

Fundamentals of UNIX
Lab 7.1.6– Redirection and Piping
(Estimated time: 45 min.)

Objectives:

- Become familiar with input / output (I/O) redirection
- Redirect standard output to create a File
- Prevent overwriting files with redirection
- Append output to an existing file
- Pipe the output from one command to another

Background:

In this lab, you will use advanced UNIX commands to accomplish **redirection** and **piping**. Every UNIX command has a source for **standard input** and a destination for **standard output**. The input to a command is normally from the **keyboard**, although it can come from a file. The output from a command normally goes to the **monitor** or screen. The UNIX computing environment enables command I/O to be controlled using **redirection**. This is useful when attempting to save the output of a command to a file for later viewing. By **piping**, you can take the output from one command and use it as input to another command for further processing.

Tools / Preparation:

- a) Before starting this lab, review Chapter 7, Section 1 – Advanced Directory and File Management Using the Command Line.
- b) You will need the following:
 1. A login user ID (e.g. user2) and password assigned by your instructor.
 2. A computer running the UNIX operating system with CDE
 3. Networked computers in classroom with class file system installed

Notes:

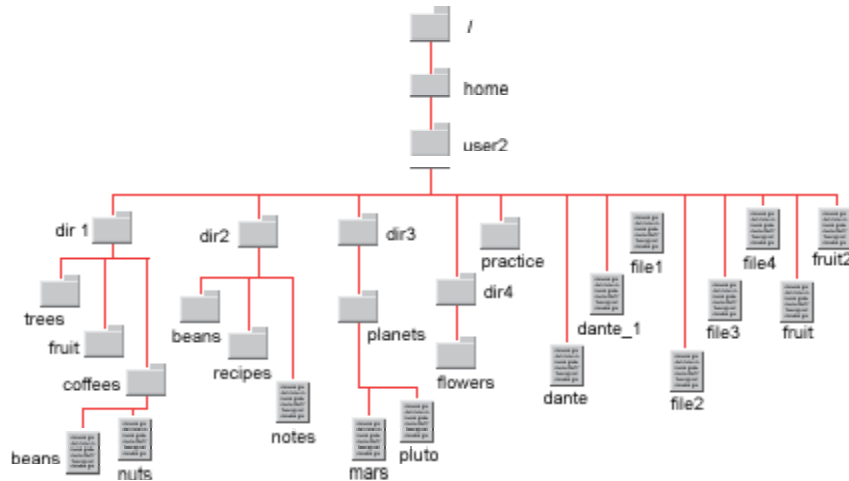
Fundamentals of UNIX

Lab 7.1.6– Redirection and Piping

Worksheet

Use the diagram of the sample Class File System directory tree to assist with this lab.

Class File Tree Structure



Step 1. Log in to CDE

Login with the user name and password assigned to you by your instructor in the CDE entry box.

Step 2. Access the Command Line

Right click on the **workspace** backdrop and click on **Tools**. Select **Terminal** from the menu to open a terminal window.

Input / Output Redirection - Overview

There are several **metacharacters** used as **redirection** symbols. Output redirection uses the **right angle bracket (>)**, which is also referred to as the greater-than symbol. Input redirection uses the **left angle bracket (<)** or the less-than symbol. Error output redirection uses the **right angle bracket preceded by the number two (2>)**. This lab will focus on output redirection.

General format: *Command* *Redirection-Symbol* *File* (text file or device file)

Step 3. Redirect Standard Output to Create a File

Standard output is redirected much more frequently than standard input or standard error. Many commands, such as **ls**, **cat**, **head**, and **tail** generate standard output to the screen and it is frequently desirable to redirect this output to a file for future viewing, manipulation or printing. By substituting a file name, you can capture the output of a command rather than letting it go to the default monitor. This is a good way to create a sizable test file for practice.

Fundamentals of UNIX
Lab 7.1.6– Redirection and Piping
Worksheet – Cont.

The **right angle bracket (>)** or greater-than symbol allows the command to send output to a file. Using the single right angle bracket will create a new file if the file name specified does not exist. If the file name exists it will be overwritten. Note: the spaces between the command, the redirection symbol and the file name are optional.

