[PDF] Portable Shell Programming: An Extensive Collection Of Bourne Shell Examples

Bruce Blinn - pdf download free book

Books Details:
Title: Portable Shell Programming: A
Author: Bruce Blinn
Released: 1995-10-29
Language:
Pages: 288
ISBN: 0134514947
ISBN13: 978-0134514949
ASIN: 0134514947

---

pdf, mobi, epub, azw, kindle

Description:

From the Publisher This complete guide shows how to use the shell to develop shell scripts, using the shell more like a programming language than a command interpreter. Covers shell syntax, portability on different UNIX systems, using shell scripts to catch or ignore signals, executing commands using the remote shell command, and using the shell's redirection syntaxes.

From the Inside Flap The most frequently asked question in shell programming is:

Where can I find examples of shell scripts?

Most engineers find that the easiest way to learn shell programming is to learn from the examples of others. Similarly, the easiest way to write a shell script is to use an existing shell script as a template. This stems from the fact that there are innumerable commands and techniques that can be used to solve any particular problem. Seeing an example helps you sift through the chaff to get to a solution quickly. Oftentimes shell programming is used to solve a short-term problem, and, therefore, it is important to get something working quickly; the task might not be worth doing if it requires a significant amount of learning or trial and error to come up with a solution.

Seeing examples also helps you to remember the peculiarities of the shell syntax and the idiosyncrasies of the UNIX commands. Since writing shell scripts is not an everyday task for most engineers, it is easy to forget those infrequently used details. It is usually easier to modify a shell script that contains many of the ideas that you need than it is to develop a new shell script from scratch.

Shell programming is too useful to avoid simply because you find the syntax difficult to remember or the number of UNIX commands overwhelming. Besides executing simple commands such as echo, date, cp, and so forth, the shell defines a command language e that allows you to combine commands into complex shell scripts. The shell supports variables, control flow statements (such as if, while, and for), functions, parameter passing, and even interrupt handling. Coupling the capabilities of the shell with the wide variety of commands available on UNIX, very powerful shell scripts can be written in just a few lines. In addition, since the shell is an interpretive language, there is no compiling or linking to worry about. You can create simple and powerful shell scripts in just a few minutes.

When I first began to write shell scripts, I found it difficult to find useful examples of shell scripts, and when I did find them, they were often difficult to understand. To this end, I believe it is useful to have a reference of relevant, well-doc umented examples, which is what I have tried to provide in this book. I have also tried to provide enough variety in the examples and their explanations that you will become familiar with many of the common practices used in writing shell scripts, an d, therefore, it will be easier for you to understand the code in other shell scripts that you find. This is not a book on how to use UNIX, as most books on shell programming are. This book is intended for people who are already familiar with UNIX and, for one reason or another, have decided to write a shell script. This book is intended for people who already know what an if statement is, but they do not know or cannot remember how an if statement is implemented in the shell. If they could just see an example, they would be off and programming. The examples in this book cover everything from simple shell statements to complex shell scripts, and unlike most books on shell programming, this book contains examples that you are likely to use.

This book is intended to be useful as a reference book. The style of presentation is straightforward. Each topic is self-contained, which allows you to skip directly to the information that you are interested in. This book provides an extensive collection of examples that are organized and indexed so that you have a ready reference to a wide variety of shell programming tasks. If you choose to read the book from cover to cover, you will be exposed to a variety of techniques and practices that would normally take you years to discover.

All of the examples in the book are written using the standard Bourne shell1 (sh or /bin/sh), which allows these examples to be used on any UNIX system. While there are several shells to choose from on most UNIX systems, and many of them are more suitable table than the Bourne shell for interactive work, the standard Bourne shell is available on all modern UNIX systems, and therefore,

is the most common choice for writing shell scripts. In addition, the Korn shell and the POSIX shell contain all of the e syntactic constructs of the Bourne shell, which makes the examples in this book compatible with those shells as well. Some of the examples in this book have dependencies on the flavor of UNIX (e.g., System V or BSD) and some of the examples are dependent upon the particular implementation of a command by the vendor, but these dependencies are explained in the text, and in each case a portable solution is shown. This makes the examples in this book especially useful to those who write shell scripts that must execute on more than one type of UNIX system.

When reading this book, it is important to remember that there are many ways to perform most tasks on UNIX systems. I have chosen implementations for the examples in this book that are easy to understand, but not necessarily the most efficient solution on, or in some other way, the most desirable solution. I will sometimes comment on alternatives, but for the most part, I leave it up to you to determine when an example from this book should be altered for your particular situation.

There is no warranty of any kind on the examples in this book. I have tested each example on several different UNIX systems, but because of the wide variety of UNIX systems and configurations available, it is possible that some examples will not work exactly as I have indicated. I welcome any comments that you might have about the content of this book, including problems with examples and suggestions for improvements.

How This Book Is Organized

The first four chapters in this book describe the syntax of the Bourne shell.

Chapter 1, "Shell Syntax," begins by introducing you to the basic syntax of the Bourne shell. This chapter describes simple commands, quotes and quoting, file name expansion n, and control flow statements.

Chapter 2, Shell Variables, continues the presentation of shell syntax. It describes how to assign values to variables, how to retrieve those values, variable initialization techniques, and variables that are built into the shell.

Chapter 3, Shell Functions and Built-in Commands, presents the syntax for shell functions and discusses techniques for using them. This chapter also lists the commands that are built into the shell and gives a brief description of each one.

