uCertify

Course Outline

Advanced Programming in the UNIX Environment

04 Aug 2025

1. Exercises, Quizzes, Flashcards & Glossary

   Number of Questions
2. Expert Instructor-Led Training
3. ADA Compliant & JAWS Compatible Platform
4. State of the Art Educator Tools
5. Award Winning Learning Platform (LMS)
6. Chapter & Lessons

   Syllabus

   Chapter 1: Preface

   Chapter 2: UNIX System Overview

   Chapter 3: UNIX Standardization and Implementations

   Chapter 4: File I/O

   Chapter 5: Files and Directories

   Chapter 6: Standard I/O Library

   Chapter 7: System Data Files and Information

   Chapter 8: Process Environment

   Chapter 9: Process Control

   Chapter 10: Process Relationships

   Chapter 11: Signals

   Chapter 12: Threads

   Chapter 13: Thread Control

   Chapter 14: Daemon Processes

   Chapter 15: Advanced I/O

   Chapter 16: Interprocess Communication

   Chapter 17: Network IPC: Sockets

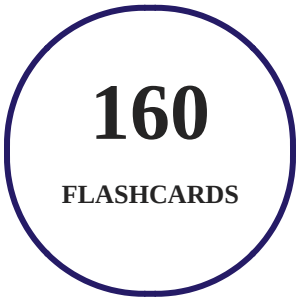
   Chapter 18: Advanced IPC

   Chapter 19: Terminal I/O

   Chapter 20: Pseudo Terminals

1. flashcards

Flashcards are effective memory-aiding tools that help you learn complex topics easily. The flashcard will help you in memorizing definitions, terminologies, key concepts, and more. There is no limit to the number of times learners can attempt these. Flashcards help master the key concepts.



2. Glossary of terms

uCertify provides detailed explanations of concepts relevant to the course through Glossary. It contains a list of frequently used terminologies along with its detailed explanation. Glossary defines the key terms.

3. Expert Instructor-Led Training

uCertify uses the content from the finest publishers and only the IT industry's finest instructors. They have a minimum of 15 years real-world experience and are subject matter experts in their fields. Unlike a live class, you can study at your own pace. This creates a personal learning experience and gives you all the benefit of hands-on training with the flexibility of doing it around your schedule 24/7.

4. ADA Compliant & JAWS Compatible Platform

uCertify course and labs are ADA (Americans with Disability Act) compliant. It is now more accessible to students with features such as:

- Change the font, size, and color of the content of the course
- Text-to-speech, reads the text into spoken words
- Interactive videos, how-tos videos come with transcripts and voice-over
- Interactive transcripts, each word is clickable. Students can clip a specific part of the video by clicking on a word or a portion of the text.

JAWS (Job Access with Speech) is a computer screen reader program for Microsoft Windows that reads the screen either with a text-to-speech output or by a Refreshable Braille display. Student can easily navigate uCertify course using JAWS shortcut keys.

5. State of the Art Educator Tools

uCertify knows the importance of instructors and provide tools to help them do their job effectively. Instructors are able to clone and customize course. Do ability grouping. Create sections. Design grade scale and grade formula. Create and schedule assessments. Educators can also move a student from self-paced to mentor-guided to instructor-led mode in three clicks.

6. Award Winning Learning Platform (LMS)

uCertify has developed an award winning, highly interactive yet simple to use platform. The SIIA CODiE Awards is the only peer-reviewed program to showcase business and education technology's finest products and services. Since 1986, thousands of products, services and solutions have been

recognized for achieving excellence. uCertify has won CODiE awards consecutively for last 7 years:

- **2014**
    1. Best Postsecondary Learning Solution

- **2015**
    1. Best Education Solution
    2. Best Virtual Learning Solution
    3. Best Student Assessment Solution
    4. Best Postsecondary Learning Solution
    5. Best Career and Workforce Readiness Solution
    6. Best Instructional Solution in Other Curriculum Areas
    7. Best Corporate Learning/Workforce Development Solution

