Understanding System Concepts

CERTIFICATION OBJECTIVES

1.01 Defining System Administration Terms
1.02 Defining the Effects of Various man Command Options in Viewing Online Manual Pages

☑ Two-Minute Drill
Q&A Self Test
To truly understand any machine, you must first understand its function and parts. Then you must understand how those parts work together to perform the machine’s function.

This chapter touches on all three topics—function, parts, and how the parts work together—in very general terms. The rest of the book goes into specific detail, but first you need to understand some common terms. This chapter covers the basic terminology in system administration and the use of the man command. This command is a starting point designed to give you the foundation on which the rest of the book builds.

CERTIFICATION OBJECTIVE 1.01

Defining System Administration Terms

This section introduces you to the basic terminology that you’ll need to understand in your role as system administrator.

Daemons

Many sysadmins are quick to point out that a daemon is not a demon. Even though it might sometimes seem like it, your server is not possessed by a devil. (Some of your users might be, but not your server.) A daemon is defined as a program that runs in the background, disconnected from a terminal or a particular login session. It is often used to manage system functions.

The term “daemon” is derived from Greek mythology. The ancient Greeks called a supernatural being that acted as an intermediary between the gods and man a daemon. This is an accurate description of what a UNIX daemon does for you. Basically, a daemon is always there, waiting to be called on to perform some action or service. Daemons are running on a system for every service the system provides. However, don’t confuse the terms “daemon” and “process.”

A process is an instance of a running program. So although a daemon is a process, a process isn’t necessarily a daemon. You can see all the processes that are running on your system by using the ps command, which displays all the processes at once. These processes can comprise one big list, even on a little-used system. Naturally,
with the use of command options, you can use the ps command to display only
the information you're looking for. The ps command is covered in more detail in
Chapter 7, but for now, let's take a look at some of the daemons that are running
on your system.

EXERCISE 1-1

Viewing the Processes on a System

Let's first take a look at what is going on behind the scenes on your Solaris system.
For this exercise, your system's hostname is enterprise.timgibbs.net.

1. Log in to the system:

   Trying 192.168.0.2...
   Connected to enterprise.
   Escape character is '^]'.
   SunOS 5.9
   login:gibbst
   password: **********
   Last login: Tue Apr  3 19:12:13 from :0
   Sun Microsystems Inc.   SunOS 5.9
   Welcome to enterprise.timgibbs.net
   [enterprise: gibbst] $

2. List the processes.

   Now you've logged in to enterprise as the user gibbst and seen a message from
   the very conscientious system administrator regarding a planned outage.

3. Now look at the processes, including the daemons, that are currently running
on enterprise by using the following ps command:

   [enterprise: gibbst] $ ps -ef | more

   UID   PID  PPID  C    STIME TTY      TIME CMD
   root     0     0  0   Mar 20 ?        0:03 sched
   root   196     1  0   Mar 20 ?        0:16 /usr/sbin/nscd
   root   206     1  0   Mar 20 ?        0:00 /usr/lib/lpsched
   root   177     1  0   Mar 20 ?        0:01 /usr/sbin/syslogd
   root   268     1  0   Mar 20 console  0:00 /usr/lib/saf/ttymon -g -h
   daemon   160     1  0   Mar 20 ?        0:00 /usr/lib/nfs/statd
   root   158     1  0   Mar 20 ?        0:01 /usr/sbin/inetd -s
   root   157     1  0   Mar 20 ?        0:00 /usr/lib/nfs/lockd
   root   186     1  0   Mar 20 ?        0:01 /usr/sbin/cron
   --More--
The ps command is used here with the e and f options. This command displayed every process in a full listing. Because the list would easily cover more than one screen, we piped (|) the display into the more command. This way, you can see one screen at a time, and by pressing the SPACEBAR, you see more of the display. The ps command is covered in more detail in Chapter 7. For now, let’s go daemon hunting!

4. Examine the following line, which is bolded in the preceding ps command output:

```
root  206     1  0   Mar 20 ?        0:00 /usr/lib/lpsched
```

This is one of the daemons running on the system named enterprise. It’s the printer spooling daemon, lpsched. This daemon waits for print requests, and then sends the print job to the requested printer. The lpsched daemon is always running, waiting for a print job to come in.

