Instructor       Prof. Rafael Ubal (ubal@ece.neu.edu)
Office           140 The Fenway, 3rd floor (see detailed directions below)
Phone            617-373-3895
Office hours     Tuesday, 10-11am
                 Wednesday, 2:30-3:30pm
                 Or by appointment outside of these time slots.
                 Please show up at the beginning of the time slot!
Class schedule   Monday, 11:45am-1:25pm
                 Thursday, 2:45pm-4:25pm
                 201 Forsyth Building
Teaching assistant Zhongliang Chen (zhonchen@coe.neu.edu)

Overview

This course covers the fundamentals of operating systems (OS) design, including theoretical, OS-
generic design considerations, as well as 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 active implementations and extensions 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 hands-on implementation of new features on 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:

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.

- Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 2 - Instruction Set Reference

Grading

Weights

- Homework 20%
- Quizzes 20%
- Midterm exam 20%
- Final exam 20%
- Course projects 20% (+5% extra credit)

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 require you to upload a PDF file with your answers, which you can produce from most common word editors (Microsoft Word, LibreOffice, LaTeX, …).

Homework due dates are strict deadlines with no exceptions. The exact due dates are specified at the end of this document. 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, problems with the submission link, etc.

To add some flexibility to this policy, the average grade for homework assignments will be calculated by discarding either that which received the lowest grade or which 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 date of the midterm is available at the end of this document. The date of the final exam will be announced during the second half of the semester.

**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.

**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 extending an open-source miniature operating system (`xv6`) with new functionality. The new features should be fully functional and tested on the QEMU full-system simulator. Even though you will be using the emulator for convenience, your OS extensions would also be able to run on a real machine. This assignment will be due by the date of the final exam, which will be announced during the second half of the semester.

Homework assignments will include the development of some modules that you will be able to reuse in the course projects. You can earn up to 5% extra credit with outstanding course projects. 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.
Grade conversion

Your final grade is calculated as a numeric grade between 0 and 100 based on the percentages shown above, and then converted into a letter grade using the following scale:

<table>
<thead>
<tr>
<th>High</th>
<th>Low</th>
<th>Grade</th>
<th>High</th>
<th>Low</th>
<th>Grade</th>
<th>High</th>
<th>Low</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>95</td>
<td>A</td>
<td>84.99</td>
<td>80</td>
<td>B</td>
<td>69.99</td>
<td>65</td>
<td>C</td>
</tr>
<tr>
<td>94.99</td>
<td>90</td>
<td>A-</td>
<td>79.99</td>
<td>75</td>
<td>B-</td>
<td>64.99</td>
<td>60</td>
<td>C-</td>
</tr>
<tr>
<td>89.99</td>
<td>85</td>
<td>B+</td>
<td>74.99</td>
<td>70</td>
<td>C+</td>
<td>59.99</td>
<td>0</td>
<td>F</td>
</tr>
</tbody>
</table>

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 classmates and myself. If any particular reason forces you to come in late to class, please let me know in advance by email or in person.

If you come to office hours, please show up at the beginning of the corresponding time slot. It is expected that multiple students have similar questions and benefit from the same answers. This will allow us to use everyone's time more efficiently.
### Teaching Assistant

#### Responsibilities

The Teaching Assistant has the following responsibilities:

- Grading the 10 homework assignments
- Grading the 2 class projects
- Grading the 4 quizzes
- Holding office hours by appointment

#### Office hours

The Teaching Assistant will be available for office hours by appointment, as requested by students through an email. The Teaching Assistant's availability will not include University holidays, such as Spring break, or Thanksgiving break. Students are encouraged to use the Teaching Assistant's office hours to discuss both content and grading of assignments, as well as to review any part of the class material.

#### Grading report

For each grading assignment, the Teaching Assistant will submit a grading report to the instructor no later than one week after the assignment submission deadline for the students. For example, assuming that an assignment is due on Feb 2 at 11:59pm, the grading report must be submitted by the Teaching Assistant no later than Feb 9 at 11:59pm.

A grading report should be sent to the course instructor in plain text in the body of an email. Each grading report should be sent in a separate email. The subject of a grading report should be formed of the course code in square brackets, the literal text “Grading report - ”, followed by the assignment name. For example:

- [EECE7376] Grading report - Homework 1
- [EECE7376] Grading report - Quiz 2
The body of the email containing a grading report should have the following sections—notice that this is confidential information between the Teaching Assistant and the course instructor:

1. Best grade, worst grade, and average grade.
2. Common mistakes from the students, including suggestions of material that should be reviewed or emphasized in class by the instructor.
3. Suggestions on how to modify, extend, or rephrase the assignment to improve clarity and fairness. These suggestions will be considered by the instructor in the design of future assignments.
4. Instances of possible plagiarism. As soon as the Teaching Assistant senses a slight hint of plagiarism, the corresponding assignment should be cited here. The instructor will later review each case individually.

**Assignment pick-up and drop-off**

Assignments involving physical hand-outs on paper, such as quizzes, will be picked up by the Teaching Assistant at the end of the class when the assignment was given, and at the same location where the lecture is held. The Teaching Assistant should be present at least 5 minutes before the lecture ends by the classroom's door, ready to pick up the material as soon as class ends.

Similarly, the Teaching Assistant is responsible for dropping off the graded material one week after the assignment was picked up, at the latest. The drop-off process is identical: the Teaching Assistant should be ready at least 5 minutes before class ends, and should come in as soon as the instructor finishes class. The instructor will give the assignments back to the students before they leave the classroom. The Teaching Assistant should feel free to drop off graded assignments at the end of a previous lecture, in those cases where the assignments were graded in less that one week.
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.

![Map of office location]

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:

```bash
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

<table>
<thead>
<tr>
<th>Week</th>
<th>Date(s)</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/10</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>1/17</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>1/24</td>
<td>Wednesday 1/27 – Homework #1 due</td>
</tr>
<tr>
<td>4</td>
<td>1/31</td>
<td>Wednesday 2/3 – Homework #2 due</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thursday 2/4 – Quiz #1</td>
</tr>
<tr>
<td>5</td>
<td>2/7</td>
<td>Wednesday 2/10 – Homework #3 due</td>
</tr>
<tr>
<td>6</td>
<td>2/14</td>
<td>Wednesday 2/17 – Homework #4 due</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thursday 2/18 – Quiz #2</td>
</tr>
<tr>
<td>7</td>
<td>2/21</td>
<td>Wednesday 2/24 – Homework #5 due</td>
</tr>
<tr>
<td>8</td>
<td>2/28</td>
<td>Wednesday 3/2 – Homework #6 due</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thursday 3/3 – Midterm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Saturday 3/5 through Sunday 3/13 – Spring break</td>
</tr>
<tr>
<td>9</td>
<td>3/13</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>3/20</td>
<td>Wednesday 3/23 – Homework #7 due</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thursday 3/24 – Quiz #3</td>
</tr>
<tr>
<td>11</td>
<td>3/27</td>
<td>Wednesday 3/30 – Homework #8 due</td>
</tr>
<tr>
<td>12</td>
<td>4/3</td>
<td>Wednesday 4/6 – Homework #9 due</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thursday 4/7 – Quiz #4</td>
</tr>
<tr>
<td>13</td>
<td>4/10</td>
<td>Wednesday 4/13 – Homework #10 due</td>
</tr>
<tr>
<td>14</td>
<td>4/17</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Friday 4/22 through Friday 4/29 – Final exams (exact date TBD)</td>
</tr>
</tbody>
</table>