

Fall 2015

CSCI 2467

John Finigan
University of New Orleans

Follow this and additional works at: <https://scholarworks.uno.edu/syllabi>

This is an older syllabus and should not be used as a substitute for the syllabus for a current semester course.

Recommended Citation

Finigan, John, "CSCI 2467" (2015). *University of New Orleans Syllabi*. Paper 168.
<https://scholarworks.uno.edu/syllabi/168>

This Syllabus is brought to you for free and open access by ScholarWorks@UNO. It has been accepted for inclusion in University of New Orleans Syllabi by an authorized administrator of ScholarWorks@UNO. For more information, please contact scholarworks@uno.edu.

CSCI 2467 - System Programming Concepts - Fall 2015

Instructor

John Finigan

Email

john@cs.uno.edu

Phone

(504) 280-7360 - if you leave a message, I will return your call.

Class Time

3:30 PM - 4:45 PM Tuesday and Thursday in MATH 226

Office Hours

2:00 PM - 3:30 PM Monday through Thursday or by appointment in MATH 327

Prerequisites

CSCI 2120 and 2450 or consent of department. **Fluency in Java is assumed.**

Required Textbook

Advanced Programming in the UNIX Environment by W. Richard Stevens - any edition. First edition ISBN is 0201563177. Second edition ISBN is 0201433079. Third edition ISBN is 0321637739. Newer editions are somewhat better but potentially much more expensive.

Required Download

Essential C - PDF available at <http://cslibrary.stanford.edu/101/EssentialC.pdf>

Useful Electronic Materials

Will be posted at <http://www.cs.uno.edu/~jfinigan>

1. Course Summary

Introduction to the concepts and tools used in systems programming. Detailed examination of computer architecture and computer system services from a user's point of view. Topics include accessing system services such as process control, file management, and input-output, through system calls and shells.

2. Student Learning Outcomes

At the conclusion of the course you should be able to confidently write simple C programs that use UNIX system calls. Given a UNIX systems programming problem, you should be able to identify which UNIX system facilities you may need to use to solve it, and explain why they are appropriate.

3. Final Grade Calculations

Tests - 60% - comprised of three tests weighted equally

Assignments - 40% - weighted equally

A = 90% - 100%, B = 80% - 89%, C = 70% - 79%, D = 60% - 69%, F = 59% and below.

Final grades will be rounded to the nearest integer.

4. Late Assignment Policy

An assignment is "late" if turned in after its due date and time. I will accept a late assignment at 90% credit until solutions for that assignment are distributed. As soon as solutions are distributed, I will still accept the assignment for one additional week, but at a maximum of 50% credit. After one week, I will not accept the assignment and its grade will be recorded as 0.

5. Grading Criteria and Attendance

Nearly all of your grade will be based on short answer questions, essays, and programming assignments. Unlike multiple choice testing, it is impossible to grade these mechanically. Points assigned or deducted must thus depend in part on my judgment. Sometimes I will give you an automated test case that your submissions have to pass for full credit; I will deduct up to 25 points if your submission fails the test case. I work hard to apply all grading criteria to all students equally. This means that the same error should be worth the same number of points on all submissions. I will take attendance, and attendance is strongly recommended since subject material may be presented verbally and on the blackboard and not being aware of this material may affect your performance on assignments and tests.

It is important that the programming assignments you turn in compile without warnings. I will deduct 25 points from an assignment if that assignment does not compile with gcc as installed on cook.cs.uno.edu, and I will deduct up to 10 points from an assignment that compiles with warnings when compiled with gcc -Wall. Please do not leave commented out code in your submissions; I will deduct 5 points for this. We will cover the use of these tools in class. Learning these tools is an important goal of the course.

6. Test and Assignment Dates

Test 1 - On or around October 6

Test 2 - On or around November 5

Test 3 (final exam) - Tuesday, December 8, 3:00 PM - 5:00 PM

Approximately five homeworks will be assigned, and will be due one week after they are assigned.

7. Academic Integrity

Academic integrity is fundamental to the process of learning and evaluating academic performance. Academic dishonesty will not be tolerated. Academic dishonesty includes, but is not limited to, the following: cheating, plagiarism, tampering with academic records and examinations, falsifying identity, and being an accessory to acts of academic dishonesty. Refer to the Student Code of Conduct for further information. The Code is available online at <http://www.studentaffairs.uno.edu>.

I will sometimes use automated software to detect cheating on homework. Please be sure the work you turn in is your own.

8. Accommodations for Students with Disabilities

It is University policy to provide, on a flexible and individualized basis, reasonable accommodations to students who have disabilities that may affect their ability to participate in course activities or to meet course requirements. Students with disabilities should contact the Office of Disability Services as well as their instructors to discuss their individual needs for accommodations. For more information, please go to <http://www.ods.uno.edu>.

9. Computing Resources

You should be able to do all of your work by connecting to **cook.cs.uno.edu** using a ssh client. Thus, the only software you should have to install on your computer is a ssh client. I recommend **cygwin** if you are using a Windows machine. Please see my instructions for installing cygwin and ssh at http://www.cs.uno.edu/~jfinigan/Installing_Cygwin.pdf. If you are using a Mac or Linux, you likely already have a ssh client installed. If you don't want to use your own computer, you can use the computers in one of the Department's labs, for instance in MATH 317 or 320.