Remember that most daemons end with the letter d. Most are started by initialization scripts at boot time, but this is configurable. Daemons can be stopped or restarted as needed. You often have to restart a daemon after making a configuration change to its service, or, to fix a stuck print queue, you might have to stop and start the print spooler daemon.

**Shells**

A shell is the interface between the user and the kernel. It’s the means by which you communicate commands. There are many kinds of shells. This section focuses on three of the most popular: Bourne, C, and Korn.

The **Bourne shell** was written by S. R. Bourne and can be found on nearly all UNIX systems. It lacks some of the bells and whistles of others, but it remains one of the most prevalent shells in UNIX. The shell is located in /bin/sh and uses a dollar sign ($) at the user prompt. Shell scripts are most often written to be run by the Bourne shell because of its excellent input/output control and expression-matching features. However, many people feel it lacks strong interactivity features.

To answer this need, pioneers at the University of California at Berkeley developed the **C shell**. This shell, located in /bin/csh and using a percentage symbol (%) at the user prompt, provides features not found in Bourne, such as job control, command history, and aliasing. The C shell is designed to use command syntax that resembles C, the programming language used to write UNIX. However, many people feel that it lacks strong control of input and output.
In an effort to combine the best of both the Bourne and C shells, David Korn developed the Korn shell, which can be found in /bin/ksh and uses a dollar sign ($) at the command prompt. The Korn shell is a superset of the Bourne shell, but it adds features such as aliasing, history, and the popular command-line editing. These three shells are provided as part of the Solaris operating environment. They are covered in more detail later in Chapter 14, but for now, let’s take a look at how to try a new shell.

**EXERCISE 1-2**

**Changing the Shell**

When you create a user, the default shell assigned to that user is the Bourne shell. Let’s walk through the steps you as the sysadmin can use to change a user’s shell. Once again, we’ll use a Solaris 9 system named enterprise as our example.

1. Log in to the system and find out which shell you are using:

```
Trying 192.168.0.2...
Connected to enterprise.
Escape character is '^]'.
SunOS 5.9
login:gibbst
password: **********
Last login: Tue Apr  4 17:22:10 from :0
Sun Microsystems Inc.   SunOS 5.9
Welcome to enterprise.timgibbs.net
[enterprise: gibbst] $ echo $SHELL
/bin/sh
```

Here we have logged in as the user named gibbst. Using the echo command, you can see that the SHELL variable is set to /bin/sh. This tells you that you are currently using the Bourne shell. Now, let’s change the shell.

2. Switch to the root account:

```
Su -
password: **********
Sun Microsystems Inc.   SunOS 5.9
BUGGER – Not Happy Jan!
[enterprise: root] #
```

Now you are the superuser, or root, of enterprise. Notice how the command prompt has changed from the user prompt dollar sign ($) to the root prompt pound sign (#).
3. Execute the command `passwd -e` to change the shell from the user named `gibbst`:

```
[enterprise: root] # passwd -e gibbst
```

The current shell will be displayed and you will be prompted for the new shell:

```
old shell/bin/sh
New shell:/bin/csh
[enterprise: root] #
```

Now, the next time you log in as the user `gibbst`, you will use the C shell to do so.

---

**On the Job**

*By default, users can’t use the `passwd -e` command to change their shells; only the root user can do so. A user can, however, use another shell by changing the SHELL variable. This can be done by editing the user’s `.profile` file in his or her home directory or from the command line.*

### File Systems

A **file system** is a collection of directories and files organized in a hierarchical structure. A file system has a one-to-one correspondence with physical storage, such as a disk partition, or a virtual device presented by Solstice DiskSuite or Veritas Volume Manager (either of which may contain one or more physical devices). The **root directory** or root (`/`) file system is the absolute parent of all other directories and file systems.

Under the slash (`/`), other file systems are connected to the root file system. Before a disk partition or virtual disk can be accessed, it must first have a file system written on it. The `newfs` command does this for you. Once the partition contains a file system, the `mount` command attaches the new file system to the `mount point`: the location (directory) on the parent file system where a separate file system may be attached. File systems can be identified by the special **lost+found** directory that exists under the

---

<table>
<thead>
<tr>
<th><strong>Scenario &amp; Solution</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>How can you tell which shell you’re using?</td>
</tr>
<tr>
<td>How can you find out more about your shell’s capabilities?</td>
</tr>
</tbody>
</table>
One benefit of creating separate file systems is that data written to one cannot overflow into another. Having /export/home as a separate file system prevents users from filling up the root file system and crashing the server.

