Matthew Stevens / Jason Alonzo

CSCE 313

Dr. Tyagi

30 March 2017

MP5 Report:

Potential Changes:

We believe that adding a promotion policy would indeed prevent starvation, and a dynamic policy for changing time quantums would be useful for increasing fairness. These would allow for processes to execute differently based on their size. However, this would need to be done in a very thorough way, for it would need to keep processes from starvation without granting too much of a time quantum to bigger processes. Another thing to think about is the size of the context switch itself. Our program is quite bulky, as it is loaded with many if statements, so if we were able to get the simplest form, or a very simple form, of a context switch where it does not need to take very much time to switch, then we are able to reduce the overall time as well. Perhaps we could integrate a MLFQ that will run the first queue based on the size of the data of each process, instead of the number of processes. This could lead to micro-managing that potentially could save more run time.

Findings:

We found that SRTF had a better wait time than FCFS. This is because processes that have shorter run-times do not have to wait on the longer processes to complete. We found that RR was interesting because the average times went down for larger quantums, but this may be only for our case, because the worst case scenario would be when a quantum of 32 seconds runs a process at clock arrival 0 and the process takes 1 second, the wait-time would skyrocket, so for the larger quantums, smaller processes are disadvantageous. On the other hand, for smaller quantums, smaller processes are advantageous because the queue gets cleared up very quickly. Lastly, we found that MLFQ had a significantly worse run time when the time quantum would increase. This is probably due to the fact that our processes are very small, so for quantums 24 and 48 and such, there was too much waiting in between processes.

FCFS

The first table displays the results from running our FCFS scheduling algorithm on the Task.txt file. (Note: I changed the arrival time of P1 from 2 to 0.)

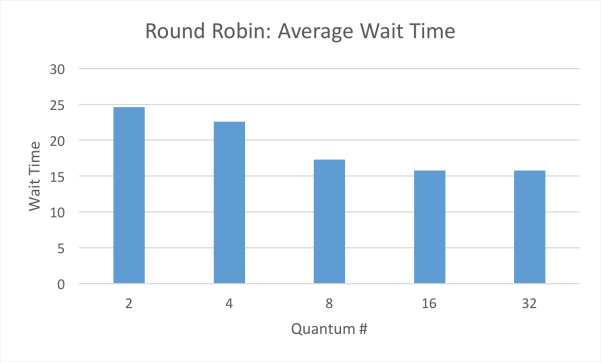
FCFS	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Wait Time	0	0	9	3	13	15	31	37	34	23
Avg. Waiting TIme	16.5									
Avg. Response Time	24.5									

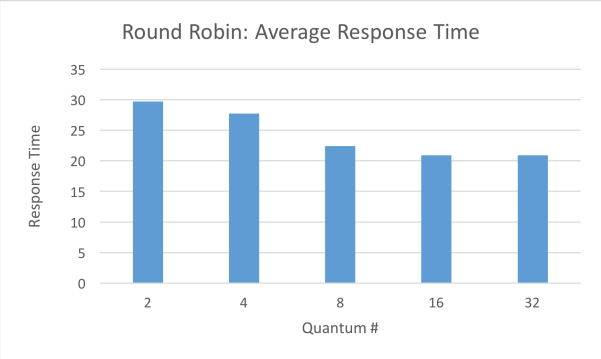
SRTF: (*note: SRTF was broken so I manually calculated this table.)

SRTF	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Wait Time	0	0	3	24	13	35	5	0	13	25
Response Time	3	4	8	30	18	43	9	2	18	33
Avg. Waiting TIme	11.8									
Avg. Response Time	16.8									

Round Robin

The data below was gathered from running our Round Robin scheduling algorithm over the original Task.txt file.





RR, quantum = 2	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Avg. Waiting Time	24.6									
Avg. Response Time	29.7									

RR, quantum = 4	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Avg. Waiting Time	22.6									
Avg. Response Time	27.7									

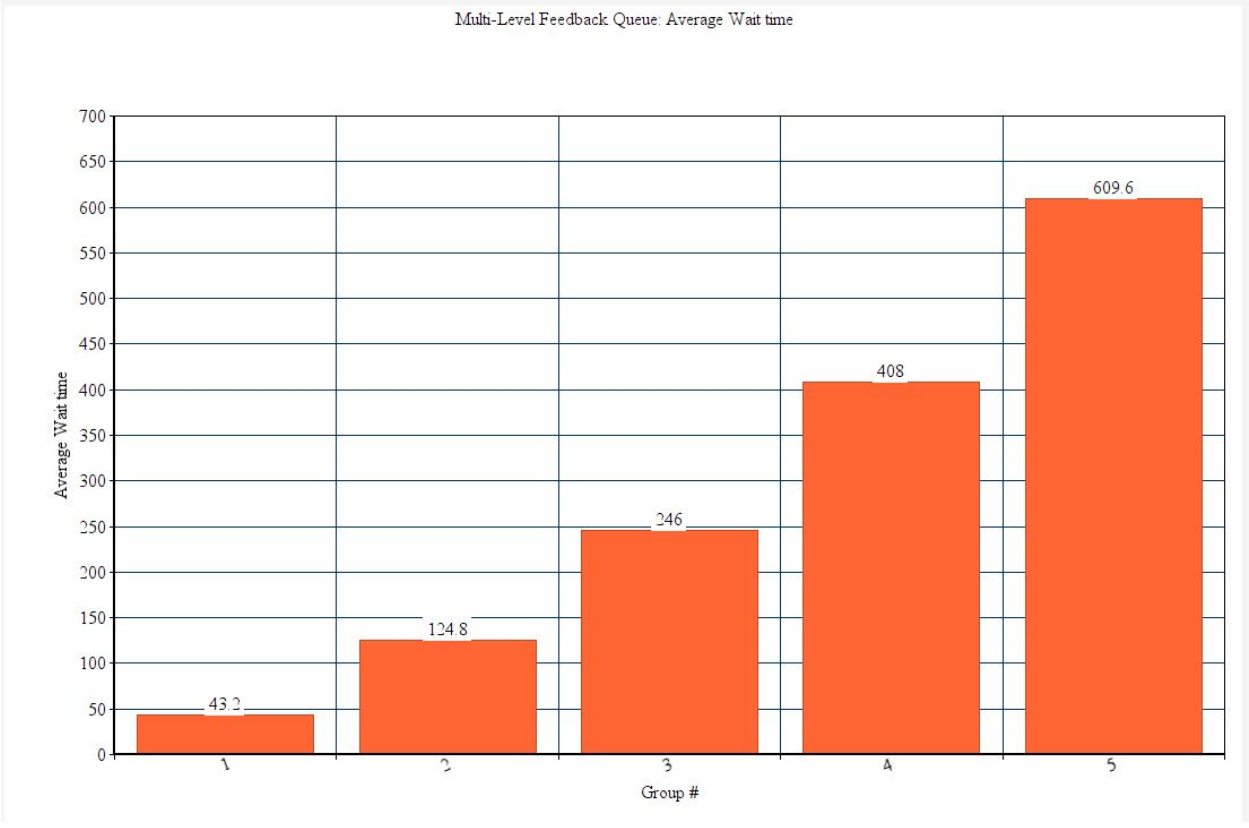
RR, quantum = 8	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Avg. Waiting Time	17.3									
Avg. Response Time	22.4									

