LX3

Advanced Linux System Programming

Advanced Linux System Programming

LX3 - Version: 2

🕐 4 days Course

Description:

The course introduces the participant to programming in the C programming language in a Linux environment. The course focuses on Linux system calls and library functions, how to use them, and their underlying mechanisms. The course deals with many facets of the Linux operating system, including: Linux kernel structure, I/O, Signals, Timers, Processes, Threads, IPC mechanisms: Pipes, Shared memory, Message Queues, Semaphores, and Sockets using TCP and UDP. The course includes many programming examples that can serve as a reference or basis for the participants' work in the future. Time is given to writing and testing several programs to help the participants comprehend the complexity of Linux system programming, along with the satisfaction of knowing how to do it well.

Intended audience:

The course is for programmers, software designers, and project managers.

Prerequisites:

A working knowledge of the C programming language.
Basic knowledge of the Linux/Unix operating system on a user level is mandatory.

Objectives:

The participant will know about the various mechanisms available to the programmer in a Linux environment. This will allow the programmer to write a wide variety of applications using standard Linux system calls and library functions.

Topics:

Introduction.

- Short History
- Why is Linux So Successful?
- UNIX Flavors - BSD, SysV, Linux
- Standards
- System Architecture
- The Kernel
- The Shell
- Utilities - Tools and Applications

Programming under Linux

- Privileged Execution Mode
- Kernel Mode Vs. User Mode
- System Calls
- Using System Calls
- What Happen When a System Call Fails?
- Printing Error Messages - perror
- Printing Error Messages - strerror

Files

- Using Files
    º File Descriptors
    º read/write
- Links

Timers

Process

Signals

Threads

- Thread-Specific Data
- Synchronization and Critical Sections
- GNU/Linux Thread Implementation
- Processes Vs.Threads

Resource Usage

- Resource Management.
- Measuring Resource Utilization.
- Upper Levels of Resource Management.
- Context Switch.
- Algorithms for Process Management.
- Process Management in UNIX.
- Priorities.

Inter-Process Communication (IPC)

- Pipes
  - º Create pipe
  - º The Shell and Pipes
  - º Pipes and Standard I/O Files
  - º Closing Pipes
  - º Popen
  - º pclose
- Named Pipes
- Shared Memory
  - º shmget
  - º shmat
  - º shmdt
  - º shmid_ds

- º shmctl
- Message Queue
    - º msgget
    - º msgsnd
    - º msgrcv
    - º msqid_ds
    - º msgctl
- csh Level Commands
    - º ipcs
    - º ipcrm

Synchronization Mechanisms

- File locking
- Semaphore

Sockets

- What Is A Socket?
- A Brief History Of Sockets.
- Communication Protocols.
- Communication Capabilities.
- Endpoint Addresses.
- The Internet Protocol (IP).
- Internet Addresses.
- Address Classes.
- Connection Oriented Communications And TCP
- Connectionless Communications And UDP.
- Stream Sockets - The Basic Model
    - º Server side

º Client side

- Sending Data – send.
- Receiving Data - recv.
- Shutting Down A Socket.
- Related files.
- The select( ) System Call.
- Broadcast and Datagram Sockets