CMSC 412 Midterm #1 (Spring 2002) - Solutions

1.) CPU Scheduling (20 points)

a) Explain why Round-Robin scheduling tends to favor CPU bound processes over I/O bound ones.

```
Each process gets put back at the end of the queue no matter how
much or how little of the quantum was used. I/O bound processes
tend to run for a short period of time and then block which means
they might have to wait in the queue a long time.
```

b) CPU scheduling quanta have remained about the same over the past 20 years, but processors are now about 1,000 times faster. Why haven't CPU scheduling quanta changed?

```
The length of the scheduling quanta is based on the overhead of
context switching a processor and the need to move between
processes within the time of human perception.  The overhead of
context switching due to the need to invalidate caches has
remained relatively constant, and the time of human perception has
also not evolved much in the past 20 years.
```

c) List 4 events that might occur to cause a user process to be context switched off the processor.

```
Timer Interrupt  (time for another process to run)

Blocking to wait for another event (e.g. wait system call)

Process termination

I/O Interrupt
```

2.) (10 points) Explain how a process differs from a thread.

```
Processes have address spaces associated with them.  A collection of
threads shares the same address space.  Processes are always visible and
scheduled by the OS.  Threads may be user space scheduled.
```

3.) (20 points) Given a system that provides binary semaphores (semaphores whose values is either 0 or 1). Show the code to implement counting semaphores using binary semaphores.

$P_{counting}$:

```
P(mutex)
if (sem.count > 0)
      sem.count—-;
      V(mutex)’
else
      P(sem.ready)
      V(mutex)
```

$V_{counting}$:

```
P(mutex)
if (sem.count == 0)
      V(sem.ready)
else
      sem.count++;
V(mutex)
```

4.) (15 Points) Deadlock

a) List the four necessary conditions for deadlock

Mutual Exclusion

Hold and Wait

Non-preemption of resources

Circular Waiting

b) Explain how the Resource-Request (Banker's) algorithm prevents deadlock.

It ensures that there is always a way to meet the resource requirements of the processes in the system (i.e. there exists a safe sequence) without getting into deadlock. When a resource request is given, the system pretends that the request has been satisfied, and verifies that there still exists a safe sequence that allows all existing processes to make their remaining resource requests.

5.) (20 points) Project

a) After you setup a user mode process, do you still need the stack created for kernel mode threads? Explain your answer.

Yes, this stack is used to store the interrupt for system calls, and for the stack records of the kernel functions called as part of processing system calls.

b) What would happen if your _Entry function in libuser.a returns rather than calling Exit.

The user process would attempt to find the return address for the _Entry function on the stack. However, the stack is empty at this point and the processor would generate an exception for a stack out of range. This would result in a GPF which would cause the kernel to terminate the process.

6.) (15 points)

a) Explain the difference between policy and mechanism in an operating system. Given an example of each.

Policy is the set of rules that define what should be allowed. Mechanism is the code that is able to implement a policy. For example, the policy might be student's should be able to read each other's files. The mechanism is that the file system provides both user and group permissions on files and the each student is a different user and that the file protection is set by default to allow only the owner of the file to access it.

b) One of the usability goals for an operating system is "proportionality." Explain what is meant by this term.

Proportionality is the need for an operating system to make the common and safe things easy, yet make the complex (or dangerous) things complex or at least not easy to invoke by mistake. For example, re-formatting the disk should be hard, but deleting a file should be easy.