COSC 4330 SECOND MIDTERM JUNE 18, 2002

This exam is closed book. You can have one page of notes. People caught cheating will expelled from UH.

1. Answer in one sentence to each of the following questions: (6×5 points)

(a) What should a process do when it receives a retransmission of a message it has already received and acknowledged?

It should acknowledge the retransmission in order to avoid a second retransmission.

(b) What is the major limitation of the ***round-robin*** scheduling policy?

It performs poorly at high CPU utilization rates as it keeps reducing the time slice duration in order to maintain its response time, thus increasing the context switch overhead.

(c) How can you implement the ***at most once*** semantics in a remote procedure call package?

Each client will associate with each of its request a serial number, which will allow the server to detect and refuse to act on retransmissions of previous requests.

(d) What is the main disadvantage of non-preemptive scheduling policies?

Non-preemptive scheduling policies cannot prevent a running process to keep running when other more time-critical processes wait in the ready queue..

(e) What is the advantage of the all-or-nothing semantics for remote procedure calls?

The all-or-nothing semantics guarantees that each process will either be completely executed or not at all, thus eliminating the possibility of partial executions,

(f) What is the major disadvantage of ***virtual circuits*** over ***datagrams***?

Virtual circuits require the setup of a virtual connection between the sender and the receiver before any message can be sent.

2. What are the differences between a ***private mailbox*** and a ***public mailbox***? (10 points)

A private mailbox is owned by a process, messages addressed to it can only be retrieved by that process and the mailbox terminates when its owner process terminates.

A public mailbox is a kernel object, messages addressed to it can be retrieved by all the process having that permission and the mailbox will survive the termination of the process that created it.

3. The Amoeba system included a special **`do_it()`** primitive for remoter procedure calls. It was used by the client and combined the **`send_request()`** and **`receive_reply()`** into a single system call . What was the main advantage of this approach? (10 points)

Less context witches as two system calls are replaced by a single call.

4. Consider the following System V Release 4 scheduler and select the most appropriate choices for the missing parameters W, X, Y and Z. (4×5 points)

```
#ts_quantum ts_tqexp ts_slpret ts_maxwait ts_lwait LEVEL
     500        0        3        4000        1   #  0
     200        0        3        2000        2   #  1
       W        X        Y        1500        3   #  2
      50        2        3         800        Z   #  3
```

W = _100_+_ **X** = _1__ **Y** = _3__ **Z** = _3__

5. Add the missing parts to the following pseudocode to obtain a correct solution to the mutual exclusion problem. (4×5 points)

```
//Global variables
global int tie_breaker;
global int intent[2] = {0, 0};
void enter_region(int pid) {
     int other;
     other = 1 – pid;
     tie_breaker = pid;
     intent[_pid____] = _1______;
     while (intent[other] && tie_breaker _==__ pid);
} // enter_region
void leave_region(int pid) {
     intent[_pid____] = _0______;
} // leave_region
```

6. What is the problem raised by the co-existence of big-endian and little-endian CPU architectures? (5 points) How would you solve it? (5 points)

Big-endian and little-endian architectures use different orderings to describe the contents of a word. As a result, binary values will not be correctly transmitted between big-endian and little-endian machines and vice versa.

The best solution is to define a network order and require all binary values to be sent in that order. (In fact, the Internet Protocol defines a standard big-endian network byte order.)