Summer 2020

CS 288-450: Intensive Programming in Linux

Andrew Sohn

CS288 Intensive Programming - Syllabus - Summer 2020 (5/18-8/3/2020)

- 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 as it is a private company address.
- Office Hours: Tues,Thur 3-4 pm on Webex after class. If you want to see me outside the office hours, send me an email.
- Teaching assistant: Mehtab Sidhu, mss87 _at_ njit _dot_ edu, office hours: Mon,Wed 1-2 pm, Webex.
- Class time and location: Tues,Thur, 1 pm - 3 pm, Webex. Get on Webex. I take attendance.
- 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 class.
- Recommended book: 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 as of today, 5/19/2020, Python, mySql, DOM, PHP, JavaScript, 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, 6/9/2020 (25%) - 1-3 pm (2 hours) on Canvas
  - Test 2, Thur, 7/2/2020 (25%) - 1-3 pm (2 hours) on Canvas
  - Final exam, Tues, 7/30/2020 (36%) - 1-3 pm (2 hours) on Canvas
- **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 to be fair with everyone else in class.
  - Do your homework from scratch and on your own. Be prepared to spend two hours a day on homework.
  - Homework must be your own work. Do not show your code and/or copy other's code.
  - 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 solely 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. Once you submit, you can'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) contact the Dean of students, (b) explain your situation as to why you had to miss, and (c) ask to issue a memo to me. 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.

· See Academic Integrity

Lecture schedule - Contents may change according to the class pace

STAGE 1 - learning the most basic and fundamental knowledge

1. Intro to Linux, LAMP (Linux, Apache, mySql(maria), Python/Php/Perl), virtualization/containers/lambda/ec2
2. Intro to Bash shell scripting - variables, assignments
3. Intro to Bash shell scripting continues - arrays, lists, functions
4. Recursive directory traversal in Bash - depth first and breadth first creating-tree.pdf
5. Pattern matching with regular expression (grep)
6. Introduction to C pointers, ref/dereferences, hello-memap.pdf
7. Pointers to pointers, array of pointers, function pointers
8. Malloc/free and basic structure handling with simple linked list, wrong-right-pointers.pdf
9. Structure handling - swap and push, structure handling with multiple links

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

10. Sorting - fast radix sort for integers/longs radix-sort-handout.pdf
11. Sorting - introduction to floating point representation
12. Sorting - fast radix sort for floats/doubles
13. State space search - depth first, breadth first search search-notes.pdf
14. State space search - heuristic-based intelligent search
15. State space search - intelligent bi-directional search

16. Matrix computation: a system of linear equation solvers
17. Matrix computation: introduction to iterative methods
18. Matrix computation: application to spectral graph partitioning for clustering in machine learning

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

19. Web processing - fetching with wget using Bash scripting, intro to DOM tree, properties, methods
20. Web processing - DOM tree navigation, data extraction using Python minidom
21. Web processing - getting up and running mySql/maria DB server, DB construction, data injection
22. Web processing - getting up and running Apache server, reading DB using PHP, constructing clickable/sortable table
23. Web processing - formulating samples for training and predicting stock trading with scikit learn package

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

24. Introduction to multicore/parallel computing using MPI - point to point communication
25. Introduction to MPI - collective communication
26. Simple matrix computation for multicore/multiple machines using MPI
