

BIBLIOGRAPHY

Most of the information in this book has been extracted from the kernel sources, which are the best documentation about the Linux kernel.

Kernel sources can be retrieved from hundreds of FTP sites around the world, so we won't list them here.

Version dependencies are best checked by looking at the patches, which are available from the same places where you get the whole source. The program called *repatch* might help you in checking how a single file has been modified throughout the different kernel patches; it is available in the source files provided on the O'Reilly FTP site.

On *sunsite.unc.edu* and all its mirrors you can also find several device drivers, which can surely help in writing your own.

Linux Kernel Books

Bar, Moshe. *Linux Internals*. McGraw-Hill. 2000. This terse book by *Byte* columnist Moshe Bar covers much of how the Linux kernel works, and includes a number of 2.4 features.

Bovet, Daniel P., and Marco Cesati. *Understanding the Linux Kernel*. O'Reilly & Associates. 2000. Covers the design and implementation of the Linux kernel in great detail. It is more oriented toward providing an understanding of the algorithms used than documenting the kernel API.

Maxwell, Scott. *Linux Core Kernel Commentary*. Coriolis. 1999. Mostly a large listing of the core kernel code, with 150 pages of commentary at the end. It can be useful for trying to figure out what is happening in a particular part of the kernel.

Nutt, Gary J. *Kernel Projects for Linux*. Addison-Wesley. 2000. Written to be used in college-level classrooms; as such, it is not a full introduction to the Linux kernel in its own right. For those looking to play with the kernel, though, this book can be a good aid.

Bibliography

Unix Design and Internals

Bach, Maurice. *The Design of the Unix Operating System*. Prentice Hall. 1987. This book, though quite old, covers all the issues related to Unix implementations. It was the main source of inspiration for Linus in the first Linux versions.

Stevens, Richard. *Unix Network Programming*. P T R Prentice-Hall. 1990. Perhaps the definitive book on the Unix network programming API.

Stevens, Richard. *Advanced Programming in the UNIX Environment*. Addison-Wesley. 1992. Every detail of Unix system calls is described herein, making it a good companion when implementing advanced features in the device methods.

Professional Linux Kernel Architecture is an excellent book as well. Both these books are a bit dated, but they touch on essential topics that should help you get started. If you have a specific interest in drivers, or networking, there are other books that can help you there. However the 2 books listed above are excellent to help you get started.Â Linux Kernel Development details the design and implementation of the Linux kernel, presenting the content in a manner that is beneficial to those writing and developing kernel code, as well as to programmers seeking to better understand the operating system and become more efficient and productive in their coding. The book details the major subsystems and features of the Linux kernel, including its design, implementation, and interfaces. General Linux Books. Ellen Siever, Aaron Weber, Stephen Figgins, Robert Love, Arnold Robbins. Linux in a Nutshell, Fifth Edition . O'Reilly & Associates, 2005. This book can be has the most complete and authoritative command reference for Linux. It covers almost every different command that you will ever need to use. Yaghmour, Karim. Building Embedded Systems . O'Reilly & Associates, 2003. This book, although mainly oriented toward the embedded Linux developer, has a great section on how to build up a cross-compiler toolchain and kernel. It is highly recommended for that, as This book is an exploration of the Linux Kernel. The first part of the book is a guide for you on how to work with the Initial RAM Disk (initrd). This simply provides us with an easy way to load the RAM disk using the boot loader. The necessary steps which can help you achieve this, and the necessary tools for you have been discussed. The tools which can be used for kernel development are discussed in this book. The first tool discussed in this book is the kcov.Â This means that you will learn how to create modules for the Linux kernel on your own. The following topics are discussed in this book: - Initial RAM Disk (initrd) - Kernel Development Tools - Writing Linux Kernel Modules. Download from free file storage. Resolve the captcha to access the links!