CS 2510: ADVANCED SYSTEM SOFTWARE
FALL 2017

Syllabus

Course Website

`http://www.cs.pitt.edu/~jacklange/teaching/cs2510-f17`

Instructor

Jack Lange, Assistant Professor
Office: Sennott Square, Office #5407
Phone: 412-648-0168
Email: jacklange@cs.pitt.edu
Office Hours: Tues. 2-4PM

Location and Time

Mon./Weds., 3:00 - 4:15 PM
Sennott Square, Room 5313

Prerequisites

Required	Knowledge of C and C++
Required	Unix development experience (gcc, gdb, make, etc)
Highly Recommended	Unix systems programming experience
Highly Recommended	CS1550

TextBooks

Required

Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
Operating Systems: Three Easy Pieces
Available online at `http://pages.cs.wisc.edu/~remzi/OSTEP/`

Recommended

Intel X86 Architecture manuals
Vols. 2 and 3 are necessary for Projects

Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman
Linux Device Drivers
Available as a free set of PDFs

Robert Love
Linux Kernel Development
Addison-Wesley Professional, 2010
If you think you are interested in kernel development read this first

Daniel P. Bovet and Marco Cesati
Understanding the Linux Kernel, 3rd edition
O'Reilly, 2005

Overview

CS 2510 is a graduate level course covering the design and implementation of past, current and future system software architectures. Throughout the course we will examine a wide range of architectures and features, as well as explore a variety of research topics. This class will consist of four main components:

- Lectures covering fundamental OS topics
- Reading research papers and other course material
- Completing assigned projects throughout the course
- Both a midterm and a final on the material covered

Lectures will be presented each Monday and Wednesday and will introduce and cover advanced OS topics and concepts. It is assumed that students have already taken an undergraduate level OS course. The material in this course will cover new advanced topics as well as going into greater detail and depth on OS and system design and implementation.

Projects

Over the course of the quarter, you will implement a set of OS functionality in a variety environments. The projects will include implementing synchronization primitives in user level, virtual memory and paging in the Linux kernel, and thread scheduling and synchronization. Each of these projects will be doable inside a standard Linux environment, but project 2 will require root level access. We highly recommend that all students install and configure a virtual machine environment such as VMWare or KVM and install an Ubuntu LTS 16 environment into it. This approach will make working on the projects much easier. Each project will consist of a code skeleton which you will flesh out to provide the required functionality.
Warning: If you do not know how to install and configure a Linux environment on your own, then you should either drop this course or be prepared to spend a considerable amount of time gaining the necessary background knowledge.

- 10% Project 1 (Synchronization primitives)
- 20% Project 2 (Virtual memory)
- 20% Project 3 (Scheduling)

Reading Papers

You will be responsible for reading and being prepared to discuss any assigned paper before each class. You may find the following brochure useful: Efficient reading of papers in Science and Technology by Michael J. Hanson, 1990, revised 2000 Dylan McNamee.

Grading

- 50% Projects
- 25% Midterm
- 25% Final

Projects will be completed in groups of two students each. While it is permissible to discuss high level details of the course project, sharing or copying another student's code is prohibited.

Communication

- **Website** - Announcements will be made both in class and via the course website. Please check it regularly for clarifications and corrections. Project materials will also be published there.
  Web address: `http://www.cs.pitt.edu/~jacklange/teaching/cs2510-f17`

Topics

- Introduction - OS/HW Background
- Processes and Threads
- Scheduling
- Locking
- Synchronization
- Time Sources and Timing
- Consistency Models
- Virtual Memory
- Paging and TLBs
- IPC and RPC
- Local File Systems (Fat32, FFS, LFS)
- Network File Systems (NFS and AFS)
- Distributed File Systems (GoogleFS and HDFS)