- **2016**
    1. Best Virtual Learning Solution
    2. Best Education Cloud-based Solution
    3. Best College and Career Readiness Solution
    4. Best Corporate / Workforce Learning Solution
    5. Best Postsecondary Learning Content Solution
    6. Best Postsecondary LMS or Learning Platform
    7. Best Learning Relationship Management Solution

- **2017**
    1. Best Overall Education Solution
    2. Best Student Assessment Solution
    3. Best Corporate/Workforce Learning Solution
    4. Best Higher Education LMS or Learning Platform

- **2018**
    1. Best Higher Education LMS or Learning Platform

2. Best Instructional Solution in Other Curriculum Areas
3. Best Learning Relationship Management Solution

- **2019**
  1. Best Virtual Learning Solution
  2. Best Content Authoring Development or Curation Solution
  3. Best Higher Education Learning Management Solution (LMS)

- **2020**
  1. Best College and Career Readiness Solution
  2. Best Cross-Curricular Solution
  3. Best Virtual Learning Solution

7. Chapter & Lessons

uCertify brings these textbooks to life. It is full of interactive activities that keeps the learner engaged. uCertify brings all available learning resources for a topic in one place so that the learner can efficiently learn without going to multiple places. Challenge questions are also embedded in the chapters so learners can attempt those while they are learning about that particular topic. This helps them grasp the concepts better because they can go over it again right away which improves learning.

Learners can do Flashcards, Exercises, Quizzes and Labs related to each chapter. At the end of every lesson, uCertify courses guide the learners on the path they should follow.

Syllabus

Chapter 1: Preface

- Introduction

- Changes from the Second Edition

Chapter 2: UNIX System Overview

- Introduction
- UNIX Architecture
- Logging In
- Files and Directories
- Input and Output
- Programs and Processes
- Error Handling
- User Identification
- Signals
- Time Values
- System Calls and Library Functions
- Summary
- Exercises

Chapter 3: UNIX Standardization and Implementations

- Introduction
- UNIX Standardization

- UNIX System Implementations
- Relationship of Standards and Implementations
- Limits
- Options
- Feature Test Macros
- Primitive System Data Types
- Differences Between Standards
- Summary
- Exercises

Chapter 4: File I/O

- Introduction
- File Descriptors
- open and openat Functions
- creat Function
- close Function
- lseek Function
- read Function

- write Function
- I/O Efficiency
- File Sharing
- Atomic Operations
- dup and dup2 Functions
- sync, fsync, and fdatasync Functions
- fcntl Function
- ioctl Function
- /dev/fd
- Summary
- Exercises

Chapter 5: Files and Directories

- Introduction
- stat, fstat, fstatat, and lstat Functions
- File Types
- Set-User-ID and Set-Group-ID
- File Access Permissions

- Ownership of New Files and Directories
- access and faccessat Functions
- umask Function
- chmod, fchmod, and fchmodat Functions
- Sticky Bit
- chown, fchown, fchownat, and lchown Functions
- File Size
- File Truncation
- File Systems
- link, linkat, unlink, unlinkat, and remove Functions
- rename and renameat Functions
- Symbolic Links
- Creating and Reading Symbolic Links
- File Times
- futimens, utimensat, and utimes Functions
- mkdir, mkdirat, and rmdir Functions
- Reading Directories

- chdir, fchdir, and getcwd Functions
- Device Special Files
- Summary of File Access Permission Bits
- Summary
- Exercises

Chapter 6: Standard I/O Library

- Introduction
- Streams and FILE Objects
- Standard Input, Standard Output, and Standard Error
- Buffering
- Opening a Stream
- Reading and Writing a Stream
- Line-at-a-Time I/O
- Standard I/O Efficiency
- Binary I/O
- Positioning a Stream
- Formatted I/O

- Implementation Details
- Temporary Files
- Memory Streams
- Alternatives to Standard I/O
- Summary
- Exercises

