11	5	1.833
P_5	4	0	1

CPU Idle time = 0

SRTF

	Turnaround time	Waiting time	TE/TS
P_1	3	0	1
P_2	9	4	1.8
P_3	2	0	1
P_4	11	5	1.833
P_5	4	0	1

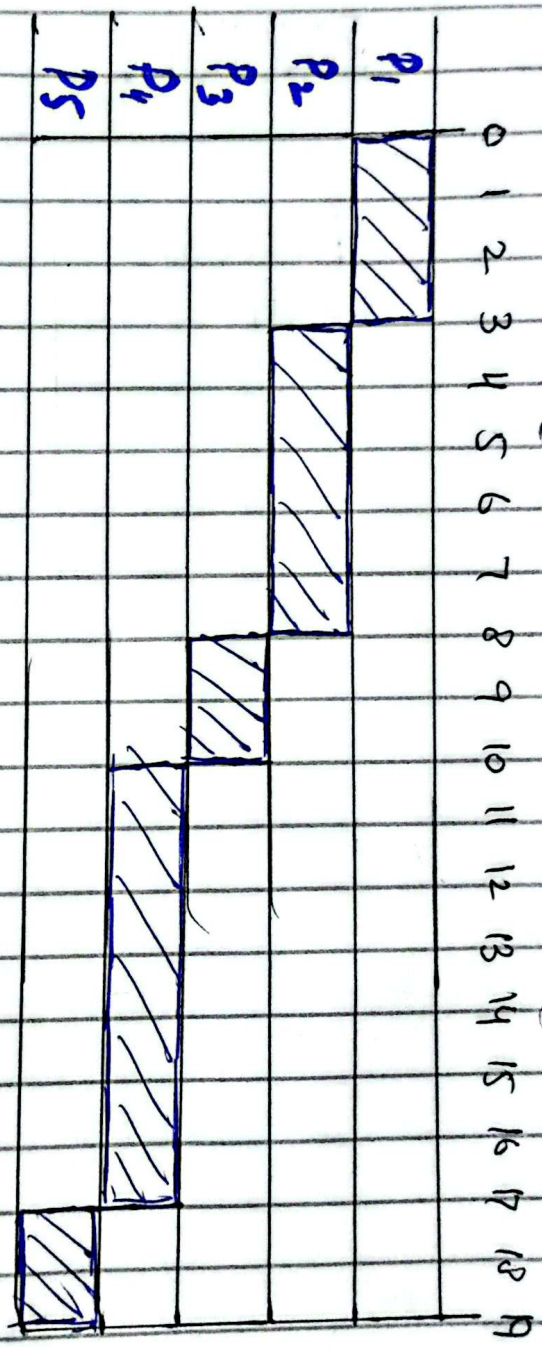
CPU Idle time = 0

PART-C

Process	Arrival T	Service T
P_1	0	3
P_2	2	5
P_3	5	2
P_4	7	7
P_5	11	2

FCFS

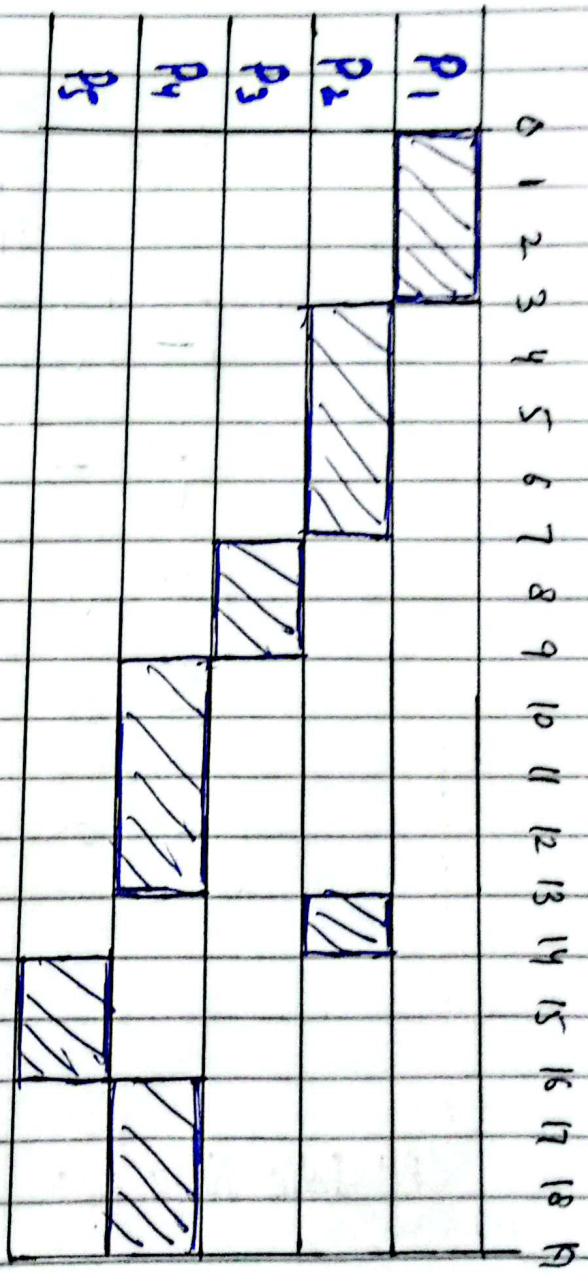
	Turnaround Time	Waiting time	TR/TS
P_1	3	0	1
P_2	9	1	1.2
P_3	5	3	2.5
P_4	10	3	1.42
P_5	8	9	4



