

Course Outline

General Information

COURSE ID (CB01A AND CB01B)

CIS 18B

COURSE TITLE (CB02)

Advanced UNIX/LINUX

COURSE CREDIT STATUS

Credit - Degree Applicable

EFFECTIVE TERM

Fall 2021

COURSE DESCRIPTION

Expanded coverage of regular expressions and grep. Advanced topics in Unix/Linux include egrep, find, sed, awk, file archiving, compression, and conversion, version control, makefile, basic shell scripts and installation of a Linux distribution.

FACULTY REQUIREMENTS

COURSE FAMILY

Not Applicable

Course Justification

This course belongs on the Unix/Linux Operating System Certificates. It is CSU and UC transferable. This course teaches advanced Unix/Linux commands. It is the second course in a three-course sequence preparing students for using the Unix/Linux platform for mobile and cloud computing, network servers and database.

Foothill Equivalency

DOES THE COURSE HAVE A FOOTHILL EQUIVALENT?

No

FOOTHILL COURSE ID

Formerly Statement

Course Development Options

BASIC SKILL STATUS (CB08)

Course is not a basic skills course.

GRADE OPTIONS

- Letter Grade
- Pass/No Pass

REPEAT LIMIT

0

Transferability & Gen. Ed. Options

Information below is subject to change. For the official listing of courses, their approval dates, and transfer credit limitations, check the De Anza catalog (by academic year), [ASSIST.ORG \(https://assist.org/\)](https://assist.org/) and [C-ID.NET \(https://c-id.net/\)](https://c-id.net/).

TRANSFERABILITY

Transferable to both UC and CSU

Units and Hours

Summary

MINIMUM CREDIT UNITS

4.5

MAXIMUM CREDIT UNITS 4.5

Weekly Student Hours

Type	In Class	Out of Class
Lecture Hours	4.0	8.0
Laboratory Hours	1.5	0.0

Course Student Hours

COURSE DURATION (WEEKS)

12.0

HOURS PER UNIT DIVISOR

36.0

Course In-Class (Contact) Hours

LECTURE

48.0

LABORATORY

18.0

TOTAL

66.0

Course Out-of-Class Hours

LECTURE

96.0

LABORATORY

0.0

NA

0.0

TOTAL

96.0

Prerequisite(s)

CIS 18A

Corequisite(s)

Advisory(ies)

CIS 14A, CIS 22A, CIS 36A or CIS 40

Limitation(s) on Enrollment

Entrance Skill(s)

General Course Statement(s)

Methods of Instruction

Lecture and visual aids

Discussion of assigned reading Discussion and problem solving performed in class Quiz and examination review performed in class Homework and extended projects Collaborative learning and small group exercises Laboratory discussion sessions and quizzes that evaluate the proceedings weekly laboratory exercises

Assignments

- A. Reading textbook and lecture notes
- B. 6-10 homework assignments to be done on the Unix/Linux system, covering the Lab Topics specified in X. below, half completed in the computer lab, half completed as homework.

Methods of Evaluation

- A. Successful completion of homework assignments, which are evaluated for completeness and correctness
- B. In-class problems, group collaborative problems, exam questions and/or online assignments used to demonstrate the ability to effectively and efficiently use the appropriate Unix/Linux utilities to solve problems
- C. At least one written midterm examination requiring the appropriate use of Unix/Linux utilities to solve problems. The exam is to be evaluated for correctness and efficient use of Unix/Linux utilities.
- D. A comprehensive final examination requiring the skills to use appropriate Unix/Linux utilities to solve problems. The exam is to be evaluated for correctness and efficient use of Unix/Linux utilities.

Essential Student Materials/Essential College Facilities

Essential Student Materials:

- None.

Essential College Facilities:

- Access to a computer running the UNIX/LINUX operating system

Examples of Primary Texts and References

Author	Title	Publisher	Date/ Edition	ISBN
Sobell, Mark; A Practical Guide to Linux Commands, Editors, and Shell Programming; 4th Edition; Prentice-Hall, 2018; ISBN-13: 978-0-13-477531-9				

Examples of Supporting Texts and References

Author	Title	Publisher
None.		

