CS4410 - Fall 2008
Homework 1
Due September 9, 2008 11:59PM

Q1. Give three examples of an explicit hardware mechanism that is motivated by specific OS services.

Q2. Can a process transition from waiting for an I/O operation to the terminated state? Why or why not?

Q3. What are the differences between user-level and kernel-level threads? Under what circumstances is one type better than the other? What is the essential cause of the difference in cost between a context switch for kernel-level threads and a switch that occurs between user-level threads?

Q4. Suppose that a process scheduling algorithm favors those processes that have used the least processor time in the recent past. Why will this algorithm favor I/O-bound processes, but not starve CPU-bound processes?

Q5. Define makespan as the total time to complete a set of jobs. An Operating System wants to minimize its makespan for a given set of jobs. Imagine our Operating System has only 2 jobs: A and B. Provide a scenario where running the jobs sequentially will provide better performance (measured by having a smaller makespan) compared to running them in parallel. If such a scenario does not exist, explain why. Otherwise, explain the particulars of jobs A and B and how it performs better in the sequential environment.

Q6. Explain why system calls are needed to set up shared memory between two processes. Does sharing memory between multiple threads of the same process also require system calls to be set up?

Q7. Suppose that we have a single-core, uniprocessor system which supports multiprogramming. At any given time, how many processes can be in the running state in this system?