66-411 Operating System Fundamentals

Test 1: 10/3/94

Name:
Login Name:

This is a open book, open notes exam. You have 50 minutes to answer as many questions as possible. The number in parenthesis at the beginning of each question indicates the number of points given to the questions (and that also indicates approximately how many minutes you should spend on the question). Write all your answers directly on this paper. Make your answers as concise as possible.

Problem	Points Scored	Points Possible
1	______	10
2	______	10
3	______	10
4	______	10
5	______	10
6	______	10
Total		

1. (10 points) Indicate whether each of the statements below is True(**T**) or False(**F**) and explain your answer `briefly`.

(a) A thread and a process are one and the same.

(b) In UNIX, system call signal is similar to a semaphore V operation.

(c) Context switches are done entirely in hardware.

(d) In a fork() operation the original process and the forked process are **absolutely** identical.

(e) Multiprocessing means that you should have a machine which could handle vectors.

(f) Test and Set operations are useful to implement semaphores.

(g) You can read the value of semaphore in your program.

(h) Bakery algorithm (page 171) solves mutual exclusion problem for n processes using busy wait.

(i) Semaphore operations P and V if not used correctly might cause deadlocks.

(j) In monitors the code for mutual exclusion is automatically inserted by the Operating System.

(2a) (5 points) Implementing mutual exclusion by busy wait can be very tricky. Is the following implementation correct? Explain. Need is an array of 2 shared variables.

```
mutexbegin:
need(me) = true;
while (1) {
   if( ! need(other)) break;
   need(me) = false;
  while (1) {
    if (!need(other)) break;
          }
  need(me) = true;
}
mutexend:
need(me) = false;
```

(2b) (5 points) What is meant by context switching? Why are context switches desirable? What are some of the considerations in the context switch program?

(3a) (5 points) Under what circumstances would a user be better off using a time shared system rather than a personal computer or a single user workstation?

(3b) (5 points) What mechanisms are used for overlapped Input, Output and Compute operations? What are some of the considerations in such mechanisms?

(4a) (4 points) What is the meaning of the term *busy-waiting*? What other kinds of waiting are there?

(4b) (6 points) A museum is to be shared among different processes (persons). The museum can be visited simultaneously by several processes, subject to the following constraint: The number of processes currently visiting the museum must be less than n. Write a monitor (in C-style format) to coordinate the visit to the museum.

(5a) (2 points) What are the operations permitted on *condition-variables*?

(5b) (7 points) A spooling system consists of an input process I, a user process U, and an output process O, which exchanges data through two bounded-buffers In an Out (see the following Figure). Write a program to implement such a scheme, using semaphores, assuming that the read and write operations allow you to read from the card reader and write on to printer.

```
process I
{
while (1) { read(card);
deposit(In);
}

process U
{
while (1) { fetch(In);
compute;
deposit(Out);
}

process O
{
while (1) { fetch(Out);
write(printer);
}
```

Put appropriate semaphores to make this work.

(6a) (3 points) Each resource is in one of the classes 1,2 up to n. A process must always request resources in order, from class up to class n. Putting this another way, a process that holds a resource from class i is not allowed to request a resource from the same class or a lower numbered class. Provide a brief explanation, why this convention prevents deadlock.

(6b) (7 points) Solve problem 7.11 page 244 in Silberschtaz and Galvin.