New Jersey Institute of Technology

Digital Commons @ NJIT

Computer Science Syllabi

NJIT Syllabi

Fall 2020

CS 288-001: Intensive Programming in Linux

Andrew Sohn

Follow this and additional works at: https://digitalcommons.njit.edu/cs-syllabi

CS288 Intensive Programming in Linux - Syllabus - Fall 2020

8/31/2020: The format of this course is new to some or perhaps most of us. It's new to me. This new course format requires not just a variety of new equipment but more importantly patience and understanding. I verified in person the equipment at KUPF 210A with the equipment director/maintainer/manager last Tuesday. While the equipment all worked fine, the new class format requires me to perform the following activities simultaneously for successful delivery of materials:

1. **execute programs in class in realtime,**
2. **write course materials on my USB-writing pad (I do not present slides),**
3. **respond to in-class questions,**
4. **watch the chat window for questions from Webex,**

which are not an easy task. While the effectiveness of this mode of delivering course materials remains to be seen, we all need to exercise patience and be more understanding in the meantime.

8/31/2020: I use a Linux box for in-person class. But at this time, the first time, I am forced to use a Win box with Cygwin (Linux/Unix for Windows) to deliver ironically Linux materials. Get this: Webex does not support 64-bit Linux machines. So when I need it, mostly in the second half of the semester, I will bring a separate Linux box. For you, DO NOT use Cygwin for this course. Use a Linux box. Cygwin does not behave exactly the same as Linux Bash. You are warned.

- Class Web page: http://web.njit.edu/~sohna/cs288 and http://canvas.njit.edu
- 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. Do not email to Canvas which is a private company address. Emails sent to Canvas are read intermittently and irregularly. Expect delay of my response if you sent emails to Canvas.
- Office Hours:
  - 3:30-4:30 pm, Mon, on Webex,
  - 1-2 pm, Thur, in person, either in my office or larger room to be announced. Until further notice, come to my office for in-person office hours. Make sure to have a face-covering mask on while maintaining 6 feet distance.
  - by appointment. If you want to see me outside the office hours, send me an email.
- Teaching assistant: not assigned at the moment (Mon, 8/31/2020) and it's not certain we'll get one.
- Graders - more will be announced in coming weeks
  - Gerry Agbay, on Webex, Office hours: TBA
  - Philip Poretsky, on Webex, Office hours: TBA
  - More graders TBA in coming weeks
- Class time and location:
  - Section 001: Tues, 2:30 pm - 5:20 pm, KUPF 210A and Webex. See below for attendance.
  - Section 003: Thur, 2:30 pm - 5:20 pm, KUPF 210A and Webex. See below for 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. Sites with useful and appropriate materials change semester to semester. 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 (CS350 textbook): 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, multi-booted on bare-metal recommended; Virtual machine is **not** recommended for this course as it is 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, 9/1/2020, 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
  - Test1 (25%) on Canvas - **See below for taking exam on Canvas.**
    - Section 001: Tues, 9/29/2020 (25%) - 2:30-4:00 pm (1.5 hours)
    - Section 003: Thur, 10/1/2020 (25%) - 2:30-4:00 pm (1.5 hours)
  - Test2 (25%) on Canvas - **See below for taking exam on Canvas.**
    - Section 001: Tues, 11/3/2020 (25%) - 2:30-4:00 pm (1.5 hours)
    - Section 003: Thur, 11/5/2020 (25%) - 2:30-4:00 pm (1.5 hours)
  - Final exam (36%) on Canvas, Date and Time TBD, See the registrar's page. - **See below for taking exam on Canvas.**
- **Homework**:
  - Homework is posted on http://web.njit.edu/~sohna/cs288
  - 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 every one 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.
- **Attendance:** I am required to verify your presence. For everyone, regardless of where you are, in-person or on Webex, get on Webex. Type in "last name, first name, here" in the chat pane. I will scrape this chat record as attendance. The name you type in must match your Webex ID. If not matched, you'll be marked absent for that lecture. Character case will be ignored.
- **Taking exam on Canvas. Read carefully:**
  - Sample exams will be posted on the class web site for your reference. Do not rely soley on them. The contents and format can/will be completely different.
  - Exam questions will be derived from programming assignments. Do your homework from scratch and on your own. Be prepared to spend at least an hour 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 submit, you cannot go back. Exam will start and end at the designated time. You will receive a message regarding the number of exam questions and their points before exam. Budget your time accordingly.

- o **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.
  - o **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.
  - o **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 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.
- **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"

15-Week Lecture Schedule - Contents may change according to class pace

(26 1.5-hour lectures (13 weeks) + two 1.5-hour exams (1 week) + 2.5 to 3-hour final = 15 weeks)

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 (Optional, depending on class pace)
17. Matrix computation: introduction to iterative methods (Optional, depending on class pace)
18. Matrix computation: application to spectral graph partitioning for clustering in machine learning (Optional, depending on class pace)

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