Syllabus

CMSC 242: Introduction to network and systems programming

Spring 2022

Time:	MWF 12noon
Room:	Rotunda G54
Websites:	`https://canvas.longwood.edu/courses/1304102`
	`https://www.cs.longwood.edu/courses/cmsc242`

Introduction to network and systems programming. 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 filesystem access and memory management, communication using signals, socket programming, and the TCP/IP network stack. Prerequisite: CMSC 162. 3 credits.

Professor:	Don Blaheta
Office:	Rotunda 334
Phone:	x2191
Email:	`blahetadp@longwood.edu`
100% office hours:	Mondays 2–3pm; Tuesdays 11am–12:30pm; Wednesdays 3-4pm; Thursdays 2–3:30pm

Overview

In this class, we study principles and implementations of operating systems and networking. The operating system manages hardware resources and provides a simplified interface for programs to use these resources. Networking allows different computers to communicate and potentially act as a larger virtual system. These topics are closely related; networking is often managed by the operating system (and always requires use of the hardware it manages) and the operating system uses the network to provide services like the file system. To facilitate our study of these topics, we will write programs in the C language, which provides low-level access to the hardware and is often used in operating systems and networking.

Textbook and resources

The book for this class is *OpenCSF: Computer Systems Fundamentals* by the OpenCSF project. It is free and online.[1]

The other main resource is provided by us: you'll be given an account on the department Linux machines (if you don't already have one), and you'll do your programming work there.

You will be expected to have a computer that can connect to the internet and various reference websites, and run PuTTY or another ssh client to connect to the department Linux machines.

You will be expected to have a device (your computer, or a phone or tablet) that is capable of recording and playing audio and video, and connecting to a live meeting via Zoom.

You will be expected to have reasonable bandwidth to connect to meetings, work on assignments, and occasionally upload video, *at your home* or wherever you plan to go in the event the campus closes down. If this is likely to be a problem, contact me early to see if we will be able to work around it.

You will be expected to have (and wear) a mask or the equivalent.

Covid-19 notes

There are a number of policies specific to running a class in a pandemic that I wanted to put early in the syllabus to get your attention.

Attending class. There are two ways you can attend class: in person, or via Zoom link. Either mode of attendance is equivalent for purposes of evaluating your presence and participation; if you attend via Zoom link,

- you must have a reason, and
- you must say what it is,

but I don't need any medical detail and if it's not directly covid-related I'm not going to police that. (Basically: be an adult and make good choices.) Unlike last year, I can't promise that the Zoom experience will be anywhere near equivalent to the in-person experience; in fact, it almost certainly won't

[1] `https://w3.cs.jmu.edu/kirkpams/OpenCSF`

be. But if you are quarantined, or otherwise just can't attend in person on a particular day, zooming is better than total absence.

Important note: I plan to, in general, turn on Zoom every day; but it's possible I won't if I don't know for sure someone will be attending that way. Try to fill out the "why am I zooming today" link as soon as you know you'll need to.

Medical needs. There are a number of medical reasons why attending class in person may not be appropriate for you. Obviously, if you receive a positive Covid-19 test, you will need to remain in isolation and attend class via Zoom link. Even without a positive confirmation, if you are feeling some symptoms or have been exposed or are awaiting test results, attending via Zoom link from quarantine is most appropriate (*especially* if you are unvaccinated).

> **More serious medical needs.** If you are feeling serious symptoms of Covid-19 (or some other sickness), your priority should be on dealing with that. If you end up missing class sessions and/or assignments due to being sick, notify me when you can and then let me know when you're on the upswing so we can plan out how to get you caught up.

What if the professor gets sick? Same as for students: if I'm feeling sick, I'll zoom myself into the class. If necessary I can teach from a zoom window on the projector screen; I'll send an email with instructions as soon as I know I need to do this.

Wearing a mask. As of the start of the semester, all students attending class in person must be wearing a mask or other appropriate face covering. Coverings that are acceptable include some kinds of folded bandannas, gaiters, or scarves, as long as: it covers both your nose and mouth, with two layers of cloth, fitted relatively snugly around the edges, and reduces aerosols (i.e. it's relatively tightly woven, not very stretchy, and doesn't have an "exhaust port"). If you show up to class without a face covering, you will be required to put one on or leave. If you are medically unable to wear a mask, please contact the Accessibility Resources Office to help you work out an accommodation.

It is possible that the university's masking rules will relax at some point. Even if/when they are not generally required, I will encourage you to wear a mask if you are unvaccinated, if you are mildly symptomatic (e.g. "it's probably allergies but just in case"), or if it makes you feel safer or more comfortable to do so.

All-online? It's still slightly possible that at some point in the semester we'll have to move all-online to handle an outbreak (or that *I* will land in quarantine). Should that happen, we will migrate the course to Zoom meetings but otherwise carry on. I expect that this course will remain largely synchronous (i.e. we meet at our regular class time) even if we go remote, but some calendar dates may be adjusted.

Learning outcomes

At the end of this 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 control functionality of the operating system and network stack,
3. implement algorithms using parallel and multi-threaded programming, and
4. write client and server applications that communicate using TCP/IP sockets.

