Online Linux trainings

Stable: 08.03.2021 • 08:04 | Revision: 08.03.2021 • 10:20

A quality version of this page, approved on *9 March 2021*, was based off this revision.

Contents

1 Online Linux training

- Learn the ways of Linux-fu, for free

Check this Linux online training with the **very** basics and a **pleasant** look & feel.

- kernelnewbies.org community

Website focusing on Linux kernel for newbies, containing useful pages such as:

  - Kernel hacking Tutorials
  - Useful Documents for Kernel developers
  - Kernel glossary
  - Kernel changes per version
  - Kernel changes per feature

- Bootlin

Bootlin is an engineering company specialized in embedded Linux and more generally in free and open source software for embedded systems.
Bootlin offers training services on embedded Linux, Linux kernel, hardware driver development and system development with Yocto Project / OpenEmbedded.

Bootlin is proposing dedicated training on MP1 platform : [1] Yocto Training will come soon.

Bootlin also offer Free online complete trainings

  - Embedded Linux training : About 500 slides, with practical labs
  - Linux kernel and driver development training: About 500 slides, with practical labs
  - Android system development course : About 400 slides, with practical labs
  - Yocto Project and OpenEmbedded development course : about 250 slides, with practical labs

  In addition to the complete courses, the main page contains also many other links to more detailed trainings.

- Linux Foundation Free Web trainings (registration needed)

The Linux Foundation offers many online video trainings. They are organized by sections on their training portal:

  - Getting Started with ...
  - System Administration
  - Linux Kernel
  - Embedded Development
  - Open Source Strategy
  - Networking
  - Cloud

- Kernel.org documentation portal

Official Kernel documentations such as:

  - The Linux kernel user’s and administrator’s guide
  - Working with the kernel development community
  - Development tools for the kernel
  - How to write kernel documentation
  - The Linux driver implementer’s API guide

- Core API Documentation
- Linux Media Subsystem Documentation
- Linux GPU Driver Developer’s Guide
- Security documentation
- Linux Sound Subsystem Documentation
- Linux Kernel Crypto API

Korean and Chinese translations of the documents are also available.

2 Online Books

- The following list is a subset of the kernel documents provided by Linux community. The complete list is available at the following locations:
  - Location: in kernel tree: **/Documentation/kernel-docs.txt**
  - URL: kernels-docs.rst

2.1 Linux Kernel

- **"Linux Kernel in a Nutshell"**
  - Author: Greg Kroah-Hartman
  - URL:Linux Kernel in a Nutshell
  - Date: 2006
  - Keywords: kernel configuration, kernel build, kernel usage, kernel upgrade, kernel install, kernel boot, kernel command-line, kernel options
  - Description:

  Written by a leading developer (also involved in the Linux kernel maintenance), Linux Kernel in a Nutshell is a comprehensive overview of kernel configuration and building, a critical task for Linux users and administrators.
  From the author:
  "If you want to know how to build, configure, and install a custom Linux kernel on your machine, buy this book. It is written by someone who spends every day building, configuring, and installing custom kernels as part of the development process of this fun, collaborative project called Linux."

- **"The Linux Kernel"**
  - Author: David A. Rusling.
  - URL for ebook version: The Linux Kernel
  - URL for pdf version: The Linux Kernel
  - Date: 1999
  - Keywords: everything!, book.
  - Description:

  Online 200 pages book describing most aspects of the Linux Kernel. Probably, the first reference for beginners. Lots of illustrations explaining data structures usage and relationships.
  - Contents:

  1. Hardware Basics
  2. Software Basics
  3. Memory Management,
  4. Processes
  5. Interprocess Communication Mechanisms
  6. PCI,
  7. Interrupts and Interrupt Handling
  8. Device Drivers
  9. The File system

10. Networks
11. Kernel Mechanisms
12. Modules
13. Processors
14. The Linux Kernel Sources
15. Linux Data Structures
16. Useful Web and FTP Sites
17. The LPD Manifesto
18. The GNU General Public License
19. Glossary

- **"Linux as a Case Study : Its Extracted Software Architecture"**
  - Author: Ivan T. Bowman.
  - URL: Linux as a Case Study : Its Extracted Software Architecture
  - Date: 1999
  - Keywords: conceptual software architecture, extracted design, reverse engineering, system structure.
  - Description:

  Detailed conceptual software architecture of the Linux kernel, automatically extracted from the source code, including useful figures. Gives good overall kernel understanding.