Chapter 7: System Data Files and Information

- Introduction
- Password File
- Shadow Passwords
- Group File
- Supplementary Group IDs
- Implementation Differences
- Other Data Files
- Login Accounting
- System Identification
- Time and Date Routines

- Summary
- Exercises

Chapter 8: Process Environment

- Introduction
- main Function
- Process Termination
- Command-Line Arguments
- Environment List
- Memory Layout of a C Program
- Shared Libraries
- Memory Allocation
- Environment Variables
- setjmp and longjmp Functions
- getrlimit and setrlimit Functions
- Summary
- Exercises

Chapter 9: Process Control

- Introduction
- Process Identifiers
- fork Function
- vfork Function
- exit Functions
- wait and waitpid Functions
- waitid Function
- wait3 and wait4 Functions
- Race Conditions
- exec Functions
- Changing User IDs and Group IDs
- Interpreter Files
- system Function
- Process Accounting
- User Identification
- Process Scheduling
- Process Times

- Summary
- Exercises

Chapter 10: Process Relationships

- Introduction
- Terminal Logins
- Network Logins
- Process Groups
- Sessions
- Controlling Terminal
- tcgetpgrp, tcsetpgrp, and tcgetsid Functions
- Job Control
- Shell Execution of Programs
- Orphaned Process Groups
- FreeBSD Implementation
- Summary
- Exercises

Chapter 11: Signals

- Introduction
- Signal Concepts
- signal Function
- Unreliable Signals
- Interrupted System Calls
- Reentrant Functions
- SIGCLD Semantics
- Reliable-Signal Terminology and Semantics
- kill and raise Functions
- alarm and pause Functions
- Signal Sets
- sigprocmask Function
- sigpending Function
- sigaction Function
- sigsetjmp and siglongjmp Functions
- sigsuspend Function
- abort Function

- system Function
- sleep, nanosleep, and clock_nanosleep Functions
- sigqueue Function
- Job-Control Signals
- Signal Names and Numbers
- Summary
- Exercises

Chapter 12: Threads

- Introduction
- Thread Concepts
- Thread Identification
- Thread Creation
- Thread Termination
- Thread Synchronization
- Summary
- Exercises

Chapter 13: Thread Control

- Introduction
- Thread Limits
- Thread Attributes
- Synchronization Attributes
- Reentrancy
- Thread-Specific Data
- Cancel Options
- Threads and Signals
- Threads and fork
- Threads and I/O
- Summary
- Exercises

Chapter 14: Daemon Processes

- Introduction
- Daemon Characteristics
- Coding Rules
- Error Logging

- Single-Instance Daemons
- Daemon Conventions
- Client–Server Model
- Summary
- Exercises

Chapter 15: Advanced I/O

- Introduction
- Nonblocking I/O
- Record Locking
- I/O Multiplexing
- Asynchronous I/O
- readv and writev Functions
- readn and writen Functions
- Memory-Mapped I/O
- Summary
- Exercises

Chapter 16: Interprocess Communication

- Introduction
- Pipes
- popen and pclose Functions
- Coprocesses
- FIFOs
- XSI IPC
- Message Queues
- Semaphores
- Shared Memory
- POSIX Semaphores
- Client–Server Properties
- Summary
- Exercises

Chapter 17: Network IPC: Sockets

- Introduction
- Socket Descriptors
- Addressing

- Connection Establishment
- Data Transfer
- Socket Options
- Out-of-Band Data
- Nonblocking and Asynchronous I/O
- Summary
- Exercises

Chapter 18: Advanced IPC

- Introduction
- UNIX Domain Sockets
- Unique Connections
- Passing File Descriptors
- An Open Server, Version 1
- An Open Server, Version 2
- Summary
- Exercises

Chapter 19: Terminal I/O

- Introduction
- Overview
- Special Input Characters
- Getting and Setting Terminal Attributes
- Terminal Option Flags
- stty Command
- Baud Rate Functions
- Line Control Functions
- Terminal Identification
- Canonical Mode
- Noncanonical Mode
- Terminal Window Size
- termcap, terminfo, and curses
- Summary
- Exercises