Graded work

I figure that I have on average about 9 hours of your time every week, including class time as well as reading, practice, homework, and projects. If you find you're regularly spending substantially more time than this, please do come discuss it with me, so that we can ensure you're making the most effective use of your time. The work you do for this course will be evaluated as follows:

Engagement. You need to be actively engaged in this class. Engagement comes in many forms, but I expect that you will be interacting with your classmates, and with me, both in class (in-person or Zoom) and in the Slack channel. General engagement will be evaluated in two-week blocks—so you don't need to artificially say a thing every day—and it's ok if most of your engagement is via Slack as long as *some* of it is in class (spoken or in the Zoom chat). In addition, there will be occasional required interactions via Canvas that will be considered part of the engagement grade.

Labs and Projects. I draw a distinction between a "lab" (roughly a week's worth of work) and a "project" (multiple weeks, typically with intermediate goals and some design component), but both comprise work that is chiefly programming and are grouped together here for grading purposes. (Projects will be worth more.) In both cases, you can talk amongst yourselves as long as you aren't writing each other's code; see the collaboration policy.

Homework. For some of the conceptual stuff that is not really assessed via the project work, there will be a few homework assignments. These will be due after a few days, but you will have a chance to revise it; it's also group work, so you can hand in a single copy for the whole group.

Exam. There will be one exam, a final, at the end of the semester. It will be take-home, and you will be given several days to work on it. It will be non-collaborative: **You are not permitted to discuss the exam, *at all,* with anyone other than me.**

Breakdown

Labs and projects	60%
Homework	15%
Exam	20%
Engagement	5%

Grading scale

I tend to grade hard on individual assignments, but compensate for this in the final grades. The grading scale will be approximately as follows:

A−	[85, 90)	A	[90, 95)	A+	[95, 100]
B−	[70, 75)	B	[75, 80)	B+	[80, 85)
C−	[55, 60)	C	[60, 65)	C+	[65, 70)
D−	[40, 45)	D	[45, 50)	D+	[50, 55)

While there will be no "curve" in the statistical sense, I may slightly adjust the scale at the end of the term if it turns out some of the assignments were too difficult. Final grades of A+ are recorded as an A in the grading system. Final grades below the minimum for D− are recorded as an F.

Note that *individual* grades recorded in Canvas should be accurate (and you should let me know if there's a data entry error!), but *averages* as computed

by Canvas sometimes are not, if the averaging is complex or (especially) if an individual student has a special case scenario. The reference gradebook is my own spreadsheet, and while I will try to make Canvas reflect it (including averages) as well as I can, Canvas can't always handle it.

Schedule

Labs and homeworks will will go out intermittently and typically be due 3–5 days later.

The larger assignments (exams and projects) are tentatively scheduled as follows:

	Out	**Due**
Project 1	4 Feb	28 Feb
Project 2	2 Mar	4 Apr
Project 3	6 Apr	29 Apr
Final exam	29 Apr	5 May

If any of those dates need to change, I will give advance notice.

Topics

Wk	M	W	F
	January		
1		**12** — Introductions; C vs C++; printf, scanf	**14** §§10.3–10.5 Pointer basics; C strings and other arrays; Functions, pointers as out params **Lab 1 out**
2	[MLK Day **no class**]	**19** — More work with C	* **21** §10.6, Ch 1 Heap allocation in C; Models and semantics; System architectures **Lab 1 due** **Lab 2 out**
3	**24** §2.1 Memory and the kernel	**26** §§2.2–2.5 Files and directories in C (high-level); Kernel and system calls; Processes	**28** §§10.8, 2.7 C function pointers; Events and signal handling **Lab 2 due** **Lab 3 out**
		February	
4	**31** §§3.1–3.2 Direct file I/O; File descriptors, stdin, stdout; Overview of IPC	**2** §3.3 Pipes and FIFO	**4** — Processes and pointers **Lab 3 due** **Proj 1 out**
5	**7** §3.4 Memory-mapped files	**9** — mmap, cont'd **Proj 1 prep**	**11** §§3.5–3.7 POSIX vs System V; Message passing; Shared memory **Hwk 1 out**
6	**14** — Project design work **Proj 1 design**	**16** — Design, cont'd **Hwk 1 due**	**18** ** §§3.8, 4.1–4.3 Semaphores, take 1; Networking fundamentals
7	**21** §§4.3, 4.5 Application layer protocols; HTTP	**23** §4.6 DNS	**25** §4.4 (exc 4.4.4) TCP/IP socket programming

* **20 January**: Deadline to add/drop classes (5pm)

