Roll no :- T21287

Page No.
Date

Assignment No 3

CPU scheduling

AIM : Implement the C program for CPU scheduling Algorithms shortest Job first (preemptive) and Round Robin with diffrent arrival time.

OBJECTIVES : To study

* CPU scheduling
* Round Robin

THEORY :

To calcute the average waiting time in the shortest job first algorithm the sorting of the process based on their burst time in ascending order then calculate the waiting time of each process as the sum of the bursting times of all the process previous or before to that process.

ALGORITHM :

step 1 : start the process

step 2 : Accept the number of processes in the ready queue

step 3 : for each process in the ready Q assign the process id & accept the CPU burst time

step 4 : start the Ready Q according the shortest Burst time by sorting according to lowest to highest burst time

step 5 : set the waiting time of the first process as '0' & its turnaround time as its burst time

step 6 : sort the processes name based on their Burst time

step 7 : for each process in the ready queue, calculate a) waiting time (n) = waiting time (n-1) + Burst time (n-1) b) Turnaround time (n) = waiting time (n) + Burst time (n)

step 8 : Calculate c) Average waiting time = Total waiting time / number of process d) Average turnaround time = Total turnaround time / number of process

step 9 : stop the process

To aim is to calculate the average waiting time There will bt a time slice each process

should be executed within that time slice & if not it will go to the waiting state so first check whether the burst time is less than the time-slice If it is less than it assign the waiting time to the sum of the total times

ALGORITHM :

step 1 : start the process

step 2 : Accept the number of processes in the ready queue & time quantum (or) time slice

step 3 : for each process in the ready q. assign the process id & accept the cpu burst time

step 4 : Calculate the no. of time slices for each process where No of time slice for process (n) = burst time process (n) / time slice

step 5 : If the burst time is less than time slice then the no. of time slices = 1

step 6 : Consider the ready queue is a circular q. Calculate a) waiting time for process (n) = waiting time of

* Decrements semaphore value
* If value becomes negative, process is blocked
* otherwise it continues execution

* Increment (semsignal)

* process executive it to transmit a signal via semaphore
* increments semaphore value
* If value is less than or equal to zero, process blocked by sem wait is unblocked

Conclusion :

Thus, we have implemented producer consumer problem using 'C' in Linux