

CSCI 340: Operating Systems

Fall 2016

Brent C. Munsell

Department of Computer Science

College of Charleston

Exam Review

- ▣ Format
- ▣ Time
- ▣ Topics Covered
- ▣ Possible Questions

Format

- ▣ Pencil and Paper
- ▣ Do not need calculator
- ▣ Open book and notes
 - ▣ May use laptop if notes/book are electronic
- ▣ May not use laptop to
 - ▣ Search for answers on the internet
 - ▣ Compile and run code during exam

Format

- Roughly 10 questions
 - Short answer
 - Evaluate programs written in C language
- Short answer
 - If the question says “show all work to receive full credit” then you must show all work 😊

Time

- 60 minutes
- Test will start promptly at the beginning of class
- Sam Brokaw (my graduate assistant) will be proctoring the exam

Three Big Topics Covered

1. Operating Systems Overview

- ▣ Understand and explain OS architectures including monolithic, layered, virtual machine, client-server, kernel, multiprocessor [Chapter 2]
- ▣ Understand the relationship between an OS and hardware. This includes: processor, memory, disks, I/O devices, and buses. [Chapters 1 and 2]

At least one question from each topic will be on the exam.

Three Big Topics Covered

1. Processes [Chapter 3]

- ▣ Fork, Exec, and Wait System Calls: Given a code segment, explain the purpose of each operation, provide some example function parameters and the expected return types, complete the code or identify coding errors with suggested fixes.
- ▣ UNIX Process State Diagram: Given the state diagram in Figure 3.17 in the course textbook, identify the different process states, explain the purpose of each state, given a scenario use the state diagram to describe the sequence of state transitions the process will go through.
- ▣ Process Modes: Discuss the difference between Kernel and User mode, and why do at least two modes exist, i.e. why not use a single mode? What types of operations cause a mode switch, and use the state diagram in Figure 3.17 to show the sequence of states a process goes through when this occurs.
- ▣ Process Switching: Explain the difference between an interrupt and a trap, and when a interrupt occurs describe the sequence of operations that occur for switching the mode, and changing the process state.
- ▣ Interrupts: Give three examples of an interrupt and explain the difference for each identified interrupt (seepage 136 in course textbook).

At least one question from each topic will be on the exam.

Three Big Topics Covered

1. Threads [Chapter 4]

- ▣ Process vs. Thread: What is the major difference between these two concepts, how are they related, and why choose a thread over a process or vice-versa?
- ▣ Types of Threads: Discuss the difference between kernel-level and user-level threads, and why do two thread levels exist?
- ▣ Thread Mode Switching: List and discuss the reasons why a mode switch between threads may be cheaper than a mode switch between processes.
- ▣ General Questions: Problems similar to 4.5 and 4.7 in the course textbook.

At least one question from each topic will be on the exam.

Possible Questions

- Any question (or programming concept) given in a homework assignment or quiz is fair game
- Any topics/problems/questions discussed in class are fair game
- I may simply change the question a small amount and then ask again.

Operating Systems Overview

- What are the advantages and disadvantages:
 - Monolithic kernel design
 - Modular kernel design
- What is the difference between an interrupt and a trap?
Give examples of each.
- Using pointers in the C language to access resources on your computer (memory, file, etc.)
 - Simple coding questions use and manipulate pointers

Processes

- Difference between kernel and user mode. Why is it necessary for an OS to have these two modes?
- Process state transition diagram
 - Given a code segment identify the sequence of process states
 - Format very similar to quiz question, and handout discussed in class.
- fork-wait and fork-execute-wait scenarios. Given a code segment:
 - Explain the purpose of each operation
 - Provide some example function parameters and the expected return types
 - Complete the code or identify coding errors with suggested fixes.
 - Format very similar to quiz question, and the different coding examples discussed in class.

Threads

- What makes a thread different than a process?
- What are some advantages of using threads?
- What are some disadvantages of using threads?
- Why is concurrency an issue with threads but not processes?
Provide an example.
- Discuss the difference between kernel-level and user-level threads, and why do two thread levels exist?