DEPARTMENT OF DATA SCIENCE & COMPUTER APPLICATIONS

MCA 4223 OPERATING SYSTEMS: MISAC-1 ANSWER SCHEME SET-1

DATE: 08-02-2024　　DURATION: 20 MINS　　MAX. MARKS: 5

Q) Consider a scenario in which every process executes as follows:

- Initially it will run in the CPU for a specific amount of time (CPU burst).
- After the CPU burst, it will wait for I/O to complete (I/O burst).
- Once it completes the I/O burst, it will again run in CPU and then terminate (CPU burst).

Please Note: The I/O bursts will be happening concurrently, meaning no process needs to wait for other processes to complete their I/O.

Given a set of 4 processes whose arrival time and burst time are listed as follows:

PID	Arrival Time	Burst Time		
		CPU Burst	I/O Burst	CPU Burst
101	0	3	2	2
102	0	2	4	1
103	2	1	3	2
104	5	2	2	1

1) Draw the Gantt Charts that illustrate the execution of the following algorithms.

a) FCFS (Break the tie, if any, by considering the process with smaller value of PID first).

b) Shortest Remaining Time First (Break the tie, if any, by considering the process that has arrived first. If again a tie exist, consider the process with smaller value of PID first)

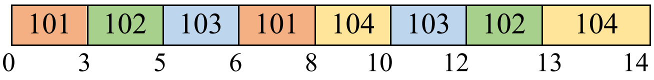
[3 marks]

2) Which among the above 2 scheduling algorithms performs better for the above given scenario. Justify your statement with necessary computations.

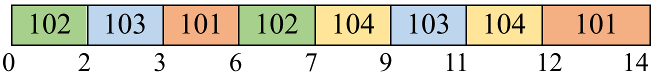
[2 marks]

Answer:

1) a) Gantt chart for FCFS:



1) b) Gantt chart for SRTF:



2)

Process	**Arrival Time**	**CT (FCFS)**	**TAT (FCFS)**	**WT (FCFS)**	**CT (SRTF)**	**TAT (SRTF)**	**WT (SRTF)**
101	0	8	8	1	14	14	7
102	0	13	13	6	7	7	0
103	2	12	10	4	11	9	3
104	5	14	9	4	12	7	2
		Average:	10	3.75	**Average:**	9.25	3

Note: I/O time is not considered as waiting time. So, WT = TAT – CPU time – I/O time

As per the above table, SRTF performs better than FCFS for the given scenario with lesser values for Average TAT and WT.

DEPARTMENT OF DATA SCIENCE & COMPUTER APPLICATIONS

MCA 4223 OPERATING SYSTEMS: MISAC-1 ANSWER SCHEME SET-2

DATE: 08-02-2024 DURATION: 20 MINS MAX. MARKS: 5

Q) Consider a scenario in which every process executes as follows:

- Initially it will run in the CPU for a specific amount of time (CPU burst).
- After the CPU burst, it will wait for I/O to complete (I/O burst).
- Once it completes the I/O burst, it will again run in CPU and then terminate (CPU burst).

Please Note: The I/O bursts will be happening concurrently, meaning no process needs to wait for other processes to complete their I/O.

Given a set of 4 processes whose arrival time, burst time and priority are listed as follows:

PID	Priority	Arrival Time	Burst Time		
			CPU Burst	**I/O Burst**	**CPU Burst**
101	2	0	3	2	2
102	3	0	2	4	1
103	1	2	1	3	2
104	4	5	2	2	1

1) Draw the Gantt Charts that illustrate the execution of the following algorithms:

a) Non-Preemptive Priority (smaller number has higher priority)

b) Shortest Remaining Time First (Break the tie, if any, by considering the process that has arrived first. If again a tie exist, consider the process with smaller value of PID first)

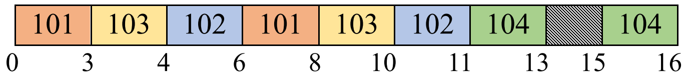
[3 marks]

2) Which among the above 2 scheduling algorithms performs better for the above given scenario. Justify your statement with necessary computations.

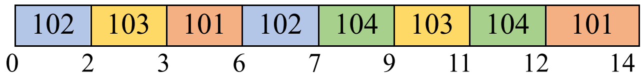
[2 marks]

Answer:

1) a) Gantt chart for Non-Preemptive Priority:



1) b) Gantt chart for SRTF:



2)

Process	**Arrival Time**	**CT (Priority)**	**TAT (Priority)**	**WT (Priority)**	**CT (SRTF)**	**TAT (SRTF)**	**WT (SRTF)**
101	0	8	8	1	14	14	7
102	0	11	11	4	7	7	0
103	2	10	8	2	11	9	3
104	5	16	11	6	12	7	2
		Average:	9.5	3.25	**Average:**	9.25	3

Note: I/O time is not considered as waiting time. So, WT = TAT – CPU time – I/O time

As per the above table, SRTF performs better than Non-Preemptive Priority for the given scenario with lesser values for Average TAT and WT.