

LINUX INTERNALS

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.

Course Length: 5 Days

Course Tuition: \$1250 (US)

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.

Course 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
- 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

System booting	Interfaces to modules
Timer interrupt	Adding/removing modules to the Kernel
Scheduler	Insmod
System call implementation	Modprobe
Description of system calls	Rmmod
Practical examples	Implementation details
Adding new system calls	• Network
• Memory Management	Layer model
Architecture-dependent memory model	Network communications
Pages	Data structures
Virtual address space	Socket
Linear address conversion	sk_buff
Page table and page directory	Inet socket
Middle page directory	proto
Virtual address space model	Devices
User segment	Ethernet
Virtual memory	SLIP and PPP
The brk system call	Loopback
Mapping functions	Dummy device
Kernel segment	Protocols
Static and dynamic memory allocation in the kernel	Arp
Block device caching	IP
Block buffering	Functions
Update and bdflush	Routing
List structures for the buffer cache	Multicasting
How to use the buffer cache	Packet filters
Paging in Linux	Accounting
Page cache management	Firewalls
Finding free pages	UDP
Page exceptions	Standard and extra functions
• Inter-Process Communication	TCP
Synchronization	Standard functions
Communication via files	Communication details
Pipes	• SCSI Subsystem
Debugging using ptrace	Architecture overview
System V IPC	Names and conventions
Socket-based communications	Upper level
• File System	Block devices (hard disks, CD-ROM)
Basic aspects	Character devices (tape)
VFS	Generic drivers
Mouting a file system	Mid level (boot parameters, proc interface)
Superblock	Lower (hardware) level and pseudo drivers
The Inode concept and operations	• Boot Process
File operations	Booting details
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

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