Operating Systems, Spring 2015
Course Syllabus

Instructor:	Dr. Rafael Ubal
Email:	ubal@ece.neu.edu
Office:	140 The Fenway, 3rd floor (see detailed directions below)
Phone:	617-373-3895
Office hours:	Wednesday 10-11am and Thursday 3-4pm
Class schedule:	Tuesday and Friday, 3:25-5:05pm, 108 Snell Engineering Center

Overview

This course covers the fundamentals of operating systems (OS) design, including the theoretical, OS-generic design considerations, as well as the practical, implementation-specific challenges in the development of a real OS.

The course is organized in two parts, each accompanied with a separate course project developed by students individually or in groups. The first part deals with the system call interface between an application and the OS, the multi-process abstraction of a computing system, multi-threading libraries, and thread communication and synchronization techniques. During this part, students will get familiar with the Linux system call interface through the implementation of a command-line interpreter (Unix shell).

The second part of the course covers memory management, file systems, disk I/O, protection, and virtualization. Students will engage in a collaborative implementation of a miniature OS, developed and debugged on an x86 processor emulator, and compatible with a real x86 PC. After completing this course, students will have a first-hand experience in the development of low-level software managing modern commercial processors.

Course Objectives

- Understand the architecture of a multi-task, multi-user system, composed of the system hardware, the operating system, and the application software, together with the interfaces between them.
- Learn how multiple applications can share processing resources, and how scheduling policies affect global system performance.
- Understand how a system's physical memory is distributed across the user-level applications transparently to the programmer.
- Learn how user-level applications communicate with hardware resources through the operating system services.
- Understand how a disk device is organized to provide a secure, fault-tolerant, and intuitive view based on files and directories.
- Understand the implications in security of a multi-user system, and the motivation for protection mechanisms.
- Get familiar with the Linux system call interface through the implementation of a command-line interpreter that spawns child processes and executes built-in commands.
- Get familiar with the core features of an OS kernel through a hands-on implementation of a miniature OS compatible with a modern x86 PC.

Prerequisites

This course relies on a proficient knowledge of the C programming language, and the GNU tool set for C programming and debugging on Unix operating systems. Experience is recommended in the use of a Unix operating system (such as Linux or OS-X) at a user level, and the use of basic shell commands (`ls`, `cd`, `ssh`, `gcc`, `gdb`, ...). Although a lack of user-level experience in Unix can be easily overcome during the first weeks of the course, strong prior C programming skills are indispensable to complete every homework and project assignment. The following book is a recommended review material:

- B. Kernighan, D. M. Ritchie, "The C Programming Language", ISBN 007-6092003106

References

Most of the material presented in class is based on the references below. These references can be considered as recommended reading. All material strictly needed to complete the course will be available through class notes, or as additional electronic handouts on Blackboard.

- R. H. Arpaci-Dusseau and A. C. Arpaci-Dusseau , “Operating Systems: Three Easy Pieces ”, May 2014 (Version 0.8) , http://www.ostep.org
- A. Silberschatz, P. B. Galvin, G. Gagne, "Operating System Concepts", ISBN 978-0470128725
- T. Anderson, M. Dahlin, "Operating Systems: Principles and Practice", ISBN 978-0985673512
- R. Love, "Linux Kernel Development", ISBN 978-0672329463
- Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 2 - Instruction Set Reference

Grading

Homework	20%
Quizzes	20%
Course project	20% (+10% extra credit)
Midterm exam	20%
Final exam	20%

Homework assignments

There will be a total of 10 weekly homework assignments. Assignments will be posted on Blackboard at least 7 days before their due date, and must be submitted on Blackboard as well. Each homework assignment will typically require you to upload a PDF file with your answers (either hand-written and scanned, or electronically typed) plus a tarball (.tgz file) with your source code ready to be built and run.

Homework due dates are strict deadlines with no exceptions, specified at the end of this document. Homework solutions will be available on Blackboard automatically after the due date, so late homework will not be accepted under any circumstances. Please make sure that you submit your assignments in advance in order to avoid unexpected submission problems due to Internet connectivity issues, trouble with PDF document generation, etc.

To add some flexibility to this policy, the average grade for homework assignments will be calculated by discarding either the one that received the lowest grade or was not submitted on time at all. This exception is aimed at covering any inevitable situation that prevented you from submitting a homework assignment on time, while it also benefits those students with no missing assignment.

