Course : Unix/Linux system developer

Practical course - 4d - 28h00 - Ref. LIS
Price : 2650 CHF E.T.

This training session will teach you how to make use of the wide and precise UNIX system interface to develop performant applications. You will learn to manage processes and threads, file systems, and memory allocation. At the end of this session, you will be familiar with network communication, signals, Posix and System V IPC.

Prerequisites

Good knowledge of UNIX/Linux system and C programming language.

Practical details

Hands-on work

Progressive practicals and case studies will enable you to fully understand the presentation of the UNIX/Linux system programming interface.

Course schedule

1 Development method and tools

- The linux distributions, open source software, licenses.
- Compilers and associated tools, profilers and debuggers.

Hands-on work

Use of GDB, cscope and profiling with a simple application.

2 Processes

- The UNIX processes, scheduling processes, priorities and processor affinity.
- Live and death of a process. Fork, exec, exit and wait.
- Security issues. Root and standard user, user ids.
- Namespaces and application mobility.

Hands-on work

Creation of a simple multi-process application. Tests of some security issues like process running wild and handling with setrlimit(2).

PARTICIPANTS

PREREQUISITES

Good knowledge of UNIX/Linux system and C programming language.

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.

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.

3 Posix Threads

- Programing with threads. Overview of Posix 1c threads.
- Thread creation and termination. Thread scheduling.
- Synchronizing Threads. Mutex and data protection, priority inversion.
- Condition variable and flow control. Using signals and threads.

Hands-on work
A simple multi-thread application using mutexes and condition variables.

4 Files and filesystems

- File handling.
- Filesystems. Accessing metadata. Accessing directories.
- I/O Multiplexing. Using poll and select.
- Signals and events with multiplexed I/O

Hands-on work
A small program using fcntl(2) for file locking and accessing a directory.

5 Inter Process Communication

- Message queues.
- Shared memory.
- Semaphores. Handling multiple semaphore sets.
- Pipes. Standard I/O redirections.
- Signals. UNIX signals implementation.

Hands-on work
Implementation of a client/server use case with the help of various technologies: message queues, shared memory and semaphores, pipes and signals.

6 Network

- Socket Interface.
- Address and protocol management. TCP/IP interface.
- Network daemons.
- mplementation of network servers and super servers.

Hands-on work
Implementation of our client/server use case with the network interface.

7 Memory management and Time

- Virtual memory.
- Memory allocation.
- Advanced use
- Date and time. Timers and timeout.
- Latencies and determinism.

Hands-on work
Test of various allocations schemes using malloc(3), brk(2) or mmap(2). Implementing good practice for real-time applications. Using the UNIX time interface to measure scheduling latency.

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.

8 Advanced linker control

- Security, real-time and multithreading.
- Shared libraries.

Hands-on work
A multi-thread real-time application with a time share thread.
Implementation of a memory allocation tracer with dlopen.

Dates and locations

REMOTE CLASS
2026 : 16 June, 16 June, 15 Sep., 15 Sep.