College of San Mateo
Official Course Outline

1. **COURSE ID:** CIS 121 **TITLE:** UNIX/Linux
   Units: 3.0 units **Hours/Semester:** 48.0-54.0 Lecture hours; and 96.0-108.0 Homework hours
   Method of Grading: Grade Option (Letter Grade or Pass/No Pass)
   Recommended Preparation:
   Completion of CIS 110.

2. **COURSE DESIGNATION:**
   Degree Credit
   Transfer credit: CSU; UC

3. **COURSE DESCRIPTIONS:**
   Catalog Description:
   Introduction to the UNIX/Linux operating system. Includes a brief introduction to operating systems and UNIX history. Covers UNIX file systems, common shell features including 1/0 redirection, piping, command substitution, and simple job control. Introduces shell-specific facilities, including use of environmental and local variables, and common UNIX utilities. Also includes advanced topics such as shell scripting, communications, and system administration. Intended for students with previous programming experience.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**
   Upon successful completion of this course, a student will meet the following outcomes:
   1. Describe the functions of an operating system.
   2. Employ common UNIX shell features including I/O redirection, piping, command substitution, and simple job control.
   3. Explain shell-specific facilities including the use of environmental and local variables, and the built-in programming language.
   4. Analyze problems and design UNIX solutions using shell command files and scripts.
   5. Describe how UNIX supports processes, memory management, input/output, and the file system.
   6. Set up a UNIX or Linux environment.
   7. Use common and advanced UNIX utilities.
   8. Describe the main UNIX system administration tasks.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**
   Upon successful completion of this course, a student will be able to:
   1. Describe the functions of an operating system.
   2. Employ common UNIX shell features including I/O redirection, piping, command substitution, and simple job control
   3. Explain shell-specific facilities including the use of environmental and local variables, and the built-in programming language.
   4. Analyze problems and design UNIX solutions using shell command files and scripts.
   5. Describe how UNIX supports processes, memory management, input/output, and the file system.
   6. Set up a UNIX or Linux environment.
   7. Use common and advanced UNIX utilities.
   8. Describe the main UNIX system administration tasks.

6. **COURSE CONTENT:**
   Lecture Content:
   Objectives 1, 2
   1. Introduction
      A. Introduction to Operating Systems
         a. UNIX
         b. Linux
      B. Accessing the UNIX system
      C. Using Commands
         a. Logging on and off
         b. Changing the password

- c. Getting online help
- d. Stopping a program
- e. Using the file system
- f. Editing files
  - i. Entering text
  - ii. Common editing features
  - iii. Moving the cursor
  - iv. Deleting, Pasting, and Undoing
  - v. Searching
  - vi. Saving/Loading files

Objectives 3, 4

1. UNIX/Linux Shells and Commands
   - A. UNIX/Linux shells
     - a. Common shell features
       - i. I/O redirection
       - ii. Piping
       - iii. Command substitution
       - iv. Simple job control
     - b. Shell-specific facilities
       - i. Use of environment and local variables
       - ii. Built-in programming language
       - iii. Advanced I/O redirection
     - c. The Bourne shell
       - i. Start-up files
       - ii. Shell variables
       - iii. Arithmetic
       - iv. Conditional expressions
       - v. Control structures
       - vi. Built-in commands
     - d. The Korn shell
       - i. Start-up files
       - ii. Aliases
       - iii. History mechanism
       - iv. iv and emacs line editors
       - v. Arithmetic
       - vi. Functions
       - vii. Job control
     - e. The C shell
       - i. Start-up files
       - ii. Variables and lists
       - iii. Expressions
       - iv. Aliases
       - v. History mechanism
       - vi. Control structures
       - vii. Job control

Objectives 2, 3, 4, 5

1. Shell Scripting and Processes
   - A. Shell Scripting
     - a. Creating a Simple Shell Script
     - b. Command Separation and Grouping
     - c. Redirecting Standard Error
     - d. Job Control
       - i. List
       - ii. Foreground
       - iii. Background
   - B. Processes
     - a. Process Structure
     - b. Process Identification
     - c. Executing a Command
     - d. Invoking a Shell Script

i. Specifying a Shell
ii. Start-up files
C. Parameters and Variables
a. Keyword Variables