The UNIX file system is hierarchical; therefore, files are identified by their full path. Executing a command against the hosts file may produce unpredictable results. However, using the hosts file full path of /etc/inet/hosts guarantees that the operation is applied to the correct file. If you want to use the cp command to copy a file named foobar.tim to your home directory, you have to tell the cp command exactly where the file is located, and exactly where you want it to go:

```
[enterprise: root] # cp /export/home/gibbst/foobar.tim /export/home/tim
```

This is an actual conversation I had with a coworker recently:

```
Me: Hey, find /1st_floor -name “Scott.”
Coworker: cd /1st_floor/breakroom; ls | grep coffee.
```

For this reason, it often seems like UNIX sysadmins are speaking another language.

### Kernels

A **kernel** is the very heart of an operating system. It communicates instructions to the hardware and schedules and executes all the system commands. It manages all the daemons, devices, and system resources such as memory, swap space, and file systems. When you are running a shell program, such as the Bourne shell, you are using that program to communicate with the kernel.

The kernel is loaded during the boot process and must be running for the system to operate. The kernel-specific files can be found in the following directories:

- **/kernel**  This directory contains all the common kernel components that are needed for booting the system.
- **/platform/<platform-name>/kernel**  This directory contains the components that are needed for a specific type of platform. For example, on an Intel system, this directory is /platform/i86pc/kernel.
- **/platform/<hardware-class-name>/kernel**  This directory contains the kernel components specific to this hardware class.
- **/usr/kernel**  This directory contains kernel components common to all platforms within a particular instruction set.
On occasion, you might have to reconfigure the parameters of a Solaris kernel to tweak it for optimal performance—the installation of a database application, for example. Make sure that you have read the instructions carefully and tested the changes on a test system before you do it on a mission-critical system! Without a bootable kernel, you don’t have a bootable system, which could get you booted out the door.

Operating Systems

An operating system is a set of programs that manage all system operations and provide a means of communication between the user and the resources available to that user. An operating system consists of the kernel and the shell.

This chapter already defined the kernel and the shell. To fully understand operating systems, it is also essential that you understand hardware, which is discussed in later chapters.

The definitions of a shell and an operating system are similar. Both are used to communicate instructions, but remember that a shell is a component of an operating system.

Sun Microsystems refers to Solaris 9 as an operating environment. What’s the difference between an operating environment and an operating system? Consider the analogy that both are like a shelter in which you live. An operating system provides you with the bare essentials. It’s like a tent that contains food, a sleeping bag, and a T-1 connection to the Internet. If you’re talking bare essentials, you could probably get by without the sleeping bag. In contrast, an operating environment is like a house with running water, electricity, home entertainment system, walk-in meat locker, hot tub, and so on. An operating environment includes all the bells and whistles that make running the system easier and more fun. Solaris 9 is designed from the ground up as a server operating environment. You won’t find games, cute sound effects, or 3-D wallpapers included with the installation. However, it does have some very handy administrative applications and features, such as these:

- **IPv6 compatibility**  The latest version of Internet Protocol allows for better security and increased available addresses.

- **Solaris smart card**  This allows a sysadmin to add more security by requiring users to be validated with a personal ID card.
PDA synchronization Using this feature, users can transfer contact, calendar, and mail data between their Solaris 9 system and their personal digital assistants (PDAs).

GNU tools Perl, Apache, bash, and a few other GNU free popular utilities are now included with Solaris 9.

X Server This feature has been upgraded to allow for better mobility and ease of use, compatibility with EnergyStar, better power management, and web-based access.

These are just some of the features of the Solaris 9 operating environment that make it an industry standout.

Sun Microsystems announced it would drop the CDE and OpenWindows GUIs in favor of GNOME 2 in Solaris 9, but it may not be ready in time for the Solaris 9 release. Although these changes might affect how users access their workstations, you shouldn’t get bogged down trying to learn how to use one kind of GUI over another. Your worth as a sysadmin will be measured at the command line, not among icons.

Certification Objective 1.02

Defining the Effects of Various man Command Options in Viewing Online Manual Pages

Without a doubt, no matter how far your journey into Solaris takes you, the beacon that you can depend on is the man command.

