COMP 242 Final Exam

Spring 2006

There are 4 questions; answer all of them. If you need to make an assumption to clarify a problem, *write your assumption down*. Only reasonable assumptions get full credit. *Explain all answers.* You have three hours to finish the exam. *Good Luck!*

1. If OS/360 is a dinosaur, what is XINU? (1 pt)

2. Protection: (33 pts.)

(a) Describe the simplifications made in Unix to the general concept of an access list. What are the benefits and drawbacks of these simplifications? (15 pts.)

(b) Describe two ways in which a Unix process can dynamically change its protection domain. (7 pts.)

(c) Consider the following scenario: Alice, a user of Unix, has created a bibliography file consisting of references to papers and her comments on them. She wants to allow other users to read the references but not the comments. Explain what she should do to protect the file in this way. Your solution should use the Unix protection facilities directly— it should not modify the operating system. (11 pts.)

3. Multiprocessor Operating Systems: (33 pts.)

(a) Distinguish between an application (or job), a virtual processor, and a thread. (11 pts.)

(b) Distinguish between time and space scheduling. Explain why space scheduling has lower response time (the time required to complete a set of jobs.) than space scheduling. (11 pts.)

(c) Why is the RRJob time-scheduling approach called uncoordinated? Describe various forms of coordination you could add to RRJob to improve its response time. (11 pts)

Continued ==>

4. Process Coordination: (33 pts.)

(a) Consider a monitor that defines the signal operation to be a ‘‘hint’’ to a waiting process. What queues are associated with such a monitor and what are the rules governing the placement and removal of processes from these queues? (14 pts)

(b) Does the programmer of a monitor have to be aware of whether the ‘‘hint’’ or ‘‘absolute’’ semantics are supported? (4 pts.)

(c) Now show that the rules given in (a) for hint semantics can lead to starvation when the following monitor is accessed: (15 pts.)

```
monitor StarvationProblem
var
    A, B: condition;
    AOK, BOK: boolean;

entry procedure GetAB;
begin
    while not AOK do
         wait (A);
     AOK := false;
    while not BOK do
         wait (B);
     BOK := false;
end GetAB;

entry procedure ReleaseAB;
begin
    AOK := true;
    signal (A);
    BOK := true;
    signal (B);
end ReleaseAB;

entry procedure GetB;
begin
    while not BOK do
         wait (B);
     BOK := false;
end GetB;

entry procedure ReleaseB;
begin
    BOK := true;
    signal (B);
end ReleaseB;

begin /* initialization */
    AOK := BOK := true;
end StarvationProblem.
```