Page: 1
Date: / /

Assignment C1

Title : Scheduling Algorithms

Date of Completion : 26/12/19

Problem Statement :
Write a Java Program (using OOP features) to implement the following scheduling algorithms : FCS, SJF (Preemptive), Priority (Non-Preemptive) & RoundRobin (Preemptive)

Objectives :
i. Process Scheduling in Multitasking & Multiuser in OS
ii. Implementation of scheduling algorithms

Outcomes :
Student should be able to :-
i. Implement the above mentioned Scheduling Algorithms.
ii. Apply the concepts of OOP
iii. Understand the key differences between the Scheduling Algorithms.

Software & Hardware Requirements :-
Fedora 64 bit OS
Eclipse IDE for Java
i5 processor
4GB RAM
500GB HDD

Open Source Software (FOSS) tools.

Theory :

Process Scheduling & Scheduler

Process Scheduling is a set of policies & mechanisms supported by operating system that controls the order in which the task to be done is completed. It is a fundamental operating System function.

A scheduler is an operating system program that selects the next job to be admitted for execution.

Performance Criteria for process scheduling

- CPU Utilization
- Turnaround Time
- Response Time
- Throughput
- Waiting Time
- Fairness.

	Preemptive Scheduling	Non-Preemptive Scheduling
i	The CPU can be taken away from the process	The CPU cannot be taken away from the process
ii	Process can be temporarily suspended due to: IO or Time Slab	Process runs to its completion
iii	Relatively Complex	Simple & Easy to implement
iv	It is suitable for multi-user environment	It is not suitable for multi user environment

Page: 3
Date: / /

First Come First Serve Scheduling (FCFS)

FCFS is a long term scheduling algorithm.
Jobs are executed in the order of arrival.
A queue of arriving jobs is maintained.
Whenever CPU is free, a new job from the ready queue is assigned CPU.
Once the process has CPU, it runs to completion

Advantages :

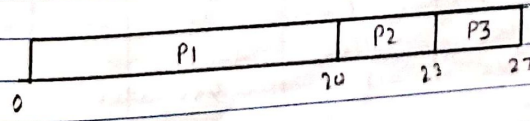
1. FCFS is simple to implement
2. FCFS meets fairness criteria for scheduling

Disadvantages :

1. FCFS' turnaround & waiting time are high
2. FCFS algorithm can't be used in time sharing system.

Example.

Process	Start time	CPU time	Finish time	Turnaround time	Waiting time
P1	0	20	20	20	0
P2	20	3	23	23	20
P3	23	4	27	27	23
				= 70/3	= 43/3
				= 23.33	= 13.33



Shortest Job First Scheduling (SJF)

In SJF, a job with the shortest execution time is selected for execution

SJF can either be preemptive or non preemptive i.e. either the job is completed till execution or jobs are suspended temporarily to complete a shorter job.

SJF is an optimal scheduling algorithm in terms of minimizing the average waiting time & average turnaround time.

Advantages :

It minimizes average waiting time & average turnaround time

Throughput at any instant is higher

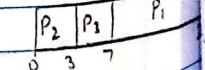
Disadvantages :

A relatively long job might have to wait for a long time.

Optimal Performance of SJF depends on future knowledge of job behaviour.

Example :

Process No.	CPU Time	Start Time	Finish Time	Turnaround Time	Waiting Time
P1	20	7	27	27	7
P2	3	0	3	3	0
P3	4	3	7	7	3
				37/3 = 12.33	10/3 = 3.33



