Homework 2 (due 10/5/2006 at 12:05pm)

via email to ada@cc or hardcopy in class

1. Consider a system running ten I/O-bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context switching overhead is 0.1millisecond and that all processes are long-running tasks.

a) What is the CPU utilization for a round-robin scheduler when the time quantum is 1 millisecond?

b) What is the CPU utilization for a round-robin scheduler when the time quantum is 10 milliseconds?

c) Explain how may the following scheduling criteria conflict in certain cases: I/O device utilization and CPU utilization.

Note: If you are making any assumptions about the behavior of the system scheduler, please state them clearly.

2. Consider the following serializer code:

```
serializer {
  queue a, b, c;
  crowd cr;
  int i = 0; /* initially zero */
  void foo() {
      enqueue(a) until i != 0;
      join_crowd(cr) {
        ... /* critical section */
      }
  }
  void bar() {
      enqueue(b) until a.empty() && i == 0;
      join_crowd(cr) {
        ... /* critical section */
      }
      i = 1;
  }
  void baz() {
      enqueue(c) until i == 1;
      join_crowd(cr) {
        ... /* critical section */
      }
      i = 0;
  }
} // end of serializer
```

Which of the following path expression(s) describe(s) only executions for which the above serializer does not deadlock?

a) path { foo + bar + baz } + bar end *b*) path { bar ; foo ; baz } end *c*) path bar ; { foo } ; baz end
d) path { baz + foo } + bar end *e*) path { bar ; baz } ; foo end

3. What happens if a Solaris user-level thread, as described in Stein and Shah, needs to be preempted while executing a blocking system call? What about a user-level thread running on top of the Psyche kernel?