E ou tramengesA

CPU scheduling

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

OBJECTIVES . To study

- * CPU scheduling
- * Round Robin

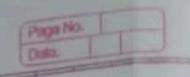
THEORY

to calcule the average wouling time in the shortest Joh first algorithm the sarting of the process based on their burst time in ascending arder then calculate the waiting time of each process as the sam of the bursting times of all the process previous or before to that process.

ALCORITHM :

step 1: start the process

step 2 : Accept the number of processes in



- step 3 for each process in the ready a assign the process id P accept the
- step 4 start the Ready & according the shortest Burst time by sorting according to howest to highest burst time
- step 5 set the cuceiting time of the first process as -0' & it's turnaround time as it's barst time
- step 6 sort the processes name based on
- step 7: for each process in the ready queue,

 coloulate a) cuceiting time (n) = cocuting

 time (n-1) + Basst time (n-1) b)

 Turneround time (n) = cocuting time (n)

 Basst time (n)
- step 8 (calculate c) Average waiting time =

 Total auculting time I number of process

 d) Average tarnarroad time = Total

 turnround time I number of process

step 9 stop the process

To ceim is to calculate the average waiting time these will be a time slice each process

should be executed within that time slice & if not it will go to the cociting 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 told times

ALGORITHM:

step 1: start the process

step 2 : Accept the number of processes in

step 3 for each process in the ready a assign the process id & accept the

step 4 concluded the no of time stices for each process where No of time stice for process (n) = burst time process

(n) I time stice

step 5 : If the burst time is less than time slices = 1

step 6: consider the ready queue is a circular a calculate a) waiting time for process (n) = waiting time of

Page No.

- * Decements semaphore value
- * If value becomes negative, process is blocked
- * otherwise it continues exerction

* Innement (sem signed)

- * process exercetive it to transmit a signal via semaphone
- * increments semaphore value * If value is less than or equal to Zero, process blocked by sem wait
- · is unblocked

Conclusion

Consumer problem using 'C' in Linux