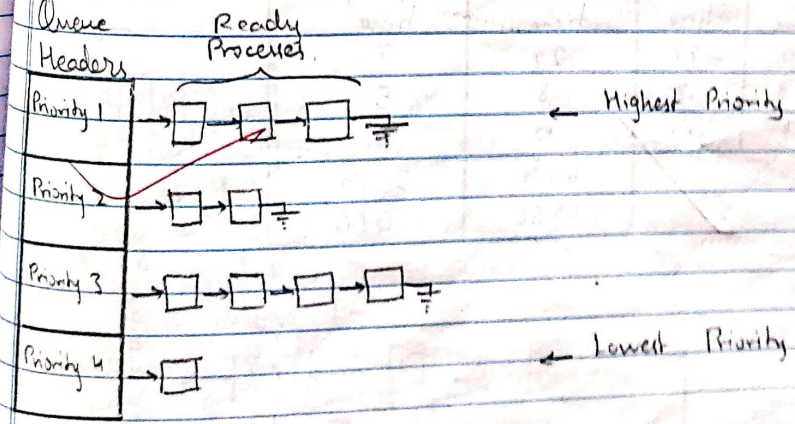
Page: 5
Date: / /

Priority Scheduling

Priority Scheduling is preemptive scheduling method.
Each Process is assigned a priority
Among the runnable processes, one with the highest priority is allowed to run first
It is often necessary to group processes into priority classes
Round Robin Scheduling is used within each priority class

Advantages :
High Priority Tasks can be completed early
Good for Real Time System

Disadvantages :
A long low priority process will wait indefinitely



Robin Robin Scheduling

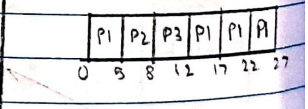
Round Robin Scheduling Algorithm is primarily used in time-sharing & a multi-user system.
CPU time is divided in slices / quantum.
Each process is allocated one time-slice, while it is running.
After the end of time slice, the process goes to the end of ready queue & a process from the front of ready queue is selected for execution.

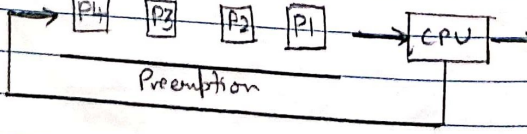
Advantages:
Good Response Time.
Sharing of System is done fairly.

Disadvantages:
Setting Quantum too short lowers CPU efficiency.
Overhead due to switching might incur.

Example

Process No.	CPU Time	Finish Time	Turnaround time	Waiting time
P1	20	27	27	7
P2	3	8	8	5
P3	4	12	12	8
			47/3 = 15.66	20/3 = 6.66





Convoy Effect in FCFS:

Convoy Effect is a phenomenon associated with FCFS algorithm, in which the whole Operating System slows down due to a few slow processes.

It is also known as Starvation.

If a Process with a large burst time is alloted CPU, then the Process will take a considerable time to finish. Because of this, the succeeding jobs with ~~longe~~ shorter burst times will have to wait, since FCFS is preemptive.

Thus the average waiting time increases, which leads to lower throughput.

Algorithms

1) FCFS

Jobs to be processed are stored in arrival queue.
1. If both wait queue & ready queue are empty then goto step 9.
2. For every job P in wait queue.
   If P.arrival_time ≤ time elapsed then
   P is deleted from wait queue
   P is inserted in ready queue.
4. A new job P is selected from ready queue for execution
5. Finish time, Turnaround Time & Waiting Time are calculated.
6. elapsed time = elapsed time + P.CPU_time

7. Job P is inserted in finish queue
8. Goto step 2
9. Generate Report
10. Stop.

2) SJF - Preemptive

1. Jobs to be processed are stored in arrival queue
2. Elapsed = 0
3. If both arrival queue & ready queue are empty then goto step
4. For every job P in arrival queue
   if P.arrival time ≤ elapsed then
   P is deleted from arrival queue
   P is inserted in ready queue
5. Jobs in ready queue are sorted with smallest in the front. (Priority Queue according to Burst Time)
6. If the newly arrived job P' is smaller than the remaining burst time for the current process P then
   P' becomes the currently executing process
   P is temporarily suspended.
7. P is selected from ready queue for execution
8. Finish time, Turnaround Time & Waiting Time are calculated
9. P is inserted in finish queue
10. Go to step 3
11. Generate Report
12. Stop

Page: 9
Date:

