Part I

Sun Certified System Administrator for Solaris 8
Study Guide, Exam 310-011

CHAPTERS

1. Understanding System Concepts
2. Administering the Boot PROM
3. Installing the Solaris 8 Operating Environment
4. Initialization and Shutdown
5. User Administration
6. Solaris File Security
7. Understanding Process Control
8. Working with Files Systems
9. Working with Files and Directories
10. Understanding the Boot Process
11. Configuring Disks
12. Working with the Format Command
13. Performing Backup and Recovery
14. Utilizing Basic Command Syntax
15. Working with the vi Editor
16. Working with a Remote Connection
Understanding System Concepts

CERTIFICATION OBJECTIVES

1.01 Defining system administration terms

1.02 Defining the effect of various man command options in viewing online manual pages

☑️ Two-Minute Drill

Q&A Self Test
In order to truly understand any machine, you must first understand its function and parts. Then you must understand how those parts work together in order to perform the machine’s function.

In this chapter, we touch on all three topics—function, parts, and how the parts work together—in very general terms. The rest of the book goes into very specific detail, but first we need to define 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

In this section, we want to introduce 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.) We define a *daemon* 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 word literally means *divinity*. The poet William Butler Yeats had a fascination with the concept of daemons and their representation of the duality of existence. He described them as beings that represent the opposing sides of human nature. You can’t know what good is with out evil’s existence. How could you recognize a great piece of pie, for example, unless you’ve experienced a not-so-great one? Actually, 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 a very 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. There are daemons
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. You can use the ps command to display
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. We cover the ps
command in more detail in later chapters, but for now, let’s take a look at some of
the daemons that are running on your system right now.

**EXERCISE 1-1**

**Viewing the Processes on a System**

Let’s first take a look at what is going on behind the scenes on our Solaris system.
For this exercise, our system’s hostname is sol.

1. Log on to the system:

   Trying 192.168.1.10...
   Connected to sol.
   Escape character is '^]'.
   SunOS 5.8
   login:ra
   password: **********
   Last login: Tue Apr 3 19:12:13 from :0
   Sun Microsystems Inc. SunOS 5.8
   Welcome to Sol - This system will be down for 4 hours this
   Sunday for upgrades. Call ext. 3649 if you have any questions.
   [sol: ra] $

2. List the processes.

   Now we’ve logged in to sol as the user ra and seen a message from the very
   conscientious system administrator regarding a planned outage. Next, let’s look at
   the processes, including the daemons, that are currently running on sol by using the
   following ps command.
Chapter 1: Understanding System Concepts

We used the ps command with the `e` and `f` options. This command displayed every process in a full listing. Since the list would easily cover more than one screen, we piped (|) the display into the more command. This way we can see one screen at a time, and by pressing the SPACEBAR, we see more of the display. We cover the ps command in more detail later. For now, let’s go daemon hunting!

The output of the ps command includes the following line, which is highlighted in the preceding code:

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

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

---More---

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, many kinds of shells. We focus on three of the most popular: Bourne, C, and Korn Shells.
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 at /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 felt it lacked strong interactivity features.

To answer this need, pioneers at the University of California at Berkeley developed the C Shell. This shell, located at /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. The Korn Shell can be found at /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. We cover these shells in more detail later in the book, but for now let’s take a look at how we can 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 8 system named sol as our example.

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

```
Trying 192.168.1.10...
Connected to sol.
Escape character is '^]'.
SunOS 5.8
login: ra
password: ********
Last login: Tue Apr 5 17:22:10 from :0
Sun Microsystems Inc. SunOS 5.8
```
Chapter 1: Understanding System Concepts

Welcome to Sol - Thanks for your cooperation during our upgrade process. This system now has two additional CPU’s, 4 more gigs of RAM and a new UPS. Call ext. 3649 if you have any questions.

[sol: ra] $ echo $SHELL
/bin/sh

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

2. Switch to the root account:

    password: **********
    Sun Microsystems Inc. SunOS 5.8
    ALL HAIL THE SUPERUSER!!!!!
    [sol: root] #

Now, you are the superuser, or root, of sol. Notice how the command prompt has changed from the user prompt $ to the root prompt #.

3. Execute the command passwd -e to change the shell for the user named ra:

    [sol: root] # passwd -e ra

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

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

Now, the next time you log in as the user ra, 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 ‘/’ file system is the absolute parent of all other directories and file systems.

Under ‘/’, 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 us. Once the partition contains a file system, the mount command attaches the new file system to the **mount point**. A **mount point** is 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 mount point for every file system. 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 ‘/’ 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 wanted to use the **cp** command to copy a file named badda-bing.employee.list 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.”

[sol: root] # cp /export/home/silvio/badda-bing.employee.list /export/home/tony
Chapter 1: Understanding System Concepts

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 in order 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 in order 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 following components:

- The kernel
- The shell

We’ve already defined the kernel and the shell in this chapter. To fully understand operating systems, it is also essential that you understand hardware, which will be discussed in later chapters.

**The definition 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 8 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 we live. An operating system provides us with the bare essentials. It’s like a tent that contains food, a sleeping bag, and a T-1 connection to the Internet. If we’re talking bare essentials, we 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 are included to make running the system easier and more fun. Solaris 8 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 include some very handy administrative applications and features, such as:

- **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 8 system and their personal digital assistants (PDAs).
Inclusion of GNU Tools  Perl, Apache, bash and few other GNU free
popular utilities are now included with Solaris 8.

