INDIA's one & only
INDUSTRIAL Embedded Systems Training Institute
KERNEL MASTERS

Online Training on

Linux System Programming

By Kishore Kumar Boddu

Session Highlights:

- Participants will develop a deep understanding the Linux System Programming with Real Time Examples.
- Online Sessions will be an assignment driven model so that participants can have a deep understanding of system programming well as kernel mode programming practices.
- Adds the following skill set to your profile**: Linux System Programming, Socket Programming, Threads, Concurrency, Synchronization Mechanisms.**

Prerequisites (Part of the course - complementary):

- Operating Systems Concepts covers in course.
- We assume that attendees are fully fluent in C, data structures
- Should be familiar with Linux/Unix command line

Audience:

- This session is mainly intended for those looking to start their career in Linux system programming or application programming or for those already working in Linux platform.

Syllabus Summary: (Detailed agenda in Next page)

1. Introduction to Linux Architecture
2. GNU Toolchain
3. Operating Systems Concepts
4. File operations
5. Linux Process Implementation
6. Signals
7. POSIX Threads
8. Synchronization Mechanisms
9. IPC Mechanisms
10. Socket Programming

Authored and Compiled By: Kishore Kumar Boddu
Email: kishore@kernelmasters.org
Reach us online: www.kernelmasters.org
Contact: 9949062828

Note: All Sessions are highly interactive hands-on-sessions.

Linux System Programming Detailed Agenda

Session 1: Introduction to Linux Architecture

Objective: To understand Linux Architecture and types of Linux Programming.

Topics to be covered :

1.1. Types of Kernel
1.2. Types of Programming
1.3. Linux System Programming vs Linux Application Programming
1.4. Linux Kernel Programming vs Linux Device Driver Programming

Session 2: GNU Toolchain

Objective: To understand how to setup Linux development environment.

Topics to be covered :

2.1. GCC
2.2. GDB
2.3. GNU Makefile
2.4. GNU Binutils
2.5. GNU Build system
2.6. Static Library vs Dynamic Library

Session 3: Operating Systems Concepts

Objective: To understand OS concepts

Topics to be covered :

3.1. Types of Kernel
3.2. OS Operations
3.3. Process Management
3.4. IPC & Synchronization

Session 4: File operations

Objective: In Linux, Everything is a File. Here understood all file operations.

Topics to be covered :

4.1. File system layout
4.2. Super block & Inode
4.3. System Call vs Standard Library
4.4. System call debugging using strace
4.5. Device File operations
4.6. Advanced File operations

Hands-On-Session:

Implementation of evtest application.

Session 5: Linux Process Implementation

Objective: To understand Linux Process management how to create a process and terminate a process.

Topics to be covered :

5.1. Using system() to Create a Process
5.2. Using fork() to Create a Process
5.3. Using exec() to Create a Process
5.4. Synchronization System calls (__exit() & wait())
5.5. Daemon Processes
5.6. Zombie process vs Orphan Process.
5.7. Using clone()

Hands-On-Session:

Implementation of system() library function.

Linux System Programming Detailed Agenda

Session 6: Signals

Objective: A **signal** is an event generated by the UNIX and Linux systems in response to some condition. Upon receipt of a signal, a process may take action.

Topics to be covered:

6.1. What are Signals
6.2. Signals Available
6.3. How to Raise a Signal & Dispatching Signals
6.4. Alarm, Pushing and Sleeping
6.5. Setting up a Signal Handler
6.6. Signal Sets & Sigaction()

Hands-On-Session:

Various example programs on Signal raising and catching.

Session 7: POSIX Threads

Objective: To learn, how to create multi-threaded tion

Skillset: Multi-Threaded Programming.

Topics to be covered :

7.1. Process Vs Thread Vs Task
7.2. What is POSIX Threads?
7.3. How to create a thread and join a thread.
7.4. Thread and Management.
7.5. Signals vs Threads

Hands-On-Session:

- Multithreaded application
- Producer consumer problem using POSIX Threads and Signals.

Session 8: Synchronization Mechanisms

Objective: To learn, how to use semaphore and mutex to handle concurrency and what are the synchronization mechanisms.

Topics to be covered :

8.1. Producer Consumer Problem
8.2. Critical Section, Race around condition
8.3. Semaphore vs Mutex

Hands-On-Session:

- Producer consumer problem using POSIX Threads and Semaphores and Mutex.

Session 9: IPC Mechanisms

Objective: To learn, IPC mechanisms and practical usage in real-time scenarios.

Topics to be covered :

9.1. IPC Methods
9.2. Pipes vs FIFO
9.3. System V Message Queues
9.4. System V Shared Memory
9.5. Advantages and disadvantages of IPC Mechanisms

Hands-On-Session:

- Create a unnamed pipe across fork() system call.
- Client server program using named pipe.
- Producer consumer problem using Message Queues and Shared Memory.

Linux System Programming Detailed Agenda

Session 10: Socket Programming

Objective: Socket programming is a way of connecting two nodes on a network to communicate with each other.

Topics to be covered :

10.1. What is Socket?
10.2. Socket Types
10.3. OSI Layer
10.4. Server and client system calls
10.5. Client server program using TCP and UDP
10.6. Netlink sockets

Hands-On-Session: Client server program using local and network