CMSC 242: Introduction to Network and Systems Programming (3 credits)
Spring 2014

http://marmorstein.org/~robert/Spring2014/cs242.html

Instructor: Robert Marmorstein (marmorsteinrm@longwood.edu) 434-395-2185
Lecture: 9:30-10:45am (TR) Ruffner 350 or 2:00-3:15pm (TR) Ruffner 354
Office Hours: 1:00-2:00pm MTWRF or by appointment (Ruffner 329)
My schedule is posted near my office door. To make an appointment, please check the schedule to see which times I am free, then contact me by e-mail and list some possible times we could meet.

Course Description:
A programming-intensive class covering the fundamentals of operating systems and networking. Emphasizes the use of programming using an Application Programming Interface (API). Topics covered include threading and parallelism, low-level file system access and memory management, communication using signals, socket programming, and the TCP/IP network stack.

Prerequisite: CMSC 162.

Course Objectives:
The student will:

1. Learn to install, configure, and maintain a Linux-based development environment

2. Develop proficiency with system development tools

3. Use system calls and other functions to request and manage resources from the operating system

4. Master complex command-line and shell commands

5. Create projects that take advantage of parallel and multi-threaded programming libraries

Textbook and Other Resources:
The textbook for this class is "Advanced Programming in a Unix Environment", Third Edition, by W. Richard Stevens and Steve Rago (Addison-Wesley, 2013, ISBN: 978-032-1637-734). Since there will be extensive reading assignments, you should have your book by the first week of class. We will also make heavy use of the Unix Programmer's Manual (which you can access from any of the Linux systems in the lab using the “man” command) and other online resources available from the course web site.

Course Requirements:
This class will have both a strong programming and a strong homework component. The projects will comprise 40% of your grade. Homework and quizzes will comprise another 40%. The midterm and final exams will be worth 10% each.

Grading Policy:
Late work will not be accepted unless you have a serious medical or family condition which prevents you from completing the assignment on time. You do not need a doctor's note, but you must send an e-mail to my Longwood account within twelve hours of the assignment due date explaining the circumstances and asking to make arrangements for the work to be completed.

Slip Days:
You may extend the due date of one or more *programming labs* using slip days. You will be allocated a fixed number of slip days at the start of the semester. You may use all of your slip days on one assignment or you may use them on multiple assignments. Keep in mind, however, that once you use them up, they are gone for good.

Slip days are calculated from the minute the assignment is due until you turn it in. The number of slip days used is rounded *up* to the nearest integer value. That means that if you turn an assignment in 24 hours and 1 minute late, you will use up *two* slip days. The slip day clock runs over weekends and holidays, so if a lab is due on Friday and you turn it in on Monday, you will have used three slip days, not just one. Slip days cannot be shared, divided, traded, bought, or sold.

Grading Scale:		100-91:	A	90:	A-
89:	B+	88-81:	B	80:	B-
79:	C+	78-71:	C	70:	C-
69:	D+	68-64:	D		
63 or lower: F		(There is no grade of D- in this course.)			

Attendance:
I expect you to attend class unless you are sick or engaged in a school-sponsored sport or extracurricular activity. I will rely on your honor for enforcement of the attendance policy. You do not need a doctor's note, but you should contact me within 12 hours of the absence to explain why you missed and arrange to make up any missed work.

In accordance with Longwood policy, missing more than 10% of scheduled class time (4 class sessions) to unexcused absences may result in loss of one letter grade. Missing 25% of class or more (14 sessions), whether excused or not, may, at my discretion, result in an automatic failing grade.

Food and Drink:
Please do not eat in class (it distracts me and the other students). You may bring non-alcoholic beverages to class. Violations of this policy will be considered an unexcused absence.

I occasionally grant exceptions to this rule for students who must otherwise forgo lunch or have medical needs that require them to eat in class. If you feel that you need such an exception, you must make arrangements with me IN ADVANCE (that is, before bringing food to class).

Cell Phones and Laptops:
Cell phones, music players, and laptops are to be turned off and put away during class, except for use during the lab sessions. Violations of this policy will be considered an unexcused absence.

Honor Code:
All work in this class should be considered pledged work.