- **"Overview of the Linux Virtual File System"**
  - Author: Richard Gooch.
  - Location: in kernel tree: **/Documentation/filesystems/vfs.txt**
  - URL: Overview of the Linux Virtual File System
  - Date: 2007
  - Keywords: VFS, file system, mounting file systems, opening files, dentries, dcache.
  - Description:

  Brief introduction to the Linux Virtual File System with a description of how it works. Provides a list of the operations that take place when opening a file or mounting a file system and a description of the important data structures and the purpose of each of their entries.

- **"Tracing the Way of Data in a TCP Connection through the Linux Kernel"**
  - Author: Richard Sailer
  - URL: Tracing the Way of Data in a TCP Connection through the Linux Kernel
  - Date: 2016
  - Keywords: Linux Kernel Networking, TCP, tracing, ftrace
  - Description:

  A seminar paper explaining ftrace (a tracing framework) and how to use it for understanding linux kernel internals, illustrated at tracing the way of a TCP packet through the kernel.
  - Abstract:

  This short paper outlines the usage of ftrace as a tool to investigate a running Linux system. ftrace produces a trace-log with information on the source code execution and the context.
  A detailed example is provided that traces the data path in a TCP Connection through the kernel.
  Finally this trace-log is used as a basis for a more conceptual exploration and description of the Linux TCP/IP stack implementation.

- **"Linux IP Networking. A Guide to the Implementation and Modification of the Linux Protocol Stack."**
  - Author: Glenn Herrin.
  - URL: Linux IP Networking
  - Date: 2000
  - Keywords: network, networking, protocol, IP, UDP, TCP, connection, socket, receiving, transmitting, forwarding, routing, packets, modules, /proc, sk_buff, FIB, tags.
  - Description:

  This paper is devoted to the Linux IP Networking. It provides a large panel of source code analysis and explanations ranging from kernel code to "user space configuration tools" code. It provides an overview of the kernel networking implementation and helps understanding all the steps a data packet undergo, from network device reception, all the way to delivery to the application. The version of the kernel code under study is 2.2.14. An example code for a packet dropper feature is also provided.

- **"Design and Implementation of the Second Extended Filesystem"**
  - Author: Rémy Card, Theodore Ts'o, Stephen Tweedie.
  - URL: Design and Implementation of the Second Extended Filesystem
  - Date: 1998
  - Keywords: ext2, linux fs history, inode, directory, link, devices, VFS, physical structure, performance, benchmarks, ext2fs library, ext2fs tools, e2fsck.
  - Description:

  Covers Linux file systems history, ext2 motivation, ext2 features, design, physical structure on disk, performance, benchmarks, e2fsck's passes description... A must read!
  - Notes:

  This paper was first published in the Proceedings of the First Dutch International Symposium on Linux, ISBN 90-367-0385-9.

- **"Analysis of the ext2fs structure"**
  - Author: Louis-Dominique Dubeau.
  - URL: Analysis of the ext2fs structure
  - Date: 1994
  - Keywords: ext2, filesystem, ext2fs.
  - Description:

  Description of ext2's blocks, directories, inodes, bitmaps, invariants...

- **"A Linux vm README"**
  - Author: Kanoj Sarcar.
  - URL: A Linux vm README
  - Date: 2001
  - Keywords: virtual memory, mm, pgd, vma, page, page flags, page cache, swap cache, kswapd.
  - Description:

  Telegraphic, short descriptions and definitions relating the Linux virtual memory implementation.

- **"I/O Event Handling Under Linux"**
  - Author: Richard Gooch.

- [URL: http://web.mit.edu/~yandros/doc/io-events.html I/O Event Handling Under Linux]
- Date: 1999
- Keywords: IO, I/O, select(2), poll(2), FDs, aio_read(2), readiness event queues.
- Description:

Extract from the Introduction: "I/O Event handling is about how your Operating System allows you to manage a large number of open files (file descriptors in UNIX/POSIX, or FDs) in your application. You want the OS to notify you when FDs become active (have data ready to be read or are ready for writing). Ideally you want a mechanism that is scalable. This means a large number of inactive FDs cost very little in memory and CPU time to manage".

- **"The Kernel Hacking HOWTO"**
  - Author: Various Talented People, and Rusty.
  - Location: in kernel tree: */Documentation/kernel-hacking/hacking.rst'*
  - URL: kernel-hacking
  - Keywords: HOWTO, kernel contexts, deadlock, locking, modules, symbols, return conventions.
  - Description:

  Extract from the Introduction: "Please understand that I never wanted to write this document, being grossly underqualified, but I always wanted to read it, and this was the only way. I simply explain some best practices, and give reading entry-points into the kernel sources. I avoid implementation details: that's what the code is for, and I ignore whole tracts of useful routines. This document assumes familiarity with C, and an understanding of what the kernel is, and how it is used. It was originally written for the 2.3 kernels, but nearly all of it applies to 2.2 too; 2.0 is slightly different".

