UNIT 7: LINUX FILE HANDLING COMMANDS

UNIT STRUCTURE

7.1 Learning Objectives

7.2 Introduction

7.3 File and Directory Handling Commands

- 7.3.1 ls
- 7.3.2 cd
- 7.3.3 cp
- 7.3.4 mv
- 7.3.5 rm
- 7.3.6 mkdir
- 7.3.7 rmdir
- 7.3.8 pwd

7.4 Commands for Creating and Viewing Ordinary Files

- 7.4.1 cat
- 7.4.2 more
- 7.4.3 less

7.5 Let Us Sum Up

7.6 Further Readings

7.7 Answer to Check Your Progress

7.8 Model Questions

7.1 LEARNING OBJECTIVES

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

- learn various file and directory handling commands
- understand commands for creating and viewing ordinary files

7.2 INTRODUCTION

In the previous unit, we have discussed the organisation of the Linux file structure. We have also discussed various disk related commands. In this unit, we will discuss various File and Directory Handling Commands

with examples. We will also discuss the commands for creating and viewing ordinary files in the subsequent sections.

7.3 FILE AND DIRECTORY HANDLING COMMANDS

As you get started out in Linux, you'll spend a great deal of time interacting with files and directories. In the following sections, we will discuss various file and directory management commands that are used mostly by a system administrator or a normal Linux user.

7.3.1 ls

The ls is the list command in Linux. It shows the full list or content of your directory.

```
localhost:~# ls
bench.py    hello.c     hello.js    readme.txt
localhost:~#
```

Various options can be used along with the ls command. Let us discuss the options below:

- **ls -a command**

In Linux, hidden files start with a dot (.) and can't be seen in the regular directory. The ls -a command displays the whole list of a directory including the hidden files also.

```
localhost:~# ls -a
.             .ash_history  .mozilla     bench.py     hello.js
..            .cache        .wine        hello.c      readme.txt
localhost:~#
```

- ls -l command

It displays the list in a long list format.

```
localhost:~# ls -l
total 16
-rw-r--r--    1 root     root           114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root            76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root            22 Jun 26  2020 hello.js
-rw-r--r--    1 root     root           151 Jul  5  2020 readme.txt
localhost:~#
```

- ls -lh command

Size of the file is very difficult to read when displayed in terms of bytes. This command displays the data in terms of Mb, Gb, Tb, etc.

```
localhost:~# ls -lh
total 16K
-rw-r--r--    1 root     root         114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root          76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root          22 Jun 26  2020 hello.js
-rw-r--r--    1 root     root         151 Jul  5  2020 readme.txt
localhost:~#
```

- ls -lhS command

This command displays the files in descending order (highest at the top) according to their size.

```
localhost:~# ls -lhS
total 16K
-rw-r--r--    1 root     root         151 Jul  5  2020 readme.txt
-rw-r--r--    1 root     root         114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root          76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root          22 Jun 26  2020 hello.js
localhost:~#
```

- ls -r

This command displays the list in reverse order.

```
localhost:~# ls -r
readme.txt  hello.js    hello.c     bench.py
localhost:~#
```

- ls -R

This command displays the subdirectories also.

```
localhost:~# ls -R
.:
bench.py    hello.c     hello.js    readme.txt
localhost:~#
```

- ls -lX

This command groups the files with same extensions together in the list.

```
localhost:~# ls -lX
total 16
-rw-r--r--    1 root     root          76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root          22 Jun 26  2020 hello.js
-rw-r--r--    1 root     root         114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root         151 Jul  5  2020 readme.txt
localhost:~#
```

- ls - lt

This command sort the list by displaying recently modified filed at

top.

```
localhost:~# ls -lt
total 16
-rw-r--r--    1 root     root           151 Jul  5  2020 readme.txt
-rw-r--r--    1 root     root           114 Jul  5  2020 bench.py
-rw-r--r--    1 root     root            76 Jul  3  2020 hello.c
-rw-r--r--    1 root     root            22 Jun 26  2020 hello.js
localhost:~#
```

