CSE 3100 Systems Programming – Spring 2022

Updated 03/24/22

Lecture

Mon/Wed 10:10am–11:00am, Storrs Hall WW16

Lab

Section 001L: F 12:20PM - 2:10PM, E2 306
Section 002L: F 10:00AM - 11:50AM, E2 306
Section 003L: F 2:30PM - 4:20PM, E2 306
Section 011L: F 2:30PM - 4:20PM, ITE 134
Section 012L: F 10:00AM - 11:50AM, ITE 134
Section 013L: F 12:20PM - 2:10PM, ITE 134

Instructor

Ion Mandoiu
ion@engr.uconn.edu
Office Hours:
M/W/F 11:30am–12:30pm
ITE 261

Graduate Teaching Assistants

Rye Howard-Stone
rye.howard-stone@uconn.edu
Office Hours:
N/A

Yiming Zhang
yiming.zhang.cse@uconn.edu
Office Hours:
N/A

Undergraduate Teaching Assistants

Jonathan Ameri
jonathan.ameri@uconn.edu
Office Hours:
M 6-8pm, Tu 1-3pm, W 2:30-4:30pm
ITE 114

Thomas Crosby
thomas.crosby@uconn.edu
Office Hours:
Tu/W/Th 12-1pm
ITE 114

Andrew Fang
andrew.fang@uconn.edu
Office Hours:
M 5-6pm, F 5-7pm
ITE 114

Alexander Gmuer
alexander.gmuer@uconn.edu
Office Hours:
F 12-1pm
ITE 114

Aniruddah Manikandan
aniruddah.manikandan@uconn.edu
Office Hours:
W 12-2pm, F 1-4pm
ITE 114

Samuel Oslovich
samuel.oslovich@uconn.edu
Office Hours:
M 3-5pm, Th 2-3pm
ITE 114

Nicholas Pang
nicholas.pang@uconn.edu
Office Hours:
W 7:30-9pm, Th 5-6:30pm
ITE 114

Matthew Spinelli
matthew.spinelli@uconn.edu
Office Hours:
Tu 5-6pm, W 4:30-6:30pm
ITE 114

James Zhu
james.zhu@uconn.edu
Office Hours:
M 1-3pm, Tu 6-7pm
ITE 114

For latest office hours see Piazza.

Catalog Description

Introduction to system-level programming with an emphasis on C programming, process management and small scale concurrency with multi-threaded programming. Special attention will be devoted to proficiency with memory management and debugging facilities both in a sequential and parallel setting.
Prerequisite: CSE 2050 or 2100

Course objectives

The goal of the course is to introduce students to basic operating systems concepts, programming techniques, and modern development tools such as integrated development environments, debuggers, and profilers that enable them to design and implement efficient systems programs. Specific objectives include:

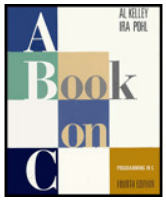
- Becoming proficient at C programming including the use of pointers and memory management.
- Understanding inter-process communication primitives such as pipes and sockets and how to use them to build client-server applications.
- Understanding intra-process concurrency primitives, in particular, POSIX threads and synchronization primitives such as mutexes and condition variables needed to write multi-threaded applications

Learning Outcomes

Upon successful completion of the course, students are be able to:

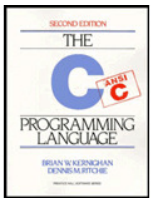
- Write C programs that use dynamic memory allocation.
- Write C programs that create, manage and terminate processes and threads on UNIX.
- Write C programs that use UNIX synchronization primitives.
- Develop simple client-server applications that communicate across a network.

Required Textbook

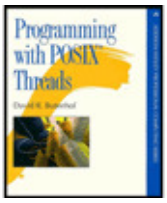


Al Kelley and Ira Pohl
A Book on C, 4th Edition
Addison-Wesley
Book website including source code for all examples:
https://users.soe.ucsc.edu/~pohl/abc4.html

Optional Textbooks



Brian W. Kernighan and Dennis M. Ritchie
The C Programming Language, 2nd Edition
Prentice Hall



