Gold Microsoft Partner

Linux Kernel Internals and Development (LFD420)

ID LFD420 **Price** 2,990.- € (excl. tax) **Duration** 4 days

Course Overview

Learn how to develop for the Linux kernel. In this course you'll learn how Linux is architected, the basic methods for developing on the kernel, and how to efficiently work with the Linux developer community. If you are interested in learning about the Linux kernel, this is absolutely the definitive course on the subject.

Who should attend

This course is for anyone interested in learning how to write and/or debug Linux kernel code.

Prerequisites

Students should be proficient in the C programming language, basic Linux (UNIX) utilities such as ls, grep and ta, and be comfortable with any of the available text editors (e.g. emacs, vi, etc.) Experience with any major Linux distribution is helpful but not strictly required.

Course Content

Introduction

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

Preliminaries

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

Kernel Architecture I

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

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

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

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 **

Kernel Initialization

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

Kernel Configuration and Compilation

- Installation and Layout of the Kernel Source
- Kernel Browsers
- Kernel Configuration Files
- Kernel Building and Makefiles
- initrd and initramfs
- Labs

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

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

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

SMP and Threads

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

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

Process Limits and Capabilities **

- Process Limits
- Capabilities
- Labs

Monitoring and Debugging

- Debuginfo Packages
- Tracing and Profiling
- sysctl
- SysRq Key
- oops Messages
- Kernel Debuggers
- debugfs
- Labs

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

Completely Fair Scheduler (CFS)

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

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

Huge Pages

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

Memory Allocation

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

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

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

Device Drivers**

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

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

** These sections may be considered in part or in whole as optional. They contain either background reference material, specialized topics, or advanced subjects. The instructor may choose to cover or not cover them depending on classroom experience and time constraints

About Fast Lane

The worldwide Fast Lane Group specializes in high-end technology and business training and consulting. Fast Lane offers complete, authorized training solutions for leading technology vendors, including AWS, Barracuda, Cisco, Google, Microsoft, NetApp, Oracle, Palo Alto Networks, Red Hat, SUSE, Symantec, Veeam, Veritas, VMware and more. In addition to those vendor's standard curricula, Fast Lane also develops advanced technology, sales and business transformation curricula. Fast Lane's vendor-independent consulting services solve a wide range of issues from conducting preliminary analyses and assessments, to designing future-focused IT solutions.

Vendor Expertise

- Amazon Web Services
- Apple
- Arista
- Aruba
- Barracuda
- Brocade
- Check Point
- Cisco
- Citrix
- Cloudera
- EMC
- F5 Networks
- GE
- Gigamon
- Google Cloud
- HP
- IBM
- Intel
- Juniper
- Kaspersky Lab
- Microsoft
- NetApp
- Oracle
- Palo Alto Networks
- Pivotal
- Red Hat
- Ruckus
- Salesforce
- SUSE
- Symantec
- Veeam
- Veritas
- VMware

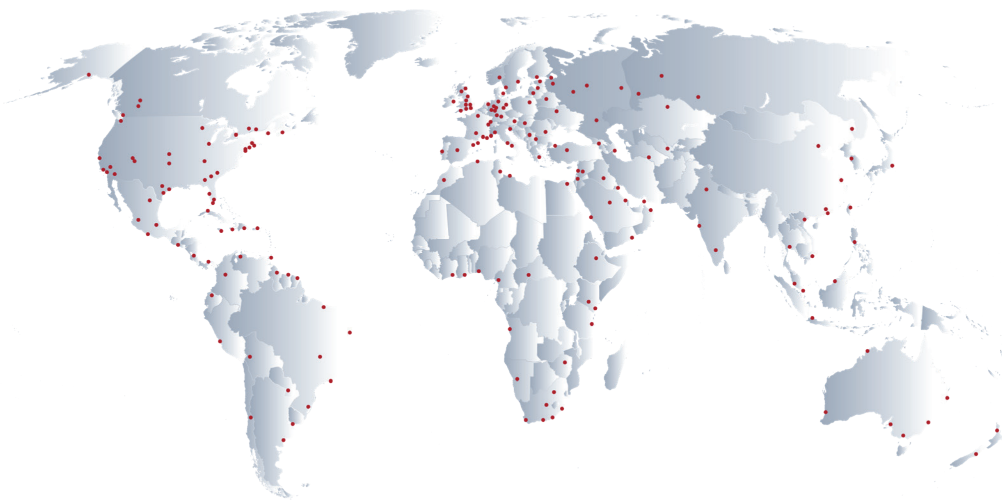
Technology Know-how

- Digital Transformation
- Big Data and Analysis
- Cloud-Solutions: Private, Public & Hybrid Cloud
- Data Center & Virtualisation
- Internet of Things (IoT) / Internet of Everything (IoE)
- IPv6
- Netzwork Infrastructures
- Security
- Software Defined Networking (SDN)
- Unified Communications, Collaboration & Video
- Wireless & Mobility
- Software Development
- Artificial Intelligence (AI)

Fast Lane Services

✓ High-end Technology Training
✓ Business & Softskill Training
✓ Consulting Services
✓ Sales & Business Enablement
✓ Managed Services
✓ E-Learning
✓ Remote Labs
✓ Content Development
✓ Guaranteed Dates
✓ Onsite Training
✓ Talent Programs

Fast Lane Training Centers



Germany
Fast Lane Institute for Knowledge Transfer GmbH

Oranienburger Strasse 66, 10117 Berlin
Tel. +49 30 81451330

Gasstrasse 4a, 22761 Hamburg
Tel. +49 40 25334610

Hansaallee 249, 40549 Duesseldorf
Tel. +49 211 5382980

Ludwig-Erhard-Strasse 3, 65760 Eschborn
Tel. +49 6196 8820410

Johann-Krane-Weg 46, 48149 Muenster
Tel. +49 251 95203000

Rotebühlplatz 21, 70178 Stuttgart
Tel. +49 711 26345990

Parkring 22, 85748 Garching
Tel. +49 89 262075100

info@flane.de / www.flane.de

Austria
ITLS GmbH
(ITLS ist ein Partner von Fast Lane)
Gutheil-Schoder-Gasse 7a, 1100 Vienna
Tel. +43 1 6000 8800
info@itls.at / www.itls.at

Switzerland
Fast Lane Institute for Knowledge Transfer (Switzerland) AG
Husacherstrasse 3, 8304 Wallisellen (Zurich)
Tel. +41 44 8325080
info@flane.ch / www.flane.ch