

Linux Device Driver Programming

Course Overview:

This course will cover Linux kernel development, Linux kernel module development, role of Virtual file system and mainly Linux char, block and network device driver programming.

Duration : 3 days

Dates : As per mutual agreement

Time : 6 Hrs/Day

Eligibility : BE/BTech/ME/MTech (CSE / E&TC / Electronics) students

Corporate : Training with enhanced labs and customization for working professionals.

Skills developed:

1. Good experience on Linux kernel development
2. Good understanding about Key principles of Linux OS
3. Expertise on Linux device driver programming

Results : At the end of training session student will get good expertise on device driver and kernel level programming

Pre-requisite: Basic knowledge Of Linux

NOTE:

Related study material, work book and lab solutions will be provide. Reference and guidelines will be provided for Linux device driver and kernel space programming.

DAY-1

1. Linux OS Architecture

- Linux Features
- Linux Kernel Source Directory Structure
- Linux Kernel Components
- User Mode Vs Kernel Mode
- System Initialization –Linux Booting Process

LAB1: Exploring on kernel directory structure and files responsible for booting

2. Kernel compilation

- Understand kernel configuration and build process
- Importance of Makefiles
- Procedure to reconfigure and recompile the kernel

LAB2: Rebuild kernel for new kernel

3. Detail study of Linux OS components

- Process Management
- Thread Management
- Interrupt Management and Signal handling

LAB3: How To create and terminate processes?

LAB4: How to control Multithreading?

LAB5: How to handle signals?

DAY-2

4. Kernel Module programming

- Introduction to Modules
- Writing Your first kernel module
- Module Related Commands
- Statically linked vs Dynamically linked
- Exporting symbols from modules
- The kernel symbol table
- Concurrency in the kernel
- Module Parameters
- Version dependency

LAB6: How to write kernel module and install it into the kernel?

LAB7: How to initialize and use kernel variables?

LAB8: How to achieve synchronization among kernel modules?

LAB9: How to control kernel parameters and peripherals, ports through kernel module programming?

5. Introduction To Device driver

- Devices and Device Driver (DD)
- Types Of DD
- Introduction to Virtual File System (VFS)

LAB10: Explore on /dev directory

6. Character Device Drivers

- Registering a character device driver
- File operations and ioctls to control char type of devices
- Reading and writing into char devices

LAB11: How to write char DD?

LAB12: How to register DD and map file operations to VFS?

LAB14: Demonstrate accessing device driver from application program.

DAY-3

7. Block Device Drivers

- Understand how block devices works
- Block DD registrations, File operations and ioctls
- Handling and controlling requests
- File system (FS) introduction and types
- Mounting and formatting FS for virtual block device

LAB15: Write RAM type of disk driver

9. Network Device Drivers

- Understand how network devices work
 - The net_device structure in detail
 - Packet transmission and reception
- LAB16:** Simulating a network device

10. Debugging tools and techniques

• Introduction to Application level debugging

• Introduction to kernel level debugging technique

LAB17: Demonstrate gdb for application program debugging

LAB18: Demonstrate kdb/kgdb for kernel space program and kernel level debugging.