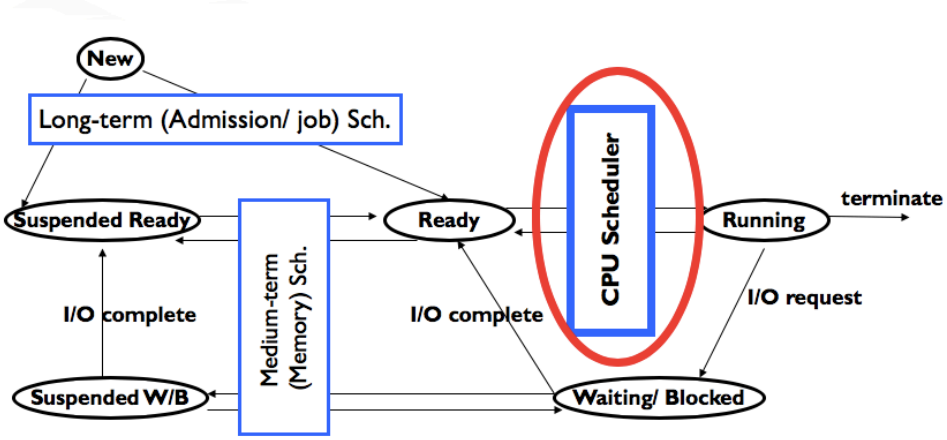
Abstract

For this assignment, the instructor designed an emulation framework that allows the simulation of processes. Students were asked to implement three CPU scheduling policies (First Come First Serve [FCFS], Shortest Job First [SJF], and Round Robin [RR]) and then evaluate and compare performance when paired with different memory management techniques (Best-Fit, Worst-Fit, and Paging).

Introduction

CPU scheduling is the sharing of the CPU among ready processes. Due to different needs, processes may react differently to different scheduling strategies. They may be broken into two large categories: CPU-bound processes and I/O bound processes. The former is characterized by long CPU bursts and few I/O operations while the latter typically has short CPU bursts and performs more I/O operations than computations. The graph below shows the CPU Scheduler in the State Transition Diagram of a process. One may observe from it how a process may move among schedulers and be put on or removed from the CPU.



Several metrics are defined for evaluating CPU scheduling strategies. These include:

Throughput: N/T

where N = Number of Completed Processes and T = Observation time

Response Time: Tf – Ta

where Tf = Time a process first gets the CPU and Ta = Arrival time

Turnaround Time: Tc – Ta

where Tc = Time a process takes to complete and Ta = Arrival time

Waiting Time: Tw

where Tw = Total time spent by a process in the ready queue

The objective of all CPU scheduling policies is to minimize turnaround time and response time while maximizing throughput. The different policies aim to achieve this thru different means and each has its own benefits and drawbacks.

For these processes, the CPU generates logical addresses that correspond to where a program exists in the memory. These addresses are mapped to corresponding physical addresses. Consequently, a program may be viewed as whole in the logical space while it actually exists in pieces in the physical space. This is done to simplify memory management so the programmer may view their program as an infinite contiguous flat space starting at Address 0.

