

Ch-6 CPU SCHEDULING

Descriptive

- 1) Write difference between preemptive and non preemptive scheduling with example.
- 2) Consider following set of processes ,

Process	Arrival Time	Burst Time
<i>P1</i>	0.0	7
<i>P2</i>	2.0	4
<i>P3</i>	4.0	1
<i>P4</i>	5.0	4

Draw Gantt chart for SJF preemptive scheduling

Calculate average waiting time .

- 3) Consider following set of processes ,

Process	Arrival Time	Burst Time
<i>P1</i>	0.0	8
<i>P2</i>	4.0	4
<i>P3</i>	1.0	1
<i>P4</i>	5.0	4

What is average waiting time for the above processes using FCFS non preemptive scheduling ?

- 4) Calculate turn around time for the given processes using RR(round robin) algorithm .
Quantum=20

Process	Burst Time
<i>P1</i>	53
<i>P2</i>	68
<i>P3</i>	17
<i>P4</i>	24

- 5) List all optimized criteria for scheduling.
- 6) Write importance of CPU scheduler

Objective

- 1) List Process scheduling algorithms
- 2) FCFS,SJF,RR,SRTF –write full forms
- 3) The smallest integer number for priority of process means that the process is having highest priority.(T/F)

- 4) The small unit of time allocated to process in RR algorithm is called _____(quantum, arrival time)
- 5) The effect we get with short process behind long process is called _____effect(Gantt,convoy)
- 6) Define :1)Response time 2)turnaround time

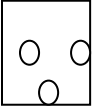
CH -8 DEADLOCK

Descriptive

- 1) What do you mean by deadlock problem?
- 2) Write Deadlock characteristics.
- 3) Write about resource allocation graph with the help of symbolic notations.
- 4) Draw the resource allocation graph with deadlock and explain
- 5) Write methods of Deadlock prevention and explain any two.
- 6) In which order the process abortion should be done for deadlock recovery?

Objective

- 1) Write an example in which two process have a deadlock
- 2) $P1 \rightarrow R_j$. What this notation does mean in resource allocation graph?

- 3)  What does this notation suggest?

- 4) Returning to some safe state and restart the process for that state during deadlock recovery is called _____.
 i) Starvation ii) return iii) rollback iv) recover