CPS 210: Graduate Operating Systems
Spring 2006

1 Basic Information

Time and place: Monday, Wednesday; 4:25 – 5:40; Room D243, LSRC
Instructor: Landon Cox; Room D304, LSRC; lpcox@cs.duke.edu
Office Hours: Tuesday, Thursday; 1:00 – 2:00 or by appointment
Course web page: http://www.cs.duke.edu/education/courses/spring06/cps210/
Course newsgroup: http://www.cs.duke.edu/phpBB2/

2 Course Overview

CPS 210 is intended to satisfy a number of objectives:

- CPS 210 is a graduate level introduction to the basic knowledge, skills, and research directions in the field of Operating Systems.
- CPS 210 is a *Quals Course*. This means that it can be used to meet one of the requirements for PhD students. To get quals credit, a student must pass the exams with a "quals pass." A quals pass is not necessarily directly related to the course grade.
- CPS 210 is a *gateway course* for prospective systems students. This means it will present a view of the research currently of interest to the systems faculty. Students who are considering pursuing systems research (either Ph.D. or Masters level) within the systems group must demonstrate research potential. It will be impossible to convince any of the systems faculty to agree to be your advisor without doing a good job on the project (see Section 3.2 below).

Because the course instructors rotate from year to year, each offering is different. This year, I have chosen to remove the smaller programming assignments used in the past. The course project is now a requirement for all students and the focus of most out-of-class effort. This should in no way discourage non-systems students; "systems" is an extremely broad area and translating an existing AI, theory, or database interest into a systems problem should be straightforward. My emphasis on a course project and research papers is meant to help students of all backgrounds work through the difficult transition from class taker to researcher. You are here because you have already demonstrated that you are very good at taking classes; one of my primary goals is to help you become very good researchers.

The prerequisite for this course is CPS 110 (or its equivalent). The intended audience is computer science graduate students and undergraduates with an interest in systems research. Undergraduates who did well in and enjoyed CPS 110 are particularly encouraged to take this course.

3 Course Requirements

The work for this course consists of:

- Reading, analyzing, and actively discussing systems research papers.
- As part of a team, designing a semester-long project, carrying it out, and writing up the results.
- Taking a midterm and a final exam

3.1 Paper Summaries and Discussion (10%)

This course has no text book; instead, we will cover between 35 and 40 original research papers. It is important that you digest each assigned paper before lecture. You will need to turn in a short *hardcopy* summary of each paper to be discussed before lecture begins. Summaries should be *no more than five sentences long*, followed by *two* questions that you would ask the authors. This is intended to fuel class discussion, so think of good questions!

3.2 Project (50%)

In addition to reading papers, you will design, carry out, and communicate the results of an operating systems project. These projects are intended to lead to published papers in peer-reviewed workshops and conferences.

I will give you broad ideas for projects, but they are purposefully vague—a question well asked is already half answered. It is perfectly fine to work on a project that contributes to your own research agenda, but you must be able to convince me that your project has some relationship to operating systems, broadly defined. If you are an AI or theory student, I do not anticipate this being a problem. Some projects may require special hardware or privileges. If you would like to pursue such a project, please talk to me so that I can make the appropriate arrangements.

Projects are carried out in teams, preferably of three people. You may want to use the class forum (http://www.cs.duke.edu/phpBB2/) to find project partners. I will make some suggestions about possible project groupings. Your project partners will have a substantial impact on your grade, so choose them wisely. There are four components to your project grade: a project proposal, a midterm status report, a final project report, and a project demo/presentation. Each group will meet with me after turning in their proposal and status report to discuss future direction.

I would like to make project proposals and reports available to students who take 210 in this and future semesters. If your group would prefer not to do this, let me know over email.

3.2.1 Project Proposal

Your proposal should explicitly state the problem your project will address, your project's goal and motivation, related work, the methodology and plan for your project, and the resources needed to carry out your project. Be sure to structure your plan as a set of incremental milestones and include a schedule for meeting them. The proposal is approximately 20% of the project grade.

3.2.2 Status Report

Your status report should contain enough implementation, data, and analysis to show that your project is on the right track. You should include your original proposal with my comments, along with any surprising results or changes in direction, schedule, etc. You should also have a refined version of the problem statement and goals, as well as a more developed related work section. The status report is approximately 10% of the project grade.

3.2.3 Final Report

Your final report should be in a form ready to submit to an OS conference or workshop. It must be well written and give convincing evidence of any conclusions you can draw from your project. It must contain the following:

- Concise abstract.
- Problem motivation.
- Design goals or performance questions.
- Design architecture or performance metrics.
- Description of code or scripts: major data structures and control flows.
- Description of difficulties in coding or performance measurement and analysis: whether, why, and how the original goals, architecture, and/or metrics needed to be changed.
- Evaluation showing achievement of goals.
- Future work, related work, summary, and references.

3.2.4 Project Presentation and Demo

You must effectively demonstrate your project through an in-class presentation and live walk-through. The presentation should be about 15 minutes long. The talk should convey to the rest of the class 1) what problem you tried to solve and why it matters, 2) why the current state-of-the-art inadequately addresses this problem, 3) your alternative vision of how to solve the problem, 4) some results demonstrating the validity and limitations of your approach. *The goal of a good talk is not to regurgitate your paper, but to convince the audience that they should read your paper*. There is a difference!

The demo happens after the talk and lasts about five minutes. This is your opportunity to call attention to aspects of your project that may not be readily apparent in the final report (a nice user interface, cute features, hideous bugs, etc). Each group receives a single grade for the report, demo and presentation—in total, the report, demo, and presentation are worth 70% of the total project grade.

