CS-362L Operating System Lab 07

Objectives:

To get familiar Scheduling Algorithms first come first serve

What is FCFS?

First in, first out (FIFO), also known as first come, first served (FCFS), is the simplest scheduling algorithm. FIFO simply queues processes in the order that they arrive in the ready queue

Processing steps:

Step 1: Input the processes along with their burst time (bt).

Step 2: Find waiting time (wt) for all processes.

Step 3: As first process that comes need not to wait so

waiting time for process 1 will be 0 i.e. wt[0] = 0.

Step 4: Find waiting time for all other processes i.e. for process i calculate

wt[i] = bt[i-1] + wt[i-1] .

Step 5: Find turnaround time = waiting_time + burst_time

for all processes.

Step 6: Find average waiting time = total_waiting_time / no_of_processes.

Step 7: Similarly, find average turnaround time = total_turn_around_time / no_of_processes.

Class activity

Implement the first come first serve by taking number of process, arrival time and brust time from input and calculate their waiting time turnaround time, completion time, average waiting time and average turnaround time.

Priority Scheduling Algorithms

-

Objectives:

To get familiar Priority Scheduling

Algorithms

What is Priority Scheduling Algorithms?

Priority scheduling is one of the most common scheduling algorithms in batch systems. Each process is assigned a priority. Process with the highest priority is to be executed first and so on. Processes with the same priority are executed on first come first served basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Processing steps:

Step 1: First input the processes with their burst time and priority.

Step 2: Sort the processes, burst time and priority according to the priority.

Step 3: Now simply apply FCFS algorithm.

Step 4: Find waiting time (wt) for all processes.

Step 5: As first process that comes need not to wait so waiting time for process 1 will be 0 i.e. wt[0] = 0.

Step 6: Find waiting time for all other processes i.e. for process i calculate

wt[i] = bt[i-1] + wt[i-1] .

Step 6: Find turnaround time = waiting_time + burst_time

for all processes.

Step 7: Find average waiting time = total_waiting_time / no_of_processes.

Step 8: Similarly, find average turnaround time = total_turn_around_time / no_of_processes.

Class activity

Implement the Priority **Scheduling** Algorithms by taking number of processes, arrival time and brust time from input and calculate their waiting time turnaround time, completion time, average waiting time and average turnaround time.

CS-362L Operating System Lab 08

Shortest job first

Objectives:

To get familiar shortest job first Scheduling

Algorithms

Entertain the process according to minimum brust time to have less waiting time for the process

- This is also known as shortest job first, or SJF
- This is a non-pre-emptive, pre-emptive scheduling algorithm.
- Best approach to minimize waiting time.
- Easy to implement in Batch systems where required CPU time is known in advance.
- Impossible to implement in interactive systems where required CPU time is not known.
- The processer should know in advance how much time process will take.

Processing steps:

Step 1: Input process with their brust time and quantum which are time slot given to each process.

Step 2: Sort all the process according to the Brust time.

Step 3: Then select that process which have minimum Burst time.

Step 4: After completion of process make a pool of process which after till the completion of previous process and select that process among the pool which is having minimum Burst time.

Step 5: Turn Around Time: Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival Time

Step 6: Waiting Time (W.T): Time Difference between turnaround time and burst time. Waiting Time = Turn Around Time – Burst Time

Class activity

Implement the shortest job first scheduling algorithm by taking number of processes, arrival time and brust time from input and calculate their waiting time turnaround time, completion time, average waiting time and average turnaround time.

Round Robbin

Objectives:

To get familiar with Round Robin Algorithm

What is Round Robin Algorithm?

- Round Robin is a CPU scheduling algorithm where each process is assigned a fixed time slot in a cyclic way.
- It is simple, easy to implement, and starvation-free as all processes get fair share of CPU.
- One of the most commonly used technique in CPU scheduling as a core.
- It is pre-emptive as processes are assigned CPU only for a fixed slice of time at most.
- The disadvantage of it is more overhead of context switching.

How to compute below times in Round Robin using a program?

- Completion Time: Time at which process completes its execution.
- Turn Around Time: Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival Time
- Waiting Time (W.T): Time Difference between turnaround time and burst time. Waiting Time = Turn Around Time – Burst Time

Steps to find waiting times of all processes:

1- Create an array **rem_bt[]** to keep track of remaining

burst time of processes. This array is initially a

copy of bt[] (burst times array)

2- Create another array **wt[]** to store waiting times

of processes. Initialize this array as 0.

3- Initialize time : t = 0

4- Keep traversing the all processes while all processes

are not done. Do following for i'th process if it is

not done yet.

a- If rem_bt[i] > quantum

(i) t = t + quantum

(ii) bt_rem[i] -= quantum;

c- Else // Last cycle for this process

(i) t = t + bt_rem[i];

(ii) wt[i] = t - bt[i]

(ii) bt_rem[i] = 0; // This process is over

Class activity

Implement the Round Robin algorithm by taking number of proceses , arrival time,and brust time from input and calculate their waiting time turnaround time, completion time ,average waiting time and average turnaround time.