CS 288: Intensive Programming in Linux

Syllabus (Fall 2023)

Instructor

- Xiaoning Ding
- **E-mail**: xiaoning.ding@njit.edu (**Note**: Before you email me, please review the email policy later in this syllabus)
- **Phone**: (973) 596-3390
- **Office location**: GITC 4203

Course

- **Title**: Intensive Programming in Linux
- **Course number**: CS 288-007
- **Location**: ECEC 100
- **Meeting time**: TR 4:00PM - 5:20 PM. First meeting: Tuesday, Sep 5, 2023. Last meeting: Tuesday, Dec 12, 2023.
- **Office hours**: TR 5:20PM - 6:20 PM

Course Description

The course covers programming topics in Linux OS using C, Python, and PHP as primary languages. The course consists of the following modules: basic tools and utilities for software development in Linux, such as Bash, C programming, and Linux system API; methods and algorithms for processing web data, such as searching trees and matrix computing; end-to-end applications such as one that constantly presents top 100 stocks; and

extending the applications to run on multiple machines. The course provides students with utilities and hands-on experience for programming relatively large applications.

- This course is **not** an introduction to specific programming languages.
- This course uses intensively Linux and Linux command line interface (shell and command lines, **not** graphic user interface). But it is **not** an introduction to Unix/Linux systems or Unix/Linux command lines and tools.
- This course is **not** an introduction about Linux operating system or how to use Linux.

Specific Goals for the Course

This course is intended for students who want to improve their programming skills and system skills in Linux. The goals of the course are to improve their programming skills by working on a variety of assignments and to familiarize students with the programming facilities and software development ecosystem in Linux.

Student Outcomes

Students will be able to

- Write shell scripts.
- Use regex to search and manipulate text strings.
- Operate binary data using bit-wise operations.
- Implement a collection of algorithms (e.g., sorting and searching) using matrices, pointers, and linked lists in C.
- Develop parallel/distributed programs that use multiple threads or multiple processes.
- Test and debug programs.

Prerequisites

- CS 100 Roadmap to Computing
- CS 280 Programming Language Concepts
- The course is about programming in Linux systems, and you will be using intensively Linux command line interface, not GUI (graphic user interface). You need to know how to use Linux systems and common Linux commands to understand course materials and finish assignments.
- The course uses intensively C language. You need to know how to read and write C programs to understand the related course materials and finish assignments.

Textbook

There is no formal textbook for this course. But you will find the following books useful.

- The C Programming Language, Kernighan and Ritchie, Prentice Hall, 2nd ed., ISBN: 978-0131103627

- Matthews, S. J., Newhall, T., & Webb, K. C. (2021). Dive into Systems: A Free, Online Textbook for Introducing Computer Systems. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 1110-1116).

  Dive Into Systems Different Versions

  Dive Into Systems Development Version (This class will use the development version)

Supplementary materials will be posted on Canvas course homepage.

Course Work and Grading

Your grade in the course will be determined by the following breakdown:

- **Practice programming assignments – 10%**

  The course includes 8 programming assignments primarily designed for hands-on practice. It's essential to fully understand the problems and develop solutions independently. Additionally, you're encouraged to explore alternative solutions, and consider variations of the problems and their corresponding solutions.

- **Quizzes (10%)**

  4 close-book close-note quizzes will be conducted in class on un-declared dates. Quiz questions will be selected from the exercises in *the C Programming Language* book. Read the book chapters. Work on the exercise problems.

- **Midterm tests – 50% (25% for each midterm test)**

  Two midterms will be scheduled in the semester. Most questions in midterm exams will be derived from examples and programming assignments.

- **Final exam – 30%**

  Final exam will be scheduled by the University. Check online for the time and location. Most questions in final exam will be derived from examples and programming assignments.

Grading Scale

Grades	Significance	Percentage Range
A	Superior	85~100%
B+	Excellent	75~84%
B	Very Good	70~74%
C+	Good	65~69%

C	Acceptable	60~64%
D	Minimal	50~59%
F	Inadequate	<50%

Course Navigation

The course is structured into a few topics, including Linux bash scripting, raw data format and organization, radix-sort, state-space searching, multi-threading and MPI parallel processing, and web-scraping. Each topic consists of lecture slides/notes, pointers to the related book chapters/sections and required online materials, homework assignments, and pointers to the supplementary materials recommended for interested students. The course will proceed by completing the topics, with each major topic taking about 1~3 weeks and having 1~2 homework assignments. Check the tentative schedule below.

Each week, students will attend two lectures in class. The lectures will mainly discuss the key points on the topics, student questions, and skills and tips needed to solve HW problems. Thus, students are expected to get prepared for the lectures by reading the related book chapters/sections and required materials before the week starts. The lecture discussions will use extensively example programs and code snippets. To get deeper understanding, students are encouraged to understand, change, and test them after each class session.

Track “Announcement” section in Canvas for any updates to the courses.

Tentative Schedule

*Note: Though I will seek to follow this schedule, please understand that I also always reserve the right to modify the schedule, depending upon class progress.

