

Kernel & Device driver Programming

- **MODULE 1: Operating System Concepts and Programming in Linux**

This module aims to introduce the Operating system concepts and focuses on programming in User Space. This module provides strong foundation to student and emphasises programming in Linux OS. Main Topics are :

- Creating Processes, Signals
- Threads and multithreading programs
- Semaphores, Mutex
- IPC mechanism Pipes, Shared memory etc.
- Sockets and client server programming
- Debugging with GDB
- Other tools like strace/ltrace etc

- **Module2 :Kernel Module Programming**

This module introduces the Operating System Kernel programming. It gives basic aspects of working in Kernel and makes comfortable in kernel programming. All the topics in this module are Completely practical oriented. They are :

- Compiling kernel
 - Configuring Kernel and compilation
 - Bootloader and boot process and booting kernel
 - Kernel code browsers
- Static linking, dynamic linking of modules
- User space, kernel space concepts
- System calls
- Writing simple modules
- Writing Makefiles for modules

- **Module3: Character Device driver Development**

This module aims to provide understanding of device structures and programming aspects for devices. It provides practicals on various drivers modules. Some of the topics are:

- Driver concepts
 - Block & character driver distinction
 - Low level drivers, OS drivers etc
- Writing character drivers
 - Device major, minor number
 - Interfaces to driver read, write, ioctl etc
 - Blocking and non blocking calls
- Synchronisation
 - Semaphores, mutexes, spinlocks
- Proc & Sysfs interfaces
- Interrupt Handling
 - Interrupts and bottom halves

- Writing interrupt driven drivers
- Implementing bottom halves
- Kernel Threads & Work Queues
- Timers
 - Kernel timers
 - Jiffies , Timer interrupts
- Kernel linked list implementation
- Interfacing with hardware
 - IOMapped IO
 - Memory mapped IO
 - Understanding DMA ops
- **Module 4: Kernel Debugging techniques**
 - Interpreting kernel OOPS
 - KDB
 - KGDB
 - Remote debugging using KGDB
- **Module 5:Linux Kernel Internals (OPTIONAL)**

This module gives indepth understanding of OS kernel. After completion of this module students have gained understanding of Operating system kernel and overall working.

 - Process and process control blocks.
 - Ready & Wait queue
 - Process states
 - Light Weight Processes/Kernel threads
 - Scheduling
 - Linux scheduling techniques
 - Memory Management
 - Virtual memory – Paging and Segmentation
 - Buddy algorithm , Slab allocators
 - Memory allocation in kernel
 - VFS –Virtual File System
 - Driver call mappings
 - File system working and operations
- **Module 6: Writing Block drivers(OPTIONAL)**

This module gives indepth understanding of block devices and its hirerachy in Linux. Module covers implementation of RAM disk driver. Following are topics covered in this module

 - Block devices design and addressing
 - Concepts of mounting and partitioning
 - Buffer cache , page cache
 - Overview of SCSI
 - Block driver development

- **Module 7: Writing USB device driver (OPTIONAL)**

This module covers the USB device interfacing , it covers implementation of USB driver for custom USB device

- USB device classes
- USB topology and interfaces
- USB protocol working
- Implementing driver in kernel

Prerequisite : C & DS

Note : Course is completely Modular