The 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 8 operating environment that
make it an industry stand-out.

Sun Microsystems recently announced it would drop the CDE and
OpenWindows GUIs in favor of GNOME. There’s also been talk of including
KDE as well. 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 Effect 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.

The command man, short for manual, 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 we use the command to get information on another helpful
command, the ls command:
EXERCISE 1-3

Using the man Command

1. Log on to the system:

   [luna: ra]$ telnet sol
   Trying 192.168.1.10...
   Connected to sol.
   Escape character is '^[].'
   SunOS 5.8
   login:ra
   password: **********
   Last login: Tue Apr  8 11:44:12 from :0
   Sun Microsystems Inc.   SunOS 5.8
   Welcome to Sol - Don't forget to log out when you're done.
   Call ext. 3649 if you have any questions.
   [sol: ra] %

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

   [sol: ra] % man ls
   Reformatting page. Please Wait... done

   User Commands
   ls(1)

   NAME
   ls - list contents of directory

   SYNOPSIS
   /usr/bin/ls [ -aAbcCdfFgilLmnopqrRstux1 ] [ file ... ]
   /usr/xpg4/bin/ls [ -aAbcCdfFgilLmnopqrRstux1 ] [ 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%)
Chapter 1: Understanding System Concepts

The output of the man command is shown one screen at a time; the preceding excerpt shows the contents of the first screen. As you see, the man command shows you a brief description of the command, its syntax, and more detailed 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 in order 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:

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

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

```
man -k PDF
```

Then the man command displays all the commands that dealt with PDF documents. Actually, it displays all the commands that have **PDF** in their man pages. In 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:

   man -M <path_name>

For example, let’s say that 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 directory /opt/app/man. To view
information on a command from the man pages in that directory, type:

   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,
then the man page for the passwd file.
To search for files:

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

\begin{verbatim}
man -f <file_name>
\end{verbatim}

For example, let's say that you want to see what man pages exist for the \texttt{passwd} command. You would type:

\begin{verbatim}
man -f passwd
\end{verbatim}

And you would see the following:

\begin{verbatim}
passwd passwd (1) - change login password and password attributes
passwd passwd (4) - password file
\end{verbatim}

This output tells you that there is not only a \texttt{passwd} command but a man page on the \texttt{passwd} configuration file as well.

To do a section search, use:

\begin{verbatim}
-s <section number>
\end{verbatim}

Now that you know that there's more than one man page on \texttt{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 we've seen, there are two versions of a man page for \texttt{passwd}. By default, the man command displays the man page on the \texttt{passwd} command, not the man page on the /etc/passwd configuration file. The \texttt{-s} option allows us to specify a section of the available man pages for the specified command. For example:

\begin{verbatim}
man -s 4 passwd
\end{verbatim}

This command displays the man page that deals with the \texttt{passwd} file, not the command. We saw that the \texttt{passwd} file was in the (4) section of the manual when we used the \texttt{-f} option.
CERTIFICATION SUMMARY

In this chapter, we covered the basic terminology you’ll need to get started with the material in the rest of the book. It’s important that you have a complete understand of them before moving on to the next chapter. We’ve also covered 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 on a regular basis to prepare for these exams.

FROM THE CLASSROOM

UNIX Isn’t Always UNIX

I have had the opportunity to work with a wide variety of UNIX OSs 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 can be considered 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 that you use the man command on a Solaris system often—even on simple, often-used commands. A switch or option that does one thing on one kind of UNIX might not do the same thing 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
TWO-MINUTE DRILL

Defining System Administration Terms

- A daemon is a process that waits and listens for a request. The lpsched daemon waits for prints 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 8: the Bourne Shell, the Korn Shell, and the C Shell.
- 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 Effect 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.
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 `$` at the command prompt. What 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 mean anything.

4. What is the correct way to for a user to change his or her shell from the following choices?
   A. Using the `passwd -shell <username>` command.
   B. Using the `passwd -e <username>` command.
   C. A user can’t change his or her shell.
   D. By changing the SHELL variable.

5. Which of the following is an example of a file’s name?
   A. `/etc/default`
20 Chapter 1: Understanding System Concepts

B. /etc/hosts
C. hosts
D. /usr/man/man.cf

6. Which statement is true in regard to 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 Effect 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 what command to use? For example, how would you find out what 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 % at the user prompt. A is incorrect since, 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, and the C Shell is neither complete or complicated. D is also incorrect.

4. ☑ D is correct. The correct way for a user to change his or her shell is by changing the SHELL variable in the users’ .profile.
   ☒ A is incorrect because there is no -shell option with the password 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 since the running of 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.
Chapter 1: Understanding System Concepts

Defining the Effect 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 on login. D is incorrect since 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 since 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 we narrow our search? We can’t use more than one keyword. Let’s try another keyword. Since we’re trying to determine the host’s name, or hostname, let’s 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
   
   ```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     gethostname (3c)    - get or set name of current host
   ```
   
   In addition, we 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 we run it.
3. Investigate a command:

```bash
# man hostname
```

We can see from the output of the command that just by typing the command `hostname`, we will see the name of the system displayed. It’s a good thing that we checked it out first, since 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, it’s not likely that you’d 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.