- **"Linux Kernel Locking HOWTO"**
  - Author: Various Talented People, and Rusty.
  - Location: in kernel tree: **/Documentation/kernel-hacking/locking.rst**
  - URL: kernel-locking
  - Keywords: locks, locking, spinlock, semaphore, atomic, race condition, bottom halves, tasklets, softirqs.
  - Description:

  The title says it all: document describing the locking system in the Linux Kernel either in uniprocessor or SMP systems.
  - Notes:

  "It was originally written for the latest 2.3.x kernel revisions (>2.3.47), but most of it applies to 2.2 too; 2.0 is slightly different". Freely redistributable under the conditions of the GNU General Public License.

- **"Linux Kernel Module Programming Guide"**
  - Author: Ori Pomerantz.
  - URL: Linux Kernel Module Programming Guide
  - Date: 2001
  - Keywords: modules, GPL book, /proc, ioctls, system calls, interrupt handlers .
  - Description:

  Very nice 92 pages GPL book on the topic of modules programming. Lots of examples.

- **"How To Make Sure Your Driver Will Work On The Power Macintosh"**
  - Author: Paul Mackerras.
  - URL: How To Make Sure Your Driver Will Work On The Power Macintosh
  - Date: 1999
  - Keywords: Mac, Power Macintosh, porting, drivers, compatibility.

- Description: The title says it all.

2.2 Linux Journal Kernel Korner articles

- **"Dynamic Kernels: Modularized Device Drivers"**
  - Author: Alessandro Rubini.
  - URL: Dynamic Kernels: Modularized Device Drivers
  - Date: 1996
  - Keywords: device driver, module, loading/unloading modules, allocating resources.
  - Description: Linux Journal Kernel Korner article. Here is the abstract:

  "This is the first of a series of four articles co-authored by Alessandro Rubini and Georg Zezchwitz which present a practical approach to writing Linux device drivers as kernel loadable modules. This installment presents an introduction to the topic, preparing the reader to understand next month's installment".

- **"Dynamic Kernels: Discovery"**
  - Author: Alessandro Rubini.
  - URL: Dynamic Kernels: Discovery
  - Date: 1996
  - Keywords: character driver, init_module, clean_up module, autodetection, mayor number, minor number, file operations, open(), close().
  - Description: Linux Journal Kernel Korner article. Here is the abstract:

  "This article, the second of four, introduces part of the actual code to create custom module implementing a character device driver. It describes the code for module initialization and cleanup, as well as the open() and close() system calls".

- **"The Devil's in the Details"**
  - Author: Georg v. Zezschwitz and Alessandro Rubini.
  - URL: The Devil's in the Details
  - Date: 1996
  - Keywords: read(), write(), select(), ioctl(), blocking/non, blocking mode, interrupt handler.
  - Description: Linux Journal Kernel Korner article. Here is the abstract:

  "This article, the third of four on writing character device drivers, introduces concepts of reading, writing, and using ioctl-calls".

- **"Dissecting Interrupts and Browsing DMA"**
  - Author: Alessandro Rubini and Georg v. Zezschwitz.
  - URL: Dissecting Interrupts and Browsing DMA
  - Date: 1996
  - Keywords: interrupts, irqs, DMA, bottom halves, task queues.
  - Description: Linux Journal Kernel Korner article. Here is the abstract:

  "This is the fourth in a series of articles about writing character device drivers as loadable kernel modules. This month, we further investigate the field of interrupt handling. Though it is conceptually simple, practical limitations and constraints make this an ``interesting *part of device driver writing, and several different facilities have been provided for different situations. We also investigate the complex topic of DMA".*

- **"Device Drivers Concluded"**
  - Author: Georg v. Zezschwitz.
  - URL: Device Drivers Concluded
  - Date: 1996
  - Keywords: address spaces, pages, pagination, page management, demand loading, swapping, memory protection, memory mapping, mmap, virtual memory areas (VMAs), vremap, PCI.
  - Description: Linux Journal Kernel Korner article. Here is the abstract:
  "Finally, the above turned out into a five articles series. This latest one's introduction reads: This is the last of five articles about character device drivers. In this final section, Georg deals with memory mapping devices, beginning with an overall description of the Linux memory management concepts".

