CS 288: Intensive Programming in Linux

Syllabus (Fall 2025)

Instructor: Ruby Kapoor
E-mail: rk28@njit.edu
(**Note**: Before you email me, please review the email policy later in this syllabus)

Course title: Intensive Programming in Linux
Course number: CS 288-101
Location: Cullimore Lecture 3
Meeting time: TR 6:00PM-8:50PM. First meeting: September 4th, 2025. Last meeting: December 11th, 2025.

Office hours: M 7-8PM (Virtual - Zoom)
Class TA: TBD

Course Description:

This comprehensive program is designed to provide students with a strong foundation in programming concepts and practical skills within the Linux environment. Over the duration of the course, we will cover a wide range of topics 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 to equip you with the necessary tools and knowledge to become proficient in the Linux environment.

- 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 to operating systems.

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

Upon completion of the course, students will be able to:
- Have a solid understanding of the Linux command line interface and be able to navigate, manipulate files, and execute commands efficiently.
- Bash scripting, enabling you to automate tasks and create powerful scripts.
- Be proficient in using regular expressions, a fundamental skill for pattern matching and text processing in Linux.

- Gain the ability to debug C programs effectively, employing various debugging techniques to identify and fix issues in your code.
- Understand the binary representation of data and be able to perform bitwise operations, enabling you to work with binary data efficiently.
- Learn radix-sort, a powerful sorting algorithm, and apply it to efficiently sort.
- C pointers and dynamic multi-dimensional arrays, essential for managing memory and manipulating complex data structures.
- Explore self-referential structures and linked lists, fundamental concepts in data structures and algorithms, and gain the ability to implement them in C.
- Acquire knowledge and skills in A* search, a widely used algorithm for pathfinding and problem-solving.
- Develop proficiency in Linux file operations and directory operations, allowing you to create, modify, and manage files and directories programmatically.
- Learn how to write programs with multiple threads, enabling you to leverage parallelism and concurrency in your applications and parallel computing with MPI (Message Passing Interface), a powerful framework for developing high-performance parallel applications.
- Explore web scraping, learning how to extract data from websites programmatically and utilize it.

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, NO 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.

Course Materials

There is no formal textbook for this course, but you will find the following books useful. All lecture slides will be made available on Canvas prior to week's material.

Dive Into Systems Development Version

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

NJIT policy on recording class materials: Recording any lecture material, including audio, video or taking photos in class is not permitted. Additionally, posting any class materials (slides, resources etc) on the internet is strictly prohibited, as it violates the University's intellectual property policy.

Course Work

Quiz and exam grades will be returned within a span of 1 Week. Feedback for assignments and exams will be provided via SpeedGrader.

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

- Practice programming assignments – 20%
  The course includes 7-8 programming assignments (can change, dependent on semester progression), which are for hands-on practice. Full independent comprehension of these assignments will be critical for application & success in this course. Additionally, you're encouraged to explore alternative solutions, and consider variations of the problems and their corresponding solutions. ***Do NOT copy or reengineer/paraphrase the programs that were not developed by you. Any violation of academic integrity will result in a zero & be reported to DOS.***

  Submit programs and screenshots of program executions to show that results are correct.

- Quizzes - 20%
  5 **close-book close-note** quizzes will be conducted in class on un-declared dates.

- Midterm test – 28%
  One midterm will be scheduled in the semester. Most questions in midterm exams will be derived from examples and programming assignments. Closed book.

- 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. Closed book.

- Attendance – 2%
  Active class participation is necessary.

Grading

Homework	20%
Quizzes	20%
Midterm	28%
Final	30%
Attendance	2%
Total	100%

Late Homework Policy

- Students are expected to submit on day of the due date, failing to submit before the due date will result in a grade of zero. No exceptions will be made unless the student provides documentation from the Dean of Students. Backup your work regularly. Losing files or accesses to any systems is **NOT** a valid excuse to late submission.

Quiz and Exam Policy

- Exams and Quizzes 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).

Grading Scale

A :	Superior	85~100%
B+:	Excellent	75~84%
B :	Very Good	65~74%
C+:	Good	60~64%
C :	Acceptable	56~59%
D :	Minimal	51~55%
F :	Inadequate	<50%

Course Navigation

The course is structured into topics. Major topics include Linux command line interface and bash scripting, regular expression, binary data representation and bitwise operations, radix-sort, pointers and dynamic multi-dimensional arrays, linked list, state-space searching, Linux file and directory operations, 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 week. Check the tentative schedule below.

Each week, students will attend one lecture 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. After each week, a homework assignment is released on a finishing topic and students finish the homework assignment to conclude the topic.

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

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.
- 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.

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.

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.

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	Assignments
1	Overview & Introduction to Linux Systems	
2	Bash Shell Scripting	
3	Bash Shell Scripting (Continuation)	
4	Pattern Matching with Regular Expressions	**Homework 1**
5	Debugging C (Ch 3.1-3.2) Binary Data Representation (4.7,4.3, 4.8) Bitwise Operations (Ch 4.6)	**Homework 2**
6	Radix Sort	**Homework 3**
7	C Pointer Variables & Memory Address (2.1~2.6, 2.9.2~2.9.4)	**Homework 4**
8	Multidimensional Arrays, Dynamic & Pointer-Pointer Variables	**Homework 5** **Midterm: October 30th, 6-7:30PM**
9	Command Line Arguments (2.1~2.6) Structures and Linked List (2.7)	**Homework 6**
10	A* Search and State Space Search	
11	Programs with Multiple Threads (14.1~14.3)	**Homework 7**
12	MPI Parallel Computing with MPI (15.2)	
13	Web Scraping	**Homework 8**
14	Web Scraping Continuation	**Homework 9**
15	No Classes	**Final Exam TBD**