“I hate black boxes” [1] certainly echoes my sentiments, and, I daresay, those of many others. Because we could think of most GNU/Linux distributions and literature as gray-to-white boxes, there is a different partition for Linux literature: concrete, strewn-together particulars that “rattle around” in the boxes, versus well-organized information for professional-level consumption. This outstanding book is most safely in the latter category. It gives the reader robust concepts and implementable knowledge of this environment. Graphical user interface (GUI)-based users and developers do not get short shrift, despite the full-power treatment of the command-line interface (CLI).


Every programmer should read the introduction’s Unix/Linux philosophy section. In contrast to many Linux books, chapter 1, the “primer,” already includes compilation and execution of programs. Wang resolves the following paradox in the best way that I’ve seen: “To learn Linux, you must use it, and, of course, to use it, you must learn it.” Highlights of chapter 2, on the Bash shell used interactively, include conceptual rendering of cookbook factoids such as control-V quoting; species of redirection; and background-foreground management, including interrupts and termination. "Each command line undergoes a number of transformations before it is executed." Elaboration lends much-needed depth to this concept. The book clearly explains the complex semantics of quoting (backslash, and single and double quotes).

Chapter 3 uses the GNU object model environment (GNOME) as its main desktop/windows illustrator, though it also gives KDE ample mention. Wang’s precision extends to pointing out that X-Windows is a GUI server, and not just a built-in operating system user interface. Wang examines this kind of client-server interface in depth later in the book.

The chapter on filters and regular expressions reinforces the book’s conceptual bent: As Linux has 700-odd commands that can be combined, a cookbook-style “mastery” would be an oxymoron. Regular expressions, which specify patterns in texts and data, are illustrated via combinations of grep, sed, vi, and awk. Page 103 enumerates optimal filter-construction rules. The treatment of files via sed, grep, sort, and awk is powerfully illustrative.

The chapter on writing shell scripts is a good summary of Bash. That “two shells are involved” in script invocation, the interactive and a subshell, reveals the underlying concepts. Wang treats clearly the semantics of Bash functions. He mentions Perl, but does not elaborate; among the few typographical errors is the name of Perl’s creator.

The chapter on the file system as “a logical organization imposed on physical data storage media … by the operating system” will teach everyone something. The networking chapter is, like the others, complete and conceptual. It instantiates the network, transport, and application layers using Internet protocol (IP), transmission control protocol (TCP) and user datagram protocol (UDP), and hypertext transfer protocol (HTTP). It also applies this abstract-
concrete combination to public key cryptography, message-digest (checksum), and digital signature software. Chapter 8 treats Linux, Apache, MySQL, and PHP (LAMP) Web hosting, and has examples of scripts and directives for each component.

Chapters 9, 10, and 11 cover Linux systems programming. “C remains special for Linux. [...] The portability of Linux is due, in large part, to the portability of C.” The masterly treatment of C includes the distinction between “exit” and “_exit” and the precept that the error value from a system call “must not be anything the (kernel) function would ever return without failure.” Recovery from nonlocal errors, via setjmp and longjmp, is treated on page 278. C code for a complete Linux command starts on page 305. Wang illustrates control of processes, including the “fork” mechanism, with the help of the state transition diagram: ready, running, waiting, and “zombie.” Interprocess and network communication includes code for pipes and sockets; the “Hello World of socket programming” is tcp_echo.c, on page 346. The Internet server daemon, xinetd, enables “any program that uses standard I/O [to] be made into an Internet TCP server without changing its code.”

The final chapter, on GUI programming using the Ruby-GTK2 combination, gives the reader the feel of the Ruby scripting language, which, along with the enumeration of GTK2’s atomic widgets and also the good example of the GUI for tape archive, gives the student a leg up on this vast topic.

This authoritative and exceptionally well-constructed book has my highest recommendation. It will repay careful and recursive study.

Reviewer: George Hacken

Would you recommend this review?  yes  no

Other reviews under "Linux":

Helmke M., Sams publishing, Indianapolis, IN, 2013. 900 pp. Type: Book (978-0-672336-93-5)  Date: Mar 26 2014

Practical Raspberry Pi

Learn Raspberry Pi with Linux

more...