Summer 2023

CS 288: Intensive Programming in Linux

Andrew Sohn

CS288 Intensive Programming - Syllabus

- Class Web page: http://web.njit.edu/~sohna/cs288 and Canvas
- Instructor: Andrew Sohn, GITC 4209, (973)596-2315, email: sohna _at_ njit _dot_ edu
- Email: Send your message to the official school email address listed above. Please, do NOT email to Canvas. I do not read Canvas emails.
- Office Hours: Tues,Thur 3-4 pm after class.
- Teaching assistant: Mehtab Sidhu, mss87 _at_ njit _dot_ edu, office hours: TBA
- Class time and location: Tues,Thur, 1-3pm, FMH 408 (classroom may change in coming weeks).
- Textbook: The C Programming Language, Kernighan and Ritchie, Prentice Hall, 2nd ed., ISBN: 978-0131103627, and a book on Linux Bash. You will need one if you plan to stay in computing. Materials used in the course are taken from various sources freely available on the Web. Pay attention to the announcements in class regarding course materials. Remember there is no such book that conveniently describes all the topics discussed in this class.
- Recommended book (if you haven't taken CS350): Computer Systems: A Programmer's Perspective, 3/E (CS:APP3e), Randal E. Bryant and David R. O'Hallaron, Pearson (July 6, 2015), ISBN-13: 978-0134123837, ISBN-10: 0134123832.
- Platform: Linux, distro Fedora 30 or above, multi-booted on bare-metal recommended; Virtual machine is not recommended for this course as they are not suitable for daily interactive computing. VMs along with containers and on-demand Lambda/EC2 are designed for backend serverless servers.
- Tools: Bash, C - the most popular language, Python, mySql, DOM, PHP, OpenMP, MPI, and possibly SciKit learn package if time allows for a simple machine learning homework.
- **Grading**:
  - Attendance (4%)
  - Programming assignments (10%) - submit on Canvas
  - Test 1, Tues, 6/13/2023 (25%) - 1-2:30 pm (1.5 hours)
  - Test 2, Thur, 7/11/2023 (25%) - 1-2:30 pm (1.5 hours)
  - Final exam (36%) - TBA
- **Homework**:
  - See Canvas for HW due dates and submission.
  - Homework is due at 11:59 pm of the posted due date.
  - Homework will not be accepted after the due date. Submit on time. Do not ask for exceptions. If you ask for an exception, I will apply that to everyone in class.
  - Do your homework from scratch and on your own. Be prepared to spend three hours a day on homework.
  - Homework must be your own work. Do not show your code and/or copy other's code. You'll be glad you did on your own from scratch. I can tell you many success stories related to doing homework on your own.
  - Copying homework will be referred to the University for disciplinary actions.
- **NJIT policy on video recording class materials**: You may not put any video/audio recorded class materials on the Web/Internet. You are violating the University policy on intellectual property.
- **Exam related. Read carefully:**

- Sample exams including final will be posted at the class web site for your reference. Do not rely soley on them. The contents and format may not necessarily be similar.
- Exam questions will be derived from programming assignments. Do your homework from scratch and on your own. Be prepared to spend two hours a day on homework. Homework will not be accepted after the due date. Submit on time.
- Exam questions will be given out in a random order with multiple versions of the same difficulty/complexity. After you sumit, you cann't go back. Exam will end at 3 pm sharp. Budget your time accordingly. I will send a message before exam.
- **Disagreement with exam marking/scores**: If you disagree with your exam scores/marks, you may dispute within a week of receiving/seeing the graded exam paper. After a week, no exams will be contested.
- **Grading dispute**: If you disagree with your grade, you may contest after the first day but within a week of the following semester. After a week of the first day of the following semester, no grading dispute will be considered.
- **NJIT policy on missed exams**: There will be no make-up exam(s). You must plan your semester accordingly, especially if you work. Should you miss the exam(s) due to emergency, (a) go to/contact the Dean of students, (b) explain your situation as to why you had to miss, and (c) ask to issue a memo. If and when I receive a memo from the Dean on your missed exam, I will copy your next exam score to the missing one. Those who miss the final exam will fail in the course unless you demonstrate a true emergency again through the office of the Dean of students. No other policy will be applied. No exceptions will be made.

- **Academic Integrity:** I am required to post this on the course syllabus.
  "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 academic code of integrity policy that is found at: http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf. 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"

Lecture schedule - Contents may change according to the class pace
STAGE 1 - learning the most basic and fundamental knowledge

1. Intro to Bash shell scripting - variables, assignments, arrays, lists, functions
2. Recursive directory traversal in Bash - depth first and breadth first creating-tree.pdf
3. Pattern matching with regular expression (grep)
4. Introduction to C pointers, ref/dereferences, pointers to pointers, array of pointers, function pointers hello-memap.pdf
5. Malloc/free and basic structure handling with simple linked list, wrong-right-pointers.pdf
6. Structure handling - swap and push, structure handling with multiple links

STAGE 2 - tools for building an end-to-end realworld application

7. Sorting - fast radix sort for integers/longs radix-sort-handout.pdf
8. Sorting - fast radix sort for floats/doubles
9. State space search - depth first, breadth first search search-notes.pdf
10. State space search - heuristic-based intelligent search such as the one used in Unity game engine
11. Matrix computation: a system of linear equation solvers, introduction to iterative methods
12. Matrix computation: application to spectral graph partitioning for clustering in machine learning if time permits

STAGE 3 - an end-to-end realworld application towards simple-minded stock prediction web-processing.pdf

13. Web processing - fetching with wget using Bash scripting, intro to DOM tree, properties, methods
14. Web processing - DOM tree navigation, data extraction using Python minidom
15. Web processing - getting up and running mySql/maria DB server, DB construction, data injection
16. Web processing - getting up and running Apache server, reading DB using PHP, constructing clickable/sortable table
17. Web processing - formulating samples for training and predicting stock trading with scikit learn package if time permits.

STAGE 4 - extending tools and applications to run on many-core machines for big data processing, machine learning and analytics

18. Introduction to multicore/parallel computing using OpenMP and MPI - point to point communication
19. Introduction to MPI - collective communication
20. Simple matrix computation for multicore/multiple machines using MPI
21. Introduction to GPU architecture if time permits.
22. Introduction to GPGPUP - General Purpose GPU Programming if time permits.