UNIT 8: LINUX FILTER COMMANDS

UNIT STRUCTURE

8.1 Learning Objectives

8.2 Introduction

8.3 Filter Commands

- 8.3.1 wc
- 8.3.2 head
- 8.3.3 tail
- 8.3.4 cut
- 8.3.5 tr
- 8.3.6 top
- 8.3.7 grep

8.4 File Compression and System Monitoring Commands

- 8.4.1 tar
- 8.4.2 unzip
- 8.4.3 kill
- 8.4.4 netstat

8.5 File Access Permission Commands

- 8.5.1 chmod
- 8.5.2 chown
- 8.5.3 chgrp

8.6 Let Us Sum Up

8.7 Further Readings

8.8 Answer to Check Your Progress

8.9 Model Questions

8.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- understand the use of wc command with various options
- learn head, tail, cut, tr and top commands
- search words in a file using the grep command

- understand to compress and decompress files
- work with various file access permission commands like chmod, chown and chgrp

8.2 INTRODUCTION

In the previous unit, we have discussed various file handling commands with examples. We have also discussed various commands for Creating and Viewing ordinary files. In this unit, we will discuss various filter commands and commands for creating and viewing ordinary files. In the last part of this unit, we will discuss various file access permission commands also.

8.3 FILTER COMMANDS

When a program takes its input from another program, it performs some operation on that input, and writes the result to the standard output. It is referred to as a filter. A large numbers of filter commands are available in Linux. In the next sections, we will discuss these commands with examples.

8.3.1 wc Command

The wc command displays the number of lines, number of characters, and the number of words in a file.

Example: To display the number of lines, words, and bytes from a file.

wc <filename>

```
localhost:~# wc kkhsou.txt
      31      933     6227 kkhsou.txt
localhost:~#
```

Example: Display the complete count information of multiple files.

wc <file1><file2>

```
localhost:~# wc hello.c kkhsou.txt
       7       10       76 hello.c
      31      933     6227 kkhsou.txt
      38      943     6303 total
localhost:~#
```

The above command displays the number of lines, words, and the number of the bytes from the files 'hello.c' and 'kkhsou.txt'. It also displays the total number of lines, words, and the number of the bytes of the two files.

Example: Display the number of lines in a file.

wc - l filename

```
localhost:~# wc -l kkhsou.txt
31 kkhsou.txt
localhost:~#
```

Example: Display the number of characters in a file

wc -m kkhsou.txt

```
localhost:~# wc -m kkhsou.txt
6215 kkhsou.txt
localhost:~#
```

Example: Display the number of bytes in a file

```
localhost:~# wc -c kkhsou.txt
6227 kkhsou.txt
localhost:~#
```

Example: Display the number of words in a file

wc -w kkhsou.txt

```
localhost:~# wc -w kkhsou.txt
933 kkhsou.txt
localhost:~#
```

Example: Display the total number of files and folders in a directory

ls | wc -l

```
localhost:~# ls | wc -l
8
localhost:~#
```

Example: Display the length of the longest line

wc -L kkhsou.txt

```
localhost:~# wc -L kkhsou.txt
941 kkhsou.txt
localhost:~#
```

8.3.2 head Command

The 'head' command displays the starting content of a file. By default, it displays starting 10 lines of any file.

head <filename>

```
localhost:~# head hello.c
#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}

localhost:~#
```

Example: Display the first 10 lines of multiple files

head <file1><file2>

```
localhost:~# head hello.c readme.txt
==> hello.c <==
#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}


==> readme.txt <==
Some tests:

- Compile hello.c with gcc (or tcc):

  gcc hello.c -o hello
  ./hello

- Run QuickJS:

  qjs hello.js
localhost:~#
```

In the above screenshot, we have seen that the lines are displayed from two files named 'hello.c' and 'readme.txt' with separate headings. (Since, the two files contains less than 10 lines, it displays the lines less than 10)

Example: Display specific number of lines from a file.

head -n filename

```
localhost:~# head -3 hello.c
#include <stdio.h>
int main(void)
{
localhost:~#
```

Example: Display specific number of bytes of a file.

head -c <number><filename>

```
localhost:~# head -c 220 kkhsou.txt
Education is the backbone of a society. In common parlance education is any act or experience that has
 a formative effect on the mind, character or physical ability of an individual. In continuing with th
e aim of buildinlocalhost:~#
```

In the above example, we have displayed the 220 bytes of content from the file 'kkhsou.txt'.