Week	Topic	Exams
1	Overview and intro to Linux systems	
2~4	Linux bash scripting: Linux command line interface, Bash shell scripting (variables, expansions, arrays, functions, etc), regular expressions	**Midterm 1: Oct 3 4:00pm~5:20pm, ECEC 100**
5~6	Raw data format and organization: binary representation of data (4.7, 4.3, 4.8), bitwise operations (4.6), C pointers and dynamic multi-dimensional arrays (2.1~2.6, 2.9.2~2.9.4), Linux file operations and directory operations,	
7~9	radix-sort: radix sort for integers/longs, radix sort for floats/doubles	**Midterm 2: Nov 2 4:00pm~5:20pm, ECEC 100**

	debugging C programs (3.1~3.2)	
10~12	State space search: self-referential structures and linked list (2.7), A* search.	
13~15	Parallel processing: pthread (14.1~14.3), MPI (15.2). Web-scraping	**Final Exam: TBD**

Canvas Course Policy

- Visit the course homepage in Canvas regularly and frequently for lecture notes, homework assignments, instructions, and latest updates.
- Course materials, including this syllabus, lecture materials, homework assignments, and exams, are protected content. Students should not make copies of course materials or distribute course materials in the public domain, including sites such as Chegg, CourseHero, etc.
- NJIT policy on video recording class materials: You may not video record the class materials. You may not put any video/audio recorded class materials on the Web/Internet. It is against the University policy.
- Turn off cell phones and alarm clocks during class meetings.

Email Policy

- Use a properly descriptive subject line that consists of the course number (“CS288”), the section number, and a very brief phrase that

summarizes the subject of your message, such as “a question on HW2 Problem 1”.

- Use your NJIT email address to send emails. Your emails may be filtered out if you don’t follow. ***Avoid sending emails using canvas***, because email threading is not supported in Canvas.
- Avoid writing emails in an informal style. Make sure you adhere to proper sentence structure, grammar, spelling, and style. Proofread for typos and spelling mistakes. Minimal requirements: capitalize the first letter in each sentence; use a punctuation mark at the end of each sentence.
- If you want to include any program code (C, Bash, etc) in the email, include it in a text file and attach the text file with the email. The text file must use a plain-text format and use .txt as file name extension. If you use windows, use notepad to generate the file. Do not use MS Word or other editors with AutoCorrect functionality enabled, which automatically replaces punctuations (e.g., “straight quotes” with “smart quotes”) and causes cause errors when the program is compiled or run.
- Most emails will be responded within 24 hours during weekdays. In some special situations, e.g., health, travels, I will try to respond student emails within 48 hours. Emails received in weekends will be responded in the next weekday.
- When you receive emails from the course, read the whole emails, not just subject lines.

Homework Policy

- Though you may discuss assignments with classmates, friends, or tutors, the discussion should be limited to comprehending the problem statement and conceptualizing potential solutions. Following your discussions, you are expected to develop **your own** code. **The final submission must be your individual work. Copying or reengineering/paraphrasing code that was not developed by you is strictly prohibited.** This policy aims to uphold academic integrity while also promoting collaborative learning. Understanding and adhering to this policy ensures that you are not only learning effectively but also being fair to yourself and your peers.

- All homework assignments for this course will require two components to be submitted through the Canvas LMS: your program files with appropriate comments, and screenshots of your program running correctly.
    - Program File: The program file should contain well-written and properly structured code, adhering to the language-specific style guidelines discussed in class. Remember to include appropriate comments in your code to describe the functionality of different sections. The filename should follow the assignment-specific naming convention, which will be provided with each assignment.
    - Screenshots: Along with the program and the documentation, please submit screenshots showing your program running correctly. These screenshots should demonstrate different use cases or scenarios to validate the successful execution of your program. You may include these screenshots in your Word document or submit them as separate image files, as per the assignment instructions.
- **The use of AI-based code generators is prohibited.**
- It's essential that your submissions show a genuine attempt to solve the assignment using the concepts learned in class. Submissions that are irrelevant or do not relate to the assignment will not receive credit. Each component of your submission contributes to the overall grade, so be sure to fulfill all the requirements.
- **Late Submission Policy:** To encourage timely submissions and enable graders to provide feedback promptly, we have a strict late submission policy:
    - Assignments submitted within 24 hours after the deadline will incur a 10% grade deduction.
    - Assignments submitted between 24 and 48 hours after the deadline will incur a 20% grade deduction.
    - **No assignments will be accepted beyond 48 hours after the deadline**, except in extenuating circumstances, such as illness or family emergencies, for which documentation must be provided.
- Homework grades will be returned within a span of 1 week. Feedback for assignments and exams will be provided via SpeedGrader.

Quiz and Exam Policy

- This course includes four unannounced, closed-book, closed-note quizzes that will be conducted during class time. The dates of these quizzes will not be declared in advance, emphasizing the importance of consistent attendance and regular review of course material.
- In calculating your final course grade, we will consider the scores from your top three quizzes. This means that the lowest of your four quiz grades will be dropped, allowing some flexibility for off days. If a quiz is missed entirely, that will be considered a zero.
- Exams will be closed-book and closed-notes. You are not allowed to take the exam of another section. Students with special needs are advised to make arrangements with OARs for exam accommodations.
- As a general rule, no makeup exams will be given, and no alternate dates for exams without a legitimate reason (e.g., jury duty, medical problem).

Academic Integrity Policy

Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the NJIT academic code of integrity policy.

Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. ***Any student found in violation of the code by cheating, plagiarizing or using any online software inappropriately will result in disciplinary action.*** *This may include a failing grade of F, and/or suspension or dismissal from the university. If you have any questions about the code of Academic Integrity, please contact the Dean of Students Office at dos@njit.edu.*

Requesting Accommodations:

If you are in need of accommodations due to a disability please contact Scott Janz, Associate Director of the Office of Accessibility Resources & Services (OARS), Kupfrian Hall 201 to discuss your specific needs. A

Letter of Accommodation Eligibility from the OARS authorizing your accommodations will be required.