Name: _____

CS 4/53201 **Exam #1** **Operating Systems**

Monday 9 October 2000

1. (10 points) Explain how each of the following hardware features is used to protect the system from malicious or erroneous processes.

   a. Privileged instructions

   b. Timer

2. (20 points) Suppose that two user-level processes are running on a machine. The wordprocessor (WP) process is currently waiting for the user's next keystroke. The spreadsheet (SS) process is currently recalculating (i.e. using the CPU).

   a. Using the 5-state process model, which state is each of the two processes in?

      WP is in state: SS is in state:

   b. Now suppose that the user presses a key (for WP), while SS is still calculating. Describe briefly what happens from the time that the key is pressed until the time that the WP process changes state. Hint: hardware is involved.

   c. Immediately after the state change at the end of the events described in part b) above, which state is each of the two processes in?

      WP is in state: SS is in state:

Name: _____

3. (10 points) Describe how the producers / consumers problem with 3 producers and 2 consumers can be solved using message passing and indirect communication. Assume that any consumer can consume an item created by any producer.

4. (10 points) Suppose an operating system implement its window manager using threads, so that a separate thread is responsible for each window on the screen. Why is it more appropriate to use threads for this purpose than processes?

5. (10 points) One way that semaphores can be implemented is by enabling and disabling interrupts (as is done in Nachos).

   a. This technique can affect the system's I/O in an adverse way. Explain.

   b. This technique does not work on multiprocessor systems. Explain.

Name: _____

6. (15 points) For each of the following code segments, where the variables are initialized as shown, and threads A and B can run concurrently, what are the possible ending values of x?

a.

```
int x = 10;
int y = 35;
semaphore m = 1;
semaphore s = 0;

thread A {
    x = x + 1;
}

thread B {
    x = y + 1;
}
```

Ending values are:

b.

```
(same
 variables)

thread A {
    P(m);
    x = x + 1;
    V(m);
}

thread B {
    P(m);
    x = y + 1;
    V(m);
}
```

Ending values are:

c.

```
(same
 variables)

thread A {
    P(s);
    x = x + 1;
    V(m);
}

thread B {
    P(m);
    x = y + 1;
    V(s);
}
```

Ending values are:

7. (25 points) Consider the code below (taken directly from Lecture 15) for solving the Readers / Writers problem with writers priority:

Reader:

```
acquire(mutex);
while (AW+WW > 0) {
  WR++;
  wait(OKToRead);
  WR– –;
}
AR++;
release(mutex);

read database

acquire(mutex);
AR– –;
if (AR == 0 &&
  WW > 0)
  signal(OKToWrite);
release(mutex);
```

Writer:

```
acquire(mutex);
while (AW+AR > 0) {
  WW++;
  wait(OKToWrite);
  WW– –;
}
AW++;
release(mutex);

write database

acquire(mutex);
AW– –;
  if (WW > 0)
    signal(OKToWrite);
  else
    br'cast(OKToRead);
release(mutex);
```

Name: _____

a. Since writers have priority, if there are active or waiting writers, a reader will have to wait. Where in the code does it wait?

b. When a reader is waiting to read as in part a) above, what happens if another reader starts to execute?

c. What is the purpose of the code after “read database” in the reader?

d. How are multiple writers prevented from writing the database at the same time?

e. (Extra Credit) Why is there a broadcast at the end of the writers code instead of a simple single?