CS 1651: Advanced System Software
Spring 2019

Syllabus

Course Website

http://www.cs.pitt.edu/~jacklange/teaching/cs1651-s19

Instructor

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

Location and Time

Tues./Thurs., 6:00 - 7:15 PM
Sennott Square, Room 5129

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

There is one required textbook for this course, plus we will be reading technical documents and research papers in the field. Readings will be assigned weekly.

TextBooks

Required

Robert Love
Linux Kernel Development, 3rd edition
Addison Wesley, 2010

Recommended

Jonathan Corbet, Alessandro Rubini, and Greg Koah-Hartman
Linux Device Drivers, 3rd Edition
O'Reilly, 2005
Available for free as downloadable PDFs

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

Richard Stevens
Advanced Programming in the Unix Environment
Addison-Wesley, 1992
A basic book for anyone writing programs that run under Unix

Maurice J. Bach
The design of the UNIX Operating System
Prentice Hall, 1986

Overview

CS 1651 is an honors leve undergraduate 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 4 assigned projects throughout the course
- A cumulative final covering all material covered

Lectures will be presented each Tuesday and Thursday and will introduce and cover basic concepts.

Projects

Over the course of the quarter, you will implement a set of low level systems projects both at user and kernel level. The projects will include implementing synchronization primitives, virtual memory and paging in the Linux kernel, and thread scheduling. Each of these projects will be doable inside a standard Linux environment, but projects 2 and 3 will require root level access. We highly recommend that all students install and configure a virtual machine environment such as VMWare, VirtualBox (cross-platform), KVM (Linux), Hyper-V (Windows), or Xen and install a Ubuntu LTS 18 environment into it. This approach will make working on the projects much easier. Furthermore, all grading will be done on an Ubuntu LTS 18 system. While you may use a different distribution for development, it is your responsibility to make sure it works with the evaluation environment before submission.
Warning: If you do not know how to install and configure a Linux environment on your own, then you should either be prepared to spend the time needed to gain the necessary background knowledge.

- 12.5% Project 1 (Synchronization primitives)
- 12.5% Project 2 (Virtual memory)
- 12.5% Project 3 (Swapping)
- 12.5% Project 4 (Scheduling)

Reading Papers

You will be responsible for reading and being prepared to discus the assigned paper before each class. Your final grade will depend on the level of participation you bring to the course which will be evaluated by the amount you contribute to the in class discussion.
When reading papers it is normally useful to write down a summary of about a page. This summary should generally include:

- Paper title and its author(s).
- Brief one-line summary.
- A paragraph of the most important ideas: perhaps a combination of their motivations, observations, interesting parts of the design, or clever parts of their implementation.
- A paragraph of the largest flaws; maybe an experiment was poorly designed or the main idea had a narrow scope or applicability. Being able to assess weaknesses as well as strengths is an important skill for this course and beyond.
- A last paragraph where you state the relevance of the ideas today, potential future research suggested by the article, etc.

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

- 60% Projects
- 10% Paper reviews & participation
- 30% Final

All course work will be completed and evaluated individually. 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/cs1651-s19`