Short for manual, the man command is the way you access information about the commands and configuration files on your system. It’s not a difficult command to learn or use.

The syntax of the man command is

```
man < command_name>
```

Let’s look at how to use the command to get information on another helpful command, the ls command.
Chapter 1: Understanding System Concepts

EXERCISE 1-3

Using the man Command

1. Log in to the system:

   [voyager: gibbst]$ telnet enterprise
   Trying 192.168.0.2...
   Connected to enterprise.
   Escape character is '^]'.
   SunOS 5.9
   login:gibbst
   password: **********
   Last login: Tue Apr  8 11:44:12 from :0
   Sun Microsystems Inc.   SunOS 5.9
   Welcome to enterprise.timgibbs.net
   [enterprise: gibbst] %

2. Use the man command to get information about the ls command:

   [enterprise: gibbst] % man ls
   Reformatting page. Please Wait... done

   User Commands                                             ls(1)
   NAME
   ls - list contents of directory
   SYNOPSIS
   /usr/bin/ls [ -aAbcCdfGillMnpqrRstux1 ] [ file ... ]
   /usr/xpg4/bin/ls [ -aAbcCdfGillMnpqrRstux1 ] [ file ... ]
   DESCRIPTION
   For each file that is a directory, ls lists the contents of
   the directory; for each file that is an ordinary file, ls
   repeats its name and any other information requested. The
   output is sorted alphabetically by default. When no argument
   is given, the current directory is listed. When several
   arguments are given, the arguments are first sorted
   appropriately, but file arguments appear before directories
   and their contents.
   --More--(10%)
information on its use. The man command also displays the revision history of the
command and, in most cases, its author.

Man pages are marked with either nroff tags or Standard Generalized Markup
Language (SGML). Both of these methods are used to mark a document with tags to
describe how to read the text. For now, let’s just say that they are similar to HTML,
used to tell a web browser how to display a web page. That’s why when you start the
man command, it says, “Reformatting page. Please Wait…done” before displaying
the page. The command reads the document’s tags to display it on your screen
correctly. The man pages are usually kept in /usr/share/man. The man pages using
SGML tags are found in /usr/share/man/sman*; the man pages using nroff tags are
kept in /usr/share/man/man*.

The directories are searched in the order specified in the /usr/share/man/man.cf file.

The man command also includes some handy options you can use:

- `-k` - The keyword switch

You use the `-k` switch to find information using a keyword search. The syntax for
using this option is

```bash
man -k <keyword>
```

For example, say you wanted to find information on viewing documents in the
PDF format. You would type

```bash
man -k PDF
```

Then the man command displays all the commands that deal with PDF documents.
Actually, it displays all the commands that have `PDF` in their man pages. Here’s
another example:

- `-M` - The path command

Use this option if you want the man command to search for a command in a
directory not specified in the /usr/share/man/man.cf file. The syntax for using this
option is

```bash
man -M <path_name>
```

For example, let’s say you’ve loaded an application that has installed its man pages into
a directory not specified by the MANPATH variable or in the /usr/share/man/man.cf
file, but into the /opt/app/man directory. To view information on a command from the man pages in that directory, type the following:

```
man -M /opt/app/man <command_name>
```

This command forces the man command to look only in the named path for the man page on that command.

With the “all” option

```
-a
```

the -a option shows you all the man pages that match the name specified. The syntax is

```
man -a <command_name>
```

The man pages will be searched in the order in which they are specified in the MANPATH variable or as specified in the /usr/share/man/man.cf file. For example, if you type

```
man -a passwd
```

you will be able to scroll through first the man page for the passwd command, and then the man page for the passwd file.

To search for files, use the -f option.

```
-f
```

The -f option gives you a brief summary of man pages that fit the name of the file you specify. The syntax is

```
man -f <file_name>
```

For example, let’s say you want to see which man pages exist for the passwd command. You would type

```
man -f passwd
```

and you would see the following:

```
passwd    passwd (1)   - change login password and password attributes
passwd    passwd (4)   - password file
```

This output tells you that there is not only a passwd command but a man page on the passwd configuration file as well.
To do a section search, use

```
-s <section number>
```

Now that you know that there’s more than one man page on passwd, how do you display the one you want? Commands sometimes have more than one version of a man page. These different versions are separated into sections. As you’ve seen, there are two versions of a man page for passwd. By default, the man command displays the man page on the passwd command, not the man page on the /etc/passwd configuration file. The `-s` option allows you to specify a section of the available man pages for the specified command, for example,

```
man -s 4 passwd
```

This command displays the man page that deals with the passwd file, not the command. You saw that the passwd file was in the (4) section of the manual when we used the `-f` option.

