Publication date : 07/29/2024

Course : Linux , performance analysis

Practical course - 4d - 28h00 - Ref. APL
Price : 2260 € E.T.

This training course will enable you to master the appropriate tools, subsystems and techniques you need to get the most out of Linux, as well as to audit a system.

Teaching objectives

At the end of the training, the participant will be able to:

- ✓ Measuring performance on a Linux system
- ✓ Auditing performance on a Linux system
- ✓ Understanding the inner workings of the core in detail

Intended audience

System administrators, network administrators, system or application operators.

Prerequisites

Good knowledge of Linux.

Practical details

The many progressive exercises and case studies will be carried out on a network of Linux servers.

Course schedule

PARTICIPANTS

System administrators, network administrators, system or application operators.

PREREQUISITES

Good knowledge of Linux.

TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS

The trainer evaluates each participant’s academic progress throughout the training using multiple choice, scenarios, hands-on work and more.
Participants also complete a placement test before and after the course to measure the skills they’ve developed.

1 Core presentation

- System overview and role of the kernel.
- Reference sites.
- Specific features of 3.x and 4.x kernels.
- Kernel development cycles, patches.
- Operating mode (supervisor and user). System calls.
- Organization of sources (Include/linux, Arch, Kernel, Documentation...).
- Principle of kernel and module compilation.
- Dependencies and symbols.
- Symbol exports.
- Kernel loading (support, arguments, etc.).
- Virtual memory management under Linux.
- File system optimization: Ext3/4, logging modes, file system attributes.
- Classic optimizations.

Hands-on work

Compiling and installing a kernel.

2 Tools for use

- Development tools (Gcc, Kbuild, Kconfig and Makefile).
- Debugging tools (GDB, KGDB, ftrace, etc.).
- Debugging environment (Linux Trace Toolkit, etc.).
- Trace system calls (ptrace...).
- Classic metrology tool for Linux.
- Performance data collection.
- Nagios, Ganglia
- Core monitoring.
- Commands: using vmstat, df, stat, cpuinfo, etc.

Hands-on work

Install all tools and sources. Data collection. Kernel monitoring. Use native commands.

3 Thread management and scheduling

- The different types of peripherals.
- Kernel operating contexts. Global variable protection.
- Thread representation (status, task_stru structure, thread_info, etc.).
- Threads, execution context.
- The Linux scheduler and preemption.
- Create a kernel thread (kthread_create, wakeup_process...).

Hands-on work

Monitor and manage threads.

TEACHING AIDS AND TECHNICAL RESOURCES

• The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.

• At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.

• A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

4 Memory, time and proc management

- Memory organization for UMA and NUMA architectures.
- User and kernel address space.
- Demand paging.
- Memory allocations, buddy allocator, kmalloc, slabs and memory pools.
- Memory access management (caches and MMU).
- Problems linked to memory over-reservation.
- Memory management on x86 and ARM, use of Hugepages.
- Optimization of system calls (IAPX32, VDSO).
- Synchronization and waiting in the kernel, waitqueues, mutexes and completions.
- Ticks and jiffies in Linux.
- Real Time Clock (RTC), timer implementation.
- High-resolution timer interface, stamping.
- Kernel-specific tools, linked lists, kfifo and container_of.
- Kernel interface with /proc via procfs.

Hands-on work

Analysis of memory and /proc files.

5 NUMA (non-uniform memory access) optimization

- Main NUMA concepts.
- CPU concepts and architecture.
- NUMA memory allocation.
- NUMA statistics.
- I/O subsystem.

6 Storage and IO

- RAID Refresher software.
- RAID levels, RAID configuration.
- Logical volumes, Volumes and volume groups, Creating logical volumes.
- Raw devices.
- Asynchronous I/O.

Hands-on work

Audit storage.

7 I/O subsystem analysis

- iostats.
- iotop, blktrace, blkparse.
- btrace, btt, blkiomon.

Hands-on work

I/O analysis and interpretation.

8 Network subsystem optimization

- Overview of the network stack.
- Optimize latency and throughput.
- Network interface hardware parameters.
- Unloading techniques.
- TCP optimization.
- Monitoring and diagnostic tools

Hands-on work
Overview of the network stack.

9 Audit

- Methods.
- A fact not to be forgotten.
- The tools.

Hands-on work
Audit a Linux system and produce a report.

Dates and locations

REMOTE CLASS
2026 : 7 Apr., 2 June, 20 Oct., 1 Dec.

PARIS LA DÉFENSE
2026 : 26 May, 13 Oct., 24 Nov.