multisoft
Systems

Linux Kernel and Device Driver Development Training

COURSE CONTENT

GET IN TOUCH

Multisoft Systems
B - 125, Sector - 2, Noida

(+91) 9810-306-956

info@multisoftsystems.com

www.multisoftsystems.com

About Multisoft

Train yourself with the best and develop valuable in-demand skills with Multisoft Systems. A leading certification training provider, Multisoft collaborates with top technologies to bring world-class one-on-one and certification trainings. With the goal to empower professionals and business across the globe, we offer more than 1500 training courses, which are delivered by Multisoft's global subject matter experts. We offer tailored corporate training; project Based Training, comprehensive learning solution with lifetime e-learning access, after training support and globally recognized training certificates.

About Course

The Linux Kernel and Device Driver Development training offered by Multisoft Systems is an immersive program designed to equip professionals and enthusiasts with in-depth knowledge and practical skills in Linux internals, kernel architecture, and device driver programming.

Module 1: Process Management (kernel side)

- ✓ The Process
- ✓ Process Descriptor and the Task Structure
- ✓ Allocating the Process Descriptor
- ✓ Storing the Process Descriptor
- ✓ Process State
- ✓ Manipulating the Current Process State
- ✓ Process Context
- ✓ Copy-on-Write
- ✓ Forking
- ✓ vfork()
- ✓ Kernel Threads

Module 2: Process Scheduling

- ✓ Multitasking
- ✓ Linux's Process Scheduler Policy
  - o I/O-Bound Versus Processor-Bound Processes
  - o Process Priority
  - o Time slice
  - o The Scheduling Policy in Action
- ✓ The Linux Scheduling Algorithm
  - o Scheduler Classes
  - o Process Scheduling in Unix Systems
  - o Fair Scheduling
- ✓ The Scheduler Entry Point

- ✓ Wait Queues
- ✓ Waking Up
- ✓ Preemption and Context Switching

Module 3: System Calls

- ✓ System Calls
- ✓ Examples of system calls
- ✓ Examples of Standard APIs
- ✓ System call Implementation
- ✓ API-system Call-OS relationship
- ✓ Types of system calls
- ✓ Processor Affinity System Calls
- ✓ Yielding Processor Time
- ✓ Communicating with the Kernel
- ✓ APIs, POSIX, and the C Library
- ✓ Accessing the System Call from User-Space

Module 4: Memory Management

- ✓ Pages
- ✓ Zones
- ✓ Getting Pages
- ✓ Getting Zeroed Pages
- ✓ Freeing Pages
- ✓ kmalloc()
- ✓ gfp_mask Flags
- ✓ Action Modifiers
- ✓ Zone Modifiers
- ✓ Type Flags
- ✓ kfree()
- ✓ vmalloc()

- ✓ Slab Layer
- ✓ Design of the Slab Layer
- ✓ Slab Allocator Interface
- ✓ Allocating from the Cache
- ✓ The Process Address Space

Module 5: The Virtual File system

- ✓ Common File system Interface
- ✓ File system Abstraction Layer
- ✓ Unix File systems
- ✓ VFS Objects and Their Data Structures
- ✓ The Superblock Object
- ✓ Superblock Operations
- ✓ The Inode Object
- ✓ Inode Operations
- ✓ The Dentry Object
- ✓ The File Object
- ✓ File Operations
- ✓ Data Structures Associated with File systems
- ✓ Data Structures Associated with a Process

Module 6: Interrupts and Interrupt Handlers

- ✓ Top Halves verses Bottom Halves
- ✓ Registering an interrupt handler
- ✓ Interrupt context
- ✓ Taslets
- ✓ Softirq

Module 7: An Introduction to Kernel Synchronization

- ✓ Kernel Synchronization Methods

- ✓ Spin locks
- ✓ Reader-writer locks
- ✓ Semaphores

Module 8: Timers and Time Management

- ✓ Jiffies
- ✓ Hardware clocks and timers

Module 9: LINUX DEVICE DRIVER

- ✓ Introduction
- ✓ Role of Device Driver
- ✓ types of Device driver
- ✓ loadable modules and its benefits
- ✓ Functions used to load and unload modules
- ✓ Passing parameters to a loadable module

Module 10: Writing a device Driver Program

- ✓ Important header files
- ✓ Writing a simple module
- ✓ Compiling and loading modules
- ✓ Device information in /proc
- ✓ Character driver
- ✓ character driver basics
- ✓ major and minor numbers
- ✓ creating device files with mknod
- ✓ registering a character device driver
- ✓ Hand- on Practice
- ✓ Character Device Driver writing
- ✓ Userspace interaction
- ✓ Proc/sys model

- ✓ Lcd implementation
- ✓ Understanding the serial device driver

Module 11: RTOS

- ✓ Introduction to RTOS
- ✓ What is Real Time System?
- ✓ Requirements of Real time System
- ✓ Hard Real-time Systems and Soft Real-time Systems

Module 12: Task

- ✓ What is a task creation?
- ✓ Unitask approach Vs multitask approach
- ✓ Task states

Module 13: Scheduling

- ✓ Multitasking Kernel
- ✓ Context switch
- ✓ Priority based scheduling
- ✓ Round Robin scheduling

Module 14: Task functions

- ✓ Task states
- ✓ Task hooks
- ✓ Task synchronization