Body

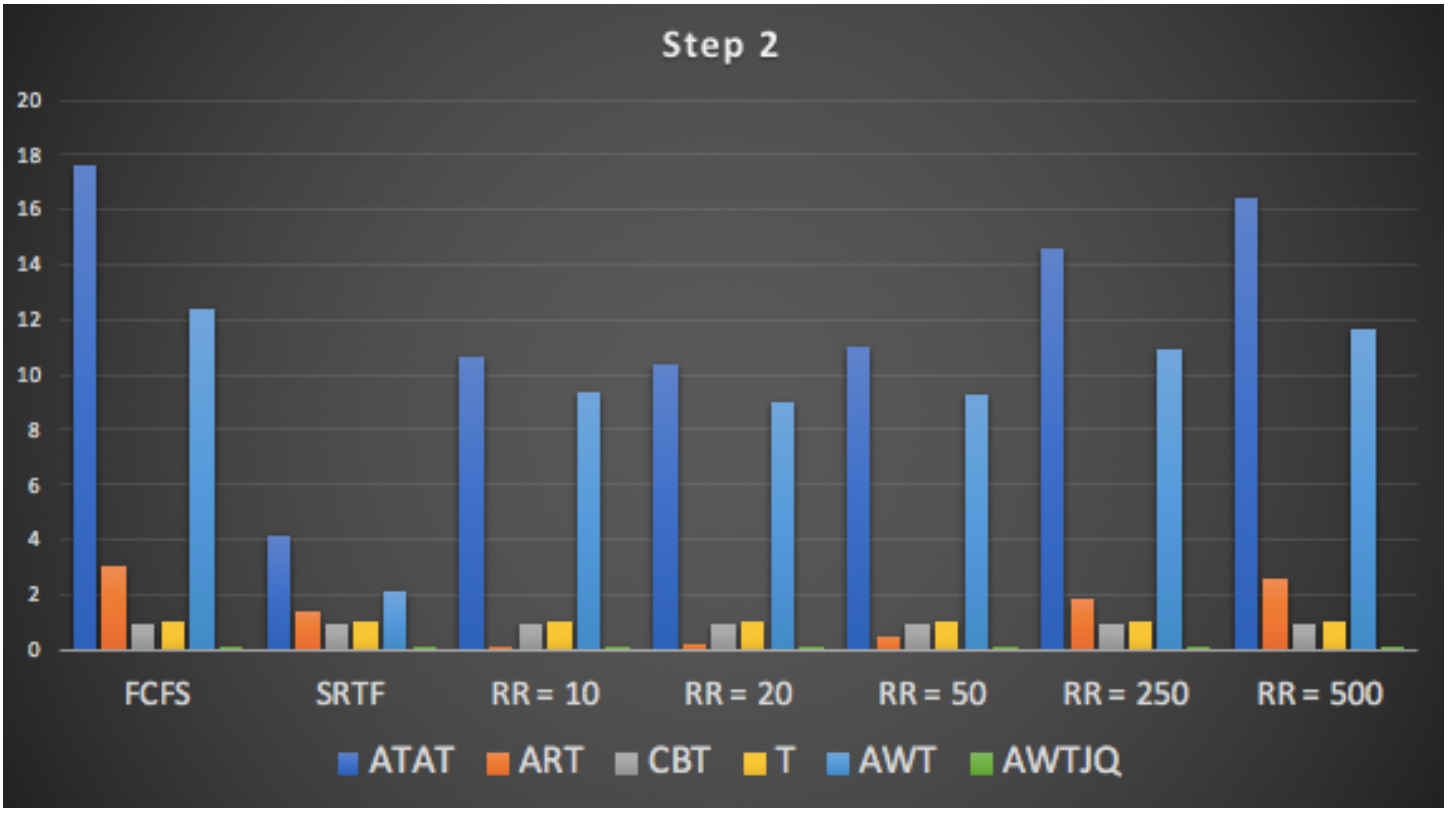
To evaluate the code for each policy, the code was instrumented to collect for each policy the average turnaround time **(TAT)**, the average response time **(RT)**, the CPU Busy time **(CBT)**, the throughput **(T)**, the average waiting time in ready state **(AWT)**, and average waiting time in the job queue **(AWTJQ).** The results for each step of the lab were as follows:

