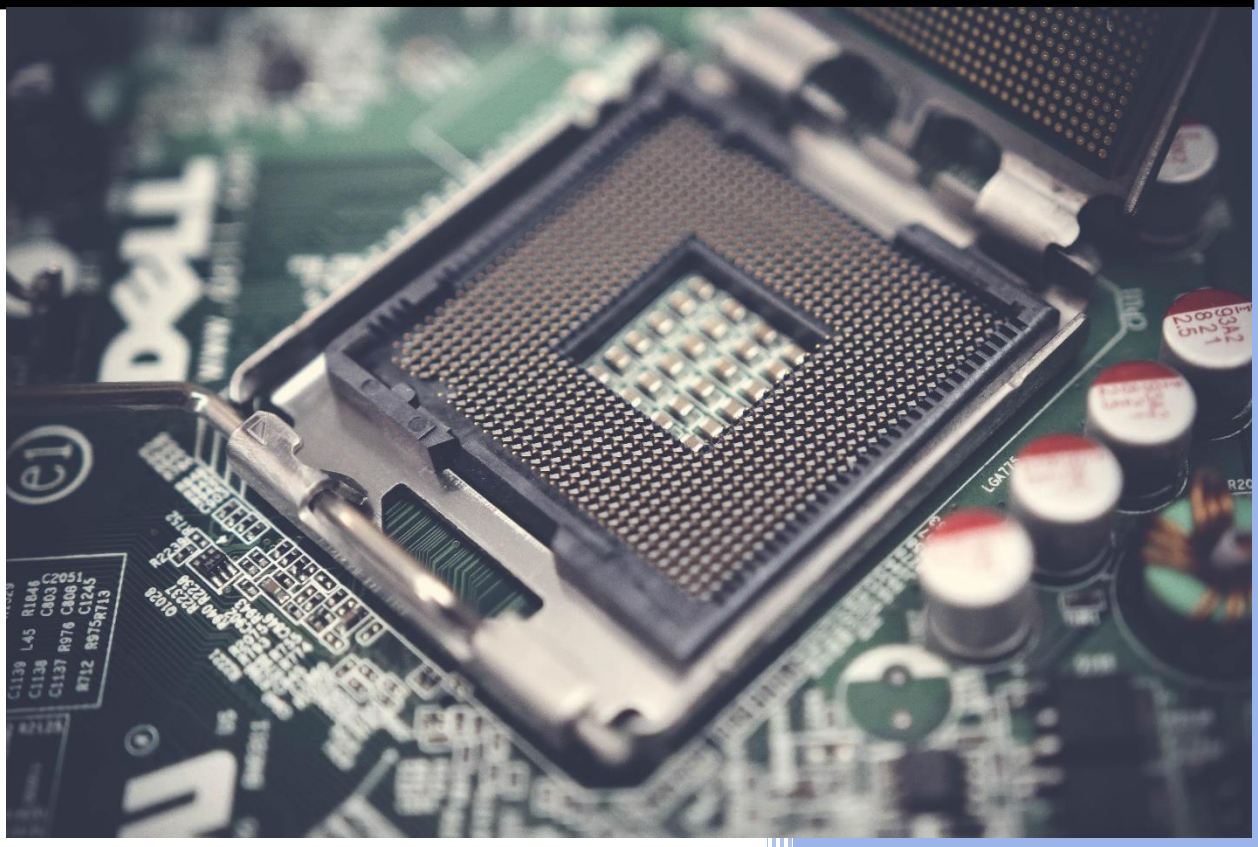
Fall 2018

COIS 3320 Lab 2 Analysis Report



Joshua Croft & Vildan Hakanaj

Trent University

Fall 2018

Introduction:

In the discussion of CPU scheduling algorithms, it is sometimes unclear as to which algorithms are the most efficient. It is also unclear as to whether there is any one algorithm that can be considered the optimal choice in all situations and conditions. The aim of this report is; to simulate the following algorithms: First Come First Serve, Shortest Job First, Shortest Job First with Pre-emption and Round Robin (using two different time quantum values); to test each of the four outlined algorithms against certain criteria, and to determine which conditions favour which algorithm in regards to performance.

Technologies used:

- The program was written in the java programming language.
- Intellij IDE was used exclusively.
- Github was used for version control.

