COSC 4330 FIRST MIDTERM OCTOBER 29, 2003

This exam is ***closed book***. You can have **one** page of notes. UH expels cheaters.

1. Assume our department wants to implement a system allowing prospective foreign students to transmit requests directly to our admission personnel. What would be the best class of service for each of the following requests and why? (3×5 points)

(a) Requesting the CS server to send a short application checklist to the applicant; this checklist will reach the applicant within a few seconds.

Unreliable datagrams because the client can always resend a request that did not go through.

(b) Sending a completed application in PDF format.

Streams (or virtual circuits) because the application must arrive intact without lost or duplicate packets.

(c) Requesting an assistantship by sending a short message whose reply could normally come several weeks or months later.

Reliable datagrams because the must know whether the request went through or not.

2. Consider the following System V Release 4 scheduler:

```
#ts_quantum  ts_tqexp  ts_slpret  ts_maxwait  ts_lwait  LEVEL
 2000        0         1          16000       1         # 0
 1000        X         2           8000       2         # 1
  500        1         3           4000       Z         # 2
  200        2         Y           2000       3         # 3
```

What are the *only correct values* for the three missing parameters **X**, **Y** and **Z**? (3×5 points)

X = __0___ **Y = __3___** **Z = __3___**

Explanations:

- X = 0 because we want to lower the priority of processes than exceed the time slice of their priority level and the process is already at level 1.
- Y = 3 because we want to increase the priority of processes than return to the ready queue from the waiting ("sleep") state and level 3 is the highest level.
- Z = 3 because we want to increase the priority of processes than have waited more than ts_lwait in the ready queue and the process is already at level 2.

3. Complete the following template to obtain a correct solution to the mutual exclusion problem for two processes whose ID's are either 0 or 1? (5×4 points)

```
shared int turn;
shared int requested[2] = {0, 0};
void enter_region(int pid) {
   requested[_pid_____________] = _1_____________;
   turn = _1- pid_____________;
   while (requested[_1- pid_____________]&& turn != pid);
} // enter_region
void leave_region(int pid) {
   requested[_pid_____________] = 0;
} // leave_region
```

4. A room has two doors and one switch at each door to turn the lights on and off. Complete the following template to ensure that (a) the room will never contain more than 20 people and (b) the light switches will always be on when there as re people in the room and off when the room is empty. (2 points for the initial values of your semaphores and 3 point for each correct semaphore call)

```
shared int people_count = 0;
semaphore mutex = _1____; // since semaphore is a mutex
semaphore room  = _20___; // same as capacity of the room
_P(&room); P(&mutex);_____________________________________;
people_count++;
if(people_count == 1) toggle_switch();
_V(&mutex);_______________________________________________;
stay_in_room();
_P(&mutex);_______________________________________________;
people_count--;
if(people_count == 0) toggle_switch();
_V(&room); V(&mutex);_____________________________________;
```

Explanation: This problem is a variant of the readers and writers problem.

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

a) We can simulate a *non-blocking receive* with a *blocking receive* inside a busy wait loop.

FALSE, you cannot simulate a non-blocking receive with a blocking receive.

b) The *all-or nothing semantics* for remote procedure calls is much harder to implement than the *at-most-once* semantics.

TRUE, the all-or-nothing semantics requires atomic transactions while the at-most-once semantics only requires adding serial numbers to requests.

c) We cannot send *linked lists* to a remote procedure.

FALSE, we can send linked lists by packing them into arrays.

d) UNIX sockets are an example of *private mailboxes***.**

TRUE, they die with the process that created them.

e) Windows NT dynamically adjusts the priorities of real-time processes to ensure that these processes always complete on time.

FALSE, real-time processes have fixed priorities (as in VMS and UNIX System V Release 4).

f) In the round-robin policy, a running process cannot be preempted before the end of its time slice.

TRUE, all processes have the same priority.