Linux System Programming Certification – 5 days

Introduction:

The Linux System Programming Certificate Training is a comprehensive program designed to equip participants with the essential skills and knowledge needed to develop software applications for the Linux operating system. Covering key concepts such as process management, file handling, inter-process communication, and system calls, this training ensures a solid foundation in Linux system programming.

Who should attend the certification:

- Software developers seeking to enhance their skills in Linux system programming
- System administrators interested in understanding the inner workings of the Linux operating system
- Computer science students or graduates looking to specialize in Linux system programming
- Embedded systems developers working with Linux-based platforms
- DevOps engineers involved in deploying and maintaining Linux-based systems
- Anyone interested in gaining a deep understanding of the Linux operating system and its programming capabilities

Why do we need to learn Linux?

- Linux is ubiquitous, powering a wide range of devices and systems.
- It dominates the internet, supercomputers, and stock exchanges.
- Used in embedded systems, mobile phones, and servers.
- Open-source software like Apache, MariaDB, and Python are incubated on Linux.
- Learning Linux enhances understanding of operating systems and software.
- Linux offers improved job prospects in IT industry.
- Essential for system programming and administration.
- Gateway to Linux programming and administration world.
- Linux is widely used by organizations and banks.
- Linux automation with bash scripting streamlines tasks.
- Enhances technical skills and job market appeal.

Certification Structure:

Timmins Linux System Programming certification program includes the following three courses that cover different aspects of Linux knowledge and skills. :

Linux Essentials: This course provides a foundation in Linux fundamentals. It covers topics such as Linux history, basic command-line operations, file management, user and group administration, permissions, and an introduction to networking. Linux Essentials is often considered an entry-level course and certification, suitable for beginners or those new to Linux.

Linux Bash Scripting: This course focuses on shell scripting using the Bash shell, which is the default shell for most Linux distributions. It covers scripting concepts, control structures, variables, functions, input/output operations, and error handling. Bash scripting enables automation and simplifies repetitive tasks, making it a valuable skill for Linux administrators and power users.

Linux System Programming: The Linux System Programming course is a comprehensive training program that provides participants with in-depth knowledge and practical skills to develop software applications for the Linux operating system. This course covers essential topics such as process management, file handling, inter-process communication, and system calls. Participants will learn how to utilize Linux system libraries, understand the inner workings of the kernel, and optimize application performance. Through hands-on exercises and real-world examples, students will gain proficiency in writing efficient and robust code for Linux environments. This course is suitable for software developers, system administrators, computer science students, embedded systems developers, and anyone interested in advancing their understanding of Linux system programming.

Course Flow:

Course 1	Linux Essentials	Duration: 1 day
Course 2	Linux Bash Scripting	Duration: 2 days
Course 3	Linux System Programming	Duration: 2 days

Detailed course content as follows on the next pages.

Linux Essentials

Duration: 8 Hours

Description:

The Linux Essentials course is designed to enhance participants' mastery of Linux command-line navigation, encompassing a comprehensive array of commands and options. It emphasizes providing learners with a deep understanding of Linux file and process abstractions, enabling them to effectively manage file permissions, employ vital Linux filters for data manipulation, and excel in file archiving and network transfers.

Course Objective:

Participants will become to be good in

- o The basics of the Linux commands & options
- o The Linux file & process abstractions
- o Understand the basics of Linux file permission
- o Understand Linux essential filters
- o Understand how to archive files and transfer through network

Prerequisite:

General Operating System Knowledge (optional)

Course Outline

Linux Architecture

Work with files and directories

Use the vi editor to create and modify files

Use commands within the default shell

View and modify file and directory permissions

Manage processes & process manipulation commands

Linux Filters

Archive files and perform remote file transfer

Linux BASH Scripting

Duration: 16.0 Hours

Description

This course includes gaining the ability to customize the Linux environment using shell scripting, making decisions using if statements and tests, controlling input and output, redirecting standard output/error, manipulating text and strings, creating functions, generating reports from log/CSV files, and connecting external applications using Here documents.

Prerequisite:

Linux Essentials Knowledge

Learning Outcome:

- Understand to customize linux environment using shell script.
- Make decisions by using if statements and performing several different kinds of tests validations.
- Control all types of input and output.
- Redirect standard output and standard error.
- Perform text and string manipulation.
- Creating functions and makes loadable script
- Reporting on log files, CSV files, and other format of data.
- Using Here document users can connect external applications.

Course Outline

About shell & execution steps

Variable & types - How to automate runtime inputs

Shell operators

Redirect shell command results to external file

Conditional statements - Test and Validate each system

operations interactively

Looping statements - perform set of system operation repeatedly

File Handling - Extend script to storage

Function Call with arguments - To make loadable script

Regular Expression

Search and Substitute operation from pipe results and input files

Sed script – line and pattern processing

awk script - search and format data for processing

Here document & trap commands

Use Cases

1. Create Customized daemon scripts - To perform run level scripts
2. Automate user management activities
3. Write customized configuration scripts more.

Linux System Programming

Description:
The Linux system calls and implementation are a focal point of the course, enabling participants to gain a deeper understanding of the underlying mechanisms and functionality of the Linux operating system

Duration: 1*6.0* Hours

Prerequisite:
- Linux Essentials & Script knowledge
- C with Data structures

Learning Outcome
Participants will become to be good in
- Illustrate file processing operations such as standard I/O and formatted I/O.
- Illustrate memory management of file handling through file/region lock.
- Design and Implement in C some standard linux utilities.
- Understand process structure, scheduling and management through system calls.
- Implement C programs to control process using system calls and indentify difference between process and threads.
- Generalize signal functions to handle interrupts by using system calls.
- Design and implement inter process communication (IPC) in client server environment by using pipe and named pipes system calls.
- Design and implement inter process communication (IPC) in client server environment by using message queues system calls.
- Illustrate client server authenticated communication in IPC through shared memory.
- Familiarity with Inter Process Communication using Semaphores.
- Demonstrate various client server applications on network using TCP or UDP protocols.

Course outline

GNU/Linux Library Commands and gcov tool

Illustrate Problems with gcov and Optimization

File Handling in GNU/Linux

Logical view of filesystem and system calls used

Process management and kernel structures

Linux signals and Handler

System V IPC concepts and use cases

Linux memory management

POSIX Threads – creation and synchronization