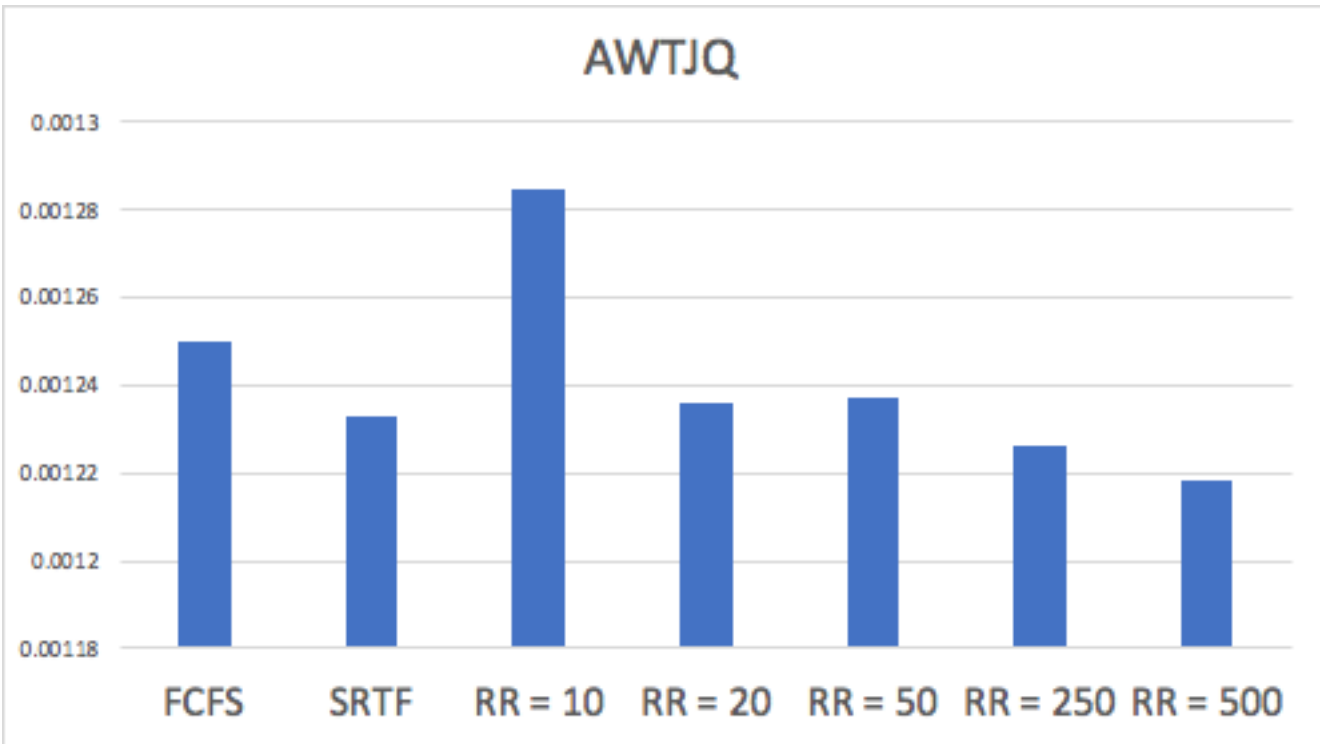
Step 1&2: Collect all metrics for FCFS, SRTF, and RR using processesmanagement2.c ASIS

Note: Here, the program assumes infinite memory, so values are optimal. Can you predict AWTJQ?

➔ Yes, it appears AWTJQ approaches 0 with infinite memory.

Scheduling Policy	Quantum	Completed Processes	ATAT	ART	CBT	T	AWT	AWTJQ
FCFS		229	17.65012	3.048353	0.971127	0.98753	12.43688	0.00125
SRTF		243	4.166786	1.420822	0.970899	1.047935	2.096872	0.001233
RR = 10	10	235	10.68799	0.104989	0.965657	1.013667	9.393024	0.001285
RR = 20	20	235	10.40861	0.195567	0.969802	1.01374	9.005154	0.001236
RR = 50	50	235	11.0099	0.46653	0.976543	1.013663	9.250038	0.001237
RR = 250	250	228	14.5907	1.863946	0.980028	0.983456	10.92357	0.001226
RR = 500	500	228	16.39048	2.630647	0.980319	0.982777	11.68181	0.001218

