CS 503 Systems Programming Syllabus

Instructor Information
Benji Lilley
email: benji.lilley@drexel.edu

***Preferred Communication Channels**:* Discord for general questions, Discord DM for personal questions, Email for emergencies or high priority notifications.
Office Hours: Will be posted on course blackboard page and on our class Discord Channel

Course Description

This course will introduce fundamental concepts of computer architecture and operating systems, covering the Unix environment from the perspective of an application developer and a systems programmer. Topics include scripting, languages and tools that are part of the Unix environment, as well as introduction to system programming topics, including OS processes and threads, memory management, concurrency, inter-process communication and networking.

Course Objective and Goals

The primary focus of this course is to provide an overview of how computers work, the connection between software and hardware, the core functionality of the operating system, and how applications work in the environment provided by the operating system, using the C language.

What Students Must Know Prior to this Course

1. An introductory level knowledge of the C programming language
2. Familiarity with software engineering best practices associated with managing and building source code assets. For example, git and make.
3. A working knowledge of the Linux/Unix environment including how to connect to a remote Linux system using ssh and navigating the Linux file system.

Students attempting to take this course who do not have the prerequisite knowledge outlined above should see the instructor ASAP.

Statement of Expected Learning

1. Understand how the compiler, linker and loader work to build and run programs on Linux.
2. Understanding how the Linux operating system coordinates running programs and protects hardware resources via services provided by the microprocessor. (e.g., Kernel vs User mode)
3. A deep understanding of the low-level interface between running applications and the operating system. This includes System Calls, Processes, Threads, File, File Abstractions, Signals, Interrupts and Virtual Memory.
4. Experience with writing concurrent programs using various services provided by the Linux operating systems that are exposed through the C runtime library.
5. Exposure to writing basic network programs using socket abstractions.
6. An introduction to some of the more advanced features of Linux that support modern workloads including – Cgroups, Namespaces, and Containers.

Textbook

There is no required textbook for this course. The instructor will make available class notes, sample code, and provide links to online resources that will complement this course. For students who prefer a more formal textbook, one that is highly recommended is:

> Randel Bryant and David O'Hallaron. Computer Systems: A Programmer's Perspective, 3rd Edition. Prentice Hall: 9780134092669

Also note that most students are unaware that Drexel does provide free access to the O’Reilly Learning Platform. You need to go to the library page and set up a cross-linked account with O’Reilly learning to get access – go to the library page on Drexel One for more information. There are several good C programming, Systems Programming and Linux resources available on that platform that you should check out.

Course Topics & Schedule (Roughly by Week)

Note that this is a proposed schedule and is subject to change by the instructor.

1. Course and Systems Programming Introduction

2. A Deep Dive into Compilers, Linkers and Loaders
3. Files and File Abstractions
4. Processes and Threads
5. Signals, Traps and Interrupts
6. Network Programming – Part I
7. Network Programming – Part II
8. (Virtual) Memory Management
9. Process and Resource Isolation – Namespaces, Cgroups
10. Virtualization (Hypervisors, Containers)

Software and Programming Environment

This course will focus on systems programming in a Linux environment. **Students must use a Linux environment for all assignments – no exceptions.** The CS department operates tux, a Linux cluster, with all the required tools that you will need. You may optionally use a personal device with a Linux virtual machine for coursework. Good options include the Windows Subsystem for Linux (on Windows) and OrbStack (a free tool for personal and educational use) on MacOS. These options will be discussed in Week 1. Note that if you run a personal VM you will be responsible for all technical support and software installation associated with your personal environment.

I believe that the use of tools to support programming is a personal choice, however **I highly encourage** https://code.visualstudio.com/download **along with the** Drexel CCI extension (you can find it by visiting https://marketplace.visualstudio.com and searching for "drexel"). The extension helps you manage assignment starter code and instructions. You are required to maintain the same filenames and directory structure as the starter code. **If you choose to use another toolchain, it is your responsibility to maintain the files in your assignment Github.com repo while maintaining the starter project structure**.

Submitting Assignments

All code assignment artifacts (code, documentation, build files) will be stored in private Gihub.com repositories. **These repositories will be configured and provided to you during the first week of class**. You will receive an email with instructions on how to access your Github assignment repository.

When an assignment is ready to be graded, you MUST submit a link to your solution in BBLearn. BBLearn is the only way that you may indicate an assignment is ready to be graded, and it is the only way for instructors to enter grades.

To recap, assignments will only be accepted if:

- The code and artifacts are pushed to the private Gihub.com repository provided to you via your Drexel email address
- The code and artifacts maintain the same structure as in the starter code (use the Drexel CCI Visual Studio Code extension mentioned above to make this easy)
- You submit a link to BBLearn indicating the assignment is ready for grading

Code Assignment Requirements

Most assignments in this course are code assignments. Be sure to validate your solutions before submitting them! In general:

- Your code must compile with "make". Starter code contains makefiles that you do not need to edit.
- Your code must pass unit tests with "make test"; we use BATS, a bash-based unit testing framework.

