

# PIPES & FILTER – UNIX SHELL PROGRAMMING

Dr. T.Logeswari

Associate professor, Department of Computer Science

New Horizon College

Kasturai Nagar

Bangalore

## Abstract

In this research paper, the concept of Pipes and Filter is discussed and various aspects of filter commands also studied. Many Commands in Unix produce output on the terminal screen or can take input from the keyboard. Actually these commands have been written to accept input from standard input file and produce output on a standard output file. Generally the standard input files are set to keyboard and the output file to the terminal screen. These special files are streams of character seen as input and the output by the commands

**Keywords:** cat, grep, sort, paste.

## I. Introduction

A shell script is a computer program to design to run by the unix shell, a command line interpreter. The various dialects of shell scripts are considered to be scripting languages[1][5]. The most influential Unix shells have been the Bourne shell and the C shell. These shells have both been used as the coding base and model for many derivative and work-alike shells with extended feature sets. The Bourne shell, sh, was written by Stephen Bourne at AT&T as the original Unix command line interpreter; it introduced the basic features common to all the Unix shells, including piping, command substitution, variables, control structures for condition-testing and looping and filename wildcarding. The language, including the use of a reversed keyword to mark the end of a block, was influenced by ALGOL 68. The C shell, csh, was written by Bill Joy while a graduate student at University of California, Berkeley. The language, including the control structures and the expression grammar, was modeled on C. The most generic sense of the term shell means any program that users employ to type commands[2]. A shell hides the information of the primary operating system and manages the procedural details of the operating system kernel line, which is the lowest-level, or "inner-most" component of most operating systems. In Unix-like operating systems, users typically have many choices of command-line interpreters for interactive sessions. When a user logs in to the system interactively, a shell program is mechanically executed for the duration of the session. In this research paper various standard files, redirection operator and filter commands are discussed with example[6].

## II. Standard Files

There are three file representing three streams each associated with a default device. i.e standard Input, standard output, Standard Error[3].

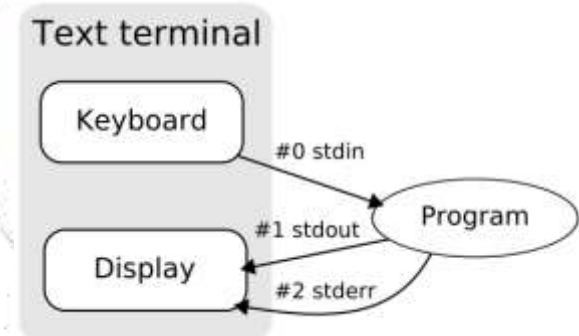


Figure 1. standard streams for Input, Output & Error

### a. I/O Redirection

Redirection is normally implemented by placing certain character between commands. The input source is redirected using < (less than) operator.

For example \$cat<file1

The cat command read the input from file1 and displays it on the terminal screen. The input therefore is not from the keyboard, but from another file

### b. Pipe and pipeline

Program can run together such that one program reads the output from another without using an intermediate file[4]

## c. Filters

Filters are UNIX commands that take their inputs from the standard input file, process it and send it to the standard output. A filter is actually a program that can read directly from files whose names are provided as argument, process it or filter it and send the results to the standard output, unless a redirection or a pipeline symbol is used. It won't process or filter & send the results to the standard output unless a pipeline (|) is used. Some commonly used filter commands are sort, cut, paste, grep, tr, wc, uniq, etc[7][8].

## i) sort

The sort command is used to sort the information in a text file in ascending or descending order. It is also used to merge sorted files.

## Sort options

-m : merges two or more given sorted files into one sorted output file

-n : sorts according to numeric order

-r : sorts in reverse order

-o outfile : Places the sorted output in the file

-d : Sorts according to dictionary & ignores the punctuation

-f : Ignores while sorting

-c : Can be used to check whether a file is sorted or not.

## Example:

Steps to create a file using cat command :

```
$cat > filename
```

```
$cat >stdlist
```

Steps to display a file using cat command :

```
$cat stdlist
```

```
001 :Deepali : 90
```

```
002 : Vidya : 68
```

```
011 : Mohena : 80
```

```
006 : Robert :72
```

```
003 : Krithika : 88
```

```
$ sort stdlist
```

```
001 :Deepali :98
```

```
002 : Vidya : 68
```

```
003 : Krithika : 88
```

```
006 : Robert :72
```

```
011 : Mohena : 80
```

- Merging of two files can be performed by the command :  
\$ sort -m -o namelist list1 list2 , where the files list1 & list 2 are merged into one namelist.