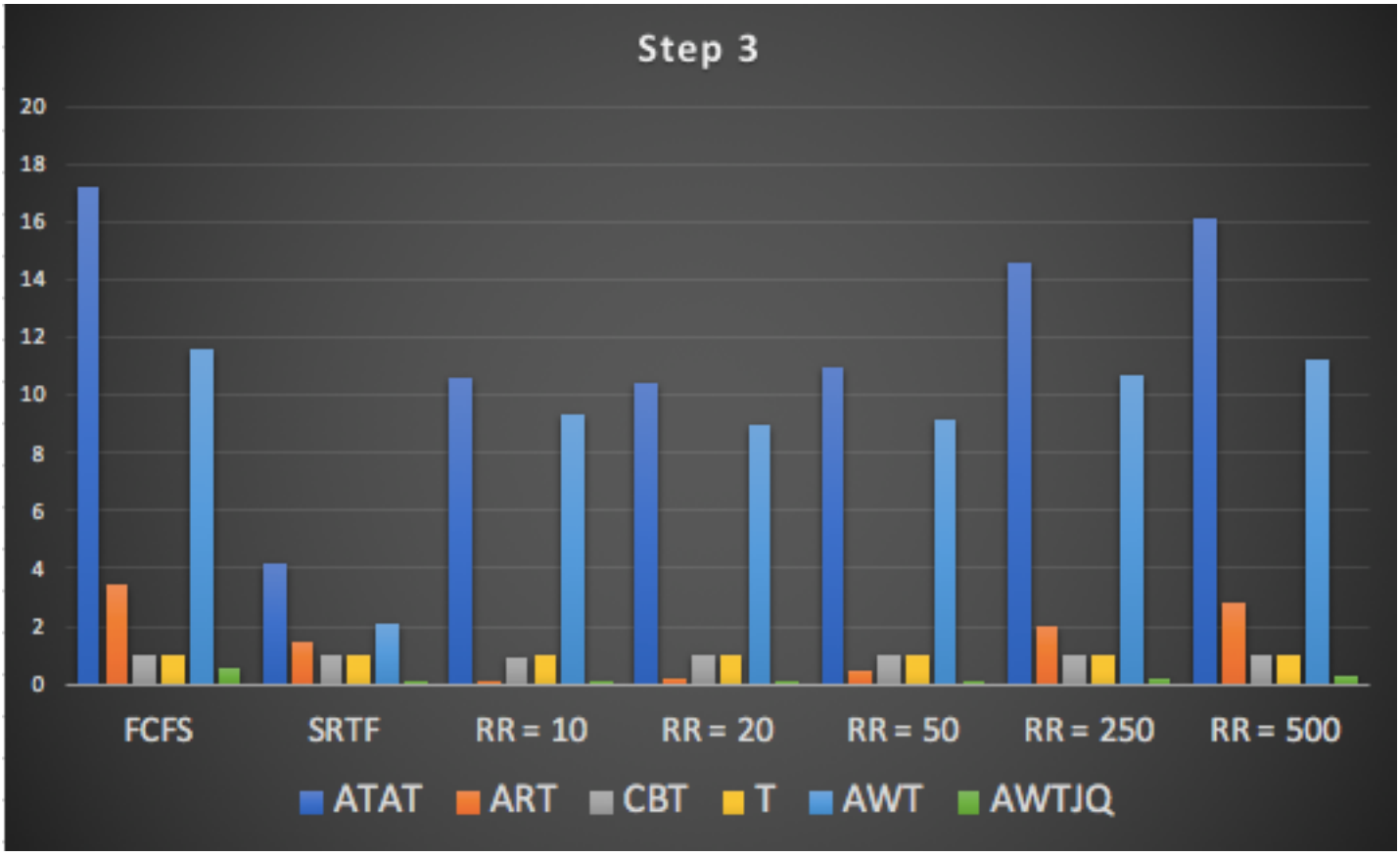




Here, the program assumes infinite memory and therefore these values are optimal. AWTJQ varied little among the different management strategies.

Step 3: Implement OMAP and collect all metrics for FCFS, SRTF, and RR

Scheduling Policy	Quantum	Completed Processes	ATAT	ART	CBT	T	AWT	AWTJQ
FCFS		229	17.20165	3.41719	0.971061	0.987463	11.61497	0.540206
SRTF		243	4.170913	1.423507	0.970804	1.047833	2.102759	0.001235
RR = 10	10	235	10.58643	0.103594	0.966926	1.013731	9.293999	0.001223
RR = 20	20	235	10.41724	0.194444	0.969745	1.013681	9.012198	0.001246
RR = 50	50	235	10.94845	0.462217	0.97674	1.013643	9.191484	0.001208
RR = 250	250	229	14.53461	1.991415	0.980743	0.987749	10.72875	0.167264
RR = 500	500	228	16.14576	2.837756	0.980221	0.983153	11.2185	0.299791

