

## BIBLIOGRAPHY

Abrahams, Paul W., and Bruce R. Larson. *UNIX for the Impatient*. 2nd ed. Reading: Addison-Wesley, 1995.

Adobe Systems Inc. *PostScript Language Reference*. 3rd ed. Reading: Addison-Wesley, 1999.

The definitive reference to the PostScript page description language.

Adobe Systems Inc. *PostScript Language Tutorial and Cookbook*. Reading: Addison-Wesley, 1985.

A guide to learning the PostScript page description language. Includes many examples.

Aho, Alfred V., Brian W. Kernighan, and Peter J. Weinberger, *The AWK Programming Language*. Reading: Addison-Wesley, 1988.

By the creators of this language.

Aho, Alfred V., Ravi Sethi, and Jeffery D. Ullman. *Compilers: Principles, Techniques, and Tools*. Reading: Addison-Wesley, 1986.

The classic textbook for compiler construction, also known as “the dragon book.” Although nearly 20 years old, this textbook still covers more than enough for a course.

Albitz, Paul, and Cricket Liu. *DNS and BIND*. 4th ed. Sebastopol: O’Reilly & Associates, 2001.

The domain name service maps names to IP addresses. If you need to administer domain names and you have no prior experience, you probably need this book.

Appel, Andrew W.. *Modern Compiler Implementation in ML*. Cambridge: Cambridge University Press, 1997.

One of the many newer compiler books on the market, Appel's guide spends less time with tokenizing and parsing, focusing instead on intermediate representation, code generation, and advanced topics such as optimization.

Beazley, David M. *Python Essential Reference*. 2nd ed. Indianapolis: New Riders, 2001.

A no-nonsense reference to the Python language and library.

Beazley, David M., Brian D. Ward, and Ian R. Cooke, "The Inside Story on Shared Libraries and Dynamic Loading." *Computing in Science and Engineering* 3, no. 5 (2001): 90-97.

An overview of how shared libraries and dynamic loading work, and the things you can do with them.

Bell Labs, *Plan 9: Programmer's Manual*. 2nd ed. Washington, D.C.: International Thompson Publishing, 1995.

Plan 9 is the operating system that the creators of Unix made after they got sick of Unix, featuring many interesting improvements in networking and the underlying filesystem infrastructure.

Dougherty, Dale, and Arnold Robbins. *sed & awk*. 2nd ed. Sebastopol: O'Reilly & Associates, 1997.

A two-for-one guide to the stream editor and the AWK programming language.

Filesystem Hierarchy Standard Group. "Filesystem Hierarchy Standard," Version 2.3 (2004), Rusty Russell, Daniel Quinlan, and Christopher Yeoh, eds. <http://www.pathname.com/fhs/> (accessed March 2004).

A document describing why the Linux directory structure looks as it does.

Friedl, Jeffery E. F. *Mastering Regular Expressions*. 2nd ed. Sebastopol: O'Reilly & Associates, 2002.

The regular expression is the basis of Unix pattern matching. You will use them in basic utilities such as grep, in configuration files, and in scripting languages such as Perl.

Hennessy, John L., David A. Patterson, and David Goldberg, *Computer Architecture: A Quantitative Approach*. 3rd ed. San Francisco: Morgan-Kaufmann, 2002.

Explains why computer hardware works as it does, and why it is designed as it is. This book is comprehensive, covering topics from economic aspects to pipelining and performance analysis.

Hopcroft, John E, Rajeev Motwani, and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. 2nd ed. Reading: Addison-Wesley, 2000.

The classic textbook for finite state automata, turing machines, and elementary complexity in computer science theory.

Hyde, Randall. *The Art of Assembly Language*. San Francisco: No Starch Press, 2003.

Covers how to program in x86 assembly language.

Kane, Gerry, and Joe Heinrich. *MIPS RISC Architecture*. 2nd ed. Englewood Cliffs: Prentice Hall PTR, 1991.

Describes MIPS RISC architecture and assembly language. An excellent overview of RISC design.

Kernighan, Brian W., and Rob Pike. *The UNIX Programming Environment*. Englewood Cliffs: Prentice Hall, 1978.

The classic guide to programming Unix system, covering the C compiler, shell scripts, and make. Although this book is showing its age (many of the programming components are outdated), it shows you the basic structure of development environment that is still used today.

Kernighan, Brian W., and Dennis M. Ritchie. *The C Programming Language*. 2nd ed. Upper Saddle River: Prentice Hall, 1988.

Probably the best computer book ever written.

Killian, Tom, "Processes as files." Proceedings of the Summer 1984 USENIX Conference, pages 203-207. Salt Lake City, 1984.

The origin of the /proc filesystem.

Kochan, Stephen C., and Patrick Wood. *Unix Shell Programming*. 3rd ed. Indianapolis: Sams, 2003.

Lamb, Linda, and Arnold Robbins. *Learning the vi Editor*. 6th ed. Sebastopol: O'Reilly & Associates, 1998.

Lendেকে, Volker. *How Samba Works*. San Francisco: No Starch Press, 2004.

A refreshingly short guide to how Samba, SMB/CIFS, and NetBIOS work.

Levine, John R., *Linkers and Loaders*. San Francisco: Morgan-Kaufmann, 2000.

A comprehensive textbook for how linkers, shared libraries, and dynamic loaders work.

Oram, Andrew, and Steve Talbott. *Managing Projects with make*. 2nd ed. Sebastopol: O'Reilly & Associates, 1991.

Peek, Jerry, Grace Todino-Gonguet, and John Strang. *Learning the UNIX Operating System*. 5th ed. Sebastopol: O'Reilly & Associates, 2002.

- Preston, W. Curtis. *UNIX Backup and Recovery*. Sebastopol: O'Reilly & Associates, 1999.  
Backups for the enterprise, including automated backup systems such as Amanda.
- Raymond, Eric S., ed., *The New Hacker's Dictionary*. 3rd ed. Cambridge: MIT Press, 1996.  
The in-print version of the so-called jargon file.
- Salus, Peter H., *A Quarter Century of UNIX*. Reading: Addison-Wesley, 1994.  
Where Unix came from and how it evolved.
- Schneier, Bruce, *Applied Cryptography*. 2nd ed. Hoboken: Wiley, 1995.  
The only guide to cryptography that you're likely to need; explains the theory of cryptographic algorithms and how to implement them.
- Schwartz, Randal L., and Tom Phoenix. *Learning Perl*. 3rd ed. Sebastopol: O'Reilly & Associates, 2001.
- Silberschatz, Abraham, Greg Gagne, and Peter Baer Galvin, *Operating System Concepts*. 6th ed. Hoboken: Wiley, 2002.  
The classic textbook for operating systems kernels, sometimes called "the dinosaur book."
- Stallman, Richard, Roland Pesch, and Stan Shebs. *Debugging with GDB*. 9th ed. Cambridge: Free Software Foundation, 2002.
- Stallman, Richard. *GNU Emacs Manual*. 15th ed. Cambridge: Free Software Foundation, 2002.
- Tanenbaum, Andrew S., *Computer Networks*. 4th ed. Upper Saddle River: Prentice Hall PTR, 2002.  
Explains just about anything you'd want to know about the theory of computer networks and how they are built. This is not a programmer's guide.
- Wall, Larry, and Tom Christiansen, and Jon Orwant, *Programming Perl*. 3rd ed. Sebastopol: O'Reilly & Associates.