Embitude

Let's have fun with Embedded

Project Oriented Online Training on Linux System Programming
By Pradeep Tewani

Course Objective
This Linux user space internals course provides a deep insight into the Linux operating system concepts. The various user space programming concepts such as process, threads, synchronization, Interprocess communication mechanisms are brought to sharp focus. The usage of programming concepts enables the candidates to design & develop the systems requiring the multitasking capabilities to efficiently manage the system resources.

Course Objective
The Linux System Programming course attempts to serve multiple objectives:

- To enable participants to effectively apply the OS cocepts to design the multiprocessing & multi-threading systems requiring sycnhronization.
- To enable participants to design the systems requiring the communication across the processes or across the system, thereby sharing the system resources as required.

Target group:
Professionals/Students looking to design & develop the Linux based advance application under the mult-threaded & multi-processing environment.

Pre-requisite
Knowledge of C Programming with comfortability in Linux environment

Methology
Every theoretical topic is accompanied by corresponding hands-on/assignment to get the deep understanding of the topic.

Assessment
Assignment Based

Session 1: Introduction to Linux User Space & System Calls
- Linux System Components
- Need for System Calls
- System Calls and Library Functions

Exercises/Assignments
- System call tracing
- Locking a file and file regions

Session 2: Processes in Linux
- Process Overview
- Process Creation & operations

- Waiting for the Process termination
- Zombie Processes

Exercises/Assignments

- Creating a processes
- Exec'ing a process
- Waiting for the child process to terminate
- Creating zombie & orphan process

+ ***Session 3: Signals***

- W's of Signals
- Types of Signals
- Signal Examples

Exercises/Assignments

- Registering a signal handler
- Masking a Signal in the Handler
- Blocking a Signal

+ Session 4 & 5: Inter Process Communication

- IPC Overview
- Pipe & Fifo
- Shared Memory
- Process Semaphores

Exercises/Assignments

- Write a program to demonstrate each of the above IPC mechanism

+ Session 6: Threads in Linux

- W's of thread
- POSIX Threads & their Internals
- Threads Creation, Operations & Usages
- Thread Joining
- Thread Cancellation

Exercises/Assignments

- Write a program to create the thread
- Write a program to demonstrate the usage of pthread_join
- Write a program to cancel the thread

+ Session 7: Synchronization in Linux

- Synchronization Overview
- Synchronization Mechanisms

Exercises/Assignments

- Write a program to solve the consumer/producer problem

+ Session 8: Linux Network Management

- Network Management Overview
- Introduction to Sockets
- Basic Socket Programming

Exercises/Assignments

- Write a programs to communicate between client & server