Command Format: `command > file`

a. Verify that you are in your home directory. What command did you use? _____ What command would you use to change to your home directory, if you were not there already?

b. You wish to capture a listing of files and directories to a file in your home directory, thus allowing you to keep track of what is in your home directory. What command would you use to redirect the output of a **long file listing** and create a new output file called **homedir.list**

c. Where was this new file **homedir.list** placed? _____ Use the **ls** command to verify that the new file is present.

d. What command could you use to view the contents of the file you just created ONE page at a time?

e. Capture the **first 10** lines of the **homedir.list** file using the **head** command and create a new file called **dhomedir.list-top-10** using redirection. What command did you use?

_____ View the contents of the file using the **more** command.

f. Capture the **last 10** lines of the **homedir.list** file using the **tail** command and create a new file called **dhomedir.list-bot-10** using redirection. What command did you use?

_____ View the contents of the file using the **more** command.

g. Capture the output from the **cal 2002** command. View the contents of the file. What have you captured? _____

Step 4. :Prevent Overwriting Files with Redirection

In the **Korn** shell, an option called **noclobber** can be set to prevent overwriting of files during redirection.

This can be done on the command line by using **\$ set -o noclobber**.

To re-enable clobbering, use **\$ set +o noclobber**.

To enable/disable clobbering with the C Shell: **% set noclobber** and **% unset noclobber**.

a. Enter the command to turn on the noclobber on with the Korn shell. What command did you enter?

b. Enter the command: **ls -l > homedir.list** What was the result?

c. Enter the command: **ls -l > homedir.list2** What was the result?

Fundamentals of UNIX
Lab 7.1.6– Redirection and Piping
Worksheet – Cont.

Step 5. Append Output to an Existing File

The **double right angle bracket (>>)** can be used if you wish to append (add to the end) to an existing file instead of overwriting it. This option creates a new file if one does not exist or appends to an existing one.

Command Format: *command >> file*

a. Enter the command to display a banner that says: **Happy Bday** and use the **double redirection** symbols to capture the output to a file called **bday4me**. What command did you use?

b. Enter the command display a banner that says: **YOURNAME!** (whatever your name is) and use the **double redirection** symbols to append the output to the **bday4me** file. View the contents of the **bday4me** file. What is in it? _____

c. Enter the command to display the calendar for the **month** and **year** you were born (If you were born in June 1965, enter: **cal 6 1965**) and use the **double redirection** symbols to append the output to the **bday4me** file. What command did you use? _____

d. View the contents of the **bday4me** file. Note that the output from three commands has been combined in the **bday4me** file. What day of the week were you born on?

Step 6. Pipe from One Command to Another

One of the most powerful metacharacters is the **pipe (|)**. The pipe takes the **standard output** of one command and passes it as **standard input** into a following command. The **pipe symbol** is sometimes referred to as a **double vertical bar** and is found below the **backspace** key. You **must** always have a command on each side of a pipe. Spaces between the commands and the pipe are optional.

Command Format: *command | command*

a. Use the pipe metacharacter to send the output of the **ls -l** command as input to the **more** command. What happened as a result of piping the output to the **more** command?

b. Look at the files listed with the **ls -l | more** command and note some of the dates created or modified. To see a listing of files or directories that were created or modified in the same month, you can use the **grep** command to search for that month. Specify the month exactly as it is displayed in the listing. (e.g. **Oct**). Enter this command: **ls -l | grep Oct** (or whatever month you are looking for). What was the result?

c. Directories always have a size of 512 bytes. Enter the command: **ls -l | grep 512**. What was in the resulting listing?

Fundamentals of UNIX
Lab 7.1.6– Redirection and Piping
Worksheet – Cont.

d. You can use multiple pipes to connect multiple commands. Enter a command that will take the output of the **long file listing** and pipe it to the **tail** command and then to the **sort** command. What command did you enter?

e. The **ps** (process status) command is used to see what processes are running a UNIX system. Pipe output of the **ps -e** command to the **more** command. The **-e** option will show you **every** process running on the system. What happened as a result of piping the output to the more command? _____

Step 7 – Remove Files and Directories Created in this Lab

Refer to the Class file system tree structure and **remove all files** and **directories** created in you home directory during this lab (including those creating under the practice directory).

Step 8. Close the Terminal Window and Logout

Double click on the dash button in the upper left corner of the screen, then click the **EXIT** icon on the front panel.