Assignment 2

Part E

1. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	6

Calculate the average waiting time using First-Come, First-Served (FCFS) scheduling

Answer-- > Averag Waiting Time = 4.333

Process	Arival Time	Burst Time	Response Time	Waiting Time	TAT
P1	0	5	0	0	5
P2	1	3	5	4	12
P3	2	6	8	6	12
avg				3.333	

Gannt Chart		P1	P2	P3	
	0	5	8	14	

2. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	3
P2	1	5
P3	2	1
P4	3	4

Calculate the average turnaround time using Shortest Job First (SJF) scheduling.
Answer-- >Average Turnaround Time = 5.5

Process	Arival Time	Burst Time	Response Time	Waiting Time	TAT
P1	0	3	0	0	3
P2	1	5	8	7	12
P3	2	1	3	1	2
P4	3	4	4	1	5
Avg					5.5

Gannt Chart		P1	P3	P4	P2
	0	3	4	8	13

3. Consider the following processes with arrival times, burst times, and priorities (lower number indicates higher priority):

Process	Arrival Time	Burst Time	Priority
P1	0	6	3
P2	1	4	1
P3	2	7	4
P4	3	2	2

Calculate the average waiting time using Priority Scheduling
Answer -- > Aveage Waiting Time = 5.5

Process	Arival Time	Burst Time	Priority	Response Time	Waiting Time	TAT
P1	0	6	3	0	0	6
P2	1	4	1	6	5	9
P3	2	7	4	12	10	17
P4	3	2	2	10	7	9
Avg					5.5	

Gannt Chart		P1	P2	P4	P3
	0	6	10	12	19

4. Consider the following processes with arrival times and burst times, and the time quantum for Round Robin scheduling is 2 units:

Process	Arrival Time	Burst Time
P1	0	4
P2	1	5
P3	2	2
P4	3	3

Calculate the average turnaround time using Round Robin scheduling
Answer -- > Average Turnaround Time =9.25

Process	Arival Time	Burst Time	Complition Time	TAT
P1	0	4	10	10
P2	1	5	14	13
P3	2	2	6	4
P4	3	3	13	10
Avg				9.25

Gannt Chart	P1	P2	P3	P4	P1	P2	P4	P2
0	2	4	6	8	10	12	13	14

5. Consider a program that uses the fork() system call to create a child process. Initially, the parent process has a variable x with a value of 5. After forking, both the parent and child processes increment the value of x by 1.
What will be the final values of x in the parent and child processes after the fork() call?
Answer -- >
· Before fork(), the parent process has a variable x = 5.
· When fork() is called:
A child process is created, and it gets its own copy of x = 5.
· Both the parent and child processes execute independently:
The parent increments x → x = 6
The child increments its own x → x = 6