## ii)cut

The cut command splits files vertically. This command can be used to extract the required fields or columns from the file. Each field is separated by a 'delimiter'

## Example:

```
$ cut -c1 -3 stdlist
```

```
001
```

```
002
```

```
011
```

```
006
```

```
003
```

cut options :

-c list : The list after -c specifies the character positions

-f list : The list after -f is a list of fields separated by a delimiter character

-d list : The character following -d is the field delimiter

## Example:

```
$ cut -d ':' -f 1,3 stdlist
```

```
001 : 90
```

```
002 : 68
```

```
011 : 80
```

```
006 : 72
```

```
003 : 88
```

## iii) paste

The paste command is used to join contexts of two files vertically. The contexts of two files can be viewed side by side by pasting them

## Example :

```
$ cat empno $cat empname $cat phone
```

```
0100 Gautam 23236549
```

```
0102 Rithish 27496536
```

```
0103 Sonam 28407862
```

```
$ paste empnoempname phone >empdet
```

```
$cat empdet
```

```
0100 Gautam 23236549
```

```
0102 Rithish 27496536
```

```
0103 Sonam 28407862
```

## iv) uniq

When duplicate entries are found in a file, they can be removed from the file by using the uniq command. This file takes only one sorted file as its argument

## uniq options:

-c : Precedes each output line with a count of the number of times the line occurred in the input

-d : Selects only one copy of the duplicate entries

-u : Selects only the lines which are not repeated

## Example

```
$cat names
```

```
Venky
```

```
Neelima
```

```
Rupali
```

```
Rupali
```

```
Vidya
```

```
Diya
```

```
Vidya
```

- \$uniq names

```
Venky
```

```
Neelima
```

```
Rupali
```

```
Vidya
```

```
Diya
```

- \$uniq -c names

```
1 Venky
```

```
1 Neelima
```

2 Rupali  
 2 Vidya  
 1 Diya  
 ▪ \$uniq -d names  
 Rupali  
 Vidya  
 ▪ \$uniq -u names  
 Venky  
 Neelima  
 Meenu

## v) wc

wc command is a filter used to count the number of lines, words & characters of one or more files. It takes one or more filenames as its arguments. It gives the output in 4 columns, the first column indicates the number of lines, second indicates the number of words, third indicates the number of characters & last column shows filename.

## wcoptions :

-l : prints the line count  
 -L : prints the length of the longest line  
 -m : prints the number of characters  
 -w : counts words delimited by the white space characters or new line characters.

## Example:

```
$ wcemplist
O/P : 5 19 90 emplist
$ wc -l emplist
O/P : 5 emplist
$ wc -L
O/P : 17 emplist
$ wc -m emplist
O/P : 90 emplist
$ wc -w emplist
O/P : 19 emplist
```

## vi) tr

tr command is used to translate characters. It is a filter that manipulates individual characters in a line  
 The general syntax is :  
 tr [ options ] [ expression1 ] [ expressions 2 ] < input file

Where tr takes the input file & replaces the expression1 with expression2.

## Example:

```
$ cat list 1
A friend in need is a friend in deed.
$ tr 'frie' 'FRI' < list 1
A FRIIndInnId Is a FRIIndInnId
Here, all f are replaced by F
all r are replaced by R
all I are replaced with I
```

## troptions :

-c : complements the set of characters specified by the first string  
 -d : Deletes all occurrences of input characters that are specified by the first string  
 -s : Replaces repeated characters with a single character.

## vii) cat

It is used to display contents of a file , create a new file , append data to an existing file & concatenate two or more files.

## cut options :

-n : Displays each line output with its line number at the beginning  
 -b : Numbers the lines , like -n but omits the line numbers of blank lines  
 -u : The output is not buffered. The default is buffered output.

## Example :

```
$cat empdet
0100 Sowmya 23246549
0102 Nikhil 24496523
0103 Deepthi 2840786665
```

## viii) grep

The grep command searches its input for a pattern and displays lines containing the pattern, the line numbers or the file names where the pattern occurs.

## Syntax is :

```
$grp [ options ] [ pattern ] [ filename ]....
```

## Example :

```
$cat stdlist
001 : Deepak : 92 : BCA
002 : Vishal : 78 : BCA
003 : Payal : 80 : BSc
004 : Peter : 60 : BSc
005 : Gopal : 45 : BCA
```

## Grep options :

-c : It is the count option  

```
$grep -c 'BSc' stdlist
2
```

 -n : This option is used to display the line numbers of matching records along with the record.  

```
$grep -n 'y' stdlist
stdlist : 3 : 003 : Payal : 80 : BSc
```

 -v : This option is known as the inverse option  

```
$grep -v BCA stdlist
003 : Aman : 80 : BSc
004 : Akhila : 60 : BSc
```

## REFERENCES

- [1] <http://linuxshellscripting.blogspot.in/2011/02/shell-scripting-advantages-and.html>
- [2] <http://www.thegeekstuff.com/2010/07/execute-Shell-script/>
- [3] [http://en.wikipedia.org/wiki/Shell\\_script](http://en.wikipedia.org/wiki/Shell_script)
- [4] <http://en.wikipedia.org/wiki/Filters>
- [5] <http://www.calpoly.edu/~rasplund/script.html>
- [6] <http://supportweb.cs.bham.ac.uk/docs/tutorials/doc-system/build/tutorials/unixscripting/unixscripting.htm> [7] Jayalakshmi ' Unix Shell Programming' ISBN: 978-93-5267-892-1, 2017.
- [8] Shivangi Shandilya, Surekha Sangwan, Ritu Yadav" SHELL SCRIPTING AND SHELL PROGRAMMING IN UNIX" IJIRT | Volume 1 Issue 11 | ISSN: 2349-6002,2014