

Linux Kernel Internals and Development - LFD420

Day 1

1. Introduction

- Objectives
- Who You Are
- The Linux Foundation
- Linux Foundation Training
- Course Registration

2. Preliminaries

- Procedures
- Things change in Linux
- Linux Distributions
- Kernel Versions
- Kernel Sources and Use of git
- Platforms
- Documentation and Links

3. Kernel Architecture I

- UNIX and Linux **
- Monolithic and Micro Kernels
- Object-Oriented Methods
- Main Kernel Tasks
- User-Space and Kernel-Space
- Kernel Mode Linux **

4. Kernel Programming Preview

- Error Numbers and Getting Kernel Output
- Task Structure
- Memory Allocation
- Transferring Data between User and Kernel Spaces
- Linked Lists
- String to Number Conversions

- Jiffies
- Labs

Day 2

5. Modules

- What are Modules?
- A Trivial Example
- Compiling Modules
- Modules vs Built-in
- Module Utilities
- Automatic Loading/Unloading of Modules
- Module Usage Count
- The module struct
- Module Licensing
- Exporting Symbols
- Resolving Symbols **
- Labs

6. Kernel Architecture II

- Processes, Threads, and Tasks
- Process Context
- Kernel Preemption
- Real Time Preemption Patch
- Dynamic Kernel Patching
- Run-time Alternatives **
- Porting to a New Platform **

7. Kernel Initialization

- Overview of System Initialization
- System Boot
- Das U-Boot for Embedded Systems**

8. Kernel Configuration and Compilation

- Installation and Layout of the Kernel Source
- Kernel Browsers
- Kernel Configuration Files

- Kernel Building and Makefiles
- initrd and initramfs
- Labs

Day 3

9. System Calls

- What are System Calls?
- Available System Calls
- How System Calls are Implemented
- Adding a New System Call
- Replacing System Calls from Modules
- Labs

10. Kernel Style and General Considerations

- Coding Style
- kernel-doc **
- Using Generic Kernel Routines and Methods
- Making a Kernel Patch
- sparse
- Using likely() and unlikely()
- Writing Portable Code, CPU, 32/64-bit, Endianness
- Writing for SMP
- Writing for High Memory Systems
- Power Management
- Keeping Security in Mind
- Mixing User- and Kernel-Space Headers **
- Labs

11. Race Conditions and Synchronization Methods

- Concurrency and Synchronization Methods
- Atomic Operations
- Bit Operations
- Spinlocks
- Seqlocks
- Disabling Preemption

- Mutexes
- Semaphores
- Completion Functions
- Read-Copy-Update (RCU)
- Reference Counts
- Labs

12. SMP and Threads

- SMP Kernels and Modules
- Processor Affinity
- CPUSETS
- SMP Algorithms - Scheduling, Locking, etc.
- Per-CPU Variables **
- Labs

Day 4

13. Processes

- What are Processes?
- The task_struct
- Creating User Processes and Threads
- Creating Kernel Threads
- Destroying Processes and Threads
- Executing User-Space Processes From Within the Kernel
- Labs

14. Process Limits and Capabilities **

- Process Limits
- Capabilities
- Labs

15. Monitoring and Debugging

- Debuginfo Packages
- Tracing and Profiling
- sysctl
- SysRq Key
- oops Messages

- Kernel Debuggers
- debugfs
- Labs

16. Scheduling Basics

Main Scheduling Tasks

- SMP
- Scheduling Priorities
- Scheduling System Calls
- The 2.4 schedule() Function
- O(1) Scheduler
- Time Slices and Priorities
- Load Balancing
- Priority Inversion and Priority Inheritance **
- Labs

17. Completely Fair Scheduler (CFS)

- The CFS Scheduler
- Calculating Priorities and Fair Times
- Scheduling Classes
- CFS Scheduler Details
- Labs

18. Memory Addressing

- Virtual Memory Management
- Systems With no MMU
- Memory Addresses
- High and Low Memory
- Memory Zones
- Special Device Nodes
- NUMA
- Paging
- Page Tables
- page structure

- Kernel Samepage Merging (KSM) **
- Labs

19. Huge Pages

- Huge Page Support
- libhugetlbfs
- Transparent Huge Pages
- Labs

20. Memory Allocation

- Requesting and Releasing Pages
- Buddy System
- Slabs and Cache Allocations
- Memory Pools
- kmalloc()
- vmalloc()
- Early Allocations and bootmem()
- Memory Defragmentation
- Labs

Day 5

21. Process Address Space

- Allocating User Memory and Address Spaces
- Locking Pages
- Memory Descriptors and Regions
- Access Rights
- Allocating and Freeing Memory Regions
- Page Faults
- Labs

22. Disk Caches and Swapping

- Caches
- Page Cache Basics
- What is Swapping?
- Swap Areas
- Swapping Pages In and Out

- Controlling Swappiness
- The Swap Cache
- Reverse Mapping **
- OOM Killer
- Labs

23. Device Drivers**

- Types of Devices
- Device Nodes
- Character Drivers
- An Example
- Labs

24. Signals

- What are Signals?
- Available Signals
- System Calls for Signals
- Sigaction
- Signals and Threads
- How the Kernel Installs Signal Handlers
- How the Kernel Sends Signals
- How the Kernel Invokes Signal Handlers
- Real Time Signals
- Labs