

CHAPTER 4

Pipes and Filters

About this Chapter: This chapter explains the usage of various types of pipes and filters available in LINUX. We shall also see how we can look for a particular pattern in a file and extract those lines only using the grep command.

Objectives

- ☛ Introduction to Pipes and Filters
- ☛ Usage of sort and grep Filters
- ☛ Usage of Other Filters

Introduction to Pipes and Filters

In LINUX, commands are used to perform only a specific task. It is not possible for the user to perform multiple tasks in one command. Redirection provides an answer to this problem. However, redirection creates a lot of temporary files that are redundant and occupy disk space. Pipes and filters are used to overcome this obstacle.

Pipes

A pipe is a mechanism in which the output from one command can be redirected as input to another command. For example,

Example 4.1

```
$ who | wc -l  
16  
$ _
```

In the above example, the output of the command "who" is passed as the input to the "wc -l" command and the result is displayed.

Pipes are considered as temporary unnamed files, which store the output of one command in memory and passes it as the input to the next command. The important advantage of pipes is that they prevent the user from making temporary files using I/O redirection. Since the temporary files are stored only in memory, they save space.

Example 4.2

```
$ cat text | head -3
```

The output of the above command will be the display of the first three lines of the file named text.

Filters

A filter accepts input from the Standard Input, processes the input and then sends the output to the Standard Output. Filters also take input from a file.

Filters are used to extract the lines, which contain a specific pattern, to arrange the contents of a file in a sorted order, to replace the existing characters with other characters, etc. Filters are also used to store the intermediate results of a long pipe. We can extract specific columns of a file and can merge two or more files together using filters.

A. Points to Ponder**State True or False**

1. LINUX commands can be used to perform multiple tasks in a single command.
2. Output of one command can be taken as input for the next command using pipes.
3. Pipes are temporary files stored in memory.
4. Particular portion of a file can be extracted using filters.
5. Filters are used to merge the contents of a file with another file.

Fill in the blanks

1. The output of the command `ls | wc -l` will be _____.
2. Filters take input from the _____ as well as the _____.
3. Filters can be used to store the _____ result in a pipe.
4. Redirection creates many _____ files.

Usage of sort and grep Filters

The sort and grep commands help us in customizing the output using various commands.

The sort Filter

The sort filter arranges the input taken from the Standard Input in the alphabetical order. The options available with sort are `-r`, `-f`, `-n`, `-b`, `-t` etc.

The following examples elucidate the usage of the various options. Type the values as given below and observe the result.

Example 4.3

```
$ sort
alice
jennifer
dorothy
Ctrl+d
alice
dorothy
jennifer
$ _
```

The above example illustrates the usage of sort command. The input is accepted from the keyboard and the result is displayed in alphabetical order.

The -r Option

When the `-r` option is used with the sort command it will display the input taken from the keyboard in the reverse alphabetical order.

Example 4.4

```
$ sort -r
alice
jennifer
dorothy
Ctrl+d
jennifer
dorothy
alice
$ _
```

The -f Option

As a rule the digits, alphabets and other special characters that are taken in as input are converted to their ASCII value. Then sort arranges the input according to their ASCII value. The ASCII values for A to Z are lesser than that of a to z. Hence there might be unpredictable results.

When the sort command is used with the -f option, the case distinction is ignored and the output is arranged in alphabetical order.

Example 4.5

```
$ sort -f
Valentino
vialli
gibson
Gonzalez
Ctrl+d
gibson
Gonzalez
Valentino
vialli
$_
```

The -n Option

We have seen that the sort command arranges the numbers, alphabets and special characters according to their ASCII value. The ASCII value of 10 for example is lesser than that of 2 and hence 10 will be displayed above 2, which is not the correct order.

The sort -n option will arrange the input according to numerical value and then display it.

Example 4.6

```
$ sort -n
9
17
2
Ctrl+d
2
9
17
$_
```