CPS 310: Operating Systems (also ECE 253)
Fall 2014

Class Meetings

MW 3:05 - 4:20 in B101 LSRC
F 3:05 - 4:20 in Physics 128 ("recitation")

Instructor

Jeff Chase
Office hours: TTh 2:00 - 3:00 in D306 LSRC, or by appointment, or try a drop-in.

Teaching Assistant

Shuyang Yao

This course gives an introduction to systems concepts and operating systems. An operating system is software that controls some programmable platform for sharing resources and data. All operating systems must deal with core issues of protection, resource management, program environment and execution, coordination, and reliable state storage and recovery. Traditionally the course emphasizes classical operating systems topics: concurrency, facilities for storage, communication, and protection, kernel services and structure, architecture/OS interaction, distributed systems, and practical application of operating system concepts in real operating systems. We also cover related principles and topics that are important for understanding modern networked software ecosystems.

Preparation. You should be familiar with undergraduate-level computer architecture and consider yourself a strong student and a good programmer. We will ask you to program in C and in Java. You should be familiar with basic data structures, e.g., lists, queues, stacks, hash tables, trees, graphs, DAGs. We will also talk a little about probability distributions and queuing theory in relation to computer system performance. You should also know the basics of the Unix command interface: see the CSL tutorials.

Readings. There is no required textbook. We will use the Web-available text: Operating Systems in Three Easy Pieces (OSTEP). The core material will be presented in lecture and notes (including almost 1000 powerpoint slides!). Students may improve their understanding by looking at one or more of the optional texts.

Base workload. In addition to the readings, there four assigned projects/labs (done in teams of 1-4), two midterms, and a final exam. Students report spending at least 10-15 hours on each lab. Many groups spend more time and a few spend much more time: as with all software development, the more bugs you create, the more time you will spend fixing them. You may choose your own groups: please follow guidelines for group submission on the course web. Here are the dates for Fall 2014:

- Sep 11: (Th) Heap manager: individual submission
- Oct 2: (Th) Shell project: group submission
- Oct 6: (M) Midterm #1
- Oct 30: (Th) Threads/elevator project: group submission
- Nov 14: (F) Midterm #2
- Nov 20: (Th) Devil filer project: group submission
- Dec 12: (F) Final exam (Friday 2:00 PM - 5:00 PM)

Late work. Graded work has deadlines. To keep things fair no extensions can be granted to individuals or groups. We may occasionally grant extensions to the entire class, or grant "free passes" for late days that all groups may "spend" as they choose. Late work receives a penalty of between 15% and 80%, depending on circumstances. It is much better to do the work and hand it in late than to receive a zero on any assignment.

Assistance. We will provide online assistance through Piazza: see the course web. Please post your questions there. Anonymous posting is allowed: please maintain a high standard of civility. In addition to

the TA we have several UTAs to assist you with the projects. The UTAs will post office hours on Piazza. The instructor holds regular office hours (posted on the course web) and is available at other times by arrangement. Drop-bys are welcome as time allows. If you are having trouble or just want to talk, please visit!

Attendance. Attendance or lack of attendance in class/recitation is not recorded.

Grading. The semester grade is determined from your exam grades (50%) and project work (50%). Additional information about grading policies, project, and exams is available on the course web. Grades are distributed through Sakai.

Topics. A list of topics and related reading is available on the course web.

Policy on collaboration for CPS 310. The Duke Community Standard applies in all aspects of this course: we value your honor and your honesty. Collaboration on lab work and project work is encouraged. Help each other. However, any work you turn in must be your own, and you may be called upon to explain (alone) your choices and approaches in more detail. You may incorporate public software into your assigned lab work and course project to a reasonable extent, but not so much as to undermine the educational purpose and spirit of the project. ***You must acknowledge any sources of your words, ideas, and software when they are not your own, and you should disclose (in advance, without any specific request) any sources you used.*** Sharing code among groups is allowable by mutual consent, however it is strongly discouraged and it will not improve your grade. All students should understand that we have software that flags copied code with a high degree of certainty and precision. (The tools do not differentiate the makers from the takers.) In this course, for practical reasons we choose to treat it as a public health problem rather than an offense under the university's academic dishonesty policy.