

PRINCIPLES OF SYSTEMS PROGRAMMING

CSCE 3600.002 – FALL 2020

Instructor: Dr. Qing Yang / Dr. Song Fu

Office: NTDP F253 / F250

Telephone: (940) 565-4899 / (940) 565-2341

E-mail Address: Qing.Yang@unt.edu / Song.Fu@unt.edu

Class Location/Time: NTDP B185 (Discovery Park), MoWeFr 1:30 – 2:20 PM

Class Zoom Link: <https://unt.zoom.us/j/89786545283>

Office Hours: WeFr 2:30 PM – 4:00 PM *or by appointment. Please use your UNT e-mail and include CSCE 3600.002 or 3600.005 (or your specific recitation section) in the subject line.* Zoom link for office hours: <https://unt.zoom.us/j/85211884719>

Optional Reference Textbook: *System Programming with C and Unix, First Edition*, by Adam Hoover, Addison-Wesley, 2010. ISBN 978-0-13-606712-2.

Grader:

Yen Pham,

Email: yenpham@my.unt.edu,

Office hours: Tue. Thu. 2:00pm - 3:00pm

Zoom link for office hours: <https://zoom.us/j/91457709404>

Recitation Schedule: Depending on which session you registered, please check [here](#) for TAs information.

Canvas: This course will use the Canvas learning management system (LMS) to distribute course materials, communicate and collaborate online, post grades, and submit assignments. You are responsible for checking the Canvas course site regularly for classwork and announcements.

Course Description

CSCE 3600 maintains a focus on systems programming, both from the standpoint of learning about how computer system software works and learning/improving students' programming skills in K&R C and bash (Bourne-again shell). The course will include one or more major programming projects that will be completed in groups as well as several smaller programming assignments to be done individually.

Course Outcomes

Course outcomes are measurable achievements to be accomplished by the completion of a course. These outcomes are evaluated as part of our ABET accreditation process.

1. Write robust, efficient, readable, and correct system software using the C programming language.
2. Demonstrate an understanding of processes and threads by developing applications using multiple processes and multi-threaded activities in a Linux environment.
3. Demonstrate an understanding of deadlocks and synchronization through the development of the application(s) that utilize a variety of mutual exclusion mechanisms.
4. Develop shell scripts and utilities that demonstrate an understanding of memory, file, and process management and interaction, including concepts such as virtual memory and disk scheduling.
5. Create a Linux-based application that utilizes inter-process communication mechanisms such as pipes and sockets to communicate information between independently running processes on one or multiple platforms.
6. Demonstrate an understanding of the use and interaction among compilers, macro processors, assemblers, linkers, and loaders through their use in creating the applications described in previous outcomes.

Attendance Policy

Lecture Section: Class attendance is regarded as an obligation as well as a privilege. All students are therefore expected to attend each class meeting. A student who misses class is still responsible to find out what was discussed and to learn the material that was covered and obtain the homework that was assigned on the missed day. The instructor is not responsible for re-teaching material missed by a student who did not attend class. Therefore, each student is accountable for and will be evaluated on all material covered in this course, regardless of attendance. Excessive student absences may have a negative impact on a student's comprehension and learning. If there are extenuating circumstances, please notify your instructor so that you can work together to ensure your success in learning the material.

Recitation Sections: Except when the programming portion of an exam is being given in which case attendance is mandatory, attendance at your scheduled recitation is considered to be optional. Students may want to take advantage of attending their recitation section to receive guidance on completing the recitation, minor, or major assignments or other course assistance, such as preparing for exams.

Acceptable Student Behavior

Student behavior that interferes with an instructor's ability to conduct a class or other students' opportunity to learn is unacceptable and disruptive and will not be tolerated in any instructional forum at UNT. Students engaging in unacceptable behavior will be directed to leave the classroom and the instructor may refer the student to the Dean of Students to consider whether the student's conduct violated the Code of Student Conduct. The university's expectations for student conduct apply to all instructional

forums, including university and electronic classrooms, labs, discussion groups, field trips, etc. The Code of Student Conduct can be found at <http://deanofstudents.unt.edu>.

ADA Statement

The University of North Texas makes reasonable academic accommodations for students with disabilities. Students seeking reasonable accommodation must first register with the Office of Disability Accommodation (ODA) to verify their eligibility. If a disability is verified, the ODA will provide you with a reasonable accommodation letter to be delivered to faculty to begin a private discussion regarding your specific needs in a course. You may request reasonable accommodations at any time; however, ODA notices of reasonable accommodation should be provided as early as possible in the semester to avoid any delay in implementation. Note that students must obtain a new letter of reasonable accommodation for every semester and must meet with each faculty member prior to implementation in each class. Students are strongly encouraged to deliver letters of reasonable accommodation during faculty office hours or by appointment. Faculty members have the authority to ask students to discuss such letters during their designated office hours to protect the privacy of the student. For additional information see the Office of Disability Accommodation website at <http://www.unt.edu/oda>. You may also contact them by phone at 940.565.4323.