Chapter 20: Pseudo Terminals

- Introduction

- Overview
- Opening Pseudo-Terminal Devices
- pty_fork Function
- pty Program
- Using the pty Program
- Advanced Features
- Summary
- Exercises

Chapter 21: A Database Library

- Introduction
- History
- The Library
- Implementation Overview
- Centralized or Decentralized?
- Concurrency
- Building the Library
- Source Code

- Performance
- Summary
- Exercises

Chapter 22: Communicating with a Network Printer

- Introduction
- The Internet Printing Protocol
- The Hypertext Transfer Protocol
- Printer Spooling
- Source Code
- Summary
- Exercises

Chapter 23: Appendix B. Miscellaneous Source Code

- B.1. Our Header File
- B.2. Standard Error Routines

8. Live Labs

The benefits of live-labs are:

- Exam based practical tasks
- Real equipment, absolutely no simulations
- Access to the latest industry technologies
- Available anytime, anywhere on any device
- Break and Reset functionality
- No hardware costs

Lab Tasks

File I/O

- Using the open, openat, and lseek Functions
- Performing File Operations
- Using File Synchronization Functions

Files and Directories

- Retrieving File Metadata Using System Calls
- Managing File Access Permissions
- Using the umask Function
- Managing Ownership of Files and Directories
- Managing and Analyzing Files
- Managing Files and Directories
- Creating and Reading Symbolic Links
- Managing File Timestamps
- Performing File Timestamp Operations
- Traversing the File Hirerachy
- Reading and Managing Directories

Standard I/O Library

- Using the fwide Function
- Handling Standard Input, Output, and Error
- Performing File Stream Operations
- Exploring File I/O Techniques
- Positioning a Stream
- Formatting I/O
- Creating Temporary Files
- Using Memory Streams

System Data Files and Information

- Retrieving System Information
- Handling Time and Data Routines

Process Environment

- Creating the main Function
- Using Command-Line Arguments
- Allocating Dynamic Memory
- Using Environment Variables
- Handling Non-Local Jumps and Resource Limits

Process Control

- Identifying Processes
- Creating and Terminating a Process
- Demonstrating Race Conditions in Process Synchronization
- Using exec Functions
- Using Process Accounting, Scheduling, and User Identification

Process Relationships

- Using the setsid and getsid Functions
- Using the tcgetpgrp, tcsetpgrp, and tcgetsid Functions
- Managing Jobs
- Handling Orphaned Process Groups

Signals

- Using the kill and raise Functions
- Handling Signals
- Handling Advanced Signals
- Using the system Function
- Using the Sleep and Signal Queue Functions

Threads

- Managing Thread Identification, Creation, Termination, and Synchronization

Thread Control

- Configuring Synchronization Attributes
- Using Threads with Signals, Forks, and I/O

Daemon Processes

- Using Error Logging
- Using Single-Instance Daemons and Daemon Conventions

Advanced I/O

- Using Record Locking
- Using I/O Multiplexing
- Using Nonblocking and Asynchronous I/O
- Performing Buffered and Vectorized I/O Operations
- Performing Memory-Mapped I/O

Interprocess Communication

- Using Coprocesses
- Using Named Pipes
- Using Interprocess Communication
- Using Semaphores
- Using POSIX Semaphores

Network IPC: Sockets

- Managing Socket Descriptors
- Performing Network Addressing and Byte Order Operations in Sockets
- Performing Data Transfer in Sockets
- Using the OOB Flag

Advanced IPC

- Using File Descriptors in Sockets

Terminal I/O

- Getting and Setting Terminal Attributes
- Using the Baud Rate and Line Control Functions
- Using Canonical Mode and Noncanonical Mode
- Using the termcap, terminfo, and curses Libraries

Pseudo Terminals

- Opening Pseudo-Terminal Devices

A Database Library

- Building a Library

Here's what you get

69	69	02:23
LIVE LABS	VIDEO TUTORIALS	HOURS