5.

(1) Priority scheduling

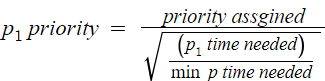
Looking into further expanding our project. One first thing we would implement in addition is priority scheduling. It is one of the most common scheduling algorithms in real world processing. Priority-based scheduling takes priority into consideration when deciding the order which process to be executed, and even when to preempt a process for higher priority ones.

(2) Multi-core processing

One of the most important upgrades for the simulator would be multiple-processor processing. Most of the hardware on market and software being designed are in favor of multiple-processor processing. Single core CPUs are hardly seen anymore. Programmers are used to taking advantages of multi-core processing to obtain a considerable efficiency improvement. Therefore, implementing multiple-processor processing can make our simulator better model real world operating systems.

(3) Long-term scheduler and job queue

We’ve included two queues in our simulator, ready queue and I/O Queue (device queue). Processes in ready queue are allocated by the short-term scheduler (or the CPU scheduler). In real world process scheduling. There is also job queue which the processes are allocated by the long-term scheduler. Ready Queue only includes the processes currently in memory. All the processes are at the ready state and waiting for execution. For job queue, processes include the ones being and will be executed. In other worlds, job queue keeps all the processes in the system. A long-term scheduler is the media between job queue and ready queue. Although what long-term scheduler does is just moving processes from job queue to ready queue, it is particularly crucial as it controls the whole process and the degree of multi-programming from a higher level than short-term scheduler. When a process is added to job queue, long-term scheduler checks for the amounts of workload of CPU-bound and I/O-bound processes each process implies. After that, long-term scheduler decides which job is to be submitted to short-term scheduler and occupy the computing power.

6. (Figure 1, calculation for revising priority)

By discounting the priority with regards to the total time a process needs, the heuristics is adding SJF’s idea to priority-based scheduling. The advantage of this is to give those shorter process with adequate level of priority to get executed without waiting for some long and high priority process by looking at all the process currently in place. This speed-up shows when we have a lot of processes, and very few of them is much longer than average and very long processes exist. The disadvantage of it is that discounting priority with the amount of time it needs will require long and extremely significant process to have a much higher priority.