

CS 192 – UNIX

SYLLABUS – FALL 2008

Lecture: F 11:00 am – noon, MLH 310
Professor: Dr. Holly Patterson-McNeill
Office: TH 224
Phone: 792-2342
E-mail: hapatterson-mcneill@lsc.edu
Web Page: <http://www.lsc.edu/hapmcneill>
Office Hours: M-Th 9:00 – 10:00 am.
Also By Appointment

I. COURSE DESCRIPTION AND PREREQUISITES

Unix is a well-designed computing system that has been around for more than 30 years. It is the base system that led to Linux. CS 192 is an introduction to the fundamental concepts of this operating system and the tools it provides. There are no prerequisites other than a quest for knowledge and a love of computing.

Credit Hours: 1

II. MATERIALS AND SUPPLIES

Textbook, Software, and Supplies:

- Das, Sumitabha. 2006. *Your UNIX: The Ultimate Guide, Second Edition*. Boston, MA: McGraw Hill Higher Education.
- We will be using the Unix provided through the CS program. Using your own version of Linux may not work quite like Unix works on TUX.
- You need to backup your programs to your own media. Your best bet is to use a flash memory device.

Resources:

- Hahn, Harley. 2009. *Harley Hahn's Guide to Unix and Linux*. Boston, MA: McGraw Hill Higher Education.
- Lamb, Linda. 1994. *Learning the vi Editor*. Sebastopol, CA: O'Reilly & Associates, Inc.
- Gilly, Daniel. 1998. *Unix in a Nutshell: A Desktop Quick Reference for System V & Solaris 2.0*. Sebastopol, CA: O'Reilly & Associates, Inc.
- Greenlaw, Raymond. 2001. *Understanding Practical Unix*. Wilsonville, OR: Franklin, Beedle & Associates, Inc.
- and many more...

III. GOALS AND OBJECTIVES

Goals of the Instructor:

Your constructive feedback is always welcome to help me achieve the following goals:

- Provide all students the tools necessary to succeed in their pursuit of a high level of understanding of the principles of programming by:
 - Shifting ownership and responsibility for learning to the students
 - Using discovery, problem-based, and other active learning techniques
- To facilitate the improvement of student problem-solving skills and processes
- To facilitate an atmosphere conducive to learning the principles of programming for all students
- To provide quality feedback to students, enabling them to gauge their progress towards achieving their goal in learning the principles of programming

Course Learning Objectives:

By the end of the course, the student should be able to:

- Use basic Unix commands for:
 - directory and file manipulation
 - environment setup
 - standard I/O
 - process and job control
- Use vi for file editing
- Apply filters using regular expressions
- Write simple shell programs

IV. IMPORTANT NOTES

Access & Accommodation Statement:

If you need course adaptations or accommodations because of a disability, if you have emergency medical information to share with me, or if you need special arrangements, please contact me immediately. I ask that you put any request in this regard in writing.

Academic Dishonesty:

In the event of academic dishonesty, those involved **will receive an "F" grade for the course** and the violation of the Student Code of Conduct will be referred to the Director of Student Life for judicial action.

As defined in the [LCSC Student Handbook](#), Academic Dishonesty is:

- a. **Cheating** - intentionally using or attempting to use unauthorized materials, information, or study aids in any academic exercise. The term "academic exercise" includes all forms of work submitted for credit hours.
- b. **Fabrication** - intentional and/or unauthorized falsification or invention of any information or the source of any information in an academic exercise.

- c. **Collusion** facilitating academic dishonesty - intentionally or knowingly helping or attempting to help another to commit an act of Academic Dishonesty.
- d. **Plagiarism** - the deliberate adoption or reproduction of ideas or words or statement of another person as one's own without acknowledgment.

According to the Student Code of Conduct,

The sanctions imposed for a violation of this section of the Code are independent of, and in addition to, any adverse academic evaluation which results from the student's conduct. The course instructor is responsible for academic evaluation of a student's work and shall make that evaluation without regard to any disciplinary action which may or may not be taken against a student under the Student Code of Conduct.

Incompletes:

A grade of Incomplete can be granted only if 80% of the course work has been satisfactorily completed (passing) and if there is a documented family or medical emergency. In addition, you must communicate with me your desire for an incomplete and you must sign the Incomplete Grade request form that includes a mutually agreed upon date of completion and a list of work to be completed. An incomplete is NOT automatically granted. I do not grant a grade of incomplete if you fail to meet any of the above criteria.

Office Hours:

Office hours are done on a first-come first-served basis. BY APPOINTMENT is just that. You must make prior arrangements to meet with me. I read my **lcsc.edu** e-mail several times daily; this is a very reliable means to contact me. My homepage contains a link to this class and its schedule of assignments. If you miss a class session, look at the class pages on the Web and BBCE to see what assignments were made.

