Linux Programming

Level: intermediate

Length: 35 – 40 hours

Course Objective: presents the resources available in the Linux operating system, how they are used in programs

What you will learn
- How to access the Linux resources
- Types of resources, how they are used in programs
- Inter-process concurrency, communication between processes
- Intra-process concurrency, threads

Who can attend: C/C++ programmers

Prerequisites
- Knowledge of C or C++ programming languages at least at medium level
- Knowledge to use Linux for the common, basic tasks

Required infrastructure: VGA projector, whiteboard, workstation with a Linux distribution installed (or a virtual Linux machine); for coding we can use an Eclipse distribution for C/C++

Bibliography

- Beginning Linux Programming, 4th Edition, Neil Matthew, Richard Stones, 2008, Wiley Publishing Inc., ISBN: 978-0-470-14762-7
- Advanced Linux Programming, Mark Mitchell & al., 2001, New Riders
- Advanced Programming in the Unix Environment, Richard Stevens & al., 2nd Edition, 2005, Addison Wesley Professional, ISBN: 0201433079

Related courses: The C programming language, The C++ programming language, Advanced C++ Programming, Advanced C Programming, Design Patterns, Introduction to Linux

Description

The course approaches Linux from programmers' perspective.

The Linux resources are presented, how they are used and integrated in applications. The focus is on parallel programming at both processes and threads level. Examples, practical problems are a good introduction for programmers who work on embedded, real-time systems or client / server systems.

Contents

1. Introduction to Linux programming. Fundamental concepts.
2. Tools used in software development; static and dynamic libraries, make
3. Input / output; performing the I/O at low & high level
4. Processes; how they are managed by the OS, how to launch them, waiting the process termination, relations between processes, zombie processes
5. Signals; types, handlers, rules for writing handlers, handlers management, sending signals
6. Timers
7. POSIX threads; starting a new thread, waiting for a thread termination, the main problems in thread programming, threads related resources - semaphore, mutex, variable condition
8. Pipes; types – process, low level, named pipes
9. Semaphores (IPC)
10. Common memory zones (IPC)
11. Messages queues (IPC)
12. Sockets; types - UNIX (locale) & Internet (TCP/IP)
13. (optional) Daemon applications
14. (optional) Logging cu syslog
15. (optional) Advanced I/O operations: asynchronous I/O operations, I/O multiplexing
16. (optional) Introduction to writing Linux device drivers