NAME: ______________________________________ (FIRST NAME FIRST) **SCORE**: ____

COSC 4330 SECOND MIDTERM NOVEMBER 5, 2001

This exam is **closed book**. You can have **one** page of notes. Cheaters face expulsion from the university.

1. For each of the statements below, indicate in one sentence whether the statement is true or false (2 points), **and why** (3 points).

(a) The initial value of a UNIX semaphore is ***undefined***.

FALSE, the initial value of a UNIX semaphore is zero (and you should know it if you have done the programming assignment).

(b) A ***blocking send*** blocks until the kernel receives the message.

FALSE, blocking sends wait until the message has been received by the process to which it was sent.

(c) Atomic transactions implement the ***at most once semantics***.

FALSE, they implement the all-or-nothing semantics.

(d) A ***mutex semaphore*** can only have two correct values.

TRUE, these values are zero and one.

(e) A message sent to a ***private mailbox*** can only be received by one of the processes having read access to that mailbox.

FALSE, only the process owning the mailbox or its children can retrieve messages from it.

(f) Starvation is always the result of a deadlock.

FALSE, it can also happen to processes that have very low priorities unless the scheduling algorithm automatically increases the priorities of processes that have remained in the ready queue above some waiting threshold.

2. Consider the function

```
transfer(int *from, int *to, int amount) {
     *from -= amount;
     *to += amount;
} // transfer
```

and assume the following calling sequence:

```
alpha = 100; beta = 200;
transfer (&alpha, &alpha, 10)
```

What will be the value of `alpha` ***after the call*** assuming that:

(a) The call was a ***regular procedure call?*** **`alpha =`**__100______

(b) The call was a ***remote procedure call?*** **`alpha =`**__110______

3. What is the simplest way to provide mutual exclusion in a kernel running on a uniprocessor architecture? (5 points) Would this solution work in a kernel running on a multiprocessor architecture? (5 points) Justify and qualify your answer. (10 points)

The simplest way to provide mutual exclusion in a kernel running on a uniprocessor architecture is to disable interrupts. The solution will also work on a multiprocessor architecture using a master/slave kernel organization but not on a multiprocessor architecture using a symmetric organization because this organization allows multiple copies of the kernel to run in parallel on each processing unit.

4. A Laundromat has 20 washing machines and 10 dryers. Assuming that all customers use one washing machine to wash their clothes then one dryer to dry them, add the necessary semaphore calls to the following program segment:

```
semaphore _washers___________ = __20__;(5 points)
semaphore _dryers____________ = __10__;(5 points)
customer (int who) {
    _P(&washer);_______________;(5 points)
    wash_clothes();
    _V(&washer);_______________; (5 points)
    _P(&dryer);_________________; (5 points)
    dry_clothes();
    __V(&dryer)_______________; (5 points)
} // customer
```

5. Consider the following solution to the critical section problem:

```
int status = 1; // Shared variable
 ...
// Enter critical section
while (status == 0);
status--;
      ...
// Leave critical section
status++;
```

Indicate when and why it does not work (10 points)

The solution will not work when two processes try to enter the critical section in lockstep. Once status equals zero, they will both enter the critical section and mutual exclusion will not be respected.