Priority

Jobs to be processed are stored in arrival queue
If both wait queue & ready queue are empty then
goto step 10
For every job P in wait queue
If P. arrival_time ≤ time_elapsed then
P is deleted from wait queue
P is inserted in ready queue.
~~A new job~~ If Ready Queue is a priority Queue
with the top most priority at the front end.
A new job P is selected from ready queue for execution
Finish Time, Turnaround Time & Waiting Time are calculated.
Elapsed time = elapsed time + P. CPU_time.
Job P is inserted in finish queue.
Goto step 2
Generate Report
Stop

4) Round Robin Algorithm

1. Jobs to be processed are stored in the wait queue. elapsed = 0.
2. If both wait queue & ready queue are empty then goto step 8
3. For every job P in wait queue
   if P. arrival time ≤ elapsed then
   P is deleted from wait queue
   P is inserted in ready queue
4. If ready queue is empty then
   elapsed = elapsed + slab
   goto step 2
5. A Process P is selected from ready queue for execution
   a. if P. elapsed time + slab ≥ P. CPU_time then
      elapsed = elapsed + P. CPU time - P. CPU time already given to process
      P. CPU time already given = P. CPU_time.
   b. else if P. elapsed_time + slab < P. CPU_time then
      P. CPU_time already given + = slab
      elapsed + = slab.
6. For every job P1 in Wait Queue
   if P1. ~~slot~~ arrival time ≤ elapsed then
   P1 is deleted from Wait Queue
   P1 is inserted in ready Queue.
7. If P has to run to its completion ~~CPU~~. then
   P. finish time = elapsed
   P. turnaround time = elapsed - P. arrival time
   P. waiting time = P. turnaround time - P. CPU time
   Insert P in finish queue
   goto step 4.
8. Generate Report
9. Stop

Scanned with CamScanner

Sr No.	Parameter ↓ / Scheduling Algorithm →	FCFS	SJF	Round Robin	Priority
1)	Definition	Scheduling Algorithm where jobs are executed in order of arrival.	Scheduling Algorithm where jobs are executed in order of burst time (Smallest)	Scheduling Algorithm where all jobs are allotted time slices.	Scheduling Algorithm where jobs are executed in order of assigned Priority
2)	Full Form	First Come First Serve	Shortest Job First	-	-
3)	Preemptive or Non Preemptive	Non Preemptive	Both	Preemptive	Both
4)	Ease of Implementation	Easy	Preemptive - ~~Hard~~ Complex ~~Easy~~ Non-Preemptive - Easy	Complex	Complex
5)	Fairness to all Processes	Fair	Not fair to long Process	Fair	Not fair to low ~~Complt~~ Priority Processes
6)	Turnaround & Waiting Time	High	~~Hi~~ Lower than SJF	Between FCFS & SJF	Cannot be predicted
7)	Suitability for Time Sharing System	~~Yes~~ No	No	Yes	Yes

Test Cases

Sr. No.	Description	Expected Output	Actual Output	Result
1)	FCFS			

Process	CPU.T	Arrival.T
P1	20	0
P2	3	~~20~~ 1
P3	4	~~23~~ 2

Waiting.T
0
~~20~~ 19
~~23~~ 21

Waiting.T.
0
~~20~~ 19
~~23~~ 21

Success

2) SJF

Process	CPU.T	Arrival T
P1	20	0
P2	3	0
P3	4	0

Waiting.T
7
0
3

Waiting T
7
0
3

Success

3) Priority

Process	CPU.T	Priority
P1	20	2
P2	3	3
P3	4	1

Waiting.T
4
24
0

Waiting T
4
24
0

Success

4) Round Robin

Process	CPU.T	Arrival T
P1	24	0
P2	4	0
P3	3	0

Slice = 4

Waiting T
7
4
8

Waiting T
7
4
8

Success

Page :13
Date : / /

Conclusion :

The mentioned Scheduling Algorithms have been studied & successfully implemented ~~in~~ in Java using the concepts of OOP.

Sneha
02/01/2020