
CIS520 Operating Systems

Course Intro

Dr. Daniel Andresen

Fall, 2009

Course Outline

1. **Process Management.**

- ❑ Interprocess communication.
- ❑ Process scheduling.
- ❑ Process synchronization.
- ❑ Deadlocks.

2. **Memory Management.**

- ❑ Virtual memory and paging.
- ❑ Pure segmentation.
- ❑ Segmentation with paging.

3. **File Systems.**

- ❑ Implementation.
- ❑ Security

4. **Distributed Systems.**

- ❑ File Systems.
- ❑ Communication.
- ❑ Synchronization.
- ❑ Distributed memory.
- ❑ Deadlock detection.

Class Organization

- Class meets Monday/Wednesday – Nichols 127
 - Lectures delivered in class (and later via Tegrity)
 - Lab sections
 - Fridays in Nichols 128
 - Quizzes unless told otherwise over lecture material
 - Quizzes will be posted the Friday prior (n-of-m questions)
 - Go over in-class exercises and project-related material
-

Studying Operating Systems

- This course deals with “classical” operating systems issues:
 - the services and facilities that operating systems provide;
 - OS implementation on modern hardware;
 - (and architectural support for modern operating systems)
 - how hardware and software evolve together;
 - techniques used to implement software systems that are:
 - large and complex,
 - long-lived and evolving,
 - concurrent,
 - performance-critical.

Nachos Projects - Maybe

- ❑ Project 1: concurrency and synchronization
 - *race conditions* with processes and threads
 - implementing/using synchronization for safe concurrent code
 - ❑ Lab 2: protected kernel with multiprogramming
 - OS kernel with system calls, memory allocation, virtual address translation, protection
 - ❑ Project 3: virtual memory
 - page faults and demand loading
 - page replacement and page cache management
 - ❑ Project 4: Filesystems
 - Extensible files, synchronization
-

We're Not in Kansas Anymore

- Be careful out there: CIS520 is For Mature Audiences Only.
 - ❑ Any OS is a complex beast with lots of moving parts.
 - ❑ Concurrency adds an unfamiliar and difficult element.
 - ❑ Virtual Machines are difficult to think about and debug.
 - ❑ For Nachos you will extend a base of **Someone Else's Code**.
 - ❑ Working in teams is a double-edged sword.
 - ❑ These labs *by design* leave more opportunity for creative interpretation than is common in introductory classes.
 - ❑ The Unix/C++ development environment is a powerful tool offering many opportunities to “shoot yourself in the foot”.

Secrets of the Nachos Projects

- We skip the hand holding; you skip the hand wringing over picky details of what you are “supposed to do”.
 - You are free to resolve ambiguity as you see fit, and you must justify your choices.
 - *It's the thought that counts.*
 - Think before you design it.
 - Think before you code it.
 - Think before you run it.
 - Think before you debug it.
 - The time needed to conceive and write the code is moderate, but debugging time is potentially unbounded.
-

Course Materials for CIS520

- ❑ *Course viewgraphs on the web site*
 - <http://online.ksu.edu/CIS520>
 - ❑ *The Nachos 4.0 instructional OS*
 - Download from KSOL
 - ❑ Silberschatz, Galvin, & Gagne: *Operating System Concepts*
 - ❑ *Other readings on the course web site*
 - Handouts, etc.
 - ❑ *Lecture notes on website*
 - *in part courtesy of Doug Chase (Duke) & Martin Rinard (MIT)*
-

Secrets of CIS520

- If you work hard, we work hard to help you.
 - Slackers are *persona non grata* (you know who you are).
- Be ready to work concurrency problems on the exams.
 - Drill it and use it in the labs.
- Pay careful attention to team management for the labs.
 - Rules: teams may eject slackers by unanimous vote of the remaining members; teams can reform for each project.
- Pay attention to the listserv, and post if you need help.
 - *cc: dan@k-state.edu*
- Have confidence in the grading.

Plagiarism & accommodations

- Plagiarism is punishable by anything from a zero for the assignment to expulsion.
 - Use online help sources, but cite sources & beware (most sites are for prior versions of Nachos)
 - Help each other out, but don't give each other code.
 - We check!
 - If you need accommodations due to a disability, let me know as soon as possible.
-

What To Do Next

1. Form teams.
 1. (2-3 people preferred)
 2. Look at the course web page and the handouts.
 - On the Web at <http://www.online.ksu.edu/CIS520>.
 - 3. Install and build the Nachos release (project 0) next week.
 - Determine where your source code will reside.
 - *Optional*: set up version control (e.g., CVS).
 - Report any problems.
-