

# CS3411 - Systems Programming

## Fall 2024

### Course Information

**Course Name:** CS3411 - Systems Programming

**Class Location:** Fisher 139

**Class Time:** Tuesday & Thursday, 9:30am–10:45am

### Instructor Information

**Instructor:** Dr. Soner Onder | Rekhi 303

**Office Hours:** By appointment only

**Email:** [soner@mtu.edu](mailto:soner@mtu.edu) (include CS3411 in subject)

*Please direct questions to the TA first if possible*

### Grade Scale

Percentage	Letter Grade
> 0.93	A
[0.88–0.93)	AB
[0.83–0.88)	B
[0.78–0.83)	BC
[0.73–0.78)	C
[0.68–0.73)	CD
[0.60–0.68)	D
< 0.60	F

### TA Information

**TA:** Caleb Swain | Rekhi 323

**Office Hours:** Tue/Thur | 10:45am-12:15pm

**Email:** [coswain@mtu.edu](mailto:coswain@mtu.edu) (include CS3411 in subject)

### Course References

**Course Website:** <https://mtu.instructure.com/courses/1516301>

#### Course Text:

**Recommended:** [Advanced Programming in the UNIX Environment](#), Third Edition, by W. Richard Stevens and Stephen A. Rago

**Further Reading:** C: A Reference Manual, Fifth Edition, by Samuel P. Harbison, III and Guy L. Steele, Jr.  
UNIX Network Programming, Volume 1–Networking APIs: Sockets and XTI, Second Edition, by W. Richard Stevens

### Course Description

This course covers basics of systems programming in a UNIX environment. We will cover operation of fundamental elements of the UNIX operating system interface, as well as certain scripting tools commonly used to manage UNIX system software resources. Systems programmers develop programs that often run in a privileged mode, with unrestricted access to system resources, and that are fundamental to correct operation of the programming platform presented to users. A primary goal of the course then is to reinforce, in a systems programming context, the skills you have already acquired to develop code that is robust.

# Learning Objectives

After completing this course, successful students should be able to:

## **C Programming:**

- Program a sophisticated program in C using provided abstractions
- Be able to program abstractions based on given requirements

## **UNIX File System Interface:**

- Use abstractions UNIX provides to interact with the filesystem
- Manipulate the filesystem in a UNIX system using C by using system calls

## **UNIX Processes:**

- Understand and use the abstractions UNIX provides to create and manage processes in a UNIX system
- Program process creation and program execution using system calls in C in a UNIX system

## **Linking and Libraries:**

- Understand how C modules are linked together to form complete programs and libraries
- Understand, use, and create static and dynamic libraries using C in a UNIX environment

## **Interprocess Communication:**

- Communicate between multiple processes in a UNIX system using signals, both within the terminal and using C
- Communicate between multiple processes in a UNIX system using pipes, both within the terminal and using C
- Communicate between multiple processes in a UNIX system using sockets, including domain sockets as well as network sockets

## **Terminal I/O:**

- Understand and use the UNIX terminal properties appropriately within a terminal, as well as from within C

## **Time and Timing:**

- Understand and use the different UNIX abstractions related to time measurement within C

# Grading Policy

All course scores will be kept in Canvas. Your performance in this class will be evaluated in programming assignments and exams. Your grade will be calculated as a percentage of points received weighted for each part of the course:

### **60% Programming Assignments**

### **40% Exams** (2 exams weighted 20% and 20%)

Your assignments and exams should be graded and returned to you within about 7-10 days. Any questions about a grade received (or request for regrades) should be made after that point except in the case of an arithmetic error in summing points or the grade was recorded incorrectly in the grade book.

The final score will determine your letter grade. See grading scale on first page for distribution. **(Note: This scale is subject to change, if a change does occur the cutoffs will only be lowered)**

# Late Assignments

You may use two (2) late passes for your programming assignments through the entire semester. Using a late pass means you may submit a programming assignment one day late with no penalties. You may not use both your late passes on the same assignment. After your late passes have expired, you may submit your assignments for grading with a 20% penalty for each additional late day.

