Linux Internals

Duration: 5 days

Course Description: This course focuses on the elements of the Linux kernel that allow programmers to build software components that are linked to the hardware level. Students gain a general understanding of basic tools and interfaces in order to successfully modify features and develop new aspects of the kernel.

Audience: Programmers, software designers, and technical managers who plan to use Linux below the application level.

Prerequisites: Advanced UNIX Programming is recommended. Strong C programming skills and intermediate knowledge of UNIX/Linux shell commands are required. UNIX/Linux application development experience is recommended.

Outline

INTRODUCTION TO LINUX INTERNALS

- Introduction and environmental setup
- Main characteristics of the Linux operating system
- Linux distributions
- Kernel configuration facilities
- Building the kernel
- Location of components
- Compiling

KERNEL OVERVIEW

- Important data structures
- Processes and tasks
- Files and inodes
- Dynamic memory management
- Queues and semaphores
- System time and timers
- Primary algorithms
- Signals
- Interrupts
- System booting
- Timer interrupt
- Scheduler
- System call implementation
- Description of system calls
- Practical examples
- Adding new system calls

MEMORY MANAGEMENT

- Architecture-dependent memory model Pages
- Virtual address space
- Linear address conversion
- Page table and page directory
- Middle page directory
- Virtual address space model
- User segment
- Virtual memory
- The brk system call
- Mapping functions
- Kernel segment
- Static and dynamic memory allocation in the kernel
- Block device caching
- Block buffering
- Update and bdflush
- List structures for the buffer cache
- How to use the buffer cache
- Paging in Linux
- Page cache management
- Finding free pages
- Page exceptions

INTER-PROCESS COMMUNICATION

- Synchronization

Linux Internals

- Communication via files
- Pipes
- Debugging using ptrace
- System V IPC
- Socket-based communications

FILE SYSTEM

- Basic aspects
- VFS
- Mouting a file system
- Superblock
- The Inode concept and operations
- File operations
- Directory cache
- Proc filesystem
- Ext2 filesystem
- Structure
- Directories
- Block allocation
- Extensions

SYSTEM CALLS

- Initialization
- Process management
- Memory management
- Communication
- Filesystem

KERNEL-RELATED COMMANDS

- ps
- top
- free
- init
- shutdown
- strace
- traceroute
- mount

DEVICE DRIVERS

- Character vs block devices
- Polling and interrupts
- Polling and interrupt mode
- Interrupt sharing
- Bottom halves
- Task queues
- Implementation
- Setup
- Init
- Open and release
- Read and write
- Ioctl
- Select
- Lseek
- Mmap
- Readdir
- Fsync and fasync

MODULE MANAGEMENT

- Interfaces to modules
- Adding/removing modules to the Kernel
- Insmod
- Modprobe
- Rmmod
- Implementation details

NETWORK

- Layer model
- Network communications
- Data structures
- Socket
- sk_buff
- Inet socket
- proto
- Devices
- Ethernet
- SLIP and PPP
- Loopback
- Dummy device
- Protocols
- Arp
- IP
- Functions
- Routing

Linux Internals

- Multicasting
- Packet filters
- Accounting
- Firewalls
- UDP
- Standard and extra functions
- TCP
- Standard functions
- Communication details

SCSI SUBSYSTEM

- Architecture overview
- Names and conventions
- Upper level
- Block devices (hard disks, CD-ROM)
- Character devices (tape)
- Generic drivers
- Mid level (boot parameters, proc interface)
- Lower (hardware) level and pseudo drivers

BOOT PROCESS

- Booting details
- LILO
- Started by MBR
- Started by a boot manager
- Structure in the MBR
- Files
- Parameters
- Start-up messages
- Error messages

DEBUGGING TOOLS

- ptrace
- SysRq
- KDB
- User Mode Linux
- kgdb