8.3.3 tail Command

Linux tail command is used to display the last ten lines of one or more files.

tail <filename>

```
localhost:~# tail test.txt
11
12
13
14
15
16
17
18
19
20
localhost:~#
```

Example: Display specific number of lines from last.

tail -n <number><file name>

```
localhost:~# tail -5 test.txt
16
17
18
19
20
localhost:~#
```

Example: Display specified number of bytes from the last.

tail -c <number><file name>

```
localhost:~# tail -10 test.txt
11
12
13
14
15
16
17
18
19
20
localhost:~#
```

From the above output, the last six bytes of the file 'test.txt' are displayed.

Example: Display the first 10 lines of multiple files

tail <file1><file2>

```
localhost:~# tail hello.c test.txt
==> hello.c <==
#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}


==> test.txt <==
11
12
13
14
15
16
17
18
19
20
localhost:~#
```

8.3.4 cut Command

It is used to cut a specific section by byte position, character, and field and writes them to standard output. To cut a specific section, it is necessary to specify the delimiter and it can be a space (' '), a

hyphen (-), a slash (/), or anything else. After '-f' option, the column number is to be mentioned.

- To cut by using the hyphen (-) as the delimiter, execute the below command:

  cut -d- -f(columnNumber) <fileName>

  Let us explain with an example.

```
localhost:~# cat >> student_marks.txt
Rajib-90
Pranjit-67
Pradip-45
Pritam-78
Prakash-89
localhost:~# cut -d- -f2 student_marks.txt
90
67
45
78
89
localhost:~# cut -d- -f1 student_marks.txt
Rajib
Pranjit
Pradip
Pritam
Prakash
localhost:~#
```

Here, we have created a new file 'student_marks' and added the content of the marks obtained by students using 'cat' command. Then, we have displayed only the marks using the 'cut' command. In the next section, we have displayed the name of the students only.

- The '-b' option is used to cut a section of line by byte. To cut a file by its byte position, execute the command as follows:

  cut -b <byte number> <file name>

```
localhost:~# cat student_marks.txt
Rajib-90
Pranjit-67
Pradip-45
Pritam-78
Prakash-89
localhost:~# cut -b 5 student_marks.txt
b
j
i
a
a
localhost:~#
```

8.3.5 tr Command

The command 'tr' stands for 'translate'. It is used to translate, like from lowercase to uppercase and vice versa.

command | tr <'old'> <'new'>

```
localhost:~# cat student_marks.txt
Rajib-90
Pranjit-67
Pradip-45
Pritam-78
Prakash-89
localhost:~# cat student_marks.txt | tr 'rai' 'RAI'
RAjIb-90
PRAnjIt-67
PRAdIp-45
PRItAm-78
PRAkAsh-89
localhost:~#
```

8.3.6 top Command

It is used to display all the running process within the environment of a system. It helps in monitoring system usage and performances. It is mainly used to detect load on the server by system administrators.

Type the following command:

