Using an RTOS ensures the rover's tasks are executed predictably and reliably, even under the demanding conditions of a distant planet.

3) <u>Blocked</u> :- if a process (e.g media player) is waiting for an event, such as reading data from the disk it enters the "Blocked" State. it won't use CPU resources untill the required event (I/O operation) is Completed.

The OS frequently switches b/w these states using Context switching to ensure responsiveness, allowing users to multitask smoothly across applications.

Q10
Ans A Real-Time operating System (RTOS) is designed to handle tasks within strict timing Constraits, ensuring that operations are performed within a specific time frame. for the rover, an RTOS is critical because it needs to make timely decisions such as navigating obstacles or processing sensor data, where delays could result in mission failure.

<u>In the Context of space exploration</u> :-

-> The rover must respond immediately to environment changes (e.g, avoiding a rock or adjusting to terrain)

-> Critical tasks, like Communication with Earth or Controlling the rover's Movement, must be prioritized to ensure real-time actions.

Real-Time Constraints :- Set real-time constraints (deadlines) for critical processes to ensure tasks like sensor input and motor adjustments are completed within astrict timeframe.

Task state management :- Continuously monitor processes in different states (ready, running, blocked) and adjust priorities dynamically based on environmental inputs.

Q9

Ans In an operating system, multiple applications (processess run concurrently, and the OS manages their states to ensure smooth user experience, Here's how the OS handles key process states :-

1.) Running :- When the CPU is executing instructions from a process (eg the text editor), the process is in the "Running" state. Only one process can be in this state per CPU core at a time.

2) Ready :- if a user switches to another application (like the web browser), the current process (text editor) moves to the "Ready" state. It is prepared to run but waiting for CPU availability. The OS keeps a list of ready processes and selects the next one based on a scheduling algorithum.

6) CPU scheduling :- The OS scheduler selects the process and assigns CPU time.

7) Execution :- The word processing application begins executing allowing the user to interact with it.

8.) Context Switching :- if necessary, context Switching occurs b/w processes to manage multitasking.

Q8

Ans In a real-time Control system for an autonomous robot, process management and prioritization are crucial for responsiveness. Here's how I would manage the processes.

3 1) Prioritize Critical Processes :- Use a priority-based Scheduling algorithum to priotize time sensitive tasks like sensor data processing and motor control over less critical tasks like decision-making.

2) Preemptive Scheduling :- Implement preemptive scheduling to ensure high-priority task can interupt lower-priority tasks when needed.

c) In many to many threading model, user-level threads are mapped to kernel-level threads (LWPs) for real-time threds it is necessary to bind them to LWPs to ensure they get the required CPU time without delays caused by the scheduling of other threads. This binding ensures that real-time threads meet their timing constraits and perform predictably.

Q8

Ans When a user opens a word processing application on a personal Computer, the operating System goes through the following process creation steps:

1. User Request : The user clicks to open the application triggering an interrupt.

2. System Call :- The operating system issues a system call to create a new process.

3. Process Creation : The OS allocates resources (memory, CPU time) and assigns a Process Control Block to manage the new Process.

4. Loading the Program :- The executable file of the word processing program is loaded into memory from disk.

5. Ready state:- The Process is placed in the "Ready" Queue waiting for CPU Scheduling.

FCFS

Process	Burst Time	Priority	A.T	C.T	T.T	W.T
Q6 (b) P_1	10	3	0	10	10	0
P_2	1	1	0	11	11	10
P_3	2	3	0	13	13	11
P_4	1	4	0	14	14	13
P_5	5	2	0	19	19	14

SJF

C.T	T.T	W.T
19	19	9
1	1	0
4	4	2
2	2	1
9	9	4

Non-Preemptive Priority

C.T	T.T	W.T
16	16	6
1	1	0
18	18	16
19	19	18
6	6	1

Round Robin

C.T	T.T	W.T
19	19	9
3	3	2
10	10	8
6	6	5
16	16	11

a) FCFS

P_1	P_2	P_3	P_4	P_5

t=0 10 11 13 14 19

SJF

P_2	P_4	P_3	P_5	P_1

t=0 1 2 4 9

P_2	P_5	P_1	P_3	P_4

t=0 1 6 16 18 19

Priority

Round-Robin

P_1	P_2	P_3	P_4	P_5	P_1	P_3	P_5	P_1	P_5	P_1

t=0 2 3 5 6 8 10 12 14 16 18 19

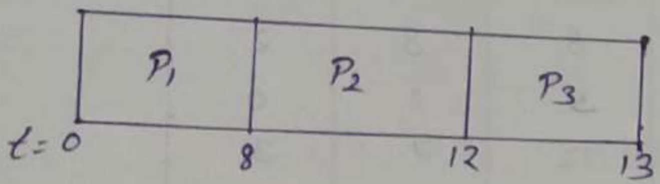
~~P1~~	~~P2~~	~~P3~~	~~P4~~	~~P5~~	~~P1~~	~~P3~~	~~P5~~	~~P1~~	P5	P1

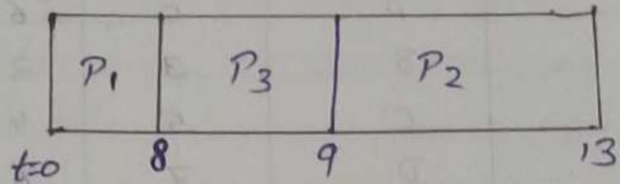
Ready que.