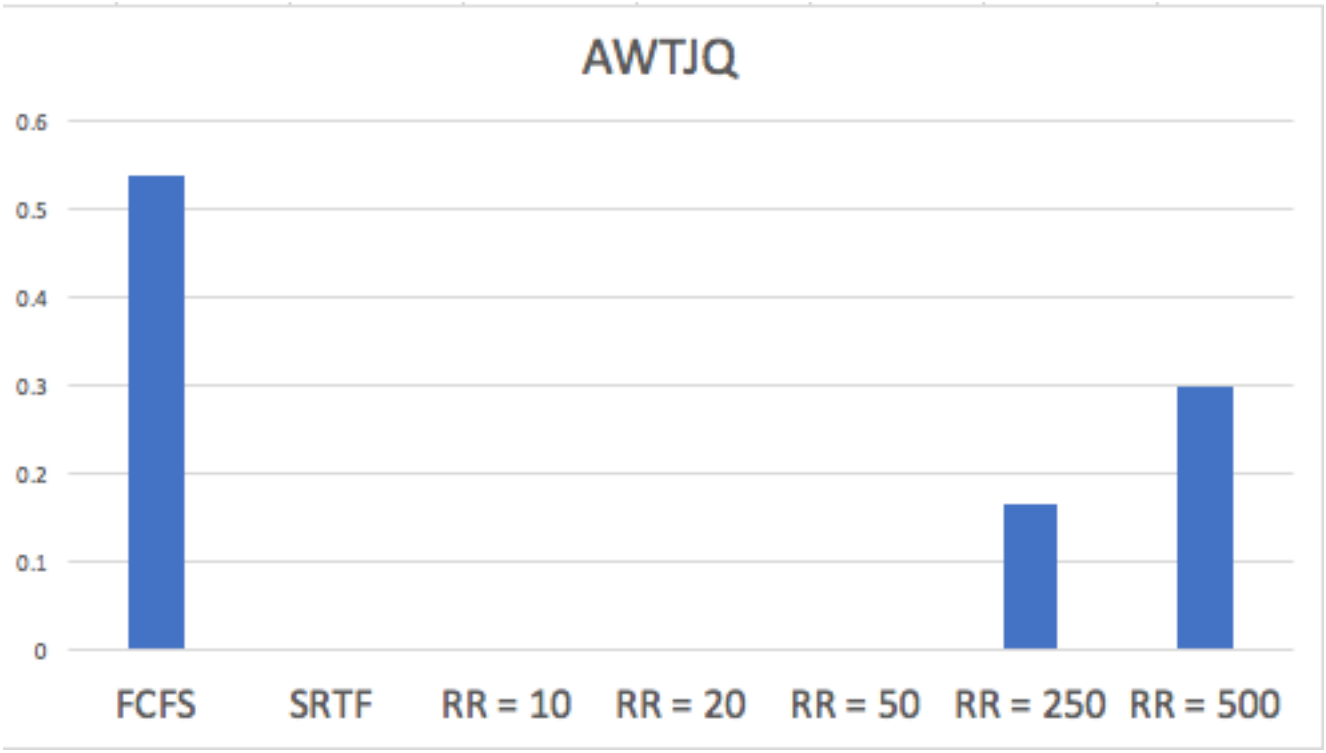


Blocks of available memory are referred to as "holes"; holes of various sizes are scattered throughout memory. When a process arrives, it is allocated a hole large enough to accommodate it (and this hole is chosen based on the process used for selection). There are several ways to satisfy a request of size n from a list of free holes. Including:

Best-Fit: where the smallest hole that is big enough is allocated

Worst-Fit: where the largest hole available is allocated

Both methods must search the entire list of holes (unless ordered by size) but Best-Fit is better than Worst-Fit in terms of speed and storage utilization. Here, Optimal Memory Allocation Policy was implemented. Memory is managed without worrying about the location of processes or the free memory blocks. These results reflect the "upper bound" for performance.