Grading Policy

Your course grade will be a weighted average according to the following:

Recitation Assignments	15.0%
Minor Programming Assignments	25.0%
Major Programming Assignments	20.0%
Midterm Exams 1 – 2	25.0% (12.5% each)
Final Exam	15.0%
Total	100.0%

The following letter grading system will be used in this class:

A [90, 100) , B [80, 90) , C [70, 80) , D [60, 70) , F [0, 60)

Grades will be posted on Canvas throughout the semester to provide an ongoing assessment of student progress, though final assessment will be measured using the weighted average above. Once a grade is assigned on Canvas, students have two (2) weeks to dispute the grade. The proper channel for grade disputes is to first go to the original grader (i.e., TA/IA) in an attempt to resolve the issue. If, however, a resolution cannot be reached between the student and the grader, the student shall then go to the instructor who will have the final say on the grade.

Most programming assignments will be due at 11:59 PM on the specified due date to

Canvas. All assignments must be completed and submitted according to their specific directives. Minor and major programming assignments (not recitation assignments) will be accepted up to 24 hours late and assessed a 30% grade reduction penalty. Any minor or major programming assignment submitted more than 24 hours late will not be accepted and receive a grade of 0.

Recitation Assignments: Recitation assignments are meant to serve as preparatory assignments for upcoming minor and major assignments that can be completed in a relatively short amount of time. Students may complete these assignments on their own or by attending their scheduled recitation where they may receive guidance from a TA/IA on completing. **No late recitation assignments will be accepted.** Please note that if any programming portion of an exam is given during each recitation section during the week of the scheduled exam, then attendance will be required.

Minor Programming Assignments: Minor programming assignments will be assigned based on the material from the lectures and textbook. These are meant to be individual programming assignments, so you should **work on these alone** unless explicitly directed otherwise by your instructor.

Major Programming Assignments: Major programming assignments will be **worked on in a group** and are meant to be more comprehensive problem-solving exercises based on the material from the lectures and textbook.

Midterm Exams: There will be **two midterm examinations** given in this course. The dates of these exams will be posted on Canvas and announced in class at least one week prior to the date of the exams. A make-up exam will be given at the discretion of the instructor when a student misses an exam with an excused absence. Unexcused absences on the date of an exam may result in a grade of 0 for the missed exam, so every effort should be made to attend class on the day of a scheduled exam.

Final Exam: There will be a comprehensive final exam on **Saturday, December 5, 2020, from 11:30 AM to 12:30 PM.** All students are expected to take the final exam during the scheduled time period.

Academic Dishonesty

This course follows UNT's policy for Student Academic Integrity that can be found at <https://policy.unt.edu/policy/06-003> as well as the Cheating Policy for the Department of Computer Science and Engineering (posted on Canvas). Specifically, the first instance of a student found to have violated the academic integrity (i.e., cheating) policy will result in a grade of "F" for the course and have a report filed into the Academic Integrity Database, which may include additional sanctions.

This course will contain both group assignments as well as individual assignments, so you should be absolutely aware of the assignment requirements **before starting an assignment**. On major programming assignments, you are to work in a group as directed by your instructor. **Minor programming assignments must be the sole work of the individual student**. You should not work with other students on shared program solutions or use program solutions found on the Internet. Specifically, you should never copy someone else's solution or code, and never let a classmate examine your code. A sophisticated program will be used to compare your work to the work of all other students (including students in past classes). If you are having trouble with an assignment, please consult with your instructor or TA. Failure to adhere to these strict standards may be cause for disciplinary action even leading to expulsion from the University.

In the case that the above description or any in-class discussion of appropriate and inappropriate collaboration does not answer all of your questions, please meet with your instructor and look at the university Student Rights and Responsibilities web page.

Student Responsibility

Students are responsible for submitting the correct assignments (i.e., uploading the proper files) for each applicable assignment submission on Canvas. In certain cases, when an assignment is verified to be completed on time, but either was submitted to an incorrect assignment location (e.g., submitting Minor 5 to Minor 4 location on Canvas) or a wrong assignment was submitted instead, the assignment may be accepted, but assessed a 30% reduction penalty if the due date has passed. Verification of completion time stamp for assignments will be done using the CSE machines, so please make sure to save your work on these departmental servers to ensure that your work can be accepted. If you have any questions or concerns about your submission, please work with your instructor or TA/IA to ensure the correct file(s) is/are submitted.

Syllabus Revisions

This syllabus may be modified as the course progresses should the instructor deem it necessary. Notice of changes to the syllabus shall be made through Canvas and/or class announcement.

Tentative Class Schedule (subject to change)

Week	Date	Subject
1	8/24 — 8/28	Systems Programming Overview

2	8/31 — 9/4	Linux Overview / Gitlab
3	9/7 — 9/11	Regular Expressions, sed/gawk
4	9/14 — 9/18	sed/gawk, Bash
5	9/21 — 9/25	Bash, Review
6	9/28 — 10/2	Processes
7	10/5 — 10/9	Processes
8	10/12 — 10/16	Threads
9	10/19 — 10/23	Threads, IPC
10	10/26 — 10/30	IPC, Review
11	11/2 — 11/6	IPC
12	11/9 — 11/13	IPC
13	11/16 — 11/20	Compilers & Compilation
14	11/23 — 11/27	Python
15	11/30 — 12/4	Python, Review
16	12/5	Final Exam 10:30 – 12:30