

CSCI 333: Operating Systems

Syllabus – Fall 2015

Nathan Sommer
sommern@moravian.edu
610-625-7786
PPHAC 213

Office Hours: MWF 10-12
or by appointment

Course Description

A survey of the structure and organization of modern operating systems. Topics include process management, memory management, concurrency, interrupts, file systems, I/O devices, and system calls. Programming projects will involve using operating system services as well as writing modules for the Linux kernel.

Course Goals

Upon completion of this course, a successful student will be able to:

- Describe the essential components of an operating system and their respective responsibilities.
- Recognize how different algorithms for core operating system components affect the performance of application software.
- Write modules for the Linux kernel using the C programming language.
- Design and implement software comprised of components that operate concurrently.

Texts

The following text is required:

- *Modern Operating Systems, Fourth Edition* by Andrew S. Tanenbaum and Herbert Bos.

The following text is recommended:

- *The C Programming Language, Second Edition* by Brian W. Kernighan and Dennis M. Ritchie.

There are two copies of the recommended text in the CS lab, and of course there are many resources available online for learning C. However, *The C Programming Language* is a classic, concise, well-written book which still stands as the definitive book on C, and has much to offer about good programming style which can be applied to most any language. It is a good book to have around.

Course Outline

- Introduction
 - OS Overview and History
 - Hardware and Organization Review
 - The C Programming Language
- Processes, Threads, and Concurrency
 - Safe Communication
 - Scheduling
- Memory Management
 - Addressing
 - Paging
- File Systems
 - Files and Directories
 - Implementation
 - Management and Optimization
- Input/Output
 - Interrupts
 - Drivers
 - Hardware
- Deadlocks
 - Resources
 - Detection, Avoidance, and Prevention
- As Time Permits...
 - Multiprocessing
 - Virtualization
 - Security

Assignments and Tests

Your grade will be calculated based on the following items:

- **Homework** – There will be regular homework assignments throughout the semester. Assignments in this category will generally be due the class session after they are assigned and will consist of some combination of small programming tasks, written exercises, and short answer questions.
- **Plicker Points** – At the beginning of each class session I will pose at least one question to the class using plickers. Each class session will be worth one plicker point, plus one point for each question that has a correct answer. You will not be able to make up plicker points.
- **Programming Projects** – Various programming projects will be assigned throughout the semester. These projects will be more involved than the homework assignments and you will be given more time to complete them.
- **Midterm Exam** – There will be a midterm exam in class. The exam is tentatively scheduled for Wednesday, October 7. You may only re-schedule a test for college approved absences or a documented illness. In either case you must contact me *before* the beginning of the test.
- **Final Exam** – There will be a cumulative final exam given in class on Monday, December 14 at 1:30 PM.

Grading

Grades will be weighted as follows:

| | |
|-----|----------------------|
| 30% | Homework |
| 10% | Plicker Points |
| 30% | Programming Projects |
| 15% | Midterm Exam |
| 15% | Final Exam |

I will use the standard 90, 80, 70, 60 grading scale with pluses and minuses. I may relax these standards as necessary but I will not raise them.

Course Policies

- **Assignment Submission** – Most homework assignments and programming projects will be distributed through the department’s grading system. For these assignments, submission through the grading system will be the only acceptable means of submission unless I indicate otherwise. Assignments that are not distributed through the grading system may be turned in through email, Google Drive, or hard copy.
- **Late Policy** – I will do my best to grade and return assignments to you as soon as I can. Receiving late submissions makes timely grading much more difficult. As such, any submissions I receive after I have started grading an assignment will be subject to a 20% penalty for each day that it is late, starting as soon as I start grading the assignment.

If an assignment’s due date has past and I want to start grading submissions but *nobody* has submitted, the 20% penalty starts at that point regardless of the fact that there is nothing to grade.

Exceptions to this policy will be granted in special circumstances. You must let me know *before* the due date if you feel you have a valid reason why you cannot turn an assignment in on time.

- **Extensions** – I will extend an assignment’s due date for the entire class if it is clear that the original time frame was unreasonable. If you are going to bring up the possibility of a due date extension for a programming project, be prepared to demonstrate that you have already made substantial progress on the project.

I will grant personal extensions under the right circumstances. If you would like an extension for personal reasons, send me an email or come see me.

- **Absences** – You are expected to attend each class, but I understand that occasionally there will be exceptional circumstances. If you miss a class or know that you will miss an upcoming class, please contact me as soon as possible to explain the situation. You will still be expected to keep up with assignments and class content.

Plicker points cannot be made up, but I will exclude a day’s plicker points for you if you provide a valid reason why you cannot attend class.

- **Academic Honesty** – You are encouraged to discuss homework assignments and programming projects with other students. However, the work you turn in should be your own. Some similarities between code submissions are inevitable. I draw the line at identical submissions, or submissions where the only differences are purely cosmetic (i.e. variable and function names have been changed to hide code copying). If in doubt, put a comment in your code acknowledging collaboration.

See the Moravian College student handbook for more on the school-wide stance on academic honesty.

- **Disabilities** – Students who wish to request accommodations in this class for a disability should contact Ms. Elaine Mara, Assistant Director of Academic & Disability Support, located on the first floor of Monocacy Hall (extension 1401). Accommodations cannot be provided until authorization is received from the Academic & Disability Support office.

Communication

Feel free to talk to me about any issues that may arise. Email is generally the fastest way to get in touch with me if you have a quick question. If you are struggling with the material, it is generally more effective to talk in person. You are most likely to find me in my office during office hours, but I will be around other times as well so don't be shy about stopping by.

The details of this syllabus are subject to change based on our progress through the material.