CS355 - Operating Systems
Spring 2024

Dianna Xu
dxu@brynmawr.edu

MW 1:10pm - 2:30pm

A practical introduction to modern operating systems, using case studies from UNIX, MSDOS and the Macintosh. Topics include computer and OS structures, process and thread management, process synchronization and communication, resource allocations, memory management, file systems, and select examples in protection and security. This is a challenging, implementation-oriented course with a strong lab component.

Textbook

Required:

1. Modern Operating Systems, 4th Edition, by Andrew S. Tannenbaum, Prentice Hall, 2015.
   Note that there is a newer 5th Edition, but there is little difference in the materials we cover and the 4th Edition is significantly cheaper.

2. The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage, by Clifford Stoll, Pocket, 2000 (or any other paperback edition)

In addition, you will need a good reference book on the C programming language, such as: The C programm Language, 2nd Edition, by Kernighan and Ritchie, Prentice Hall, 1988

Class Time and Office Hours

- Class meets MW 1:10pm-2:30pm
- Lab meets W 11:40am - 1pm
- Office hours W 2:30pm - 4pm
- TA:
  - Joon Luther, F 7pm-9pm in Park 232

Remote Learning Resources

- Class website: `www.cs.brynmawr.edu/cs355`
- CS server: lecture notes, program data and other handouts, assignment submissions
- Slack: you will receive an invitation to join a class channel.

Learning Goals

- processes and threads, Unix system calls
- Synchronizaiton, classic IPC problems such as race conditions and deadlocks,
- processor scheduling
- memory management internals
- file system internals
- I/O and disks
- other OS topic such as multi-processor, visualization and the cloud, OS security, etc

- kernel design and implementation
- performance profiling and debugging
- experience in low-level systems programming in a realistic development environment
- Programming assignments enable students to further develop the analytical skills required to manage system complexity, creative problem solving, and engages student in collaborative and in-depth application of their high-level operating system knowledge.

Learning Outcomes

- Understanding fundamental OS components and structure.
- Understanding OS design challenges and algorithms
- Koweledge of the services provided by the OS and details of major OS concepts.
- Experience in developing low-level operating system code.
- Understanding the performance and design trade-offs in complex software systems
- Experience in developing benchmarks and test suites to evaluate the performance and robustness of systems

Prerequisites

The following courses (or their equivalents at Haverford or Swarthmore) are required with a grade of 2.0 or better (or permission of the instructor).

1. CS 231
2. CS 223 (previously CS 246)

Schedule of Topics

This schedule is *tentative*. Students should expect ***at least*** **15 hours of work each week.**

Week 1. Introduction, Unix processes and system calls

- Reading: 1.1-1.4, Chapter 1 and 2.1
- lab0

Week 2. System calls, processes and threads

- Reading: 2.1 and 1.6, 2.2
- lab1, hw1

Week 3. IPC and synchronization

- Reading: 2.3, 2.5
- lab2, hw2

Week 4. Scheduling

- Reading: 2.4
- lab3, hw3

Week 5. Deadlocks

- Reading: Chapter 6
- lab4

Week 6. Memory management, address space, swapping and partions

- Reading: Chapter 3
- hw4

Week 7. Virtual Memory and Paging, Exam 1

- Reading: Chapter 3

Week 8. Spring break

Week 9. Paging and Segmentation

- Reading: Chapter 4
- hw5

Week 10. Intro to File System, File System Implemenation

- Reading: Chapter 4
- hw6

Week 11. Sample File Systems, Free Blocks, FS Consistency

- Reading: Chatper 4

Week 12. I/O and Disks

- Reading: Chapter 5
- hw7

Week 13. Multiple Processors, Virtualization and the Cloud

- Reading: Chapter 8 and 7

Week 14. Security and Protection, Case studies and Discussions

- Reading: Chapter 9, 10

Week 15. Reivew and Exam 2

Week 16. Final exam week

Total grade breakdown

Grades will be awarded based on the number of points earned and according to the percentage breakdowns shown.

Programmong Assignments	60%
Exams	30%
Written homeworks and discussions	10%

Late work policy

There are weekly or bi-weekly programming assignments as well as written homeworks (problem sets). You are encouraged to work together, but you should write up your own solutions (except for group programming assignments). A late penalty of 10% will be imposed for each day that an programming assignment is turned in late, until 7 days past due date. Programming assignments that are more than a week late will not be accepted. The late policy is my way of acknowleding the implementation challenges you will face in this class. On the other hand, please keep in mind that late begets later and it's easy to fall behind on every assignment if you start with one. Written homeworks (problem sets) are **due in class on the day specified**. If you miss class (or forget to bring it), you can hand it in to the plastic box outside of my office. Written homeworks (problem sets) handed in late without a previously granted extension may not be accepted.

Except for circumtsances beyond your control. All extensions must be requested **at least 24 hours in advance** of the deadline. Extensions will be granted based on individual circumstances. **Time-management related problems are not valid reasons for extensions.**

Attendance and Participation

Attendance at, and active participation in, all class sessions is expected of all students. Participation will be taken into account in awarding of final grades for students who are "on the edge" between two grades. For example, a student with a B+/A- average and a strong attendance and participation record would receive an A-, while a student with a weak record would receive a B+.

Collaboration

It is your responsibility to understand and follow the collaboration policy in this class. The goal of the policy is to encourage collaboration while ensuring that you and your classmates really engage in earning how to solve the challenging problems you'll see in this course. If you are ever uncertain if collaboration or certain sources are allowed, you should ask the professor.

You are encouraged to discuss the lecture material, projects and problems with other students, subject to the following restriction: the only "product" of your discussion should be your memory/understanding of it - you may not write up solutions together, or exchange written work or computer files. Group projects are the only exceptions

to this - in this case, these collaboration rules apply to students outside of your group. Collaboration is not allowed on exams or quizzes.

Code should not be copied without permission from the author. If permission is given, code should be cited at the location it is used with a comment. If your solution is inspired by any outside resources (I understand that sometimes it is hard to not see things), you MUST cite.

Learning Accommodations

Students requesting accommodations in this course because of the impact of disability are encouraged to meet with me privately early in the semester with a verification letter. Students not yet approved to receive accomodations should also contact Deb Alder, Coordinator of Accessibility Services, at 610-526-7351 in Guild Hall, as soon as possible, to verify their eligibility for reasonable accommodations. Early contact will help avoid unneccessary inconvenience and delays.