Objectives and Outcomes

This course explores the computer science subfield of operating systems. A computer's operating system is its “program that runs all other programs” — it is a very special type of program, however, and so it is grounded in common concepts, theories, and algorithms. Long after the course concludes, my hope is that you will:

1. Know the fundamental concepts and algorithms behind modern operating systems, independent of a particular platform's specific policy, implementation, or interface.

2. Have a comparative and working knowledge of real-world operating systems.

3. Be able to envision, implement, and deploy common operating system functionalities or behaviors.

In addition to the course-specific content, you are also expected to:

4. Follow academic and technical best practices throughout the course.

Prerequisites/Prior Background

Programming proficiency in a systems-level language, particularly C; a prior course in computer system organization (LMU CMSI 284 or equivalent). Familiarities with Java, shell scripting, and system administration are also beneficial.

Materials and Texts


- Assorted handouts, articles, and sample code to be distributed throughout the semester

Additional information is also available on the web; do not hesitate to look for further sources of information regarding the concepts, techniques, tools, and paradigms that we will discuss.

Course Work and Grading

This course uses standards-based grading: your proficiency in each course objective is directly evaluated according to the outcomes shown on page 4 of this syllabus. Proficiency is measured according to the following key:

| + | Advanced proficiency |
| | / | Appropriate proficiency |
| / | Approaching appropriate proficiency |
| - | Needs practice and support |
| O | No basis for evaluation |

Your submitted work is used to evaluate these outcomes (see below). Letter grades are then assigned as follows:

| + | / | / | – |
| A | many | none | none |
| B | many | none | none |
| C | some | none |
| D | some |
| F | many |

A−, B+, B−, C+, and C− grades may be assigned based on “close calls” along the proficiency measure thresholds and qualitative considerations such as degree of difficulty, effort, class participation, time constraints, and overall attitude throughout the course. You may inquire at any time about the proficiency measures that I currently have on record for you.
Homework

Homework consists of questions, exercises, and programming assignments to be given throughout the semester. Homework is one mechanism for demonstrating the proficiencies expected of the course. You will be given feedback on these proficiencies, and may resubmit your homework throughout the semester in order to improve upon them.

With great flexibility comes great accountability. First off, you must submit your homework on time. The assignment due date is encoded in the homework number. Note that one of the outcomes in this class is the ability to meet deadlines (4f). Late work will detract from this outcome (duh).

Quizzes and Tests

Some outcomes are best demonstrated by answering questions or doing exercises in class. These resemble traditional quizzes and tests, but, like homework, they are evaluated according to standards and do not produce a numerical score. They are typically spontaneous and unannounced.

Questions may include content-oriented elements as well as forward-looking, applicative portions (i.e., “use this knowledge to resolve this situation”). Tests are open-paper-everything; no sharing. “Open computer” might be allowed depending on the circumstances. You may neither solicit nor give help while a test is in progress. Late and/or missed tests are handled on a case-to-case basis; in all instances, talk to me.

Term Portfolio

Your accumulated homework and tests for the semester comprise the term portfolio — the final, definitive artifact that demonstrates the proficiencies you have reached. The term portfolio provides you with an opportunity to finish and polish the work done throughout the semester; it is how you show that you learned from your mistakes or improved on already established knowledge.

Throughout the semester, you are encouraged to improve your work based on received feedback, and show it to me for re-evaluation. Improvements in proficiency are recorded and give you a good idea of how your term portfolio will fare long before its final version is submitted.

The final version of your term portfolio is due on May 4. Late portfolios detract from outcome 4f.

Extra Credit

In terms of standards-based grading, “extra credit” takes on a different meaning; it indicates work that, if successfully performed, would indicate advanced proficiency (+). Extra credit tasks may be assigned for either homework or the term portfolio. Accomplish them successfully to rack up those +’s. You do not need to perform extra credit work to show advanced proficiency; it merely demonstrates such proficiency more readily.

Version Control

Version control is an indispensable part of today’s computer science landscape in industry, the academy, and the open source community. We use version control heavily in this course: make sure that you get the hang of it.

Attendance

Attendance at all sessions is expected, but not absolutely required. If you must miss class, it is your responsibility to keep up with the course work. Note that the last day to add or drop a class without a grade of W is January 13. The withdrawal or credit/no-credit status deadline is March 16.

