

# CMSI 587

## OPERATING SYSTEMS (GRADUATE LEVEL) Spring 2006

### ***Final Review Sheet***

The final will take place as scheduled, on May 2; I'll shoot for 90–120 minutes. This guide should help you to prepare for it properly.

### **Covered Material**

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

- Chapters 1–6, 8–11, 13, and 15 in-depth
- Major concepts in Chapters 7 and 12
- Computer organization basics, particularly when relevant to operating system concepts such as memory management, storage management, and I/O
- Knowledge of C and the POSIX APIs
- Knowledge of algorithms we have studied, including but not limited to process scheduling, paging, segmentation, page replacement, frame allocation, and storage allocation
- Knowledge of how file systems are stored on disk, particularly the Linux *ext3* file system

### **Sample Tasks and Questions**

In addition to the types of questions that were fair game for the midterm, the following represent additional types of questions or tasks that you may be asked to accomplish:

- Manually perform some operating system algorithm from any of the major areas (process management, memory management, storage management, I/O); for example:
  - Apply some form of contiguous memory allocation approach to a given set of processes
  - Translate a paged or segmented address to its linear equivalent
  - Given a page reference string, show the per-page reference frame allocation based on some page replacement algorithm
  - Given a file allocation approach, show how logical file blocks are mapped to disk blocks
- Show detailed knowledge of navigating a file system structure, particularly Linux *ext3*
- Given a system description or code fragment, identify any potential security flaws
- Answer an open-ended operating system question that synthesizes the concepts we have covered in the class: these questions will generally pose some real-world situation, and ask you to identify operating system issues relating to this situation, or perhaps propose operating system techniques or approaches that may be applicable to that situation