- **"Network Buffers And Memory Management"**
  - Author: Alan Cox.
  - URL: Network Buffers And Memory Management
  - Date: 1996
  - Keywords: sk_buffs, network devices, protocol/link layer variables, network devices flags, transmit, receive, configuration, multicast.
  - Description: Linux Journal Kernel Korner. Here is the abstract:
  "Writing a network device driver for Linux is fundamentally simple---most of the complexity (other than talking to the hardware) involves managing network packets in memory".

2.3 How to develop Linux drivers

- **"Linux Device Drivers, Third Edition"**
  - Author: Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman
  - URL: Linux Device Drivers, Third Edition
  - Date: 2005
  - Keywords: device drivers, modules, debugging, memory, hardware, interrupt handling, char drivers, block drivers, kmod, mmap, DMA, buses.
  - Description:
  A 600-page book covering the (2.6.10) driver programming API and kernel updates in general. Available under the Creative Commons Attribution-ShareAlike 2.0 license.
  - Contents:

1. An Introduction to Device Drivers
2. Building and Running Modules
3. Char Drivers
4. Debugging Techniques
5. Concurrency and Race Conditions
6. Advanced Char Driver Operations
7. Time, Delays, and Deferred Work
8. Allocating Memory
9. Communicating with Hardware
10. Interrupt Handling

11. Data Types in the Kernel
12. PCI Drivers
13. USB Drivers
14. The Linux Device Model
15. Memory Mapping and DMA
16. Block Drivers
17. Network Drivers
18. TTY Drivers

- **"Linux PCMCIA Programmer's Guide"**
  - Author: David Hinds.
  - URL: Linux PCMCIA Programmer's Guide
  - Date: 2003
  - Keywords: PCMCIA.
  - Description:

  "This document describes how to write kernel device drivers for the Linux PCMCIA Card Services interface. It also describes how to write user-mode utilities for communicating with Card Services.

- **"An Introduction to SCSI Drivers"**
  - Author: Alan Cox.
  - URL: An Introduction to SCSI Drivers
  - Date: 1999
  - Keywords: SCSI, device, driver.
  - Description: The title says it all.

- **"Advanced SCSI Drivers And Other Tales"**
  - Author: Alan Cox.
  - URL: Advanced SCSI Drivers And Other Tales
  - Date: 1999
  - Keywords: SCSI, device, driver, advanced.
  - Description: The title says it all.

- **"Writing Linux Mouse Drivers"**
  - Author: Alan Cox.
  - URL: Writing Linux Mouse Drivers
  - Date: 1999
  - Keywords: mouse, driver, gpm.
  - Description: The title says it all.

- **"More on Mouse Drivers"**
  - Author: Alan Cox.
  - URL: More on Mouse Drivers
  - Date: 1999
  - Keywords: mouse, driver, gpm, races, asynchronous I/O.

- Description: The title still says it all.

- **"Writing an ALSA Driver"**
  - Author: Takashi Iwai <tiwai@suse.de>
  - Writing an ALSA Driver
  - URL: Writing an ALSA Driver
  - Date: 2005
  - Keywords: ALSA, sound, soundcard, driver, lowlevel, hardware.
  - Description:

  Advanced Linux Sound Architecture for developers, both at kernel and user-level sides. ALSA is the Linux kernel sound architecture in the 2.6 kernel version.

- **"Video4linux Drivers, Part 1: Video-Capture Device"**
  - Author: Alan Cox.
  - URL: Video4linux Drivers, Part 1: Video-Capture Device
  - Date: 2000
  - Keywords: video4linux, driver, video capture, capture devices, camera driver.
  - Description: The title says it all.

- **"Video4linux Drivers, Part 2: Video-capture Devices"**
  - Author: Alan Cox.
  - URL: Video4linux Drivers, Part 2: Video-capture Devices
  - Date: 2000
  - Keywords: video4linux, driver, video capture, capture devices, camera driver, control, query capabilities, capability, facility.
  - Description: The title says it all.

- **"Writing Video4linux Radio Driver"**
  - Author: Alan Cox.
  - URL: Writing Video4linux Radio Driver
  - Date: 1999
  - Keywords: video4linux, driver, radio, radio devices.
  - Description: The title says it all.

Linux® is a registered trademark of Linus Torvalds.

Application programming interface

Graphics Processing Units

Virtual File System

input/output

Portable Operating System Interface based on uniX (https://en.wikipedia.org/wiki/POSIX_terminal_interface for more details)

Operating System

Central processing unit

symetric multiprocessing

Direct Memory Access

TeleTYpewriter

Advanced Linux sound architecture