CMSI 387

Operating Systems

Spring 2008

Midterm Review Sheet

The midterm will take place as scheduled, on February 26. It will be open everything: book, notes, handouts, and computer (which means that either we can have the test in the Keck lab, or everyone should have a laptop that they can use). This guide should help you to prepare for the midterm properly.

Covered Material

The midterm covers the following areas, including all handouts and sample code that have been distributed in support of this content:

- SGG Chapters 1–2, 10–11
- Working knowledge of how to configure and build a Linux kernel
- Working knowledge of an assortment of prominent operating systems (i.e., the ability to access the process, memory, I/O, and storage functions of these platforms)

Sample Tasks and Questions

The following represent the types of questions or tasks that you may be asked to accomplish:

- Perform some analysis, critique, or evaluation of an operating system concept (design choices, operating system roles [process management, memory management, file systems, I/O], the boot process) or structure (kernel, system calls, privileged vs. user mode, device controllers and drivers, file system drivers)
- Describe a real-world computer issue or activity in more precise, operating system-specific terms (e.g., computer won't boot, computer is slow, "blue screen of death," device doesn't work with a computer, laptop power management, dual-boot computers, etc.)
- "Read" a given snapshot of processes (e.g., what is a process's ID, what is a process's "lineage," which process was the first one created upon boot-up, which process is using the most CPU time, how much memory is a process using, etc.)
- Answer questions pertaining to storage devices and file systems, including but not limited to common volume structures, mounting/unmounting, file system drivers, file allocation schemes, directory structures, shortcuts/aliases/links, and consistency semantics
- Navigate or interpret specific instances of a given file system (*ext2* in detail, and possibly FAT, UFS, or some other hypothetical implementation that will be described in general)