---

**FROM THE CLASSROOM**

**UNIX Isn’t Always UNIX**

I have had the opportunity to work with a wide variety of UNIX operating systems in my career. The variety of UNIX flavors out there can be confusing, but they share many similarities. I often hear fellow UNIX sysadmins use the phrase, “UNIX is UNIX,” meaning that if you know one kind, you know them all. In some ways that’s true, but for the most part, it’s not. Every brand of UNIX has its own unique way of doing things. Solaris is no different.

To prepare for these exams, make sure you use the man command on a Solaris system often—even on simple, often-used commands. A switch or option that performs one task on one kind of UNIX might not perform the same task on another. The man pages are like the hidden history of UNIX. Many of them are virtually the same as they were when they were first added to a UNIX server. However, they are updated to reflect any changes and pertain to the version of UNIX on which they’ve been installed. By browsing through man pages on a Solaris system, you’ll be able to see all the available options for the commands on that system.

—Randy Cook, SCSA
CERTIFICATION SUMMARY

This chapter covered the basic terminology you’ll need to get started with the material in the rest of the book. It’s important that you completely understand the terms before moving on to the next chapter. You also learned about one of the most helpful tools you’ll ever need in your career as a sysadmin: the man command. The man command is one of the tools you should use regularly to prepare for these exams.
TWO-MINUTE DRILL

Defining System Administration Terms

- A daemon is a process that waits and listens for a request. The lpsched daemon waits for print requests and sends them to the correct printer.

- A shell is the program used to communicate your commands or the commands of an application or script to the kernel. Three basic shells are installed with Solaris 9: Bourne, Korn, and C.

- A file system is a collection of files and directories organized in a hierarchical structure.

- A kernel is the master program that communicates the requests of applications to the system hardware. It also manages all devices, memory, and processes.

- The operating system is a set of programs that govern all operations and acts as a means for the user to communicate instructions to system resources.

Defining the Effects of Various man Command Options in Viewing Online Manual Pages

- The man command provides several options to allow for specialized searching and displaying of the online manuals provided with Solaris.

- The \(-k\) switch is used for searching for keywords.

- The \(-M\) switch allows you specify a different path to a man page.

- The \(-a\) switch allows you to view all the available pages on a command.

- The \(-f\) option allows you to search and view brief summaries.

- The \(-s\) option allows you to view specified sections of available man pages for a command.
Chapter 1: Understanding System Concepts

SELF TEST

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully because there might be more than one correct answer. Choose all correct answers for each question.

Defining System Administration Terms

1. You’ve run the command ps -ef. Which of the following is an example of a daemon?
   A. /etc/hosts
   B. /usr/dt/bin/dtscreen -mode pyro
   C. /sbin/sh
   D. /usr/lib/nfs/lockd

2. A new user logs in to a system and sees a dollar sign ($) at the command prompt. Which shell is this user using?
   A. The Money shell
   B. The C shell
   C. The Korn shell
   D. The Bourne shell

3. Where did the C shell get its name?
   A. It’s short for Complete shell.
   B. It’s short for Complicated shell.
   C. Its syntax is based on the C programming language.
   D. It doesn’t stand for anything.

4. What is the correct way for a user to change his or her shell from the following choices?
   A. Use the passwd -shell <username> command.
   B. Use the passwd -e <username> command.
   C. A user can’t change his or her shell.
   D. Change the SHELL variable.
5. Which of the following is an example of a file’s name?
   A. /etc/default
   B. /etc/hosts
   C. hosts
   D. /usr/man/man.cf

6. Which statement is true about a kernel?
   A. Once Solaris is installed, it’s not required.
   B. It’s a good idea to delete it occasionally to make sure it doesn’t get too big.
   C. It is loaded during the booting of the system.
   D. It gets its name from the fact that, like a military officer, it’s in charge of everything.