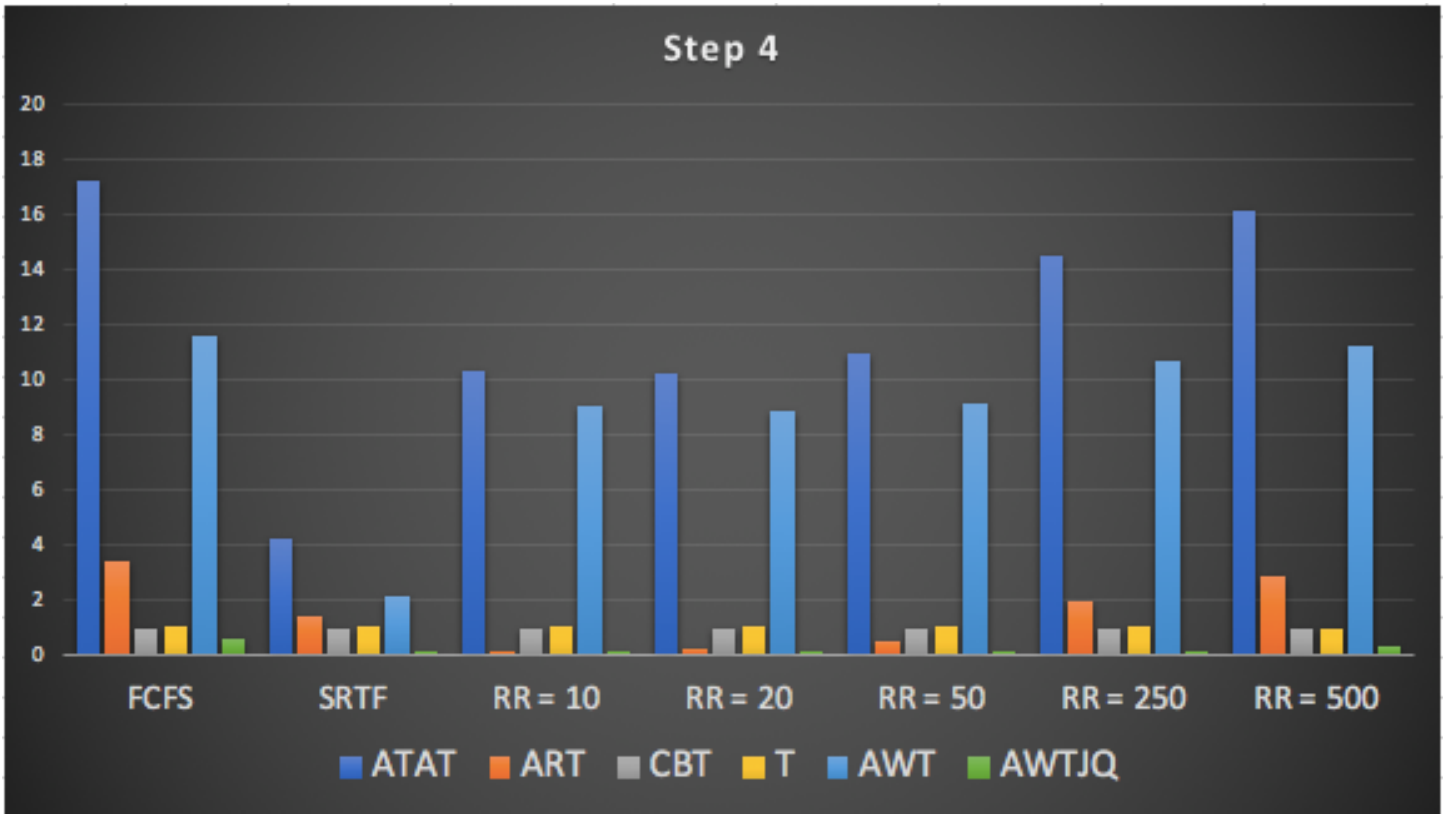


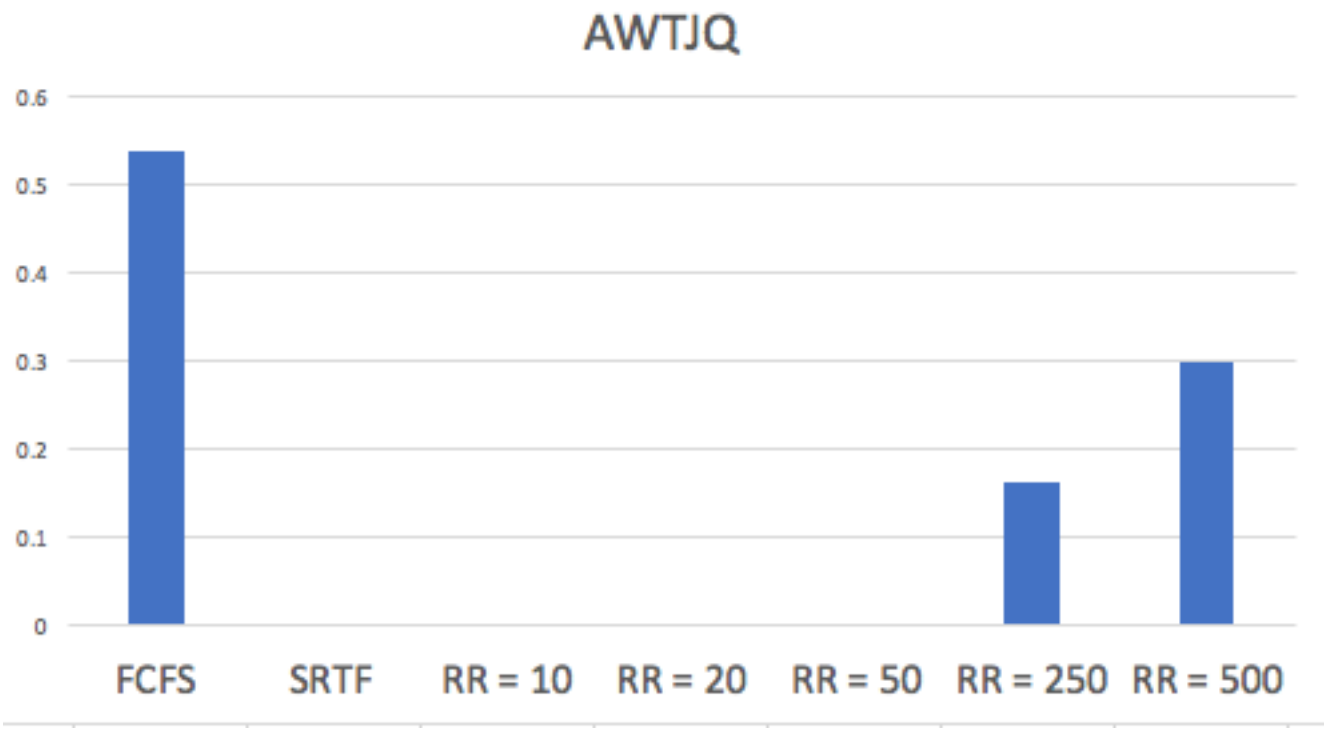
Here, AWTJQ showed the most variation for FCFS and RR with high quantum. This is due to RR behaving like FCFS in higher quantum and is expected. The non-preemptive nature of FCFS left many processes in the Job Queue waiting.

Step 4: Implement Paging and collect all metrics for FCFS, SRTF, and RR

For (256) Page Size:

Scheduling Policy	Quantum	Completed Processes	ATAT	ART	CBT	T	AWT	AWTJQ
FCFS		229	17.18799	3.411891	0.971212	0.987616	11.60685	0.540087
SRTF		243	4.1762	1.415187	0.970923	1.047961	2.100902	0.001145
RR = 10	10	235	10.3402	0.10121	0.968665	1.013764	9.054581	0.001151
RR = 20	20	235	10.21671	0.191354	0.971585	1.013825	8.819482	0.001085
RR = 50	50	235	10.90069	0.462064	0.977231	1.013705	9.148438	0.001111
RR = 250	250	229	14.51212	1.972478	0.980306	0.987139	10.69508	0.162027
