Gilberto Ramirez
vwz745

CPU Scheduling Report

Compile: make
Run all test cases with input2.txt: make run

File_Read_thread
This thread opened the specified input file and got a line from it until EOF. If it was proc, it malloced memory for the PCB structure and initialized all its members by tokenizing the line using strtok(). Once the proc was initialized, it was added into the ready queue and sem_post was called to let the sched thread that a job was ready to be executed.

If Sleep was encountered, then we simply slept for the time specified. And if STOP was received, we broke from the loop and exited the thread.

System_io_thread
This thread waited for the io semaphore to post and once it did, it dequeued the first job in the io queue. It then slept for io_burst time and added it back into the ready queue. Once everything was done and all queues were empty, it broke from the infinite loop and joined back with the main thread.

Scheduling_thread
In the scheduling, I first check whether or not the alg chosen was RR or not because the other 3 algs are pretty much the same, just differs in how we dequeue the process. So if it was one of these 3, we waited for the semaphore and if we got in CS, start time for CPU UTIL purposes. We then got the next process according to the specified algorithm and ended the read queue waiting member and calculated the time it waited for this instance and added it to the total. The cpu burst executes for the appropriate time and index was incremented. If it was the termination case, then we ended the throughput timing member and set the time from when it first started to when it exited and then added it to the finished list. If not, then it was added to the io queue and the CPU UTIL time was ended and added to the total time in cpu.

For RR, it was pretty much the same deal just that we had to check whether our cpu burst was larger/smaller than the quantum. If it was smaller then we just waited that amount, however if it was bigger then we had to wait quantum time and subtract the cpu burst by the quantum and then added back into the ready queue.

For all cases where we had to change the ready or queue lists, I had then been covered by semaphores so no more than 1 thread can get into those structures at a time. Also, instead of passing many parameters or creating structs for params, I simply had most variables as global to make it easier for threads to access data.

Results
I ran my program with the given input2.txt with all test cases and for the most part it is very much similar to the expected results. It matches most with the FIFO and RR runs.

However when we look at the PR runs the avg time in the waiting queue is a bit off. I double checked my alg and it was working as expected, but when I went based on the min pr instead of max, it would match the expected results. I am not sure if this was an error on the expected results, but I believe my implementation is correct.

Also the SJF turnaround times are a bit off as well, the way I set it up was that it would choose the smallest job for the pcb->cpuburst[cpu_index] and it was working just as expected, so I am not really sure why my results are a bit different. I am thinking the professor used a slightly different alg, such perhaps he took the total of the entire cpu bursts and chose the min from that, or just went off of total cpu bursts.But other than those 2 differences, everything else looked fine so I believe I did this correctly.

My output vs expected output

Input File Name : input2.txt	inputs/input2.txt
CPU Scheduling Alg : FIFO	FIFO
CPU utilization : 85.408758%	83.815%
Throughput : 0.014628	0.014 processes / ms
Avg. Turnaround time : 235.384064ms	234.2ms
Avg. Waiting time in R queue : 46.659464ms	45.6ms
---	---
Input File Name : input2.txt	inputs/input2.txt
CPU Scheduling Alg : SJF	SJF
CPU utilization : 78.497760%	77.54%
Throughput : 0.013451	0.013 processes / ms
Avg. Turnaround time : 245.308703ms	262.8ms
Avg. Waiting time in R queue : 42.588265ms	48.8ms
---	---
Input File Name : input2.txt	inputs/input2.txt
CPU Scheduling Alg : PR	PR
CPU utilization : 82.931310%	81.69%
Throughput : 0.014213	0.014 processes / ms
Avg. Turnaround time : 219.189462ms	228.6ms
Avg. Waiting time in R queue : 44.530593ms	52.4ms
---	---
Input File Name : input2.txt	inputs/input2.txt
CPU Scheduling Alg : RR quantum: 10	RR quantum: 10
CPU utilization : 80.990570%	81.69%
Throughput : 0.014187	0.014 processes / ms
Avg. Turnaround time : 237.817837ms	227.0ms
Avg. Waiting time in R queue : 53.327356ms	58.4ms
---	---
Input File Name : input2.txt	inputs/input2.txt
CPU Scheduling Alg : RR quantum: 30	RR quantum: 30
CPU utilization : 80.629680%	79.67%
Throughput : 0.013804	0.013 processes / ms
Avg. Turnaround time : 237.640571ms	234.2ms
Avg. Waiting time in R queue : 38.588319ms	37.8ms
---	---
Input File Name : input2.txt	inputs/input2.txt
CPU Scheduling Alg : RR quantum: 60	RR quantum: 60
CPU utilization : 85.388159%	84.302%
Throughput : 0.014631	0.014 processes / ms
Avg. Turnaround time : 235.263395ms	234.4ms
Avg. Waiting time in R queue : 46.592050ms	45.2ms