

Day - 1

- **Session 1: Linux Application Space Overview**
 - Linux System Overview
 - Types of Kernel Architectures
 - Linux System components
- **Session 2: System Calls in Linux**
 - Need for System Calls
 - System Calls & Library Functions
 - Using strace, ltrace
 - Example programs on System calls
- **Exercises**
 - System call tracing
 - Locking a file & file regions
 - Getting the time & formatting the same
- **Session 3: Linux Processes**
 - Need for Process
 - Creating & Exec'ing the process
 - Waiting for the process termination
 - Zombie & Orphan process
 - Daemonizing the process

- **Exercises**

- Creating a Process
- Creating multiple processes
- Distinguishing between the child & parent processes
- Client configuration
- ntpq command output interpretation
- ntpq command output interpretation continued
- Times Zones
- Using chrony

- **Session 4: Managing the signals**

- Need for Signals
- Overview of Signals in Linux
- Handling the signal
- Blocking/Masking the signals with signal sets

- **Exercises**

- Registering a signal handler
- Restoring the signal disposition
- Using SIGCHLD signal to notify the parent process
- Examine the signal mask for the proces
- Blocking the signal

- **Session 5: Inter-Process Communication Mechanisms**

- Need for IPCs
- Using Pipes & FIFOs
- Message Queues
- Shared Memory & Semaphores

- **Exercises**

- Pipes to communicate amount the related processes
- Using popen function for the pipes
- Pipe Communication among unrelated processes
- Using Message Queues
- Communication using Shared Memory
- Synchronizing the access to Shared Memory

- **Session 6: Threads**

- Need for Threads
- Process Vs Thread
- Creating thread with pthread APIs
- Waiting for thread termination
- Creating a detached thread
- Cancelling the threads
- Clean-up handlers

- **Exercises**

- Creating a thread
- Passing arguments to the thread
- Waiting for threads & examining the return value
- Modifying the thread attributes to create the detached thread
- Cancelling the threads
- Using the clean-up handler

- **Session 7: Thread Synchronization**

- Need for Synchronization
- Using Mutex
- Using Semaphores
- Reader / Writer Lock
- Conditional Variables

- **Exercises**

- Example for mutex usage
- Example for semaphore usage
- Case study on Synchronizing threads
- Example on reader/writer Lock
- Using the Conditional variables

- **Session 8: Socket Programming**

- Introduction to Sockets
- Socket APIs
- Client & Server
- Connectionless & Connection oriented Sockets
- Creating UDP/TCP server/Client

- **Exercises**

- Creating local Client & Server
- Creating networked Client & Server