David R. Butenhof
Programming with POSIX Threads, 1st Edition
Addison-Wesley

Online platforms

Moodle. We will use a course website hosted using Moodle at edx.engr.uconn.edu. Please use the Moodle site to access quizzes, interactive videos, programming assignments, grades, and other course materials.
Piazza. For electronic class discussions we will be using Piazza, which can be accessed from Moodle (the first access will ask you to confirm joining our Piazza class if you have not done it already). You are strongly encouraged to ask class-related questions and communicate with other students, the instructors,

and the TAs via Piazza rather than e-mail. Please observe basic etiquette by keeping your messages polite, concise, and on-topic. Before posting new messages do take a look at the postings that are already there–it is possible that your question has already been answered. Appropriate questions include general questions about the material covered in class and clarifications on the assignments. Keep in mind that the collaboration policy is in effect and you must not post extensive code fragments in public messages. For questions that are specific to your work use direct messages to the instructors or the TAs.

Code-server IDE. To ensure a consistent development environment, each student will have access to a VS Code IDE running on a 4-core virtual machine and accessible via any modern web browser at code.engr.uconn.edu after NetID login.

Grading

Asynchronous course content (graded interactive videos and quizzes) will be posted weekly on Moodle. It is essential that you review the asynchronous content and complete associated quizzes by the due date (typically before each class meeting) to ensure you are prepared to participate in class discussions. In addition to interactive videos and quizzes, grading will be based on weekly labs, programming projects, and three exams. Labs are short C programming exercises designed to give you hands-on practice with common programming tools and an opportunity to apply the concepts covered in lectures. Programming projects will require you to write more complex C programs, often building on a provided code base. The exams will consist of programming tasks similar to those in the labs and homework assignments.

Grade breakdown

Labs	10%
Quizzes & interactive videos	20%
Programming projects	25%
Three exams	15% each

The lowest lab score and lowest programming project score will be dropped from the overall grade calculation.

Collaboration policy

Unless otherwise indicated, assignments must be completed individually. All programs and documents you hand-in must be your own work. You may discuss course related topics with others, but must not share code or quizz answers. Reasonable use of published materials (including web resources) is allowed, but all sources must be explicitly acknowledged in your submissions. Violations will be reviewed and sanctioned according to the University Policy on Academic Integrity. An example of unreasonable use is submitting copied solutions with minor changes like renaming variables. If you need additional clarifications regarding the collaboration policy, please contact the instructor.

Students with disabilities:

If you have a documented disability for which you are or may be requesting an accommodation, please contact the Center for Students with Disabilities by the end of the third week of the semester to ensure that any accommodations you need can be implemented in a timely fashion.

Tentative Schedule (updated 03/24/22)

Dates	Lecture/Lab topics
Jan 19	Course introduction; C overview
Jan 21	NO LAB MEETINGS
Jan 24 & 26	Expressions and basic data types, control flow (ABC Ch. 2-4)
Jan 28	Lab1: code-server IDE; basic terminal commands; `make`
Jan 31 & Feb 2	Functions, arrays, and structures (ABC Ch. 5-6 & 9)
Feb 4	Lab canceled due to inclement weather
Feb 7 & 9	Pointers and dynamic memory allocation (ABC Ch. 6)
Feb 11	Lab2: Debugging
Feb 14 & 16	Pointer arithmetic and I/O (ABC Ch. 6 & 11)
Feb 18	Lab3: `valgrind`
Feb 21 & 23	Miscellaneous C topics and review for exam #1
Feb 25	Lab canceled due to inclement weather
Feb 28 & March 2	Processes, upgrades, & intro to redirections (ABC Ch. 11 & 12)
March 4	EXAM 1
March 7 & March 9	Inter-process communication using pipes (ABC 12.3) and I/O multiplexing using select
March 11	Lab4: pipes
March 21 & 23	Intro to sockets and client-server communication (Beej's guide) (March 21 meeting cancelled due to power outage)
March 25	Lab5: sockets
March 28 & 30	Signals (ABC Ch. 12)
Apr 1	Lab6: Debugging multi-process applications
Apr 4 & 6	Intro to threads and review for exam #2
April 8	EXAM 2
April 11 & 13	Basic thread management and synchronization
April 15	Lab7: thread management
April 18 & 20	Thread synchronization
April 22	Lab8: thread synchronization
April 25 & 27	False sharing; threads scheduling and priority inversion; review for exam #3
Apr 29	EXAM 3