Midterm and final exams

A midterm exam will cover the first part of the course material. A comprehensive final exam will focus on the second part of the course, but will also include the material corresponding to the first part. The dates for both the midterm and the final exam will be announced at the beginning of the semester.

Course projects

Students will work on two projects throughout the course, either individually or in groups of at most two people:

- The first project consists in the implementation of a Unix shell, that is, a command-line interpreter for the user to interact with the OS services. The shell will support the execution of external commands by spawning child processes, as well as the execution of internal, built-in commands (`cd`, `echo`, `exit`, ...). This assignment will be due by the date of the midterm exam, announced in the beginning of the course.
- The second project consists in the implementation of a mini-OS from the ground up. The OS will be progressively developed and debugged on QEMU, a modern and efficient emulator of a x86 processors. Although you will be using the emulator for convenience, your OS would also be able to run on a real machine. This assignment will be due by the date of the final exam, also announced in the beginning of the course.

One question in each homework assignment will guide you through the implementation of one independent module that will be part of each of these projects. Preparing the project itself will consist in putting together the pieces created in individual homework assignments, with some optional additional work that can give you up to 10% extra credit.

Additional features can be either inspired in the suggestions from open-ended homework assignments, or based on ideas of your own. Extra features in your project will be graded based on creativity, novelty, usability, and coding style. You can also choose to give a 10-minute presentation on your project during the last lecture. Due to time limitations, a selection of volunteering presenters might be needed, which will be based on the quality of the projects.

Quizzes

There will be a total of 4 quizzes during the semester, on the dates specified in the schedule at the end of this document. Quizzes will have an approximate duration of 20 minutes, and will start in the beginning of the lecture time.

Attendance and Punctuality

While attendance to the lectures is highly recommended, punctuality in class is indispensable, and constitutes a basic rule of respect toward your instructor and class mates. If any particular reason forces you to come in late to class, please notify your instructor in advance.

Course Topics

Part I

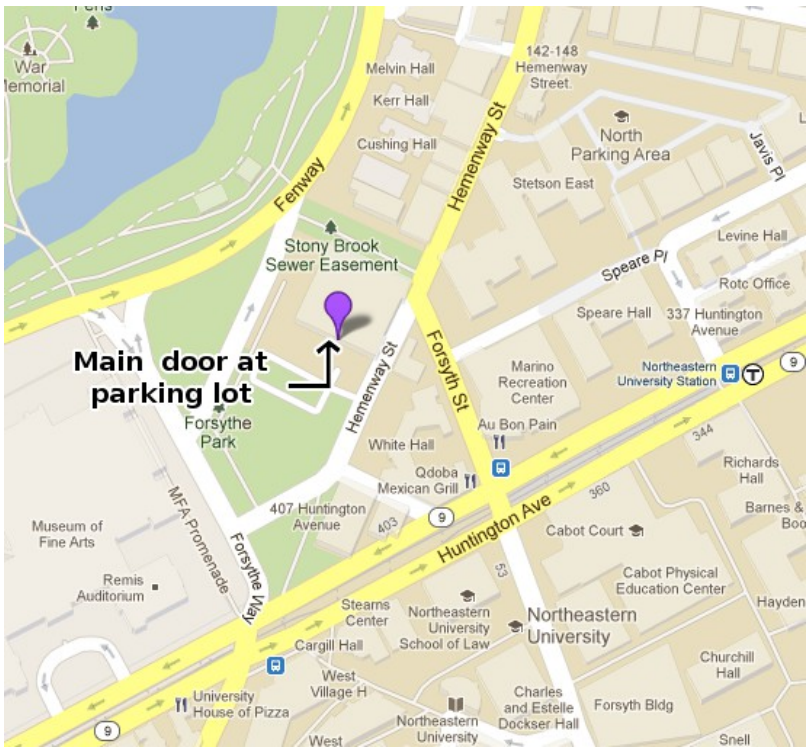
- History of operating systems, OS organization, the Linux system call interface.
- Introduction to Course Project I: the Unix shell
- Processes, context switching, interrupts, inter-process communication
- Threads, multithreading models, multithreading libraries
- Process scheduling, scheduling algorithms, priorities, real-time constraints
- Synchronization, locks and semaphores, atomic operations
- Deadlock, the dining philosophers problem, deadlock prevention, detection, and recovery

