Computer Operating System 66140

Assignment 1, due BEFORE class 9, September 30th

Consider a computer system with 10 processes in the ready queue, numbered from 0 to 9. The burst time of each process is as follows: $t_B^9=55$ms and for $i<9$ $t_B^i=1.5$ms if i is an even one-digit number and $t_B^i=3$ms if i is odd. After executing for the burst time, each process goes into disk service that takes time t_D. In each question below consider three values of t_D equal to 6, 15, 25ms, corresponding to a fast, medium and slow disk, respectively. Assume that the disk has constant response time, regardless of the number of processes it is serving. The context switching time T_s is 1ms.

For each question below find out a) a response time of interactive tasks (tasks with the id number other than 9), b) a slow down of the processor for the computational task (the task numbered 9), which of course becomes infinity if this task is starving, c) the CPU utilization. In each case consider first an abstract case in which there is no context switching time, and then include the context switching time in your considerations.

1. Consider preemptive SJF scheduling for this system (also called Shortest Remaining Time (SRT)), and use only analytical analysis to formulate your answer (3pt).
2. Consider HRRN scheduling (you can use either analytical analysis or emulation) (6pts).
3. Consider RR scheduling with time quantum T_Q=3ms (6pts).

Bonus question worth 5pts: include in your analysis/emulation cases of question 3, in which time quantum, T_Q, is 1ms, 1.5ms and 11ms, and based on these results recommend the best value for T_Q under different criteria (e.g., the CPU utilization, disk utilization, response time and slow down).

Clarifications: The slowdown can be defined as the ratio of the clock time to the execution time received by the computational process. Hence, if you emulate system to T_{end} time and during that time CPU executed the computational process for T_{comp} time, then slowdown=T_{end}/T_{comp}

The response time for interactive processes can be defined as the time from entering the ready queue of the CPU till time of the request for the disk, so, the time when a process is at the disk is not included in the response time.

The response time (like a slowdown), can become infinity if at least one of the interactive processes is starving

Hints: To verify your emulator, consider FCFS algorithm first, can you provide the answer for this case analytically? If not, see the TA during office hours.

Even more helpful could be to implement first the system discussed in class that has only 3 processes in the ready queue. Full data for this system and a sketch of the program for emulation are given in class.