BIRZEIT UNIVERSITY
Electrical and Computer Engineering Department
ENCS3390 Operating Systems 2nd Semester 2020/2021
Project

Simulation of CPU Scheduling

The task is to simulate the process of loading a new process and adding it to the ready queue, short term scheduler and the IO waiting queue in a multi-processor system.

Input file

Delay	PID	CPUBurst	IOburst	CPUBurst	IOBurst …..			
0	1	20	40	100	20	40		
2	2	100	2	40	200	100		
0	3	120	200	40	30	50	30	50

First step is to generate the input file , each line contains information about one process , the Delay tells us the arrival time relative to the first process (in the example process one arrive at 0 delay, process 2 arrives 2 time units later process 3 arrives 0 time after process2 i.e. process 2 and 3 arrive at the same time). PID: is the unique process id
Then a sequence of CPU Bursts and IO bursts.

The generated file should have a random number of processes between 20 and 100, and a random number of cpu and io bursts (between 5 and 20) , each burst should have the length of (2 to 120 time units, with exact durations defined randomly)

Please note the file should be saved and your program must have the option to choose what file to run so you can run the exact input multiple times and with different algorithms if needed

Also you need to set the number of processors as an input for your code (you should be able to run the same input file using the same algorithm using different number of processors)

Structure
You code should contain at least three types of threads
Thread one: adds new processes to the ready queue as they arrive

Thread two: the scheduler moving processes to running state and back to ready **you need a thread per processor from this type**

Thread three: the IO queue when a process has an io burst it will be set to the waiting state and placed on the io waiting queue, this thread will simulate the IO waiting queue and notify the scheduler when a process has completed the IO request (to have it moved to the ready queue).

Please note that to synchronize all threads you need a shared counter to represent the time units since simulation starts, this can be managed by one of the threads and shared with the others

Please note that any resource you need to be shared between threads require synchronization.

Each process will have its ready Queue, how to move processes from the global ready queue to each processor's ready queue is up to you

Output

You need to run the simulation with different scheduling algorithms
RR , SRTF (assume context switch time is 0.1, always)

Your output for each simulation should contain (for each algorithm):

- CPU utilization
- Average response time
- Throughput

It is preferred to have GUI for your program that also can show the different queues state at any chosen time

Groups

You can work in groups of three students (more than that is not accepted).
You need the permission of the instructor to work in smaller groups.

Good luck.