Grading and Policies

1. Assignments – Programming & Written. Note that many assignments will have extra credit components.	65%
2. Quizzes – There will be 5 online quizzes (25 pts each) in this course. Final quiz grade will be calculated: (**GRADE**: top 4 quiz scores @ 25 pts each) + (**Extra Credit:** 0.5 of your lowest quiz score). Thus, your maximum score can be 112.5%.	35%
3. Class Participation & Extra Credit. Various programming assignments will have extra credit components. Class participation, especially helping fellow students in Discord discussions, is highly encouraged.	Various
TOTAL	100%+

In general, programming assignments will be due Midnight on Fridays. Quizzes will be posted with a 48-hour availability to take them. Students are fully responsible for keeping up to date on deadlines that will be managed through Blackboard.

Final grades

A+	99 – 100+		C+	77 – 79
A	93 – 98		C	73 – 76
A-	90 – 92		C-	70 – 72
B+	87 – 89		D+	66 – 69
B	83 – 86		D	60 – 65
B-	80 – 82		F	< 60

The university's Academic Honesty policy is in effect for this course. Please read Drexel University Student Handbook found at http://www.drexel.edu/Studentlife/. On the first incident, students who share their work (even with best intentions) or otherwise violate the course or university academic honesty policy may receive a grade of F for the course (the students may not withdraw in this case). The students may be reported to the department, college, and/or University Judicial (Honesty) Board. Both the giver and the receiver will receive these penalties.

Meetings of this course are recorded

Any recordings will be available to students registered for this class. Students are expected to follow appropriate university policies and maintain the security of passwords used to access recorded lectures. Recordings, or any part of the recordings, may not be reproduced, shared with those not in the class, or uploaded to other online environments.

Assignments, Due Date and your Late Day Bank

This course will have assignments that will be spread out over the course as mentioned above. Given my previous experience, students tend to have different levels of readiness, especially with background in programming. I will adjust deadlines for the whole class as necessary, based on my observation of student effort and questions being asked over Discord, and attending office hours (mine or the TAs). One off deadline extensions are not possible, but I do realize that things come up from time to time. If you go into grade center, you will notice that you have a bank of 5 late days to use over the term. Late submissions will be accepted without penalty by deducting from your late day bank. Once you exhaust your bank, late assignments will take a 50% penalty on day 1, and will not be accepted on day 2+.

Note that the late day bank is to accommodate students “when things come up” for whatever reason. This includes work deadlines, needs of other courses, sickness, unexpected travel, and so on. They should be considered as “insurance”, because once they are gone, they are gone. You do not require any permission to submit assignments late – they will be accepted without penalty, without any reason for lateness so long as late days exist in your bank. The late day bank is intended to provide students with some flexibility in managing their academic commitments throughout the term. I often receive emails asking for extensions, while I will do my best to reply to these, if I don’t reply just assume the general response is “That is the purpose of your late day bank”.

ALSO NOTE that the late day bank is only available for assignments, it cannot be used for quizzes.

Expectations of Timely Communication for Hardships

If, for whatever reason, you find yourself struggling with keeping up with course deliverables, it is important that you let me know right away, including your plan to get back on track. I am here to help you and am willing to be as flexible as possible. It is your responsibility to be as proactive as possible in contacting me over a hardship. Do not expect any flexibility if you come to me weeks after one or more assignment deadlines have been missed with a hardship story expecting me to accept missing work without significant penalty, if even at all. I have been alarmed by the number of students who approach me towards the end of the term after they have not submitted a significant number of course deliverables with the expectation that I will accept a batch of late work. Please do not find yourself in this position, because regardless of the reason, the answer is always no, and almost all of these students ended up failing the course.

The process you need to follow: If you are having a personal hardship that is impacting your ability to submit course deliverables on time, contact me directly either by visiting my office hours, sending me a DM over Discord or via an email to discuss. What is most important to me is that you come to the discussion prepared with a plan to get back on track with your course responsibilities. Based on the outcome of our discussion, I will, at my discretion, add additional late days to your “late day bank” to accommodate your needs to the best of my ability and in fairness to the entire class. It is your responsibility to notify me as early as possible, and to get yourself back on track in the course. There is little that I can do to help you if you come to me after you have missed a significant amount of course work.

Holidays

The academic calendar will guide holidays that may, or may not happen, during the course. Changes to content and deliverables because of holidays will be communicated by the instructor.

Office of Disability Resources

Students requesting accommodations due to a disability at Drexel University need to present a current Accommodation Verification Letter (AVL) to faculty before accommodations can be made. AVL's are issued by the Office of Disability Resources (ODR). For additional information, visit the ODR website at http://www.drexel.edu/oed/disabilityResources, or contact the Office for more information: 215-895-1401 (V), or disability@drexel.edu.

University Policies

In addition to the course policies listed on this syllabus, course assignments or course website, the following University policies are in effect:

- Academic Integrity: https://drexel.edu/studentlife/community-standards/code-of-conduct/academic-integrity-policy
- Student Conduct Policy: https://drexel.edu/studentlife/community-standards/code-of-conduct/conduct-policies
- Students with Disability Statement: https://drexel.edu/disability-resources
- Course Drop Policy: https://drexel.edu/graduatecollege/forms-policies/policies-procedures-guidelines/course-add-drop-withdrawal/

The instructor may, at his/her/their discretion, change any part of the course during the term, including assignments, grade breakdowns, due-dates, and the schedule. Such changes will be communicated to students via the course web site Announcements page in BBLearn. This page should be checked regularly and frequently for such changes and announcements. Other announcements, although rare, may include class cancellations and other urgent announcements.