CS330 Final – April 23, 2010
14:00-17:00, CL435
D. Hepting

READ THE INSTRUCTIONS FIRST. Answer all questions in the exam booklet (none on this page). *Entirely complete your portion of* the cover page of the exam booklet. *Return this page* tucked inside your exam booklet.

If you have any doubts about how to interpret a question, state your assumptions in your answer and then proceed. I won't clarify questions during the exam.

You must pass the final to pass the class.

There are 2 parts to the exam, with choices in parts 1. Required questions are marked with R, optional questions are marked with O. In part 1, answer all the required questions and as many optional questions as needed. All questions are worth 5 marks. In part 1, reserve ½ page for each answer. In part 2, reserve 1 page for each answer. Therefore, you should use 10 page sides in total (the back of the exam booklet should be empty).

For all questions, I want to see that you have answered the question asked in a way that illustrates to me that you understand the concepts. You do not need to fill all available space in the exam booklet! Less can be more!

If you've followed the instructions *completely,* you will receive a bonus of 3 marks (no part marks).

Part 1 (12 questions x 5 = 60 marks)

1. R. To which states may a running process transition, and why?

2. R. What are Shortest Seek Time First and Shortest Job First? How do they compare to one another?

3. O. How are disk allocation and memory management alike, and different?

4. O. How have killing and starvation entered into our class discussions? Explain.

5. R. What is an operating system?

6. O. How is the memory hierarchy used in implementing virtual memory?

7. R. In a virtual memory system, how many frames should a process get? How does this relate to thrashing?

8. O. The optimal algorithm in many cases is one that depends on knowing the future. What are the ways that we can approximate these optimal algorithms? Explain.

9. R. What is fragmentation? Where can it occur? Can it be eliminated? Use 2 examples from our discussions.

10. R. What is a race condition and how can we deal with it? Use the producer/consumer problem as an example.

11. R. In what ways can one deal with the possibility of deadlock? Describe each briefly in the context of the dining philosophers' problem

12. O. How might we organize directories. What does it matter if a file is available locally or remotely (via NFS [Network File System])?

13. O. How might sharing of an editor (for example) between processes be accomplished? Which approaches make it easy, and which make it difficult? What structures are required?

14. R. What is different about accessing a memory location in a system with paging versus a system with virtual memory and paging? Be specific.

15. O. Describe the multi-level feedback queue algorithm for process scheduling. What are its advantages and disadvantages?

16. O. Why is the analogy of an elevator appropriate for disk access?

Part 2 (4 questions x 5 = 20 marks)

1. R. How would/did you make your (project) file system able to handle growth of files, in robust and scalable way?

2. R. What are the advantages and disadvantages between unix-style owner-group-world and access control lists for managing permissions on files. Can we have both?

3. R. Illustrate the contents of the page frames (3 available) with the following reference string under 2 FIFO and LRU page-replacement algorithms. Reference string: 701203042303. What is possible with one and not the other?

4. R. Discuss some of the tradeoffs faced in designing an operating system (use your project as an example). Include block size in your discussion.