CS 149 Report

Group 2 : David Vu, Brian Lai, Katelynn Tran, Jomel Bautista, John Lau, Tsering Dolkar

**Process Scheduling Algorithms**

*First-Come-First-Served(FCFS) [Non-Preemptive]*

* Schedules according to arrival time of processes.

*Shortest Job First (SJF) [Non-Preemptive]*

* Schedules the processes with shortest runtime first.

*Shortest Remaining Time (SRT) [Preemptive]*

* Preemptive version of Shortest Job First, schedules jobs according to the shortest remaining time.

*Round Robin (RR) [Preemptive]*

* Assigns each process a fixed time and schedules in a cyclic way

*Highest Priority First (HPF) [Non-Preemptive & Preemptive]*

* Schedules according to their priorities, highest priority first and if two processes have the same priority then schedules according to the arrival time.

**Calculated Statistics**

*Minimum Average Turnaround Time*

* *Turn Around Time = Completion Time – Arrival Time*
* *Shortest Job First* is the best process scheduling algorithm to calculate the minimum average turnaround time because it schedules the processes by the shortest runtime. This is the most optimal because the completion time of each process is at its minimum. Therefore, if you take the minimum completion time and subtract the arrival time of the process then it is the minimum turnaround time.

*Ex.*

|  |  |  |
| --- | --- | --- |
| *Process* | *Arrival* | *Runtime* |
| *0* | *0* | *9* |
| *1* | *1* | *4* |
| *2* | *2* | *9* |

*SJF Chart -> | P1 | P0 | P2 | FCFS Chart -> | P0 | P1 | P2 |*

*0 4 13 22 0 9 13 22*

*AverTurnAround = ((4-0)+(13-1)+(22-2))/3 = ((9-0)+(13-1)+(22-2))/3*

*= 12 = 13.3*

*Minimum Average Waiting Time*

* *Waiting Time = Turnaround Time - Runtime*
* *Shortest Remaining Time* is the best algoritrithm to calculate the minimum average waiting time because it handles the processes with lowest runtime first and then handles the processes with longer runtime. This will shorten the wait time because the processes with lower wait times are already handled.

*Ex.*

|  |  |  |
| --- | --- | --- |
| *Process* | *Arrival* | *Runtime* |
| *0* | *0* | *9* |
| *1* | *1* | *4* |
| *2* | *2* | *9* |

*SRT Chart -> | P0 | P1 | P0 | P2 | SRT Chart -> | P1 | P0 | P2 |*

*0 1 5 14 22 0 4 13 22*

*AverWaiting = ((5 -1) + 0 + (14-2))/3 = (0 + 4 + 13)/4*

*= 5.3 = 5.6*

*Minimum Average Response Time*

* *Response Time = Process produces first response*
* *Round Robin* is the best algorithm to calculate the minimum response time because it splits the processes into fixed time pieces. Therefore, the processes are accessed more because it has been split up.

*Ex.*

|  |  |  |
| --- | --- | --- |
| *Process* | *Arrival* | *Runtime* |
| *0* | *0* | *9* |
| *1* | *1* | *4* |
| *2* | *2* | *9* |

*RR Chart -> |P1 | P2 | P3 | P1 | P2 | P1 | P2 |*

*0 4 8 12 16 20 21 22*

*Maximum Throughput*

* *Throughput = Number of Process / Time Unit*
* *Highest Priority First* is the best algorithm to calculate the maximum throughput because the processes are sorted by highest priority. The highest priority processes are executed first then the lower, however there can be starvation – low priority may never be executed. But aging will fix the problem because the technique increased the priority of the low priority processes. Therefore , all processes are executed.