** **18 February**: Deadline to elect pass/fail option (5pm)

Wk	M	W	F
	February		
8	**28** §5.3 (exc 5.3.1) Transport layer **Proj 1 due**	**March** **2** §5.5 Internet layer (IP) **Proj 2 out**	**4** TBA
		SPRING BREAK	
9	**14** §5.4 (esp 5.4.4–5) Transport layer security	**16** §§4.4.4, 5.3.1 Datagrams and UDP; NAT vs IPv6 **Proj 2 prep**	**18** §§5.6, 5.7 Lower networking layers **Hwk 2 out**
10	**21** Project design work **Proj 2 design**	**23** §§6.1–6.4 Design cont'd Intro to threads **Hwk 2 due**	**25** — Threads demo
			April
11	**28** — Threads with arguments	**30** * §6.5 Returning values from threads	**1** §6.6 Threads in other languages
12	**4** §§7.1–7.2 The problem of synchronisation **Proj 2 due**	**6** — Synchro cont'd; Peterson's solution **Proj 3 out**	**8** §§7.3, 7.4 Synchro with locks; Semaphores, take 2
13	**11** — Other synchro abstractions; Deadlock **Proj 3 prep**	**13** — Classic sync problems **Hwk 3 out**	**15** — Project design work **Proj 3 design**
14	**18** — Design cont'd **Hwk 3 due**	[Research Day **no class**]	**22** — Thread sync design patterns
15	**25** — TBA	**27** — TBA	**29** — TBA **Proj 3 due** **Exam out**
	May		

Exam due: Thu 5th, 5:30pm

* **30 March** : Deadline to withdraw from a class (5pm)

Week 1

Introductions, C/C++ differences, man pages

Week 2

Ch 1: models and semantics, system architectures; 2.1–2.2 Kernel memory, heap allocation and structs; structs and pointers in C

Week 3

POSIX calls, files and directories in C; Rest of Ch 2: processes, signals, fork, C function pointers

Week 4

Direct File I/O, file descriptors; 3.1-3.3 IPC, pipes, FIFO

Week 5

3.4–3.7 forms of IPC, POSIX vs SysV implementations

Week 6

3.8 semaphores, take 1; 4.1–4.3 network fundamentals and application layer

Week 7

Rest of Ch 4 HTTP and other application layer protocols

Week 8

Catch-up and exam

Week 9

Ch 5: Transport and internet layers, TCP, IP; Sockets and data transfer

Week 10

6.1–6.4 Threads and writing multithreaded code

Week 11

Rest of Ch 6: More complex thread interactions

Week 12

7.1–7.3: The problem of synchronisation, early solutions

Week 13

Rest of Ch 7: Semaphores take 2; Ch 8: The classic sync problems

Week 14

Sync problems cont'd; 9.1–9.2: Parallelism vs concurrency

Week 15

Rest of Ch 9: parallelism and scaling

Week 16

Catch-up and exam

Policies

You can find several university-wide course policies at
`http://www.longwood.edu/academicaffairs/syllabus-statements/` .

Support

I'm in my office a lot (not just during posted office hours). Feel free to come in (or try the office zoom link) and ask questions (or just to talk). If you can't catch me in my office, messaging via Canvas or Slack is probably your best bet.

Honor code policy

Above all, I ask and expect that you will conduct yourself with honesty and integrity—and not to ignore the other ten points of the Honor Code, either. Take pride in what you are capable of, and have the humility to give credit where it is due.

The two main forms of academic dishonesty are "cheating" and "plagiarism". "Cheating" is getting help from someplace you shouldn't, and "plagiarism"

is presenting someone else's idea as if it's your own. If you ever find yourself inclined towards either of these, know that there are always other, better options. Persevere! See my website[2] for some discussion and examples of how to steer clear of these problems, and feel free to come talk to me if you need help finding some of those other options (even if it's for another course).

Cheating or plagiarism (on any assignment) will normally receive a *minimum* penalty of lowering the *course* grade by a full letter, and may range at my discretion up to an F *in the course*. Cases will also be turned in to the Honor Board. But: I believe in your potential, and I hope that you will, or will grow to, observe this policy not simply to evade punishment but positively as a matter of character.

Accommodations

If you have any special need that I can accommodate, I'm happy to do so; come speak to me early in the term so we can set things up. If you have a documented disability, you should also contact Longwood's Accessibility Resources Office (Brock Hall, x2391) to discuss some of the support the college can offer you. All such conversations are confidential.

Attendance and late policy

Attendance is required, and assignments must be turned in on time. That said, if you have a good reason to miss class or hand something in late, I tend to be fairly liberal with extensions if you ask in advance. (Good reasons do include assignments due for other classes.) (And medical and family emergencies are exempted from the "in advance" part, of course. But contact me ASAP.)

Frequent absence will result in a lowered participation grade; habitual absence may in extreme cases result in a failing grade for the class. *Unexcused* late assignments will normally be given a zero.

Inclement weather policy

I don't plan to cancel class for weather unless the entire college shuts down; and if the campus closes, I'm likely to hold class in some form by zoom instead

[2] `http://cs.longwood.edu/~dblaheta/collab.html`

(check your email). If you are commuting or are otherwise significantly affected by a weather event, use your own best judgement (and remember that zoom is an option); and if you do miss class for this reason (e.g.: power's out too), contact me as soon as possible to make up missed work.

Early bird policy

Nobody's perfect, and on occasion an assignment gets written a little unclearly (or, once in a while, with an actual error in it). If you catch one and bring it to my attention early, so that I can issue a clarification or correction to the rest of the class, there'll be some extra credit in it for you.