Instructor: Robert Marmorstein   **Office:** Rotunda 331
Office Phone: 434-395-2185   **E-mail:** marmorsteinrm@longwood.edu

Office Hours: 3:00pm – 4:00pm MF, 10:00am – 10:50am TR, or by appointment (Rotunda 331)
Lecture: 12:00pm – 12:50pm MWF, Rotunda 352

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:
By the end of the course the successful student will be able to:
1. Use the Linux command line and system tools to effectively develop software

2. Create programs which use system calls and library functions to access files, control threads and processes, and manage memory

3. Implement algorithms using parallel and multi-threaded programming

4. Write client and server applications that communicate using TCP/IP sockets

Textbook and Other Resources:
The textbook for this class is "Computer Systems: A programmer's perspective", Randal E. Bryant and David R. O'Hallaron, Third Edition, Pearson, 2014, ISBN: 978-0134092669.

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 on-line resources available from the course web site.

All projects and assignments in this class are to be completed on a Linux system.

You may use the systems in the Advanced Computing Lab, but I strongly recommend that you also install Linux directly on your laptop. Learning how to set up a new operating system on your computer is a valuable computing skill that you should master while at Longwood and having access to the "manpages" and other resources while at home can make the projects much easier. I am happy to help you get things working, but it is your responsibility to make sure you have access to a Linux system.

Course Requirements:
This class will have both a strong programming and a strong homework component. The projects will comprise 40% of your grade. Participation will be worth 5%. Homework and quizzes will count for 35%. The midterm and final exams will each be worth 10%.

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 and may result in a penalty against your final course 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).

Grading Policy:

Late work will not be accepted (and will receive a grade of 0%) unless you have a serious medical 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 within twelve hours of the due date) to explain your circumstances and make arrangements for the work to be completed.

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+	68-65:	D			
64 or lower: F		(There is no grade of D- in this course. Anything below a 65 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. It is your responsibility to 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 also keep a record of your attendance. In accordance with Longwood policy, missing more than 10% of scheduled class time (4 class sessions) to unexcused absences may, at my discretion, result in loss of one letter grade and missing 25% of class or more (roughly 10 sessions), whether excused or not, may result in an automatic failing grade.

Honor Code and Collaboration:

I believe wholeheartedly in the honor code. As such, **I encourage you to actively collaborate with other students and to discuss homework problems and lab projects.** 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.** All exams and quizzes will be closed-book, closed-notes unless I specify otherwise (usually by providing you material I have selected as a reference for the exam such as an ASCII table or other information source).

2. On all other assignments, everything you turn in should be something I have provided or something YOU have personally typed or hand-written. You **MAY NOT** copy code electronically from other students or the Internet. You **MAY NOT** use generative AI in **ANY** way.

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 use the Internet to look up topics related to the class. It DOES mean that you should re-type any code you find online and not just download it or copy/paste it.

It also means you **MAY NOT** transfer code using Slack, E-mail, USB flash drives, Floppy Disks, or other any other electronic storage or communication device – unless I specifically direct you to do so.

You **MAY** print out copies of your code to bring to office hours or as a personal reference. You **MAY NOT** print out copies of your code to share with other students.

3. *Limit your sharing to general topics or small pieces of code.*

The purpose of the projects and homework exercises is for you to demonstrate mastery of the material.

You **MAY** provide or get assistance with simple problems like syntax errors.

You **MAY** assist other students with their projects and homework assignments or receive help on these assignments as long as you discuss only the general problem or the process of obtaining a solution. When helping someone with an assignment, point them to the appropriate material from lecture or the textbook – or walk them through a related example that illustrates the process they need to solve the problem.

You **MAY NOT** share a complete solution to a project or homework problem.

You **MAY** compare answers once you have both worked out a solution, but you **MAY NOT** share answers unless both of you have at least attempted the problem independently. *You should never turn in any code or answers that you don't understand well enough to explain to me without help.*

Again, you **MAY NOT** use generative AI in any way.

4. **You MUST give proper attribution.**

Whenever you receive help or use an on-line resource, you must acknowledge your sources. In projects, should do this by placing comments in the code. A simple comment like:

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

or

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

is fine. The 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. On homework assignments, indicate that you have received help on a problem by making a note in the margin near the problem on which you collaborated. When in doubt, ALWAYS cite your source.

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, etc., 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.

University Policies: This course adheres to the university policies found at
http://www.longwood.edu/academicaffairs/syllabus-statements/.

Disability Accommodations Policy: If have a disability and require accommodations for this course, I am happy to work with you, but you must (prior to receiving accommodations) do two things:

      1. Register with with the Accessibility Resources Office in Brock Hall

      2. Schedule a meeting with me early in the semester to discuss a plan for your accommodations.

Should you need temporary accommodations due to an injury or illness, you should reach out to both me and ARO and we can discuss how to meet your needs while you work with ARO to address your long-term needs.

Tentative Course Schedule:
      Please check the course web site regularly for updated versions.

Week 1 (Jan. 14-16)	Introduction, Linux Shell Fundamentals, Man Pages, C Programming, Strings and Console I/O in C *Read Chapter 1*
Week 2 (Jan. 21-23)	Libraries and Linking, Makefiles, Using vim *Read Chapter 7* **Library Lab Due: Jan. 30th**
Jan. 22	**Last Day of Add/Drop (by 5pm)**
Week 3 (Jan. 26 – 30)	Processes and Jobs, Interrupts and Exceptions, System Calls, Fork and Exec, Wait and Exit *Read Sections 8.1-8.4* **Beetle Lab Due: Feb. 9th**
Week 4 (Feb. 2 – 6)	Signals and Non-local Jumps *Read Sections 8.5 – 8.8*
Week 5 (Feb. 9 – 13)	Direct (System) File I/O, Directories Debugging with gdb and valgrind *Read Sections 10.1-10.9*

Shell Lab Due: Friday, Feb. 20th

Week 6 (Feb. 16 – 20)   **Catchup, Midterm Review, Midterm Exam**

Week 7 (Feb. 23 – 27)   Streams and Buffered File I/O
Read Sections 10.10-10.12

Week 8 (Mar. 2 – 6)   Catchup and Review

                Mar. 9 – 13   **Spring Break: NO CLASS**

Week 9 (Mar. 16 – 20)   High-level Networking: HTTP and SMTP

Week 10 (Mar. 23 – 27)   Networks and Socket Programming
Read Chapter 11
Network Lab Due: Friday, Apr. 2nd

Week 11 (Mar. 30 – Apr. 2)   I/O Multiplexing, IPC, Pipes, and Domain Sockets

Apr. 1   **Deadline to Withdraw without an F**

Week 12 (Apr. 6 – 10)   Threads and Threading
Read Chapter 12

Week 13 (Apr. 13 – 17)   Shared variables, Semaphores, and Concurrency
Proxy Lab Due: Apr. 24

Week 14 (Apr. 20 – 24)   Project Work Week

Week 15 (Apr. 27 – May 1)   Catchup and Review

May. 3 (Thursday)   Final Exam (11:30am-2:00pm)