**Defining the Effects of Various man Command Options**

**in Viewing Online Manual Pages**

7. The man command can be used to search for specific keywords. Which command would search for the keyword `login`?
   A. man login
   B. man -a login
   C. man -k login
   D. man -s login

8. The man command can be used to display all the available man pages on a command. Which of the following commands displays all the man pages on the name `passwd`?
   A. man -all passwd
   B. man | grep passwd
   C. man -a passwd
   D. man -a passwd

**LAB QUESTION**

Solaris has a way or three to do just about anything. But what if you are trying to perform a particular action and don’t know which command to use? For example, how would you find out which command to use to display the name of the system into which you’re logged? How would you use the man command to help you find the command you need?
SELF TEST ANSWERS

Defining System Administration Terms

1.  ✔ D. This is one of the daemons used to provide NFS service. Daemons are often recognized by having the letter *d* at the end.  
    ✗ A is incorrect because this is the host file on the system, which you probably wouldn’t see if you ran the *ps -ef* command. B indicates that the screen saver is running on the system—that really cool fireworks one. C indicates that the Bourne shell is running.

2.  ✔ C or D. Although the user is new, the default system shell can be whatever the sysadmin set it to when the user account was created. Out of the box, the Bourne shell is the default system shell.  
    ✗ B is incorrect because the C shell uses a percent sign (%) at the user prompt. A is incorrect because, as of now, there is no UNIX shell called the Money shell.

3.  ✔ C. The wizards at Berkeley wanted a shell that would work like programming in C, so they made one.  
    ✗ B and A are incorrect because the C shell is neither complete nor complicated. D is also incorrect.

4.  ✔ D. The correct way for a user to change his or her shell is by changing the SHELL variable in the user’s .profile.  
    ✗ A is incorrect because there is no *-shell* option with the passwd command and a user won’t have access to the passwd command. Same for B; although there is an *-e* option, only the root user has access to the passwd command. C is also incorrect.

5.  ✔ B. This answer describes the complete pathname of a file, which is how UNIX names files.  
    ✗ A is incorrect because it describes the location of a directory. C is incorrect because it could be a directory or a filename. D could be correct if you moved or copied the man.cf file to this location, but the default location of the man.cf file is /usr/share/man/man.cf.

6.  ✔ C. A kernel is loaded during the booting of the system.  
    ✗ A and B are incorrect because the kernel is what keeps the system running. D is incorrect because the name *kernel* has nothing to do with the military rank of colonel. Besides, everybody knows it’s sergeants that really run things.
Defining the Effects of Various man Command Options in Viewing Online Manual Pages

7. **✓** C. This displays the man pages that have the keyword login in their description.
   - A is incorrect because it displays the man page for login. B is incorrect because the `-a` option displays all pages upon login. D is incorrect because the `-s` option is used to display sections.

8. **✓** D. This command displays all the man page manuals that cover the passwd command, one after another.
   - A is incorrect because there is no `-all` option. B is incorrect because it doesn’t follow the man command syntax. C is incorrect because “passwd” is misspelled.

LAB ANSWER

1. Use the man command to search for a keyword:

   ```bash
   # man -k name
   ```

   Did it show you the command you’re looking for? Possibly, but it’s three or four screens of commands. How can you narrow your search? You can’t use more than one keyword. Let’s try another keyword. Because you’re trying to determine the host’s name, or hostname, try using that as a keyword.

2. Narrowing the search:

   ```bash
   # man -k hostname
   ```

   Is that a more manageable list? It’s considerably smaller than the first one. The following command can determine the system’s fully qualified hostname:

   ```bash
   check-hostname check-hostname (1m) - check if sendmail
   ```

   gethostname gethostname (3c) - get or set name of current host
   gethostname gethostname (3xnet) - get name of current host
   hostname hostname (1) - set or print name of current host system
   sethostname sethostname (3c) - get or set name of current host

   In addition, you can see that one of the choices is a command called hostname. What were the odds? Now let’s see how the hostname command works before you run it.

3. Investigate a command:

   ```bash
   # man hostname
   ```
You can see from the output of the command that just by typing the hostname command, you will see the name of the system displayed. It’s a good thing that you checked it out first, because the hostname command can also be used to change the hostname of the system.

This is an example of using the man command to help jog your memory or to research your options among all the available commands on a system. Naturally, you’re unlikely to have root access to a system and not know its hostname or how to find it. This was just a simple exercise to demonstrate the use of the keyword search capabilities of the man command.