top

```
Mem: 5868K used, 181376K free, 8K shrd, 0K buff, 872K cached
CPU:   0% usr   0% sys   0% nic  96% idle   0% io   0% irq   1% sirq
Load average: 0.00 0.00 0.00 1/32 151
  PID  PPID USER     STAT   VSZ %VSZ CPU %CPU COMMAND
  151    61 root     R     1516   1%   0   1% top
   61     1 root     S     1552   1%   0   0% sh -l
   97    61 root     T     1524   1%   0   0% cat
    1     0 root     S     1512   1%   0   0% {init} /bin/sh /sbin/init
   56     1 root     S     1260   1%   0   0% dhcpcd -q
   55     1 root     S      744   0%   0   0% settime -d /
    7     2 root     SW       0   0%   0   0% [ksoftirqd/0]
   15     2 root     SW       0   0%   0   0% [kworker/0:1]
   62     2 root     SW       0   0%   0   0% [kworker/u2:1]
   10     2 root     SW<      0   0%   0   0% [writeback]
   11     2 root     SW       0   0%   0   0% [kcompactd0]
   12     2 root     SW<      0   0%   0   0% [crypto]
   13     2 root     SW<      0   0%   0   0% [bioset]
   14     2 root     SW<      0   0%   0   0% [kblockd]
    6     2 root     SW<      0   0%   0   0% [mm_percpu_wq]
   16     2 root     SW       0   0%   0   0% [kswapd0]
   17     2 root     SW<      0   0%   0   0% [bioset]
   34     2 root     SW       0   0%   0   0% [khvcd]
   35     2 root     SW<      0   0%   0   0% [bioset]
   36     2 root     SW<      0   0%   0   0% [bioset]
   37     2 root     SW<      0   0%   0   0% [bioset]
   38     2 root     SW<      0   0%   0   0% [bioset]
   39     2 root     SW<      0   0%   0   0% [bioset]
   40     2 root     SW<      0   0%   0   0% [bioset]
    8     2 root     SW       0   0%   0   0% [kdevtmpfs]
    4     2 root     SW<      0   0%   0   0% [kworker/0:0H]
```

8.3.7 grep Command

The 'grep' command stands for "global regular expression print". It filters the content of a file which makes our search easy.

- The 'grep' command is generally used with pipe (|).

Syntax: command | grep <searchWord>

Example: cat student_marks.txt | grep 9

```
localhost:~# cat student_marks.txt
Rajib-90
Pranjit-67
Pradip-45
Pritam-78
Prakash-89
localhost:~# cat student_marks.txt | grep 9
Rajib-90
Prakash-89
localhost:~#
```

- The 'grep' command can also be used without the pipe (|).

Syntax: grep <searchWord> <file name>

Example: grep 9 student_marks.txt

```
localhost:~# cat student_marks.txt
Rajib-90
Pranjit-67
Pradip-45
Pritam-78
Prakash-89
localhost:~# grep 9 student_marks.txt
Rajib-90
Prakash-89
localhost:~#
```



CHECK YOUR PROGRESS

Q 1: The syntax to display the complete count information of multiple files is _____

Q 2: _____ command displays the number of lines in a file.

Q 3: wc -w kkhsou.txt command displays the number of _____ of the file 'kkhsou.txt'

Q 4: _____ command can be used to display the starting 10 lines of any file.

Q 5: _____ command is used to translate from lowercase to uppercase and vice versa.

Q 6: To search word in a file, _____ command is used.

8.4 FILE COMPRESSION AND SYSTEM MONITORING COMMANDS

We often need to compress files and directories to save space without compromising the quality. In Linux, few handy commands are available to achieve the task. In the next sections, we will discuss those commands. We will also discuss how to work with some system related commands like kill and netstat in the next sections.

8.4.1 tar Command

The tar command stands for tape archive. This command is used for creating Archive and extracting the archive files.

- Making an uncompressed tar archive with -cvf option

  The command is:

  tar cvf file.tar *.txt

```
localhost:~# tar cvf file.tar *.txt
readme.txt
student_marks.txt
test.txt
localhost:~#
```

The above example makes a tar file known as file.tar. It is the archive of every .txt file inside the current directory.

- Extracting files through the archive with -xvf option

  The command is:

  tar xvf file.tar

```
localhost:~# tar xvf file.tar
readme.txt
student_marks.txt
test.txt
localhost:~#
```

- gzip compression over tar archive with -z option

This option makes a tar file known as file.tar.gz. It is the archive of every .txt file.

