CHAPTER 5, LAB 1: INTRODUCTION TO THE SHELL (15 MINUTES)

LEARNING OBJECTIVES AND OUTCOMES

In this lab you will learn about the terminology, structure, and processing of the command line including arguments, options, builtins, corrections, and pathname expansion.

READING

Read "The Command Line" starting on Sobell, page 126, up to the section titled "Simple Commands" on page 130.

PROCEDURE

1. On a command line, what is a *token?* What is an *argument* and how does it usually affect a command? What is an *option* and how does it usually affect a command? How can you usually distinguish an argument from an option?
2. Review the previous labs. List one or more command lines in each of the following categories: a command line with zero arguments, one argument, two arguments, one option, and one option and one argument.

3. Most utilities have many options. You can read the man page or use the **––help** option to learn which options a utility accepts. Experiment with the ls **–r** (reverse) option and try combining it with the **–l** option. Try using the cp **–r** (recursive) option to copy a directory hierarchy. Use the head ***–n*** (substitute a number for ***n***) option to display the first ***n*** lines of a file instead of the default 10; try the same option with tail. Many utilities accept the **––version** option to display version and license information. Experiment with this option on some of the utilities you are familiar with.
4. What is a builtin (Sobell, page 153)? How does a builtin differ from a utility? The **builtins** man page describes the bash builtins. Which builtins have you used so far?
5. and The echo builtin copies its arguments to the screen. Given the following command line and its output, how can you repeat the command line without retyping it (Sobell, page 31)?

```
$ echo hi there
hi there
```

After giving the preceding command, how can you edit the command to replace **hi** with **hello** (Sobell, page 31)?

6. Using pathname expansion (Sobell, page 148), list the files in the **/usr/bin** directory that have the characters **ab** anywhere in their names.

   List the files in the **/usr/bin** directory that begin with the letter **u**. Next list those that begin with **un**.

   List the files in **/usr/bin** that have names that are one character long.

   List the files in your home directory that begin with a period followed by the letters **bash** (**.bash**).

DELIVERABLES

This lab introduces you to command-line terminology and has you practice using options, builtins, command-line editing, and pathname expansion. When you use script to capture your work in the lab, you can turn the resulting file in to your instructor.

CHAPTER 5, LAB 2: STANDARD INPUT AND STANDARD OUTPUT (15 MINUTES)

LEARNING OBJECTIVES AND OUTCOMES

In this lab you will use cat, echo, and ls to learn about standard input and standard output, including learning how to use cat to create a short file.

READING

Read about standard input and standard output starting on Sobell, page 133, up to the section on noclobber on page 139.

PROCEDURE

1. "The Keyboard and Screen as Standard Input and Standard Output" on page 135 of Sobell explains how to use the cat utility to read from standard input (defaults to the keyboard) and write to standard output (defaults to the screen). Use cat as shown in Figure 5-5 on page 135 of Sobell to copy standard input to standard output. Press CONTROL-D on a line by itself to terminate cat.
2. The echo builtin copies its arguments to standard output which, by default, bash directs to the screen.

   ```
   $ echo This message goes to standard output.
   This message goes to standard output.
   ```

   The cat utility sends the contents of a file specified by its argument to standard output.

   ```
   $ cat /etc/hosts
   127.0.0.1               localhost.localdomain localhost
   ::1             localhost6.localdomain6 localhost6
   ```

   Redirect standard output (Sobell, page 136) of echo to write a short message to a file and then use cat to display the contents of the file.
3. As demonstrated in step 1, when you do not specify an argument for cat, cat copies standard input to standard output. Redirect standard input to cat to come from the file you created in the previous step. Do not redirect standard output from cat; it will appear on the screen.

Some utilities can take input either from a file or from standard input

tip The cat utility belongs to a class of utilities that can accept input from a file given as an argument (step 2) or, if you do not specify an argument, from standard input (steps 1 and 3). Refer to "Utilities that take input from a file or standard input" on Sobell, page 138.

4. Redirect the output of an **ls –l** command to a file named **ls.out**. Display **ls.out** using cat.
5. The who utility (Sobell, page 70) lists the users who are logged in on the system. Append the output (Sobell, page 140) of the who utility to the **ls.out** file you created in the previous step. Display the augmented **ls.out** file using cat.

   What happens if you omit one of the greater-than signs (use **>** in place of **>>**)? Try it and see.
6. Redirect standard output of cat to create a file named **days** that holds the names of the days of the week in chronological order, one per line. Do not redirect standard input to cat; it will come from the keyboard. Remember to press CONTROL-D on a line by itself to exit from cat.

   Use cat to read the **days** file and send it to standard output, through a pipeline, to standard input of the sort (Sobell, pages 58 and 143) utility. The result will be a list of days in alphabetical order.

   Replace sort in the preceding command with grep (Sobell, page 56) with an argument of (uppercase) **T**. The result will be a list of days that have an uppercase **T** in their names in chronological order.

   Next create a filter (Sobell, page 144) by repeating the preceding command but sending standard output of grep through a pipeline to standard input of sort. The result will be a list of days that have an uppercase **T** in their names in alphabetical order.
7. Produce a long listing of the **/etc**, **/usr/bin**, and **/sbin** directories, sending the output to a file and running the command in the background (Sobell, page 146).

DELIVERABLES

This lab gives you practice redirecting standard input and standard output on the command line. When you use script to capture your work in the lab, you can turn the resulting file in to your instructor.