CS 6210 - Final - Dec. 2001

Closed book and closed notes

Your Name: _____

Prof. Schwan

Please make time to work on each question in this exam. Partial credit will be given whenever possible.

1) Short questions: 50 mins. (questions marked with * have more weight)

a. *Assuming that a NIC (network interface card) does not have any DMA hardware, answer the following: (1) what is the effect of co-scheduling (co-scheduling means 'scheduling to make sure they run in parallel') the NIC's actions with those of the host CPU (i.e., making sure the host's driver runs at the same time the NIC is executing), and (2) for what kinds of applications is such co-scheduling most appropriate?

b. We studied the notion of cache affinity in class. Identify two other notions similar to affinity that came up in papers studied in class. Define what kind of affinity you are talking about and describe its importance.

*c. Consider a multi-media system where a message arriving at a machine should trigger some action in an application-level program. Define two interesting 'semantics'(behaviors) of such a trigger mechanism. Be specific about the mechanism's semantics and its consequent use in a sample multimedia application.

d. Describe the difference between logical and physical clocks, as defined in Lamport's paper on distributed systems.

e. Use the paper on "Integrating Security in a Large Scale Distributed System" to delineate what is meant by 'protection' vs. 'security' research. Be specific by giving two sample issues we are concerned with in security but not in protection (please be technical here...e.g., don't talk about physical security as in the military).

*f. In Global Memory Management, first, describe intuitively the algorithm used for placing pages onto the memories of the multiple machines cooperating in terms of memory management. Second, why are ‘epochs’ needed by the memory management algorithm?

g. A mail server like Porcupine must pay attention to both performance and reliability. Describe how soft vs. hard state is used in Porcupine, by providing at least two examples.

*h. Describe the use of Undo and Redo logs in transaction management, as in LRVM. Then comment on how such logs may be/may not be necessary for some uses of transactions (e.g., recall Quicksilver). Be specific here, by providing at least two examples of transaction usage.

i. What are ‘doors’ in the Spring operating system? Are they necessary in order to properly implement Spring’s object invocations? Whether you agree or disagree, you must argue why they are necessary or not necessary.

2) Medium design question. 30 mins. You are designing protection facilities for an operating system. One common issue with protection is ‘revocation’, which means the ability to withdraw at any time access rights you have given to someone.

1. In general, briefly discuss how easy/hard revocation is with a capability- vs. access control list-based implementations of protection.

a. describe why current Unix does not support revocation

2. Now develop a new mechanism for revocation in a capability-based access control system. In order to do this:

a. describe what protection you want to offer for file accesses
b. ‘upgrade’ Unix file descriptors into capabilities, then describe your new revocation mechanism
c. comment on the expected performance of your mechanism, with respect to (1) common file usage (reads, writes), and (2) cost of revocation

3. Briefly comment on how you would solve a second problem: amplification. This means that someone who has your file capability needs additional rights.

a. in what situation is this important?
b. what mechanism (other than deleting the capability and giving someone a new one) can you imagine for amplification?

3. Medium design question (30 mins.)

This question asks you to design a notion of system-wide resource reservation. Starting with the paper on CPU reservations, design an OS facility with which an application could reserve multiple resources (CPU, memory, network, other devices). Apply this to a scenario in which you want to ensure that data you are receiving from a remote sensor (e.g., a surveillance camera attached to a network) is processed and displayed by some deadline (end-to-end delay) required by its usage.

1. define the usage scenario: the machines, devices and programs involved, and the end to end delay requirements (detail the latter for at least one scenario)
2. describe your reservation mechanism's individual components for each of at least three of the devices involved in some end to end interaction
3. describe your multi-device reservation mechanism that makes combined use of the individual components in 2.
    a. its API to an application (must permit dynamic use...whenever your programs need to use it)
    b. indicate a possible algorithm it may use to determine suitable reservations
    c. is a notion of 'transactions' useful in the mechanism's internal implementation and if not, what alternative notion are you using in order to guarantee properties like 'all or nothing' reservations?
4. comment on your mechanism's simultaneous use by multiple end users
5. comment on the performance of your multi-resource reservation mechanism