Q5

Process	A.T	B.T	C.T	T.T	C.T	T.T
P_1	0.0	8	8	8	8	8
P_2	0.4	4	12	11.6	9	8.6
P_3	1.0	1	13	12	13	12

FCFS SJF



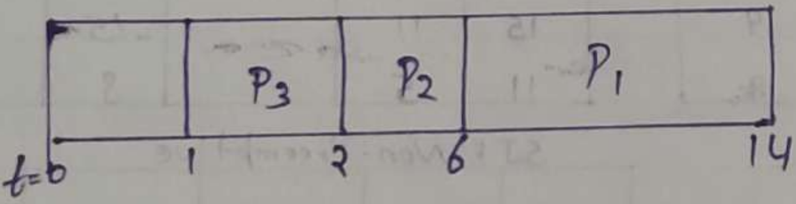


a) Ave T.T for FCFS $= \frac{32.6}{3} = 10.86$

b) Ave T.T for SJF $= 9.53$

for c)

Process	B.T	A.T	C.T	T.T	W.T
P_1	0.0	8	14	14	14
P_2	0.4	4	6	5.6	5.2
P_3	1.0	1	2	1	0



ave T.T $= \frac{20.6}{3} = 6.86$

Q3 Find Ave Turn around time using SRTF Algorithum.

Sol->

Process	A.T	B.T	C.T	T.T	W.T
A	0	6	8	8	2
B	3	2	5	2	0
C	5	4	12	7	3
D	7	6	21	14	8
E	10	3	15	5	2

Gantt Chart

A	B	A	C	E	D	
t=0	3	5	8	12	15	21

Ave T.T => $\frac{36}{5}$ = 7.2

Hence The Ave. T.T is -> 7.2 millisec.

Q4 -> Find the lowest ave. turn around time ? FCF, SJF & SRTF.

Ans-> For FCF

			FCF		SJF		SRTF	
Process	A.T	B.T	C.T	T.T	C.T	T.T	C.T	T.T
A	0	3	3	3	3	3	3	3
B	1	6 3	9	8	9	8	11	10
C	4	4	13	9	15	11	15	11
D	6	2	15	9	11	5	8	2

FCF

A	B	C	D	
t=0	3	9	13	15

SJF Non-Preemptive

A	B	D	C	
t=0	3	9	11	15

A	B	D	B	C	
t=0	3	6	8	11	15

FCF	SJF	SRTF
Ave. T.T => 7.25	Ave. T.T => 6.75	Ave T.T => 6.5

- Memory Management: Virtual memory and paging will allow programs to run even if they require more memory than is physically available, balancing memory needs across all users.

- Peripheral Management :- Peripheral devices (like, printer, external drives, GPUs for deep learning) are shared and allocated as needed, with the operating system managing access in a way that avoids bottlenecks or resource starvation.

This setup would ensure fairness in resource distribution prevent resource monopolization and maximize productivity for both students and professors.

Q2
Sol → The possible output will be two lines but the order of these is not deterministic due to the concurrent execution of the child and processes. The order in which they print can vary.

1) if the parent process executes first :-

output will be

Parent has x = 0

Child has x = 2

2) if the child process executes first :-

output will be

Child has x = 2

Parent has x = 0

Date
26-Sep-2024
Submission

B-Tech CS-(AIML & IOT)
Operating System Assignment
- 01

Manish Bhardwaj, Sec-CB, Rollno->26

Submitted to -> Mr. Hitendra Garg

Q1
Ans. In the GLA University Computer lab 330, where multiple students and researcher share computing resources for activities like programming assignments, data analysis, and training deep learning models, a time-sharing and multi-user operating system would be the most suitable choice.

Key features :-

1. Time-sharing :- This OS allows multiple users to share system resources effectively by assigning each user a small slice of CPU time. This ensures that the system remains responsive and allows multiple users to execute tasks concurrently.

2. Multitasking Capability :- The System can support several users simultaneously, ensuring that both students & Professors can access the System without Conflicts. Each other get their own environment and can performe tasks like programming or model training independently.

3. Resource Management :-

- CPU Scheduling :- Efficient algorithms (like Round Robin & Priority) can be implemented to ensure fairness. Tasks such as running deep learning models or data analysis jobs will recieve adequate CPU time without monopolizing the system