Exams and quizzes are to be completed entirely on your own. You may discuss labs, projects, and homework problems as long as the work you turn in is your own original work, typed (or handwritten) by you, with proper credit given to any sources or resources you used to complete the assignment. The purpose of these projects and assignments is for you to get first-hand knowledge of the programming libraries and techniques of this class. Any sharing of knowledge or information that defeats that purpose is cheating.

In particular, you must adhere to the following rules:

1. You must turn in a copy of your own work which YOU have typed or hand-written.

You may discuss the homework problems in the abstract, but please do not compare answers until after the assignment has been collected. On the projects, you may only turn in code that you personally have typed (except for code snippets downloaded from the course web page or otherwise distributed by the instructor).

2. You may NOT download code from the Internet (except for the course web page).

Do not download sample code, skeleton programs, or project solutions from the Internet. You may use online manuals, API documentation, or other reference material as long as you give proper credit, re-type any code you find (subject to the three line limit as described below), and do not download or copy/paste anything. You may use any format you prefer to cite a source as long as you provide sufficient information for me to find that resource and double check your work.

3. You may not share code electronically with anyone else.

This includes copying files using flash drives, cell phones, e-mail, web sites, floppies, CDs, or any other electronic storage or communication device. It also includes printouts of your code and network transfers.

4. You may not copy large blocks of code from other students or the Internet.
You MAY discuss the general design of the project with other students, but you must limit these discussions to general design details. You may also help other students with debugging, but you may not copy large blocks of code. What constitutes a “large” block of code? In general, a one-line change is acceptable, but a block of code that is more than three lines is too large.

5. You are responsible for protecting your data from copying.

You should probably set the permissions on your home directory to 711 or 700 using the “chmod” command. Source code files should probably have 600 permission. (Run “man chmod” for details)

Infractions of these policies will be dealt with harshly under the Longwood Honor Code. Any student convicted of an honor offense involving this class will automatically receive a final course grade of **F** *in addition to any penalties imposed by the Honor Board.*

Tentative Course Schedule:

Please check the course web site regularly for updated versions.

Week 1 (Jan. 14-16)	Introduction, C Review, Unix and Posix, Installing Linux *Read Chapter 1, Sections 2.1, 2.5 and 2.8*
Week 2 (Jan. 21-23)	File I/O, Files and Directories, Permissions, Basic Vim Commands *Read Chapters 3 and 4*
Jan. 21	**Last Day to Drop (by 5pm)**
Week 3 (Jan. 28-30)	Streams, Formatted I/O, and System Files, Advanced Vim *Read Chapters 5 and 6*
Week 4 (Feb. 4-6)	Linking and Loading, Processes and Process Control, Process Tracing using ltrace and strace *Read Chapters 7 and 8*
Week 5 (Feb. 11-13)	Process Groups, Signals, and Alarms, Using gdb for debugging *Read Chapters 9 and 10*
Week 6 (Feb. 18-20)	Catchup, Midterm Review, **Midterm Exam**
Week 7 (Feb. 25-27)	Threads and Threading Using valgrind for debugging *Read Chapters 11 and 12*
Mar. 3-7	**Fall Break: NO CLASS**
Week 8 (Mar. 11-13)	Daemon Processes and Advanced I/O *Read Chapters 13 and 14*
Mar. 10	**Deadline to Withdraw without an F**
Week 9 (Mar. 18-20)	IPC, Pipes, and Domain Sockets *Read Chapter 15*
Week 10 (Mar. 25-27)	Socket Programming, Advanced IPC *Read Chapters 16 and 17*
Week 11 (Apr. 1-3)	Terminal I/O and Pseudo Terminals *Read Chapters 18 and 19*

Week 12 (Apr. 8-10)	Parallel Programming with MPI, Distributed Compilation *Read tutorial linked from web site*
Week 13 (Apr. 15-17)	Reductions and Gather/Scatter *Read tutorial linked from web site*
Week 14 (Apr. 22-24)	Catchup and Review
Apr. 28	**Final Exam (Monday, 11:30am-2:00pm)** For 9:30am section
Apr. 30	**Final Exam (Wednesday, 11:30am-2:00pm)** For 2:00pm section