Methods:

The four algorithms are tested against three different types of "Job sets". The job sets are defined as collections of jobs that are created under the same possible conditions, restrictions and scope. A Job is defined as a process to be run within the CPU and to be controlled and manipulated by the scheduling algorithm.
A Job contains an initial arrival time for entry into the waiting queue and a job length representing the amount of time remaining for processing. The properties of each job are generated using gaussian distributions. All three job sets assign a random arrival time upon creation of the job using a gaussian distribution with mean = 160 and standard deviation = 15. The job sets differ in their assignment of the job length property.

For the sake of simplicity, abstract integer "time units" will be used to represent time.

- Job length distribution parameters for Job set #1
  - Mean = 150-time units
  - Standard Deviation = 20-time units

- Job set #2 and Job set #3 is made up of two "types" of Jobs: large and small.
  - Small Jobs
    - Mean = 50-time units
    - Standard Deviation = 5-time units
  - Large Jobs
    - Mean = 250-time units
    - Standard Deviation = 15-time units

- Job length distribution parameters for Job set #2
  - 20% chance of a Job to considered "large Job".
  - 80% chance of a job to be considered "small Job".

- Job length distribution parameters for Job set #3
  - 80% chance of a Job to considered "large Job".
  - 20% chance of a job to be considered "small Job".

The five algorithms to run and a brief description of each are as follows:

- First come first serve.
  - Jobs process on a first-in first-out basis dependant upon arrival time into the waiting queue. Once a job begins processing, it will run to completion (i.e. no other Job may run until this Job finishes processing).

- Shortest Job First.
  - Jobs process depending on whichever currently waiting Job has the shortest Job Length property (shortest time to process). This excludes Jobs that have not yet arrived. Once a job begins processing, it will run to completion (i.e. no other Job may run until this Job finishes processing).

- Shortest Job First with Pre-emption
  - Jobs process depending on whichever currently waiting Job has the shortest Job Length property (shortest time to process). This excludes Jobs that have not yet arrived. Once a job begins processing, it will run for a set interval of 40 time units or until completion if less time is required. At this stage, the shortest Job in the waiting queue will be the next to process.

- Round Robin with a time quantum $= 50$ time units
  - Jobs process initially on a first-in first-out basis dependant upon arrival time into the waiting queue. Once a job begins processing, it will run for an interval of 50 time units or until completion if less time is required. If a job has not completed at this stage, it will move to the back of the waiting queue, behind any Jobs which may have arrived while the current Job was processing.

- Round Robin with a time quantum $= 75$ time units
  - Jobs process initially on a first-in first-out basis dependant upon arrival time into the waiting queue. Once a job begins processing, it will run for an interval of 75 time units or until completion if less time is required. If a job has not completed at this stage, it will move to the back of the waiting queue, behind any Jobs which may have arrived while the current Job was processing.

No two Jobs run simultaneously. That is, the scheduler can only run a maximum of 1 Job at any given instance regardless of whichever algorithm is being used.

Each algorithm will output three key pieces of information: average turn around time, average response time, and the number of context switches that occurred.

- Turn around time
  - The time it takes a Job to finish processing after arriving in the waiting queue.

- Response time
  - The time it takes a job to initially start processing after arriving in the waiting queue.

- Context Switch
  - Occurs when a Job is loaded into the processor. (A new job begins processing)

Results:

The following output after running the application for 1000 trials is displayed in the following table

Job Set #1						
	FCFS	SJF	SJFP	RR with Q=50	RR with Q=75
Avg. TurnAround Time	171.727	171.727	171.672	192.627	195.866
Avg. Response Time	21.434	21.434	21.241	11.351	14.491
Num. context switch	1000	1000	1001	1747	1581

Job Set #2						
	FCFS	SJF	SJFP	RR with Q=50	RR with Q=75
Avg. TurnAround Time	109.585	109.036	101.334	112.782	112.634
Avg. Response Time	24.223	23.674	9.423	7.329	11.837
Num. context switch	1000	1000	1134	1426	1258

Job Set #3						
	FCFS	SJF	SJFP	RR with Q=50	RR with Q=75
Avg. TurnAround Time	25281.76	19565.487	19531.097	39905.275	39591.396
Avg. Response Time	25073.066	19356.793	19315.677	8166.939	11263.711
Num. context switch	1000	1000	1138	4623	3361

(Fig 1.1)

Analysis:

Job set #1

First Come First Serve (FCFS) and Shortest Job First (SJF)

The first thing that is evident when observing the first Job set is that both the First Come First Serve algorithm and the Shortest Job First algorithm share the exact same output. They both have the exact same number of context switches as there are number of Jobs in the set. This is expected, both algorithms run each Job to completion once processing has begun (i.e. No context switching besides initially loading each Job).