Part II

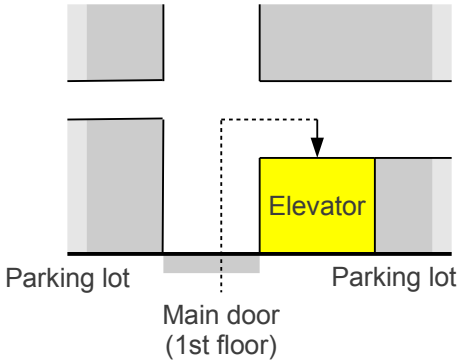
- Introduction to Course project II: implementation of a mini-OS, the x86 instruction set architecture (ISA)
- Memory management, swapping, memory allocation, paging, segmentation
- Virtual memory, page replacement policies, frame allocation, thrashing, translation look-aside buffers (TLBs)
- File systems, disk I/O, access methods, free space management, file system protection
- File system recovery, journaling, log-structured file system, network file system (NFS)
- Security, access protection, access matrix, user authentication, cryptography
- Virtualization, processor/binary emulation, virtual machine monitors (VMMs)
- Project presentations, research opportunities at Northeastern

Office Location

1) Find the office building at 140 The Fenway (TF), and enter the main door located at the parking lot.



2) Take the main elevator to the 3rd floor.



3) Once on the 3rd floor, call me at 617-373-3895. My office is in a locked research laboratory. I will meet you on the hallway right by the elevator and let you in.

Access to COE Linux Computers

The College of Engineering has currently a set of 10 Linux machines available for students in the Computer Center at 271SN (Snell Engineering). These machines can be accessed remotely through an SSH client, or physically in open lab hours. No matter which machine you choose, you will see the same files in your home folder. Information about the computer lab is available at

http://www.coe.neu.edu/computer/

To obtain a COE user account on the Linux machines, you can click on the link “Request a COE computer account” on the left. You will need to enter your name and NUID number.

In order to connect remotely, you need to connect to machine *gateway.coe.neu.edu* through an SSH client installed on your machine. On Windows, you can install Putty (or any other SSH client). On Linux/Mac, you can run command *ssh* from a terminal in order to connect to the COE machines.

Once you are connected to *gateway.coe.neu.edu* using your COE user name and password, you can run command *linux-load* to obtain a list of available machines and their current load, with an output like this:

```
Ergs: 2 users, load average: 0.05, 0.15, 0.08: last update Jun 25 13:10
Farads: 1 user, load average: 1.00, 1.00, 1.00: last update Sep  6 08:05
Grams: load average: 0.05, 0.10, 0.07: last update Aug 16 11:45
Hertz: 1 user, load average: 1.22, 1.06, 1.02: last update Sep  6 08:05
Joules: 1 user, load average: 4.15, 4.07, 4.01: last update Sep  6 08:05
Laminar: 1 user, load average: 0.33, 0.09, 0.03: last update Sep  6 08:05
Moles: 1 user, load average: 0.00, 0.00, 0.00: last update Sep  6 08:05
Nano: 0 users, load average: 0.00, 0.00, 0.00: last update Sep  6 08:05
Ohms: 1 user, load average: 0.44, 0.18, 0.06: last update Sep  6 08:05
Quark: load average: 0.00, 0.04, 0.02: last update Sep  6 08:05
```

Pick a machine which has the least users and log into it with the following command:

```
ssh -p 27 hertz
```

More information about remote connections to the Linux machines can be found at

http://help.coe.neu.edu/coehelp/index.php/Linux_Machine_Help

Important Dates

Week 1 1/11	–
Week 2 1/18	–
Week 3 1/25	Tuesday 1/27: HW #1 due
Week 4 2/1	Tuesday 2/3: HW #2 due Tuesday 2/3: Quiz #1
Week 5 2/8	Tuesday 2/10: HW #3 due
Week 6 2/15	Tuesday 2/17: HW #4 due Tuesday 2/17: Quiz #2
Week 7 2/22	Tuesday 2/24: HW #5 due
Week 8 3/1	Friday 3/6: Midterm exam
3/8 through 3/14: Spring break	
Week 9 3/15	–
Week 10 3/22	Tuesday 3/24: HW #6 due
Week 11 3/29	Tuesday 3/31: HW #7 due Tuesday 3/31: Quiz #3
Week 12 4/5	Tuesday 4/7: HW #8 due
Week 13 4/12	Tuesday 4/14: HW #9 due Tuesday 4/14: Quiz #4

Week 14 4/19	Tuesday 4/21: HW #10 due
4/27 through 5/2: Final exams (exact date TBD)	