RR = 500	500	228	16.13685	2.823261	0.98106	0.983397	11.21111	0.299595


Step 4
20
18
16
14
12
10
8
6
4
2
0
FCFS
SRTF
RR = 10
RR = 20
RR = 50
RR = 250
RR = 500
ATAT
ART
CBT
T
AWT
AWTJQ


AWTJQ
0.6
0.5
0.4
0.3
0.2
0.1
0
FCFS
SRTF
RR = 10
RR = 20
RR = 50
RR = 250
RR = 500

For (8192) Page Size:

Scheduling Policy	Quantum	Completed Processes	ATAT	ART	CBT	T	AWT	AWTJQ
FCFS		191	14.08771	3.887929	0.619044	0.644372	8.184058	1.867947
SRTF		243	4.169554	1.422272	0.970841	1.047872	2.1015	0.001224
RR = 10	10	219	10.35219	0.657114	0.671839	0.717301	8.481959	0.565166
RR = 20	20	206	10.06259	0.551339	0.81608	0.860324	8.266705	0.37635
RR = 50	50	207	10.49883	0.795419	0.693677	0.73278	8.385067	0.383595
RR = 250	250	204	13.20904	2.982514	0.413837	0.42912	8.478267	1.62599
RR = 500	500	193	13.48767	3.196843	0.515916	0.537484	8.310865	1.420342