Respect for Others:

Please demonstrate respect for the other students in this class:

- Turn off all audible pagers, cell phones and other such equipment while in the classroom.
- If you must take care of personal issues during class, please leave quietly and return quietly.
- If you must be late to class, please enter quietly, take your seat and get to work.
- If you know you must leave early, please let me know. Then leave quietly.

There is a pencil sharpener around the corner to the right of the stairwell. There is a uni-sex bathroom around the corner to the right of the stairwell. There is a ladies' bathroom to the right just before you enter this room.

Computer Lab:

The computer lab (MLH310) is open two nights a week starting after Labor Day.

Important dates:

Aug 29	Last Day to Register/Add/Drop Classes Online
Sep 1	Labor Day – no classes
Sep 2	Instructor's Written Approval Required to Add Class
Sep 8	Last Day to Add Class(es) or to Drop without 'W' Grade
Oct 1	Last Day to Apply for Graduation Spring 2009
Oct 24	Midterm Grades Posted
Oct 31	Last Day to Withdraw from Class(es) or College for the Semester
Nov 3-6	Advance Registration for Spring 2009
Nov 24-28	Thanksgiving Break – no classes

The last date to drop this class without a permanent entry on your transcript is **Sept. 8**. The last date to withdraw from this course with a W on your transcript is **Oct. 31**. After that date, withdrawal is permitted by petition only and approval of the division chair, Dr. Matt Johnston. The division chair requires documentation of **extraordinary** circumstances that prevented you from withdrawing by the deadline.

V. METHODS OF INSTRUCTION

I utilize a lecture/lab format. Programming and other work are assigned each week. Meeting in the lab affords us the opportunity for hands-on work to test and apply material from lecture. You will have activities which require you to work through the material, learning to understand it in your own way.

Tentative Weekly Schedule

Week	Topics
1	Introduction to Course
2	File System
3	vi
4	Workday
5	Workday
6	vi
7	Shell
8	Filters
9	Filters
10	Filters
11	Shell programming
12	Shell programming
13	Thanksgiving Break
14	Process control

15	Process control
16	Process control

A detailed schedule is available to you on BBCE.

VI. PERFORMANCE CRITERIA FOR STUDENTS

- Demonstrate persistent and active learning through an organized approach that includes completion of activities, attending every class and lab period and bringing appropriate materials to support quality and active participation in lab activities, and appropriately using available resources to aid in learning course content.
- Develop high-level modeling skills where appropriate design elements can be quickly and correctly identified.
- Professionally perform computer skills and functions when applying computer science principles through correct use of equipment and software; and quality documentation of procedures such that others can easily read and assess the work performed.
- Exemplify effective communication skills as demonstrated through the use of appropriate technical language and as documented in assignments and problem solutions.

Grading Breakdown for Activities Designed to Meet the Above Performance Criteria:

Evaluation Methods:

Programming	100%
TOTAL	100%

Grading Scale (expressed as percentages):

A	≥ 94
A-	≥ 90 to < 94
B+	≥ 87 to < 90
B	≥ 83 to < 87
B-	≥ 80 to < 83
C+	≥ 77 to < 80
C	≥ 73 to < 77
C-	≥ 70 to < 73
D+	≥ 67 to < 70
D	≥ 60 to < 67
F	< 60

Description of Methods for Evaluation

Programming (100%): You will be assigned exercises throughout the semester. Many of our assignments will be graded during lab. Regular attendance is important. Under normal circumstances, if you have not completed your assignment by the due date, you should submit the work you have done for partial credit. For the sake of your grade, you should ALWAYS turn in SOMETHING. **Be sure to keep ALL graded material.**

Be sure to backup your work. USB devices fail; hard drives fail. Keep copies of your work on different devices. Failure to have your program ready because your drive crashed is not an acceptable excuse.

Profile of Good Problem Solvers:

Computer scientists must be good problem solvers. Problem solvers:

- Have a strong desire and ability to identify and define the current problems confronting a process, system, person or group.
- Enjoy the problem solving process and as much as the solutions generated by the process.
- Seek to improve their skills by self-assessing their use of the problem solving process.
- Work to increase their ability to manage frustration so they can take on more challenging problems.
- Generate activities that produce opportunities to solve more intellectual problems.
- Adapt to and make effective use of changing technological environments.
- Seek out, learn, and use technological tools to improve the quality of their use of the problem solving process and the presentation of problem solutions.
- Logically identify key issues by utilizing previous problem solutions and/or outside expertise.
- Seek to improve the quality of problem solutions through the selection and use of better analytical modeling and tools.
- Partition a problem into a clear set of manageable subproblems.
- Produce multiple problem solutions
- Effectively integrate subsolutions into a cohesive solution.
- Assess the problem solution to make sure that every underlying assumption has been identified, tested, and documented.
- Make sure that every solution is tested for both reliability and robustness.
- Produce a documented problem solution that others can accept for quality.