3.3 Midterm Exam (15%)

The midterm will be a take-home, open book, open note exam. You may consult any written source of material during the exam, including on-line sources, but you **MAY NOT** discuss the exam with anyone else—including faculty members, family, or others who are not enrolled in the course. The midterm exam should take a well-prepared student two hours to complete.

3.4 Final Exam (25%)

Similar to the midterm, the final exam will be a take-home, open book, open note exam. You may consult any written source of material during the exam, including on-line sources, but you **MAY NOT** discuss the exam with anyone else—including faculty members, family, or others who are not enrolled in the course. The final is tentatively scheduled to be handed out at noon on Friday, May 5th and will be due no later than noon on Saturday, May 6th. Note that this time includes the official exam period for the course. The final exam should take a well-prepared student two hours to complete.

4 Reading List

Here is a list of the papers we will cover in this course, sorted by topic and in the order they will be discussed in class. All papers are available electronically.

Basics
BA-1 Levin: An evaluation of the Ninth SOSP submissions
BA-2 Lampson: Hints for computer system design

Trends
TN-1 Rosenblum: The impact of architectural trends on operating system performance
TN-2 Vogels: File system usage in Windows NT 4.0

Concurrency/Scheduling
CS-1 Anderson: Scheduler activations: Effective kernel support for the user-level management of parallelism
CS-2 Waldspurger: Lottery scheduling: Flexible proportional-share resource management
CS-3 Lamport: Time, clocks, and the ordering of events in a distributed system

Virtual Memory and I/O
VMe-1 Appel: Virtual memory primitives for user programs
VMe-2 Chase: Sharing and protection in a single-address-space operating system

Virtual Machines
VMa-1 Chen: When Virtual Is Better Than Real
VMa-2 Rosenblum: Complete Computer System Simulation: The SimOS Approach
VMa-3 Barham: Xen and the Art of Virtualization
VMa-4 Dunlap: ReVirt: Enabling Intrusion Analysis Through Virtual-Machine Logging and Replay

Squishing Bugs
BU-1 Savage: Eraser: A dynamic data race detector for multi-threaded programs
BU-2 Engler: Bugs as deviant behavior: a general approach to inferring errors in code
BU-3 Qin: Rx: Treating Bugs As Allergies -- A Safe Method for Surviving Software Failures

Networking Implementations
NW-1 Bershad: Lightweight remote procedure call
NW-2 Morris: The Click modular router

Local File Systems
LF-1 Rosenblum: The design and implementation of a log-structured file system
LF-2 Seltzer: File system logging versus clustering: a performance comparison
LF-3 Lumb: Towards higher disk head utilization: Extracting free bandwidth from busy disk drives
LF-4 Chang: Automatic I/O hint generation through speculative execution

Distributed File Systems
DF-1 Howard: Scale and Performance in a Distributed File System
DF-2 Kistler: Disconnected Operation in the Coda File System
DF-3 Nightingale: Speculative Execution in a Distributed File System

Reliability
RE-1 Satyanarayanan: Lightweight recoverable virtual memory
RE-2 Lowell: Free transactions with Rio Vista
RE-3 Rodrigues: BASE: Using abstraction to improve fault tolerance
RE-4: Swift: Improving the reliability of commodity operating systems

Security
SE-1 Burrows: A logic of authentication
SE-2 Myers: A decentralized model for information flow control
SE-3 Costa: Vigilante: End-to-end Containment of Internet Worms
SE-4 King: Backtracking intrusions
SE-5 Efstathopoulos: Labels and Event Processes in the Asbestos Operating System

Burning Issues of the Day
(Readings to be announced later)

5 Lecture Schedule

Week	Monday	Wednesday	Deadlines
9-Jan		BA-1,2; Intro	
16-Jan	MLK Day	TN-1,2	Group declaration: 20-Jan
23-Jan	CS-1,2	CS-3	
30-Jan	VMe-1,2	VMa-1,2	Project proposal: 3-Feb
6-Feb	VMa-3	VMa-4	
13-Feb	BU-1,2	BU-3	
20-Feb	NW-1,2	LF-1,2	
27-Feb	LF-3	DF-1	Midterm exam: 1-Mar/2-Mar
6-Mar	DF-2	DF-3	
13-Mar	Spring break	Spring break	
20-Mar	RE-1,2	RE-3	Status report: 22-Mar
27-Mar	RE-4	SE-1	
3-Apr	SE-2,3	SE-4	
10-Apr	SE-5	BI	
17-Apr	BI	Talks/Demos	Demos/final report: 19-Apr
1-May	Finals	Finals	Final exam: 6-May, 9a-12p

Paper summaries (except for BA-1 and BA-2) are due on the day shown on this schedule, though we may not always complete (or even begin) discussion of a particular paper on its assigned day. This schedule is subject to change as the course develops; changes will be announced in class, on the newsgroup, and on the web site.

6 Late Work, Exam Conflicts

Paper summaries must be turned in before *you come to class, in hardcopy*. No late or electronic summaries will be accepted. All project deadlines are 6:00 PM on the day each component is due. Work submitted after 6:00 PM but before midnight is penalized 5%. Thereafter, each day is charged an additional 10%; weekend days count as "days." If you have a conflict with an exam, you must see me no later than three weeks prior to the exam to make arrangements.

7 Cheating and Collaboration

Acts of cheating and plagiarism will be reported to the appropriate administrative bodies. You are responsible for knowing, and will be held to, the University Honor Code. With the exception of exams, discussion of course material is not considered cheating and is strongly encouraged. If you receive substantial help from another person you must acknowledge them in your work. If you use any published or unpublished source in any of your own work, you must give full citation.