FCFS and SJF also share the same average turnaround time as well as average response time. This is due to the nature of the gaussian distribution for arrival times of the Jobs. As jobs are so spaced out in their arrival times, there is no instance where there are two or more Jobs waiting for a third Job to process. This negates the potential performance benefits that SJF may have over FCFS. If only a single Job arrives and is waiting for another process to finish to completion, there will be no difference in performance between the two algorithms as it is guaranteed to be the next Job to run. If it were the case that there were two Jobs waiting, it would be uncertain as to which one would end up running in SJF as either of them may have a shorter job length, and of course in FCFS the first job to arrive will always run next.

Shortest Job First with Preemption (SJFP)

In the shortest job first with pre-emption algorithm there is only 1 more context switch than the number of Jobs in the set. This is because, in the rare case that the distribution allows two jobs to be in the ready queue at the same time, the newest job would have to have a smaller job length than the job that was already running. This is unlikely to happened, not only because of the jobs being created under the same conditions but also because the job that ran first has already had its time decreased due previous execution.

The average turn around time and average response time are both ever so slightly smaller than those seen in FCFS and SJF due to the minimal performance benefit of that 1 preemption.

Round Robin (RR)

Round robin turnaround time (in both cases with different quantum times) is slightly higher than the previous three algorithms. This is due to the fact that each arrived job must share the processor with every other arrived jobs.

The response time is reduced almost in half compared with the previous three algorithms. This is due to each arrived job not having to wait for a previous job to completely finish processing before getting a chance to do some processing itself.

The Round Robin with a smaller quantum time has a slightly better response time than the Round Robin with a bigger quantum time. This makes sense, as Jobs waiting for processing have to wait for a smaller time slice for a chance to initially process.

Job Set #2

First Come First Serve (FCFS) and Shortest Job First (SJF)

Similar to the results seen in the job set #1, we can observe that the output for these two algorithms are nearly identical. The difference between the Job set #2 and #1 is that the SJF has a slightly better response time and turnaround time than FCFS within Job set #2. This is likely because there actually happened to be at least one instance where two or more Jobs were waiting for another Job to finish processing.

These two algorithms have a smaller average turn around time in job set #2 than they do in job set #1. This is due to the nature of job set #2 having a greater probability of having small jobs than large jobs. This means that, if jobs are smaller, than there will naturally be less time between their arrival time and finish time in algorithms where Jobs run to completion once processing has begun.

Response time is comparable in this job set with Job set #1.

Shortest Job First with Preemption (SJFP)

This Algorithm we can see that the response time has an abrupt decrease due to the context switching because jobs will run within a time slice and check if a shorter job is ready to process so it can switch it. This will result in the jobs running more frequently and finish processing much faster, as if a short job arrives while a large job is processing it will have to wait for it to finish first, but in this case the job just has wait for a time slice and than switch with the shorter job in the queue.
Also the context switching happens more often.

Round Robin (RR)

For Round Robin algorithm in this job set we can observe that the response time is lower than the other 2 job sets. Also, the response time is lower when run with the 50 time quantum then 75 because the jobs will run for a shorter time and the job context will switch more often whereas with a time quantum of 75 the job will run for a longer time and also makes room for possibilities that the job will run to completions and resulting in fewer context switching.

Job set #3

In job set #3 the average response time, turn-around time and number of context switches are increased drastically over all algorithms compared to the last two job sets. The cause of this is that 80% of the jobs are large jobs which means longer waiting time for FCFS, SJF and the jobs take more time to run. For Round Robin in both cases the context switches are much larger because of the time the job takes to run so that mean the job runs for more than once.

Conclusion:

It is our opinion that the best overall algorithm across all job sets is Shortest Job first with preemption. We have come to this conclusion based on the fact that across all three Job sets, SJFP has the smallest turnaround time, making it the most accommodating and fair in all tested conditions. Its response time is always comparable if not smaller than the response times of FCFS and SJF algorithms across all Job sets. Its response time is beaten only by round robin by a factor of ~2 in the first and third question sets, however in the second question set round robin's response time is comparable to SJFP's response time. As far as number of context switches, SJFP has slightly more than both FCFS and SJF, however SJFP has significantly less context switching than round robin, which is much preferable as there will be much less overhead and wasted resources.