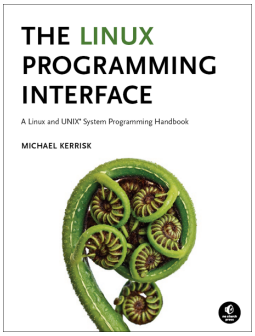
man7.org
Training and Consulting

Linux/UNIX IPC Programming

Course code: M7D-IPC01

This two-day course provides a thorough introduction to the interprocess (IPC) techniques that Linux and UNIX systems provide for use by user-space programs. Using these features allows the creation of complex multiprocess applications that coordinate their actions and exchange information with each other. Detailed presentations coupled with many carefully designed practical exercises provide participants with the knowledge to write such applications. The course dives into some specifics of the Linux system, but makes careful and frequent reference to the POSIX standard, so that it is also valuable to developers working on other UNIX systems.



Audience and prerequisites

The audience for this course includes programmers developing and porting system-level and network applications for Linux and UNIX systems, embedded application developers, security engineers, site reliability engineers, and DevOps engineers.

To get the most out of the course, participants should have:

- Good reading knowledge of the C programming language
- Solid programming experience in a language suitable for completing the course exercises (e.g., C, C++, Go, Rust, or Python)
- Knowledge of basic UNIX/Linux shell commands
- Working knowledge of the fundamental system programming topics covered in the *Linux System Programming Fundamentals* course (M7D-SPINTRO01). In particular, participants should have a good understanding of concepts such as: file descriptors, open file descriptions, file I/O using system calls, process lifecycle (*fork()*, *exec()*, *wait()*), and programming with signals.

Course duration and format

Two days, with up to 50% devoted to practical sessions.

Course materials

- Course books (written by the trainer) that include all slides and exercises presented in the course
- An electronic copy of the trainer's book, *The Linux Programming Interface*
- A source code tarball containing around 30,000 lines of example code written by the trainer

Course inquiries and bookings

For inquiries about courses and consulting, you can contact us in the following ways:

- Email: training@man7.org
- Phone: +49 (89) 2155 2990 (German landline)

Prices and further details

For course prices and further information, please visit the course web page, *http://man7.org/training/ipc_prog/*.

About the trainer

Michael Kerrisk has a unique set of qualifications and experience that ensure that course participants receive training of a very high standard:

- He has been programming on UNIX systems since 1987 and began teaching UNIX system programming courses in 1989.
- He is the author of *The Linux Programming Interface*, a 1550-page book widely acclaimed as the definitive work on Linux system programming.
- He is actively involved in Linux development, working with kernel developers on testing, review, and design of new Linux kernel–user-space APIs.
- Since 2004, he has been the maintainer of the Linux *man-pages* project, which provides the manual pages documenting the Linux kernel–user-space and GNU C library APIs.

Linux/UNIX IPC Programming: course contents in detail

Topics marked with an asterisk (*) are optional, and will be covered on an as-needed basis and/or as time permits

1. **Interprocess communication overview**
   - Taxonomy of IPC facilities
   - Comparison of IPC facilities
2. **Pipes and FIFOs**
   - Creating and using pipes
   - Using pipes to connect filters
   - FIFOs
   - I/O semantics
3. **Introduction to sockets**
   - Socket types and domains
   - Sockets system calls
   - Stream sockets
   - Datagram sockets
4. **UNIX domain sockets**
   - UNIX domain stream and datagram sockets
   - Socket permissions
5. **Internet domain sockets**
   - Internet socket addresses
   - Data representation issues
   - *getaddrinfo()*, *getnameinfo()*
   - Socket options
   - Other sockets-specific APIs
6. **Alternative I/O models**
   - Nonblocking I/O
   - Signal-driven I/O
   - *poll()* and *select()*
   - The *epoll* API
7. **Introduction to POSIX IPC**
   - Common features of POSIX IPC
   - Contrast with System V IPC
8. **POSIX semaphores**
   - Named semaphores
   - Semaphore operations
   - Synchronizing access to a shared resource
   - Unnamed semaphores
9. **POSIX shared memory**
   - Opening SHM objects
   - Using SHM objects
   - Synchronizing access to SHM
   - Unmapping and removing SHM objects
10. **POSIX message queues (*)**
    - Open and closing MQs
    - Message queue attributes
    - Sending + receiving messages
    - The *mqueue* filesystem
    - Message queue limits
    - Message notification
11. **Other IPC methods (*)**
    - *eventfd()*
    - Pseudoterminals
    - File locks
    - Cross-memory attach
    - Shared file mappings