Chapter 4, Using Files, presents the shell syntax for input-output redirect. It shows a variety of examples that use the shell redirection syntax to redirect the standard input and the standard output of commands, and it also discusses how to use the shell redirection syntax to read and write other files.

Chapter 5, The Environment, describes the process environment and how it affects shell scripts. This chapter discusses how parent and child processes can affect each other, how signals can be used in shell scripts, and it presents a variety of examples on how to execute commands on remote systems.

Chapter 6, Parsing Command Line Parameters, explains the conventions used to pass parameters to shell scripts and presents examples of common techniques that are used to parse the command line parameters.

Chapter 7, Using Filters, explains what a filter is, how to write one, and how they are used. This chapter contains many examples of filters that can be used to process text files.

Chapter 8, Shell Utilities, has examples of how to perform many common tasks that are frequently used in larger shell scripts. This chapter covers such things as doing arithmetic, manipulating strings, manipulating files and directories, asking questions, presenting output, and much more.

Chapter 9, Examples of Shell Functions, has examples of complete shell functions. These functions demonstrate a variety of reasons to use shell functions, and each function in this chapter contains a line-by- line explanation describing exactly how the function works.

Chapter 10, Examples of Shell Scripts, has examples of complete shell scripts. Each example is complete with comments and demonstrates good shell programming practices. Each shell script in this chapter contains a line-by-line explanation describing exactly how the shell script works

Chapter 11, Debugging, describes the debugging options in the shell and other techniques that are useful for debugging shell scripts. This chapter also describes a variety of common problems and unusual coding styles that can lead to errors.

Chapter 12, Portability, explains the most common issues surrounding the portability of shell scripts and how to avoid portability problems. This chapter also discusses some of the portability issues that arise with specific UNIX commands.

Chapter 13, Common Questions and Problems, is structured as a question and answer discussion. It explains many of the problems that inexperienced users are likely to encounter, and it answers many of the questions they are likely to ask.

Appendix A, Comparison of UNIX Shells, provides a brief description of each of the major shells available on UNIX systems

Appendix B, Syntax Summary, is a complete but brief summary of the syntax of the Bourne shell.

Conventions Used in this Book

To differentiate the text of the explanations in this book from the examples of shell code, typewriter font is used to represent the examples of shell code. Typewriter font is also used to indicate command names, file names, or anything else that must t be entered literally as it is shown in the text.

```
echo Hello world.
```

Text in italics is used to indicate text that should be replaced with an appropriate value rather than entering the text literally as it is shown.

```
rm file
```
When expressing command syntax, square brackets () indicate an optional entry, a vertical bar () separates multiple selections, and ellipses (...) indicate that the previous entry may be repeated.

```
cp file1 file2 ... target
```
Nonprinting characters are represented by the name of the character in italics and surrounded by angle brackets. This notation is only used where the additional clarity is necessary.

Even though this book does not cover using the shell instructively, it is frequently useful to use the

shell interactively to test new ideas. Some of the examples in this book are presented this way. These examples will use the dollar sign ($) as the shell prompt and the greater than sign (>) as the prompt for continuation lines.

```
$ if command
> ...
> fi
$
```

Software Included with this Book

Many of the examples in this book are complete, ready-to-use shell scripts. These examples (which include all of the examples in Chapter 9, Examples of Shell Functions, and Chapter 10, Examples of Shell Scripts) are contained on the floppy diskette attached to this book and they are also available electronically from the Prentice Hall FTP server.

The files on the floppy diskette are stored as a tar archive. To extract the files from the floppy diskette, enter the following command:

```
tar -xvf device
```

Where device is the name of the floppy diskette device on your system (for example, /dev/fd0). This command will create a subdirectory named shbook that contains the examples from this book. To obtain the examples electronically from the Prentice Hall FTP server, you can use the ftp command to connect to ftp.prenhall. The files are contained in the compressed, tar archive file /pub/blinn/shbook.tar.Z. The following example shows a sample dialog to obtain this file from the Prentice Hall FTP server. The underlined text is the portion of the dialog that you enter, the remainder of the text is the information displayed by the ftp command.

```
$ ftp ftp.prenhall
Connected to ... 220 ... FTP server (...) ready.
Name (ftp.prenhall:username): anonymous 331 Guest login ok, send ident as password.
Password: username@xxx.xxx.xxx (Your user name and full host name; it will not echo) 230 Guest
login ok, access restrictions apply.
ftp> cd /pub/blinn 250 CWD command successful.
ftp> binary 200 Type set to I.
ftp> get shbook.tar.Z 200 PORT command successful.
150 Binary data connection for shbook.tar.Z (n.n.n.n,n) (n bytes).
226 Transfer complete.
n bytes received in n.n seconds (n.n Kbytes/s)
ftp> quit
221 Goodbye
$
```

After executing this command, the file shbook.tar.Z should be in the current directory. To unpack this file, enter the following commands: uncompress shbook.tar.Z tar -xvf shbook.tar

This will create a subdirectory named shbook that contains the examples from this book.

- Title: Portable Shell Programming: An Extensive Collection of Bourne Shell Examples
- Author: Bruce Blinn
- Released: 1995-10-29
- Language:
- Pages: 288
- ISBN: 0134514947
- ISBN13: 978-0134514949
- ASIN: 0134514947