

Gordon Lu

Professor Daniel Mosse

CS 1550

06 Feb. 2020

CS 1550 Project 1 Report:

By implementing a priority with a semaphore, we allow processes with higher priorities based on the **task_nice** values to be allowed onto the critical section and be scheduled onto the CPU. The benefit of this approach is that processes will be allowed into the critical section based on the **task_nice** value and accordingly be moved to the front on the queue, rather than solely on waiting time. The downsides of this approach lie in starvation and deadlock. When processes can be preempted, allowing processes onto the critical section becomes more troublesome. When two processes are calling down and up multiple times, if one process gets preempted, the other processes will end up waiting indefinitely, until the other process finishes up or down, and thus cannot make further progress, until the process is no longer suspended. As for starvation, if processes with higher **task_nice** values are continuously placed onto the semaphore's queue, lower priority processes may never get an opportunity to be on the critical section, and the lower priority processes may never get the chance to be removed from the semaphore queue.