Compsci 577: Operating System Design and Implementation

Midterm Exam Study Guide

General Guidelines:

1) Material from Lectures 1 to 13 will be part of the test.
2) You will be tested on material covered in class lectures. You are not expected to know about concepts not covered in class (e.g., from the book but not covered in class). As a general staretegy, start with the Lecture Slides and be sure to review all of the material on the slides carefully. Use Lecture notes, Class videos and possibly the book to supplement the Lecture Slides.
3) The exam will consist of several short answer questions (that test basic understanding of concepts covered in class) and other questions that ask you to apply the concepts or ask for a bit more details. In all cases, questions are derived from concepts covered in class lectures (and slides).
4) There are no coding questions on the test. There are no math questions either. There is no need to memorize arcane details of Minix - although you are expected to know general Minix concepts that were taught in the class.

Lecture-specific Study Guidelines

Lecture 1: Review OS definitions and various kinds of OS architectures and their pros and cons.

Lecture 2: Review hardware/architectural features that were included to help the OS implement various functionality. Lec 2, slide 7 summarizes all the hardware concepts covered during this lecture.

Lecture 3: We covered three easy to read papers in this lecture (listed on lecture 3, slide 3). You are not expected to memorize any history details from the first paper. Review the key concepts from the other two readings. The slides give you most of the key details.

Lecture 4 (continuation of lecture 3) Review the key components of the Minix kernel (summarized on Lec 3, page 23) and also review the high level functionality of each component.

Lecture 5: Review the six step process for the generic OS boot process and what each step does. There is no need to memorize arcane details but you should understand the high level ideas.

Lecture 6: System and Kernel Calls: A high level understanding of system calls, kernel calls, and Minix-specific details of these concepts. There is no need to memorize arcane details of sys call numbers etc - instead focus on how these concepts work and are realized in OS/Minix.

Lecture 7: CPU Scheduling: Review various CPU scheduling policies and their key pros and cons. Understand pre-emptive and non-preemptive scheduling.

Lecture 8: Review Minix scheduling concepts (which you also learnt in lab 2). Review IPC concepts. Review process concepts, thread concepts, fork/exec.

Lecture 9: Review high-level concepts about Linux process management, threads and Linux CPU scheduling. No need to memorize low-level details.

Lec 10: Memory management: Review concepts of memory addresses (physical, virtual). Key memory allocation schemes (contiguous, paging, segmentation, demand-paging etc), hardware support needed for each, and the pros and cons of each approach.

Lec 11: Review Minix memory management concepts. No need to memorize low-level data structures etc.

Lec 12: Virtualization I: Review what is virtualiation and different types of virtualization. Review type 1, type 2, and paravirtualization (what are they, how do they work, what hardware support, if any, is needed for each). Review high-level concepts behind virtualizing memory, disk and network from the slides.

Lec 13: Virtualization II: Review OS level virtualization and high-level ideas such as namespaces, cgroups, fair share proportional schedulers, docker, and live VM migration.