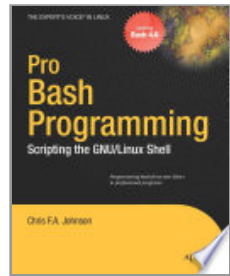


GET PRINT BOOK

No eBook available

- Apress.com
- Rediff Books
- Flipkart
- Infibeam

Find in a library
All sellers »



Pro Bash Programming: Scripting the Linux Shell

By Chris Johnson

Search in this book Go

About this book



Shop for Books on Google Play

Browse the world's largest eBookstore and start reading today on the web, tablet, phone, or ereader.

Go to Google Play Now

Introduction

Although most users think of the shell as an interactive command interpreter, it is really a programming language in which each statement runs a command. Because it must satisfy both the interactive and programming aspects of command execution, it is a strange language, shaped as much by history as by design.

Brian Kernighan and Rob Pike, *The UNIX Programming Environment*, Prentice-Hall, 1984

The shell *is* a programming language. Don't let anyone tell you otherwise. The shell is not just glue that sticks bits together. The shell is a lot more than a tool that runs other tools. *The shell is a complete programming language!*

When a Linux user asked me about membership databases, I asked him what he really needed. He wanted to store names and addresses for a couple of hundred members and print mailing labels for each of them. I recommended using a text editor to store the information in a text file, and I provided a shell script to create the labels in PostScript. (The script, `ps-labels`, appeared in my first book, *Shell Scripting Recipes: A Problem-Solution Approach*.)

When the SWEN worm was dumping hundreds of megabytes of junk into my mailbox every few minutes, I wrote a shell script to filter them out on the mail server and download the remaining mail to my home computer. That script has been doing its job for several years.

I used to tell people that I did most of my programming in the shell but switched to C for anything that needed the extra speed. It has been several years since I have needed to use C, so I no longer mention it. I do everything in the shell.

A shell script is as much a program as anything written in C, Python, or any other language. Just because shell scripts are easier to write doesn't mean they should take a backseat to compiled programs or other scripting languages. I use the terms *script* and *program* interchangeably when referring to tasks written in the shell.

Why the Shell?

Some Linux users do all of their work in a GUI environment and never see a command line. Most, however, use the shell at least occasionally and know something about Unix commands. It's not a big