# Programming Expectations

The grading criteria for submitted projects includes readability, robustness, and efficiency. Correct operation on normal input will not achieve an A grade. Submitted programs are expected to terminate normally under reasonably foreseeable errors in user input and are expected not to contain gross inefficiencies in resource usage.

# Exams

The exams for this class are worth 40% of the total and will cover topics from the lectures. The exams will be closed book. Exams are announced at least 7 days ahead of the test date.

## **Makeup exams:**

Makeup exams will not be given without prior arrangements. Excused absences as described in the Student Handbook must be emailed to the instructor 3 days prior to the test date. Unplanned events (illness, emergencies) should be brought to the instructor's attention as soon as possible to make alternative arrangements. Unexcused missed exams will result in a zero for that exam.

# Attendance

Attendance is required for the class, and will be taken. A significant amount of missed classes will affect your overall letter grade at the end of the semester. If you will be required to miss a large number of classes, you should discuss a make up plan with the instructor. Considerations will be made for emergencies.

# Class Policy and Communication

Cellphones, laptops, iPads, etc. are not to be used during class time. Class time is not a time for you to work on other things, talk with other people, or be disruptive in any manner. The exception to this policy is for the express reason of taking notes.

## **Communication:**

In addition to class time, students should refer to the course website (on Canvas) and class mailing list to receive announcements and information on the class. Make sure you're able to receive announcements on Canvas in your email inbox as soon as they are posted.

# Collaboration and Cheating

## **Programming Assignments:**

Programming projects are to be performed individually. You may neither show anyone your project code nor look at the code of anyone else (This policy extends to any external resource, including code found on the web or individuals who are not enrolled in the course.). However, you may use code found in provided lecture slides, examples shown in class as well as manual pages of UNIX. You may also engage in empty hands discussions with anyone. No participant in an empty hands discussion should leave the discussion with written or printed material. If you are unsure whether or not a particular type of collaboration is allowed, you are expected to check with the instructor before engaging in the collaboration.

## **Exams:**

You are to complete all exams with no help from other students, and with no textbooks, class notes, cheat sheets/notecards, or any electronic equipment, etc. (unless approved in advance by the instructor).

## **Getting Assistance:**

If you have questions on the course, I ask you to consider the following options:

- Email the TA/instructor a clear and detailed question
- Schedule a meeting with the TA/instructor
- Stop by the Computer Science Learning Center (CSLC).

## **Academic Misconduct:**

Academic misconduct in any form will not be tolerated. Evidence of misconduct will result in zero credit for the assignment, drop in one final course grade, and notification with the Office of Student Affairs.

# University Policy

## **Assessment Statement:**

Student work products (exams, essays, projects, etc.) may be used for purposes of university, program, or course assessment. All work used for assessment purposes will not include any individual student identification.

## **Other Policies:**

If you have a disability that could affect your performance in this class or that requires an accommodation under the Americans with Disabilities Act, please see the instructor as soon as possible so that we can make appropriate arrangements.

Michigan Tech has standard policies on academic misconduct and complies with all federal and state laws and regulations regarding discrimination, including Americans with Disabilities Act of 1990. For more information about reasonable accommodation for or equal access to education or services at Michigan Tech, please call the Dean of Students Office, (906) 487-2212 or go to [http://www.mtu.edu/ctl/instructional-resources/syllabus/syllabus\\_policies.html](http://www.mtu.edu/ctl/instructional-resources/syllabus/syllabus_policies.html)

## **Academic Integrity:**

<http://www.mtu.edu/conduct/integrity-center/students/>

## **Institutional Equity and Inclusion:**

<http://www.mtu.edu/equity/>

## **Disability Services:**

<http://www.mtu.edu/deanofstudents/disability/>