Linux Kernel Internals and Advanced Programming Training Workshop

Kernel Internals

- **Introduction to Kernel**
  - History of Linux
  - Types of Kernel
  - The Linux kernel
  - Kernel Facilities
  - Kernel Architecture
  - Kernel Version Numbers
  - Configure, Compile Kernel

- **The File System**
  - Virtual File system & its role
  - Files associated with a process
  - System Calls

- **Process management**
  - Process Defined
  - Process Descriptor Structures in the kernel
  - Process States
  - Process Scheduling
  - Process Creation

- System calls related to process management

- **Memory Management**
  - Defining and Creating secondary memory areas
  - Memory allocation & deallocation system calls malloc, calloc, alloca, free
  - Demand Paging defined
  - Process Organization in Memory
  - Virtual Memory Management
  - Address Translation and page fault handling
  - Buddy System Algorithm
  - Slab Allocator
  - Swapping Memory Areas
  - Memory Mapping
  - Zones and Pages
  - Address Space
  - Block Device Caching
  - High Memory
  - Paging

- **Interrupts**
  - About Interrupts
  - Interrupt Handlers
  - Softirqs
  - Tasklets
  - Work Queues

- **Time and Timers**

HZ and Jiffies

Time of Day

Delayed Execution

Kernel Timers

- **Device Drivers**

  Character Device Drivers

  Block Device Driers

  Network Device Drivers

- **Kernel Configuration and Compilation**
  - Kernel Building System
  - Patching the Kernel
  - Kernel Configuration
  - Compiling the Linux Kernel

- **Kernel Parameters**
  - Build-Time Parameters
  - Boot-Time Parameters
  - Run-Time Parameters
  - System Tuning
  -
- **Kernel Synchronization**
  - UP vs SMP Issues
  - Atomic Operations
  - Semaphores
  - Spin Locks

- **Boot Process - understanding Power ON to login prompt process**
  - BIOS Level
  - Boot Loader
    - Setup, startup_32 functions
    - Available Bootloaders – GRUB, UBoot, etc.
    - Role of a Bootloader

- Bootloader Phase
- Kernel Initialisation Phase
  - The start_kernel() function
- System Initialisation Phase
  - Understanding Runlevels
  - Various initialization scripts & customizing them
  - Kernel Command-Line Boot Parameters

Linux Advanced Programming

- **Arguments , the environment, and popular system functions**
  - Arguments to a program.
  - Retrieving information from the environment.
  - How to use getopt() effectively.
  - Discovering the date and time
  - Getting system information.

- **File I/O**
  - Library functions and system calls for I/O
  - Getting file status information with stat()
  - Processing directories and directory entries.

- **Shell Commands & Shell Scripting**
  - Basic Shell commands
  - Bash Shell Essentials

- **Creating Makefiles**
  - Makefile basics
  - Creating make files for single or multiple source files project

- **Creating Lilbraries**
  - Creating Static Library
  - Creating Shared Library

- **Multi Process Programming**
  - Creating child processes
  - fork(). vfork(), exec()
  - Parent synchronization with child

- **Multi Thread Programming**
  - Creating multiple threads
  - Parent synchronization with other Threads

- **Inter Process Communication**
  - Pipes, FIFO's,
  - Signals
  - System-V IPC's
  - Message queues
  - Shared memory
  - Semaphores

- **Introduction to Sockets**
  - An Overview
  - System calls related to TCP and UDP sockets
  - Using Wireshark for network sniffing

- **Network Programming**
  - TCP Server Client Programming
  - UDP Server Client Programming
  - Lab exercises

- **Programming & Debugging Tools**
  - Debugging and Analysis Tools
  - strace : Tracing System calls
  - ltrace : Tracing Library calls
  - Using gdb and ddd utilities
  - Core Dump Analysis etc

- **Applications Development and Debugging**
  Application Development

  Source Code Management

- **Source Code Version Control**
  - Understanding need of Version Control System
  - cvs
  - svn
  - git