CMSC 242 (Spring 2015)
Introduction to Network and Systems Programming (3 credits)
http://marmorstein.org/~robert/Spring2015/cs242.html

Instructor: Robert Marmorstein **Office:** Ruffner 329
Office Phone: 434-395-2185 **E-mail:** marmorsteinrm@longwood.edu
Office Hours: 1:30pm-3:00pm MWF, 10:00am-11:00am T *or by appointment*

Lecture: 3:00 - 3:50am MWF (Ruffner 354)

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. Become proficient with system development tools

3. Develop a knowledge of system calls and library functions provided by the UNIX environment

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

Textbook and Other Resources:
The textbook for this class is "Computer Systems: A Programmer's Perspective" by Randal Bryant and David O'Hallaron, 2nd Edition, Prentice Hall, ISBN: 978-0-13-610804-7.

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.

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

Grading Policy:
Late work will not be accepted unless you have a serious medical or family emergency which prevents you from completing the assignment on time. In such cases, you do not need a doctor's note, but you must send me ***e-mail*** within a reasonable amount of time (typically twelve hours from the assignment due date) to explain your circumstances and to make arrangements for the work to be completed. All other late work will receive a grade of 0%.

Slip Days:
You will be allocated a fixed number of slip days at the start of the semester. You may use your slip days to extend the due date of one or more *programming projects*. You can use all of your slip days on one assignment or you may use them over multiple assignments.

Slip days are calculated from the minute the assignment is due until you turn it in and are rounded *up* to the nearest integer value. That means that if you turn an assignment in 24 hours and 1 minute after the due date, you will use up *two* slip days. The slip day clock runs over weekends and holidays. If a lab is due on Friday and you turn it in on Monday, you will have used three slip days, not one. Slip days cannot be shared, traded, bought, or sold, but can occasionally be earned by participation in relevant campus activities I select.

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. Anything below a 64 is failing.)			

Attendance:
I expect you to attend class unless you are sick or engaged in a school-sponsored sport or extracurricular activity. Please do NOT come to class if you are sick. Instead, contact me within 12 hours of the absence to check whether you've missed any work and make arrangements to make up any missed quizzes. You should also make arrangements to get notes from another student in the class. You should also check the course web site for announcements, new assignments, and other important updates.

I will rely primarily on your honor for enforcement of the attendance policy. However, I will keep a record of your attendance as required by Longwood policy. In accordance with that policy, missing more than 10% of scheduled class time (5 class sessions) to unexcused absences may, at my discretion, result in loss of one letter grade and missing 25% of class or more (14 sessions), whether excused or not may result in an automatic failing grade.

Food and Drink:
You may bring non-alcoholic beverages, including soft drinks, to class. However, please do not eat in class (it distracts me and the other students). 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 (i.e. before bringing food to class).

Honor Code and Collaboration:
I am a strong believer in the honor code. As such, I encourage you to actively collaborate with other students and to discuss homework problems. However, there is a point at which collaboration becomes cheating and I deal harshly with cheating in my courses. To help you understand the line between acceptable discussion of a project and dishonorable behavior, I ask you to observe the following rules:

1. Exams and quizzes are to be completed entirely on your own.

2. On all other assignments, everything you turn in should be something YOU have personally typed or hand-written. You may NOT copy code electronically from other students or the Internet.

The work you submit should, in general, be your own original work or material which I have provided and you have suitably modified by yourself.

This doesn't mean you can't look online for help with a project. It just means that you must re-type any code you find and not just download it or copy/paste it. You may not share code with other students using flash drives, cell phones, e-mail, web sites, floppies, CDs, or any other electronic storage or communication device unless I specifically direct you to do so. You may not print out copies of your code to share with other students (personal copies or copies to show me are fine).

3. Do not copy large blocks of code from other students or the Internet.

You MAY assist other students or get assistance with simple problems like syntax errors, but you may NOT copy large blocks of code from each other. A good guideline of what "large" means is that copying one or two lines of code is usually okay, but copying more than three complete statements is usually too much.

4. You must give proper attribution.

Whenever you receive help or use an online resource, you should comment your code to give proper credit. A simple comment like:

/ based on http://codewarrior.com */*

or

/* Jessica helped me with the curly braces here */

is fine. This comment should go directly above or on the same line as the code on which you received help, so that it is clear exactly which parts of your program are original and which are not.

5. You are responsible for securing your code.

Helping other students to cheat is also cheating. Furthermore, it is your responsibility to make sure that other students do not use your work to cheat. Be careful with who you let access your computer and report any missing files, flash drives, or other devices to me promptly.

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. You should consider all work in this class to be pledged work, whether or not the pledge appears on the assignment.

Tentative Course Schedule:

Please check the course web site regularly for updated versions.

Week 1 (Jan. 14-16)	Introduction, C Review, Unix and Posix, Linux Fundamentals *Read Chapter 1*
Jan. 20	**Last Day to Drop (by 5pm)**
Week 2 (Jan. 21-23)	Linking and Loading, Shared Libraries, PIC Vim Commands *Read Chapter 7* *Linux Lab Due: Friday, Jan. 23*
Week 3 (Jan. 26-30)	Exceptions, System Calls, Process Control, Alarms Process Management and Tracing Commands Advanced Vim *Read Chapter 8, Sections 8.1-8.4* *Vim Lab Due: Friday, Jan. 30*
Week 4 (Feb. 2-6)	Signals and Non-local Jumps, Memory functions *Read Chapter 8, Sections 8.5-8.8*
Week 5 (Feb. 9-13)	File I/O, Files and Directories, Permissions Debugging with gdb and valgrind *Read Chapter 10* *Shell Lab Due: Friday, Feb. 13*
Week 6 (Feb. 16-20)	**Catchup, Midterm Review, Midterm Exam**
Week 7 (Feb. 23-27)	Memory Management and Memory Errors *Read Chapter 9, Sections 9.8-9.12*
Mar. 2-6	**Fall Break: NO CLASS**
Week 8 (Mar. 9-13)	Networks and Socket Programming *Read Chapter 11* *Malloc Lab Due: Friday, Mar. 13*

Mar. 9	**Deadline to Withdraw without an F**
Week 9 (Mar. 16-20)	Process-Level Concurrency, Fork and Exec, I/O Multiplexing *Read Chapter 12, Sections 12.1-12.2*
Week 10 (Mar. 23-27)	Threads and Threading, Shared Variables, Semaphores *Read Chapter 12, Sections 12.3-12.8*
Week 11 (Mar. 30-3)	IPC, Pipes, and Domain Sockets *Proxy Lab Due: Friday, Apr. 3*
Week 12 (Apr. 6-10)	Parallel Programming with MPI, Reductions and Gather/Scatter *Read tutorial linked from web site*
Week 13 (Apr. 13-17)	GPU Programming *Read tutorial linked from web site* *MPI Lab Due: Friday, Apr. 17*
Week 14 (Apr. 20-24)	Catchup and Review
Apr. 27	**Final Exam (Monday, 11:30am-2:00pm)**