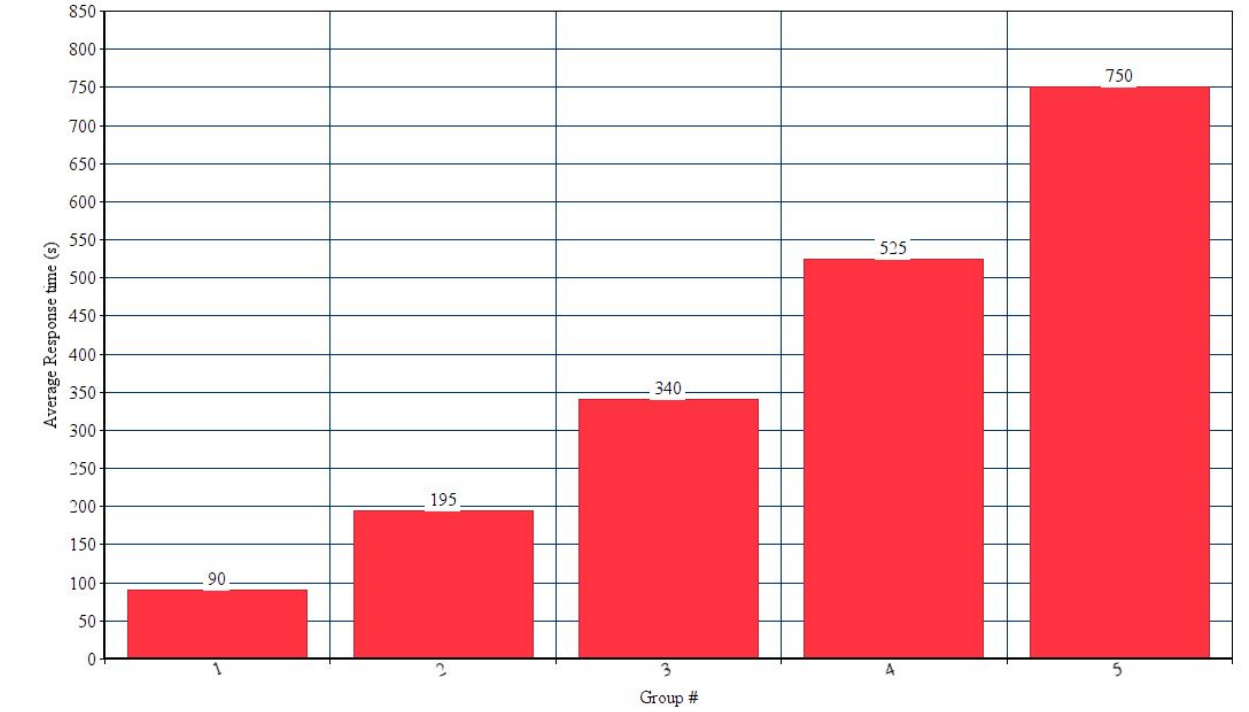
RR, quantum = 16	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Avg. Waiting Time	15.8									
Avg. Response Time	20.9									

RR, quantum = 32	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Avg. Waiting Time	15.8									
Avg. Response Time	20.9									

MLFQ:

Group 1:	Level 1 Quantum
	Level 2 Quantum
	Level 3 Quantum
Group 2:	Level 1 Quantum
	Level 2 Quantum
	Level 3 Quantum
Group 3:	Level 1 Quantum
	Level 2 Quantum
	Level 3 Quantum
Group 4:	Level 1 Quantum
	Level 2 Quantum
	Level 3 Quantum
Group 5:	Level 1 Quantum
	Level 2 Quantum
	Level 3 Quantum


Multi-Level Feedback Queue: Average Wait time
Average Wait time
Group #
43.2
124.8
246
408
609.6
1
2
3
4
5


Multi-Level Feedback Queue: Average Response time
Average Response time (s)
Group #
90
195
340
525
750
1
2
3
4
5

MLFQ and RR quantums were changed via manually overwriting the quantums in putty.