Objective 6

1. File Manipulation and Internals
A. UNIX Utilities
a. Filtering files
b. Sorting files
c. Comparing files
d. Archiving files
e. Searching for files
f. Scheduling commands
g. Hard and soft links
h. Checking for mail
i. Transforming files
j. Identifying shells
B. Overview of UNIX Internals
a. Processes
b. Memory Management
c. Input/Output
d. File System

Objective 7

1. Linux Installation
A. Linux
a. Components
b. Installation and Configuration
c. Internetworking with Other Operating Systems

Objective 8

1. UNIX/Linux Utilities
A. Advanced UNIX Utilities
a. Switching Users
b. Mounting File Systems

Objectives 8, 9

1. Communications/Networking
A. Overview of Network Concepts and Terminology
a. Building a Network
i. Ethernets
ii. Bridges
iii. Gateways
iv. Routers
b. Internetworking
i. Packet Switching
ii. Internet Addresses
iii. Naming
iv. Routing
v. Security
vi. Network Programming
c. UNIX Network Utilities
i. finger
ii. ftp/ssh
iii. wall
iv. who
B. Overview of Systems Programming
a. Error Handling
b. Regular File Management
c. Process Management
d. Signals
e. Interprocess Communication (IPC)
f. The Internet Shell

Objectives 6, 8

1. System Administration
   - A. Advanced Commands and Utilities
     - a. Starting and Stopping the System
     - b. Maintaining and backing up the File System
     - c. Maintaining User Accounts
     - d. Installing Software
     - e. Installing and Configuring Peripherals
     - f. Managing the Interface to the Network
     - g. Automating Repetitive Tasks
     - h. Performing System Accounting
     - i. Configuring the Kernel
     - j. Checking System Security

7. **REPRESENTATIVE METHODS OF INSTRUCTION:**

Typical methods of instruction may include:

- A. Lecture
- B. Discussion
- C. Other (Specify): The course will include the following instructional methods as determined appropriate by the instructor: • Lecture will be used to introduce new topics; • Teacher will model problem-solving techniques; • Class will solve a problem together, each person contributing a potential "next step"; • Students will participate in short in-class projects (in teacher-organized small groups) to ensure that students experiment with the new topics in realistic problem settings; • Teacher will invite questions AND ANSWERS from students, generating discussion about areas of misunderstanding; • Teacher will create and manage an Internet conference for discussion of course topics; and • Students will work in small groups to solve programming assignments.

8. **REPRESENTATIVE ASSIGNMENTS**

Representative assignments in this course may include, but are not limited to the following:

Writing Assignments:

Writing takes the form of completing hands-on Linux/Unix assignments. There is one assignment for each chapter. Additionally there are "Projects" that are larger in scope and require students to work in groups.

Reading Assignments:

Students read a chapter from the text every week. Chapter is 15-20 pages of technical reading. Each chapter has hands-on exercises the student is expected to complete as part of the reading. An example of this is "Installing Linux." The material is READ, but then the student is required to actually download and install the operating system.

9. **REPRESENTATIVE METHODS OF EVALUATION**

Representative methods of evaluation may include:

- A. Class Work
- B. Exams/Tests
- C. Group Projects
- D. Homework
- E. Projects
- F. Quizzes
- G. Written examination
- H. • Bi-weekly quizzes (short answer--from textbook material) to provide feedback to students and teacher; • Assessment of student contributions during class discussion and project time; • Individual programming assignments; • Midterm and Final exams - short answer from textbook material, general problem solving - similar to in-class work, short program segments - similar to programming assignments; • Assessment of group participation on course projects, including peer-assessment of participation and contribution to the group effort.

10. **REPRESENTATIVE TEXT(S):**

Possible textbooks include:

- A. Blum & Bresnahan. *Linux Command Line and Shell Scripting Bible*, 4th ed. Wiley, 2020
- B. Nemeth & Snyder. *UNIX and Linux System Administration Handbook*, 5th ed. Wiley Professional, 2017
- C. Sobell. *A Practical Guide to Linux Commands, Editors, and Shell Programming*, 4th ed. Addison-Wesley Professional, 2018

Origination Date: September 2020
Curriculum Committee Approval Date: November 2020
Effective Term: Fall 2021
Course Originator: Kamran Eftekhari