Learning Outcomes and Objectives

Course Objectives

- Apply regular expressions with different filters
- Apply the grep utility in different search applications
- Apply the sed utility on the command line and with input file
- Apply the awk utility in text processing and calculating applications
- Use the make utility to manage software projects
- Apply utilities for file compression
- Create archive files for backup and storage
- Convert text files from the Unix / Linux format to other OS formats
- Automate tasks with basic shell scripts
- Create version control in a project

CSLOs

- Use the Unix/Linux Operating System utilities, shell features, and regular expressions for advanced text file manipulation.

Outline

- A. Apply regular expressions with different filters
 - 1. Atoms and operators
 - 2. Comparison between mathematical expressions and regular expressions
 - 3. Atoms (single character, dot, class, anchor, and back-reference)
 - 4. Operators (sequence, alternation, repetition, group, and save)
 - 5. Greedy and non-greedy operations
- B. Apply the grep utility in different search applications
 - 1. Grep operation (inherent loop)
 - 2. Grep family (egrep, fgrep, and egrep)
 - 3. Atoms and operators supported by each member of the family
 - 4. Use of options with grep family
 - 5. Pattern matching with grep family
 - 6. Application examples
- C. Apply the sed utility on the command line and with input file
 - 1. Sed as a filter
 - 2. Sed script and instruction format
 - 3. Operation of sed (nest inherent loops)

4. Role of pattern space and hold space
 5. Script execution step by step
 6. Addresses and address types
 7. Commands in sed
 8. Simulation of grep commands using sed
 9. Applications
- D. Apply the awk utility in text processing and calculating applications
1. Awk as a programming language
 2. Execution of an awk command
 3. Operation of awk (inherent nested loops and selection)
 4. Fields and records
 5. Buffers and variables
 6. Awk script
 7. Patterns and actions
 8. Pattern types
 9. Actions and statements: expression statement, output statement, decision, loops, control, function call
 10. User-defined functions
 11. Simulating grep commands using awk.
 12. Simulating sed commands using awk
 13. Applications
- E. Use the make utility to manage software projects
1. Code Building Background
 - a. Building an executable from multiple source file
 - b. File dependency is a software application
 - c. The role of make in a software project
 2. Running make
 - a. Without a makefile
 - b. With a makefile
 3. The makefile
 - a. makefile instructions
 - b. Target file and source files
 - c. Dependencies in a makefile instruction
 - d. Built in recursion in the makefile
 - e. Macros
 - f. Phony targets
 - g. Common Suffixes
 4. Use of make outside of software development
- F. Apply utilities for file compression
1. Explanation and algorithm of file compression
 2. gzip and gunzip
 - a. Compress files
 - b. Uncompress files
 - c. Options
- G. Create archive files for backup and storage
1. Archiving files
 2. Using tar to archive files
 3. Archiving, compressing files and editing the archive file
- H. Convert text files from the Unix / Linux format to other OS formats
1. Unix to DOS
 2. DOS to Unix
- I. Automate tasks with basic shell scripts
1. Compare shell scripts to programs
 - a. Linux as a native OS
 - b. Linux as virtual OS
 - c. Linux on an external device
 2. Use of shell scripts
 3. Writing shell scripts
 4. Running shell scripts
 5. Running Linux on the PC
 6. Running Linux on a PC
 7. Linux as a native operating system
 8. Installation
 9. Boot
 10. Configuration
- J. Create version control in a project
1. Explore the role of version control of files in a project
 2. Use git for version control
 - a. Initialize a git directory
 - b. Configure the git directory
 3. Add and delete files in the repository
 - a. Put files in the staging area
 - b. Commit files into the repository
 4. Create branches
 - a. Switch branches
 - b. Merge branches
 5. Accessing remote files and repositories

- a. Clone a repository
- b. Push / pull files to the remote repository

Lab Topics

- A. Review utilities to manipulate files and directories, look up system and user status, and filter input data
- B. Write and describe regular expressions that match given requirements
- C. Use the grep family of utilities with regular expressions
- D. Use the sed utility on the command line, including address fields and sed commands
- E. Use the sed utility with a sed script file
- F. Use awk on the command line, with basic awk commands and regular expression
- G. Write awk scripts, complete with regular expressions, awk operators and action commands, and programming constructs such as selection and looping
- H. Write awk scripts to work with arrays and system commands
 - I. Write makefiles to build software executables
 - J. Use file compression utilities to compress and uncompress files
 - K. Use archiving utilities to compress and archive files
 - L. Use file conversion utilities to format text files and transfer files to different OS platforms
- M. Write and run a basic bash shell script
- N. Install and boot a Linux distribution
- O. Use version control to manage files in a project with multiple branches