

CSCI 0050 - INTRODUCTION TO UNIX/LINUX

Catalog Description

Prerequisite: Completion of CSCI 10 with grade of "C" or better
Hours: 72 (54 lecture, 18 laboratory)

Description: An introduction to the Unix and Linux operating systems with an emphasis on system programming. Topics include the filesystem, permissions, regular expressions, processes, networking, basic system administration, and shell scripting. Extensive hands-on experience using the command line interface. (CSU, UC)

Course Student Learning Outcomes

- CSLO #1: Explain the basic Linux shell commands and utilize piping and redirection to construct more complex commands.
- CSLO #2: Utilize regular expressions to search for, extract, and transform data from input.
- CSLO #3: Construct shell programs utilizing control structures.
- CSLO #4: Read man pages and use the man command to find Unix/Linux commands to perform a specific operation.
- CSLO #5: Explain basic networking commands of ssh, sftp and how to use email.

Effective Term

Fall 2020

Course Type

Credit - Degree-applicable

Contact Hours

72

Outside of Class Hours

90

Total Student Learning Hours

162

Course Objectives

Lecture Objectives:

1. Identify and describe prominent figures in Unix/Linux history: Ken Thompson, Dennis Ritchie, and Linus Torvalds.
2. From a written description, analyze and calculate correct permissions to set on files and directories using both numeric and symbolic representations.
3. Specify file locations using absolute and relative paths.

Laboratory Objectives:

1. Create, edit, and remove files and directories using core commands: cp, mv, mkdir, rmdir, rm, ls.
2. Construct pipelines of commands to filter data.
3. Construct correct regular expressions to match patterns in data files.
4. Create and kill processes using job control commands and signals.
5. Copy files to and from remote machines.
6. Diagnose network problems using ping, nslookup, and netstat.

7. From written descriptions, write correct shell scripts utilizing sequencing of commands, conditionals, and iteration.
8. Compile and install third-party software packages.
9. Compile and execute user-supplied programs in Java, C, or C++.

General Education Information

- Approved College Associate Degree GE Applicability
- CSU GE Applicability (Recommended-requires CSU approval)
- Cal-GETC Applicability (Recommended - Requires External Approval)
- IGETC Applicability (Recommended-requires CSU/UC approval)

Articulation Information

- CSU Transferable

Methods of Evaluation

- Objective Examinations
 - Example: Example objective question: 1. Which of the following are relative paths? a. /home/unix/bbrown b. ../lab1/readme.txt c. public_html/sierra.jpg d. ../jsmith/./overview e. /bin/grep Answer: b, c, and d 2. The output of the "ls -l lab2.txt" file shows this: -rwxr-xr- 1 jsmith engineering 253 2012-3-12 lab2.txt What can the jsmith user do with this file? a. Alter its contents b. Delete it c. Make it the current working directory d. Can't tell from the information given Answer: a (Not b, because only the containing directory will dictate whether it can be deleted; not c because it's not a directory.)
- Problem Solving Examinations
 - Example: Write a bash shell script to create a thumbnail of each image listed on the command line. Place each thumbnail in a directory called thumbs. Each thumbnail should be 50 pixels high and the filename should be the same as the original. The original remains unaffected. Rubric Grading. Solution: while [\$# -gt 0] do jpegtopnm \$1 | pnmscale -height 50 | pnmtjpeg > thumbs/\$1 shift done
- Projects
 - Example: Find at least three sources that describe the philosophies behind open source software, free software, and proprietary software. Write a two-page essay that analyzes situations in which each would be most appropriate for a) medical diagnostic software, b) voting software, and c) web site software. Rubric Grading.

Repeatable

No

Methods of Instruction

- Laboratory
- Lecture/Discussion
- Distance Learning

Lab:

1. The instructor reviews the commands that manipulate permissions: chmod, chown, chgrp, and umask. Using a computer and projector on a live Unix/Linux system, the instructor demonstrates several examples of altering permissions and their effects. The instructor puts students into groups of 3 or 4 and assigns them Unix group names. Students work in groups to share files among their group members, following written problem descriptions. The instructor

monitors progress by checking answers and visually verifying the results. (Laboratory Objective 7)

- On a prior day, the instructor requests that students bring in a Java, C, or C++ program they have been working on in another class, or a program they have written in the past. The instructor may also provide a program for students who don't have one of their own. On the day of the assignment, the instructor describes how to compile programs using the javac or gcc compilers and how to run them on the command line. The instructor also shows how to build a Makefile or build.xml file to automate the process using make or ant. In the lab, students apply the instructions to their own project (or instructor-supplied one). The instructor determines if the program has been successfully built by having the student show the intermediate object or class files and running the final executable. (Laboratory Objective 9)

Lecture:

- Assume the students have read the appropriate section in the textbook about permissions or they have watched supplemental videos (if available). The instructor reviews file permissions by explaining the numeric (e.g., 764) and symbolic (drwxrwx-r-) representation. The students work problems of converting one representation into the other, since they are equivalent. (Lecture Objective 2)

Distance Learning

- The instructor will present a video lecture describing how to compile programs using the javac or gcc compilers and how to run them on the command line. The instructor also shows how to build a Makefile or build.xml file to automate the process using make or ant. Students apply the instructions to their own project (or instructor-supplied one). The instructor determines if the program has been successfully built by having the student upload the intermediate object or class files and running the final executable. (Laboratory Objective 9)

Typical Out of Class Assignments

Reading Assignments

- Read the chapter on the AWK programming language. Pay attention to examples of extracting columnar data from text files based on numeric criteria, such as "display all ages over 18." Practice it extracting columnar data by creating a criteria of your own. Be prepared to discuss in class.
- Read web sites about the history of Unix. Take note of prominent figures over the past thirty years: Ken Thompson, Dennis Ritchie, Richard Stallman, and Linus Torvalds. Be prepared to discuss in class.

Writing, Problem Solving or Performance

Example 1: Use awk, grep, sort, and uniq to analyze a web server log file and show how many "hits" the server received each hour of the day, over the span of a month. Solution: `awk -F: '{print $2}' /var/log/httpd/access_log | sort -n | uniq -c` Example 2: Calculate the correct umask so that future directories will have permission drwxr-x-x and files will be -rwxr-x-. Solution: `umask 026` Example 3: Write a three-page report describing the similarities and differences between open source software, free software, and proprietary software. Which would be most appropriate in these situations: a) medical diagnostic software, b) computerized voting software, c) web service software?

Other (Term projects, research papers, portfolios, etc.)

Hands-on experience using the command line in a shared, server environment.

Required Materials

- Practical Guide to Linux Commands, Editors, and Shell Programming
 - Author: Sobell, Mark
 - Publisher: Prentice Hall
 - Publication Date: 2014
 - Text Edition: 4th
 - Classic Textbook?: No
 - OER Link:
 - OER:
- Your Unix/Linux: The Ultimate Guide
 - Author: Das, Sumitabha
 - Publisher: McGraw-Hill
 - Publication Date: 2012
 - Text Edition: 3rd
 - Classic Textbook?: No
 - OER Link:
 - OER:
- The Linux Command Line: A Complete Introduction
 - Author: Shotts, William E., jr.
 - Publisher: No Starch Press
 - Publication Date: 2019
 - Text Edition: 2nd
 - Classic Textbook?: No
 - OER Link:
 - OER:

Other materials and-or supplies required of students that contribute to the cost of the course.