COMP 730 Final Exam

Spring 2007

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. What does (or could) XINU stand for? (1 pt)

2. Transactions: (33)

(a) Explain what it means for a schedule of transactions to be (i) serializable, and (ii) cascade-free. Assume that the operations performed by the transactions are Reads and Writes. Give as liberal a set of constraints as you can for ensuring serializable and cascade-free transaction schedules. (11 pts.)

(b) Consider the object, `HasthTable,` which defines the operations `put(k,o)` and `get(k).` `put(k,o)` binds key, `k,` to object, `o;` overriding any existing binding between k and an object. `get(k)` returns the object associated with `k,` if `k` is bound, and null otherwise. Use these semantics to give as liberal a set of constraints as you can for ensuring (i) serializable and (ii) cascade-free schedules for transactions that perform operations on this object. (11 pts.)

(c) If we did not have object-specific information about `Hashtable,` we could have modelled `put(k,`o) as a write of the entire object, and `get(k)` as a read of the entire object; and used the conditions given in (a) for serializable and cascade-free schedules. Illustrate the benefits of (b) by giving an example of (i) a serializable schedule that would not be allowed by (a) but would be allowed by (b), (ii) a cascade-free schedule that would not be allowed by (a) but would be allowed by (b), (iii) a non-serializable schedule that would not be allowed by (b), and (iv) a serializable but not cascade-free schedule that would not be allowed by (b). (11 pts.)

3. Hierarchical Access Control (26 pts):

(a) Explain the relationship between access control and the kernel/user mode (3 pts.)

(b) Explain how stack-inspection is used in Java to limit access of downloaded code. (5 pts.)

(c) Explain why kernel/user modes and the Java stack-inspection scheme are both examples of hierarchical access control. (3 pts.)

(d) Design a single integrated hierarchical access-control scheme that supports both user/kernel modes and Java stack inspection. You can assume that each Java object is placed in a separate segment and that user and kernel code are kept in separate segments. Your scheme can consist of both hardware and software extensions. (15 pts.)

Continued ==>

4. Multiprocessor Operating Systems: (40 pts.)

(a) Consider an implementation of scheduler activations that provides preemptive space-sharing (a la Dynamic) to keep the number of virtual processors less than or equal to the number of physical processors. Explain the sequence of kernel actions that occurs when there are no idle processors and a new application needing one processor enters the system. You can assume that at least one of the existing applications has been assigned at least two processors and no processor is executing a critical section. (13 pts)

(b) Consider the following hybrid processor-sharing policy, which combines elements of space and time sharing. It assigns each application its Equipartition quota of physical processors, but allows the number of virtual processors to exceed the number of physical processors. The number of virtual processors assigned to an application is equal to its current concurrency. The virtual processors of an application are time shared among the physical processors assigned to it. Describe how a kernel could support such a policy. Give the sequence of kernel actions that takes place in the scenario descibed in part (a) where a new application enters the system. You can ignore I/O, priorities, and critical sections for this part of the question. (13 pts.)

(c) Compare the dynamic policy with the hybrid policy. Be sure to state any assumptions you make about details of the hybrid policy not specified in (b). (14 pts)