The command is as follows:

tar cvzf file.tar.gz *.txt

```
localhost:~# tar cvzf file.tar.gz *.txt
readme.txt
student_marks.txt
test.txt
localhost:~# ls
bench.py          file.tar.gz       hello.js          student_marks.txt
file.tar          hello.c           readme.txt        test.txt
localhost:~#
```

- Extracting the gzip tar archive with -xvzf option

This option can extract the files through file.tar.gz tar archived files.

The command is as follows:

tar xvzf file.tar.gz

```
localhost:~# tar xvzf file.tar.gz
readme.txt
student_marks.txt
test.txt
localhost:~# ls
bench.py          file.tar.gz       hello.js          student_marks.txt
file.tar          hello.c           readme.txt        test.txt
localhost:~#
```

8.4.2 unzip Command

To compress a file, the zip command can be used as follows:

zip myfile.zip filename.txt

To unzip the file, the following command is used:

unzip myfile.zip

8.4.3 kill Command

It is used for manually terminating the processes. The command is:

type -a kill

The kill command transfers a signal towards a process in which further terminates the process. We can use the kill command along with the -l option for getting the list of every available signal:

kill -l



8.4.4 netstat Command

netstat command stands for Network statistics. It displays information about different interface statistics, including open sockets, routing tables, and connection information.

The basic syntax of the netstat command is as follows:

netstat

- Installation of the netstat command

If the netstat command is not installed on your machine, it will display the traditional Linux installation error message "Command 'netstat' not found."

To install it, execute the below command:

sudo apt install net-tools

The above command will ask for the administration password to install the command.

netstat -a

The above command will list all the existing connections.

```
localhost:~# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
netstat: /proc/net/tcp6: No such file or directory
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
netstat: /proc/net/udp6: No such file or directory
netstat: /proc/net/raw6: No such file or directory
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING       2930 /run/dhcpcd.sock
unix  2      [ ACC ]     STREAM     LISTENING       2932 /run/dhcpcd.unpriv.sock
unix  2      [ ]         DGRAM                      2961
localhost:~#
```

8.5 FILE ACCESS PERMISSION COMMANDS

In linux, three popular file access permission commands are used. They are (a) chmod (b) chown (c) chgrp. In the next sections, we will discuss how they can be used.

8.5.1 chmod Command

Linux chmod command is used to change the access permissions of files and directories. chmod stands for change mode. In the Linux, each file is associated with a particular owner and have permission access for different users. The user classes may be:

- owner
- group member
- Others (Everybody else)

The file permissions in Linux are the following three types:

- read (r)
- write (w)
- execute (x)

Let's see how to change the file permission using the chmod command.

The basic syntax of chmod command is as follows:

chmod <options> <permissions> <file name>

To view the permission of all text file from the current working directory, execute the below command:

```
localhost:~# ls -l *.txt
-rw-r--r--    1 root     root          151 Jul  5  2020 readme.txt
-rw-r--r--    1 root     root           51 Oct  8 13:54 student_marks.txt
-rw-r--r--    1 root     root           51 Oct  8 13:09 test.txt
localhost:~#
```

To set the read and write permission for other users, execute the below command:

chmod o+w *.txt

It will set the read and write permission for other users of the text files. Consider the below output:

```
localhost:~# ls -l *.txt
-rw-r--r--    1 root     root          151 Jul  5  2020 readme.txt
-rw-r--r--    1 root     root           51 Oct  8 13:54 student_marks.txt
-rw-r--r--    1 root     root           51 Oct  8 13:09 test.txt
localhost:~# chmod o+w *.txt
localhost:~# ls -l *.txt
-rw-r--rw-    1 root     root          151 Jul  5  2020 readme.txt
-rw-r--rw-    1 root     root           51 Oct  8 13:54 student_marks.txt
-rw-r--rw-    1 root     root           51 Oct  8 13:09 test.txt
```

Numerical Shorthand

We can use the numeric values instead of letters to specify the permissions. A three-digit value is used to specify the permission. The leftmost digit represents the owner (u), and the middle digit represents the group members (g). The rightmost digit represents the others (o).

