



## Linux Device Driver Programming Workshop (4 days)

This course consists of four full days of training on Linux Device Driver Programming which covers the various aspects of Device Driver programming in Linux OS. The course starts with a session on Device driver module and related kernel programming, also cover Memory management, Character Device Driver, Block Device Driver, USB, Serial, Network Device Driver Programming.

**Prerequisites:** Linux Kernel Internal Programming Workshop is Prerequisite to this course. However those who have strong Linux Internal Knowledge will be able to start this course without the prerequisite course. Considerable knowledge of Linux System Programming concepts like signals, processes, file operations, system calls and sockets or Linux Internals & Programming are Essentials Course. *We will try to take DAY 1 of the workshop for recollecting the Linux Systems introduction.*

### Target Audience

This course is designed for C programmers who are familiar with Linux System Programming.

**Note:** The program concentrates more on hands on. The lab exercises contain developing and testing of respective programs on X86 /PC hardware introduced on day one of the program.

### Day 1

#### Introduction to Linux

- o History and open source
- o Features
- o Modular and monolithic vs micro
- o Kernel design goals

#### Understanding the kernel

- o Kernel structure
- o Kernel components and organization
- o Modes of operation
- o User library different from kernel

#### Booting and kernel initialization

- o Booting Basics
- o System Ups and Down under Linux
- o Boot-loaders (Various target specific boot loaders)
- o Understanding Linux start\_kernel in brief

#### Linux Basic Common Set

- o File and directory related command
- o Using VI effectively [contrast w.r.t Word processor]
- o Commands for filters, IO redirection and etc
- o Process related commands
- o Miscellaneous Command [zip/tar/mount etc]
- o Shell scripts

#### Simple handy utilities

#### Major script techniques

- o Compiling C codes
- System calls
- o Understanding the Layer for Communication
- o Basics

- o Common system calls
- o How is it implemented in Linux?
- o Using a system call directly in assembly
- o Linux implemented methods

#### **DAY 2:**

- \* An introduction to device drivers
  - o Role of the Device Drivers
  - o Splitting the kernel
  - o Classes of devices and modules
  - o Kernel Architecture or Model
  - o Lab exercises
  
- \* Module Concepts
  - o The kernel symbol table
  - o Module Initialization and shutdown
  - o Using memory, I/O Memory and I/O ports
  - o Module parameters and description
  - o Doing it in user-space
  - o Lab exercises
  
- \* OS Services And Data Structures
  - o Atomic functions, Bit Operations
  - o Linked List, Memory barriers
  - o Semaphores, Time intervals in the kernel
    - o Delaying execution, Task queues
  - o Tasklets, Kernel timers
  - o Basic memory Management
  - o The proc file system
  - o Lab exercises

#### **DAY 3:**

- \* Character Device Drivers
  - o Registering a character device driver
  - o File operations
  - o The file structure
  - o devfs / lseek / ioctl
  - o Lab exercises
  
- o Blocking, non blocking and asynchronous operations
  - o Lab exercises
  
- \* Block Device Drivers
  - o Registering a block device driver
  - o File operations
  - o The file structure
  - o devfs / lseek / ioctl
  - o Lab exercises

#### **DAY 4**

- \* Hardware and Interrupt Handling
  - o Top halves vs bottom halves

- o registering a interrupt handler
- o freeing a interrupt handler
  
- o Using IO Ports
  - + Installing and implementing an interrupt handler
    - o why bottom halves?
- o Tasklets and Bottom halves
- o softirq's
  - o Lab exercises
  
- \* Block Device Drivers
  - o Handling requests
    - o buffers and buffer head
  - o Ram Disk Driver
    - o Lab exercises
  
- \* Network Drivers
  - o The net\_device structure in detail
  - o Packet transmission
  - o Packet reception
  - o The Interrupt handler
  - o Simulating a network device
    - o Lab exercises
  
- \* Serial Interface Device Drivers
  - o Working of serial devices
  - o system calls related to serial device drivers

*(Outline is subject to minor tuning and changes before launch of the program)*

\* \* \* \* \*