CPU Idle time = 0

Round Robin (Q=4)

	Turnaround time	Waiting time	TR/TS
P_1	3	0	1
P_2	12	7	2.4
P_3	4	2	2
P_4	12	5	1.714
P_5	5	3	2.5



Idle time of CPU = 0

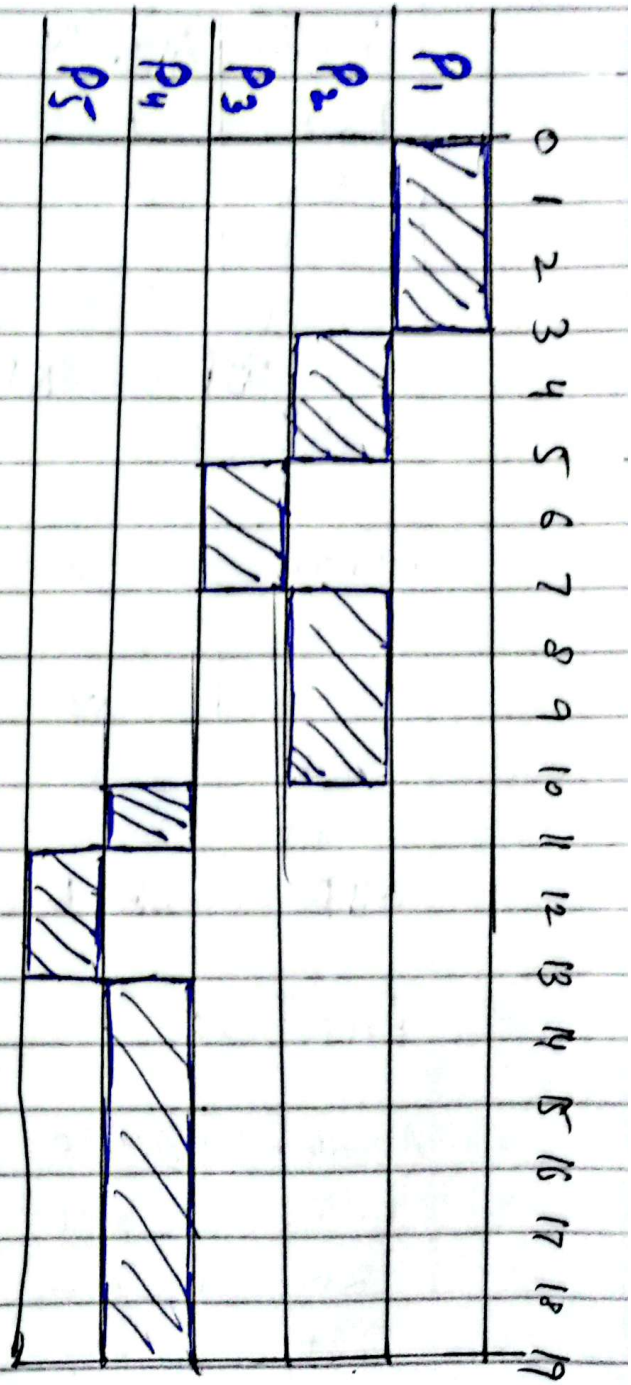
SJF

	Turnaround Time	Waiting time	TT/TS
P_1	3	0	1
P_2	6	1	1.2
P_3	5	3	2.5
P_4	10	3	1.428
P_5	9	6	4

CPU Idle time = 0

SRTF

	Turnaround time	Waiting time	TR/TS
P_1	3	0	1
P_2	8	5	2.667
P_3	2	0	1
P_4	12	5	1.714
P_5	2	0	1



CPU Idle time = 0

Comparing Average values

PART A:

Algorithm	Avg. TR	Avg. Waiting	Avg. TR/TS	Avg. idle time
FCFS	7	3.4	2.162	0
RR	7.6	4	2.196	0
SJF	5.8	2.2	1.44	0
SRTF	5.8	2.2	1.44	0

Both **SJF** and **SRTF** are best having min TR, waiting and TR/TS

PART B:

Algorithm	Avg. TR	Avg. Waiting	Avg. TR/TS	Avg. Idle time
FCFS	6.8	2.8	1.9132	0
RR	7.4	3.4	1.9266	0
SJF	5.8	1.8	1.326	0
SRTF	5.8	1.8	1.326	0

SJF and **SRTF** are best having lowest waiting time, turnaround time and TR/TS

PART C:

Algorithm	Avg. TR	Avg. Waiting	Avg. TR/TS	Avg. Idle time
FCFS	6.4	2.6	2.024	0
RR	7.2	3.4	1.9228	0
SJF	6.4	2.6	2.025	0
SRTF	5.4	2	1.4762	0

SRTF is best having lowest TR, waiting time and TR/TS ratio.