The following table represents the digits and their permissions:

Digits	**Permissions**
000	No permission
001	Execute permission
010	Write permission
011	Write and execute permissions
100	Read permission
101	Read and execute permissions
110	Read and write permissions
111	Read, write, and execute permissions

8.5.2 chown Command

The chown stands for change owner. chown command is used to change a file's ownership, directory, or symbolic link for a user or group.

- Display the UID, GID, and Groups

To display all the groups, execute the groups command as follows:

groups

To display the UID and GID, execute the id command as follows:

id

```
localhost:~# id
uid=0(root) gid=0(root)
localhost:~#
```

- Display the User and Group Ownership of a File

To display the user and group owner of a file, execute the "ls -l" command with a specific file name. Consider the below command:

ls -l hello.c

The above command will display the user and group ownership of the 'hello.c' Consider the below output:

```
localhost:~# ls -l hello.c
-rw-r--r--    1 root     root           76 Jul  3  2020 hello.c
localhost:~#
```

- Change the owner of a File (Using user name)

To change the owner of a file, pass the user name (new owner) with the chown command as follows:

sudo chown <username> <File name>

- Change the Owner of a File (Using UID)

We can change the owner of a file or directory using the UID of a user. To change the ownership by using the UID, pass the UID with chown command as follows:

sudo chown <userid> <File name>

- Change the Group of a File

To change the group of a file, specify group name or GID

with the chown command. Execute the chown command specifying the group name separated by a colon as follows:

sudo chown :<groupname> <filename>

or

sudo chown :<GID> <filename>

8.5.3 chgrp Command

chgrp command is used to change the group ownership of a file or directory.

- Change the Group Ownership of a File

To change the group ownership of a file, execute the command as follows:

sudo chgrp groupname filename

- Change the group ownership of a folder

To change the group ownership of a folder, execute the 'chgrp' command with the folder name as follows:

sudo chgrp groupname directoryname



8.6 LET US SUM UP

- The wc command displays the number of lines, number of characters, and the number of words in a file.
- The 'head' command displays the starting content of a file. By default, it displays starting 10 lines of any file.
- tail command is used to display the last ten lines of one or more files.
- Cut command is used to cut a specific section by byte position, character, and field and writes them to standard output.
- tr command is used to translate, like from lowercase to uppercase and vice versa.
- Top command is used to display all the running process within the environment of a system.
- grep command is used to search a keyword in a file.
- tar command is used for creating Archive and extracting the archive

files.

- kill command is used for manually terminating the processes.
- Netstat command displays information about different interface statistics, including open sockets, routing tables, and connection information.
- chmod command is used to change the access permissions of files and directories.
- chown command is used to change a file's ownership, directory, or symbolic link for a user or group.
- chgrp command is used to change the group ownership of a file or directory.



8.7 FURTHER READINGS

1) Das, S. (2000).UNIX: Concepts and Applications. Tata McGraw-Hill.
2) Das, S. (2012). Your UNIX/Linux: The Ultimate Guide. (Third Edition). (Science/Engineering/Math). McGraw-Hill.
3) Christopher Negus (2012). Linux Bible. (8th Edition). Wiley.
4) https://www.javatpoint.com/linux-tutorial



8.8 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: wc <file1><file2>

Ans to Q No 2: wc - l filename

Ans to Q No 3: word

Ans to Q No 4: head <filename>

Ans to Q No 5: tr

Ans to Q No 6: grep



8.9 MODEL QUESTIONS

Q 1: Explain the Linux wc command with various options.

Q 2: Differentiate between head and tail command with examples.

Q 3: What are the functions of cut command?

Q 4: Explain the function of (a) tr and (b) top commands .

Q 5: Explain the grep command with examples.

Q 6: How will you compress and un-compress files using tar and untar commands? Explain with examples.

Q 7: How will you compress and un-compress files using zip and unzip commands? Explain with examples.

Q 8: For what purpose kill command is used?

Q 9: Explain the netstat command.

Q 10: Explain the chmod command with examples.

Q 11: Explain the chown command with examples.

Q 12: Explain the chgrp command with examples.

*** ***** ***