CS-537 Spring 2004: Quiz

Last name:______________________________ First name:__________________________________

Instructions: Answer all 5 questions. All 5 questions are worth same points.

1. Write one-sentence definition of each of the following terms:

(a)Time Sharing:

(b)Resource:

(c)Race Condition:

(d)Deadlock:

(e)Rendezvous:

2. Which of the following are more like policies, and which are more like mechanisms? **For each answer, circle policy or mechanism.**

(a)The timer interrupt:	Policy / Mechanism
(b)How long a time quantum should be:	Policy / Mechanism
(c)Saving the register state of a process:	Policy / Mechanism
(d)Continuing to run the current process when a disk I/O interrupt occurs:	Policy / Mechanism

3. Assume we run the following code snippet. After waiting for a "long" time, how many processes will be running on the machine, ignoring all other processes except those involved with this code snippet? You can assume that fork() never fails. Try to add a short explaination to your answer.

```
void runMe() {
    for (int i = 0; i < 23; i++) {
        int rc = fork();
        if (rc == 0) {
            while (1);
        }
        else {
            while(1);
        }
    }
}
```

Number of processes running:____

Explaination:

4. Assume the following code snippet, where we have two semaphores, *mutex* and *signal:*

Thread 1	Thread 2
mutex.down();	mutex.down();
if (x > 0)	x++;
signal.up();	signal.up();
mutex.up();	mutex.up();
signal.down();	

We want *mutex* to provide mutual exclusion among the two threads, and for *signal* to provide a way for thread 2 to activate thread 1 when 'x' is greater than 0. What should the initial values of each of the two semaphores be? Assume 'x' is always positive or zero, and that there are only these two threads in the system.

Value of mutex:____
Value of signal: ____
Explaination:

5. For a workload consisting of 10 CPU-bound jobs of varying lengths (half run for 1 second, and the other half for 10 seconds), which policy would result in the lowest total run time for the entire workload? Why?

(a)Shortest Job First
(b)Shortest Time to Completion First
(c)Round Robin with 100 ms quantum
(d)Multi-level Feedback Queue

Your answer:

_