- ls -~

It gives the contents of home directory.

```
localhost:~# ls ~
bench.py    hello.c     hello.js    readme.txt
localhost:~#
```

- ls ../

It displays the contents of parent directory.

```
localhost:~# ls ../
bin    etc    lib    mnt    proc   run    srv    tmp    var
dev    home   media  opt    root   sbin   sys    usr
localhost:~#
```

7.3.2 cd

This command is used to change the directory. For example, if we want to change the current directory to the directory KKHSOU2, we have to type cd KKHSOU2

- To navigate into the root directory, use "cd /"
- To navigate to your home directory, use "cd" or "cd ~"
- To navigate up one directory level, use "cd .."
- To navigate to the previous directory (or back), use "cd -"

7.3.3 cp

This command is used to copy a file or a directory.

For example, to copy the file hello.c to hello_new.c in the same directory, we have to type cp hello.c hello_new.c

```
localhost:~# cp hello.c hello_new.c
localhost:~# ls
KKHSOU2       KKHSOU_NEW    hello.c       hello_new.c
KKHSOU3       bench.py      hello.js      readme.txt
localhost:~#
```

To copy a file or a directory to a different location, we have to mention the path of the destination directory.

- To copy multiple file or directories in a directory, we have to type cp file1 file 2 directory name

  For example, to copy the files hello.c and hello_new.c to the directory KKHSOU_NEW, the following command can be used.

```
localhost:~# cp hello.c hello_new.c KKHSOU_NEW
```

- cp -r command is used to copy a directory along with its sub directories.

  For example, the directory KKHSOU_NEW now contains two files named hello.c and hello_new.c. Let us copy the directory KKHSOU_NEW with all the files to the directory KKHSOU2 which is at present empty.

```
localhost:~# cp -r KKHSOU_NEW KKHSOU2
```

Now, to check the content of the directory KKHSOU2, we have to change the directory using cd command.

```
localhost:~# cp -r KKHSOU_NEW KKHSOU2
localhost:~# cd KKHSOU2
localhost:~/KKHSOU2# ls
KKHSOU_NEW
localhost:~/KKHSOU2#
```

7.3.5 rm

rm stands for remove and this command is used to remove a file. We have to be very careful while using this command as it permanently removes a file.

rm filename

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

In the above example, you can see that we have removed/ deleted the file 'file.tar.gz' using the rm command.

- rm -r

With rm '-r' option, you can delete a directory having sub directories inside it. So you don't need to delete sub-directories manually.
The syntax is: rm -r directoryname

- rm -i

The option 'i' stands for 'interactivity'. Before deleting a file it will ask you whether you want to delete it or not. The syntax is:

rm -i filename

Similarly, to delete a directory 'interactively', the following command is used:

rm -ri directoryname

7.3.6 mkdir

To create a new directory, mkdir command is used. The mkdir stands for 'make directory'. For example, to create a new directory named KKHSOU, you have to type mkdir KKHSOU

```
localhost:~# mkdir KKHSOU
localhost:~#
```

The directory KKHSOU is created. To check whether the directory is created or not, type the ls command.

```
localhost:~# ls
KKHSOU      bench.py    hello.c     hello.js    readme.txt
localhost:~#
```

We can create multiple directories with a single command. Suppose, we want to create directories named KKHSOU1, KKHSOU2 and KKHSOU3

So, we have to type mkdir KKHSOU1 KKHSOU2 KKHSOU3

```
localhost:~# mkdir KKHSOU1 KKHSOU2 KKHSOU3
localhost:~# ls
KKHSOU      KKHSOU2     bench.py    hello.js
KKHSOU1     KKHSOU3     hello.c     readme.txt
localhost:~#
```

Note: By default, your directory will be created in your current

directory. If you want to create your directory somewhere else, then provide the path of your destination directory and your file will be created there.