University Policy on Academic Honesty

Loyola Marymount University expects high standards of honesty and integrity from all members of its community. All students are expected to follow the LMU Honor Code and Process, as stated in the LMU Undergraduate Bulletin.

Americans with Disabilities Act

Students with special needs as addressed by the Americans with Disabilities Act who need reasonable modifications, special assistance, or accommodations in this course should promptly direct their request to the Disability Support Services (DSS) Office. Any student who currently has a documented disability (physical, learning, or psychological) needing academic accommodations should contact DSS (Daum Hall, Room 224, x84535) as early in the semester as possible. All discussions will remain confidential. Please visit http://www.lmu.edu/dss for additional information.
**Topics and Important Dates**

Correlated outcomes are shown for each topic. Specifics may change as the course progresses. University dates (italicized) are less likely to change.

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>Operating systems overview; operating system “power use;” networking and I/O (1a, 1f, 1g; 2a–2f)</td>
</tr>
<tr>
<td>January 13</td>
<td>Last day to add or drop a class without a grade of W</td>
</tr>
<tr>
<td>February</td>
<td>Booting and installation (1b; 3a–3c); process management (1c, 3d)</td>
</tr>
<tr>
<td>February 27 to March 2</td>
<td>Spring break; no class</td>
</tr>
<tr>
<td>March</td>
<td>Scheduling, synchronization, and deadlocks (1d, 3e); memory management (1e, 3f)</td>
</tr>
<tr>
<td>March 16</td>
<td>Withdraw/credit/no-credit deadline</td>
</tr>
<tr>
<td>April</td>
<td>Mass storage and file systems (1g, 2g); portfolio workshops</td>
</tr>
<tr>
<td>April 4 to April 6</td>
<td>Easter break; no class</td>
</tr>
<tr>
<td>May 4</td>
<td>Term portfolios due</td>
</tr>
</tbody>
</table>

You can view my class calendar in any iCalendar-savvy client such as Google Calendar or Apple iCal by subscribing to:

```
webcal://www.me.com/ca/sharesubscribe/1.9392690/M2CD-5-1-5B14D70A-E341-4026-A665-D391
D97E01B8.ics
```

(I know, it’s ugly; the link is also available on my web site for convenience.)

If necessary, this syllabus and its contents are subject to revision. Students are responsible for any changes or modifications announced in class.
Course Outcomes

1  Know the fundamental concepts and algorithms behind modern operating systems, independent of a particular platform’s specific policy, implementation, or interface.

1a  Know and understand the major parts of an operating system.
1b  Understand the operating system booting process.
1c  Understand processes, threads, and algorithms for scheduling and managing them.
1d  Know what process synchronization is, its classic scenarios, and its associated mechanisms and algorithms.
1e  Understand main memory, virtual memory, and algorithms for allocating and managing memory.
1f  Know how operating systems model and manage input and output, particularly redirection and pipes.
1g  Know the major concepts relating to files, including directories, file systems and drivers, permissions, and allocation.

2  Have a comparative and working knowledge of real-world operating systems.

2a  Know how to monitor and manage foreground and background processes in Linux, Mac OS X, and Windows.
2b  Know how to monitor memory use in Linux, Mac OS X, and Windows.
2c  Be familiar with the layouts and features of Linux, Mac OS X, and Windows file systems.
2d  Interact with operating systems across the network using ssh and vnc.
2e  Be able to set up and run virtual machines.
2f  Use ssh to reroute network connections and services.
2g  Know how to create a Linux ext virtual disk and to navigate it at the byte level.

3  Be able to envision, implement, and deploy common operating system functionalities or behaviors.

3a  Build and deploy a new Linux kernel.
3b  Create a new Linux system call.
3c  Write and run a program that invokes a new system call.
3d  Write a simple operating system shell.
3e  Implement standalone demonstrations of process synchronization scenarios and algorithms.
3f  Implement standalone demonstrations of memory management algorithms.

4  Follow academic and technical best practices throughout the course.

4a  Write syntactically correct, functional code.
4b  Provide clear, appropriate inline documentation (i.e., comments).
4c  Write code that is properly indented and spaced for human readability.
4d  Use available resources and documentation to find any required technical or developer information.
4e  Commit to version control early and often, accompanying committed changes with informative messages.
4f  Meet all designated deadlines.