7.3.7 rmdir

To delete a directory, rmdir command is used. But it can delete an empty directory only. If the directory contains sub-directories, it will not be able to delete the directory.

For example, to delete the directory KKHSOU, we have to type rmdir KKHSOU

```
localhost:~# rmdir KKHSOU
localhost:~# ls
KKHSOU1     KKHSOU3     hello.c     readme.txt
KKHSOU2     bench.py    hello.js
localhost:~#
```

7.3.8 pwd

This command displays the current working directory. It displays the whole path starting from the root ending to the directory.

```
localhost:~/KKHSOU2# pwd
/root/KKHSOU2
localhost:~/KKHSOU2#
```



CHECK YOUR PROGRESS

Q 1: _____ command displays the whole list of a directory including the hidden files.

Q 2: To display the whole list of content of a directory including the sub-directories also _____ command is used.

Q 3: To display the content of the parent directory, _____ command is used.

Q 4: _____ command is used to copy a directory along with its sub directories.

Q 5: _____ command displays the current working directory.

7.4 COMMANDS FOR CREATING AND VIEWING ORDINARY FILES

7.4.1 cat

The 'cat' command is one of the most frequently used commands in Linux. Let us discuss the various uses of the cat command.

- The 'cat' command can be used to display the content of a file. For example, to display the content of the file hello.c, we have to type

  cat hello.c

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

localhost:~#
```

To display the content of multiple files at once, type file names in one single line like "cat file1 file2 file3... filen".

For example, if we want to display the content of the files named bench.py and hello.c, we have to type the command as

cat bench.py hello.c

```
localhost:~# cat bench.py hello.c
def fib(n):
    if n < 2:
        return n
    return fib(n-2) + fib(n-1)

n = 25
print("fib(", n, ")= ", fib(n))
#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}

localhost:~#
```

- The 'cat' command can be used to create a new file with greater

than sign (>).

For example, to create a new file named "file1.txt", we have to type the command

cat > file1.txt and press 'enter'. You will be directed to the next line. Press 'enter' after every line and you will be directed to the next line. To save your file, go to the next line, press 'ctrl+d' and your file will be saved.

```
localhost:~# cat > file1.txt
```

- The 'cat' command with double greater than sign (>>) append (add something in the last of a file) something in your already existing file.

```
localhost:~# cat >> file1.txt
This is line no1.
Welcome to KKHSOU.
```

Here, we have added two lines in the already existing file file1.txt. In this case, since the file1.txt is an empty file, the two lines will be appended. After typing the lines, press ctrl + d to save the file.

- The 'cat' command can be used to copy the content of a file into another file.

  For example, to copy the file readme.txt to another file named readme_copy.txt, we have to write the following command:

  cat readme.txt>readme_copy.txt

```
localhost:~# cat readme.txt > readme_copy.txt
localhost:~# cat readme_copy.txt
Some tests:

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

  gcc hello.c -o hello
  ./hello

- Run QuickJS:

  qjs hello.js

- Run python:

  python3 bench.py
localhost:~#
```

- The 'cat' command can be used to concatenate the contents of multiple files in a single new file.

```
localhost:~# cat file1.txt hello.c > new_file.txt
```

In the above example, we have concatenate the files file1.txt and hello.c to a new file new_file.txt

Now, after concatenation, to view the content of the file new_file.txt, we have to write

cat > new_file.txt

```
localhost:~# cat new_file.txt
This is line no1.
Welcome to KKHSOU.

#include <stdio.h>
int main(void)
{
        printf("hello world\n");
        return 0;
}

localhost:~#
```

- The 'cat -n' option displays line numbers in front of each line in a file.

```
localhost:~# cat -n hello.c
     1  #include <stdio.h>
     2  int main(void)
     3  {
     4          printf("hello world\n");
     5          return 0;
     6  }
     7
localhost:~#
```

- The 'cat -b' option removes the empty lines.

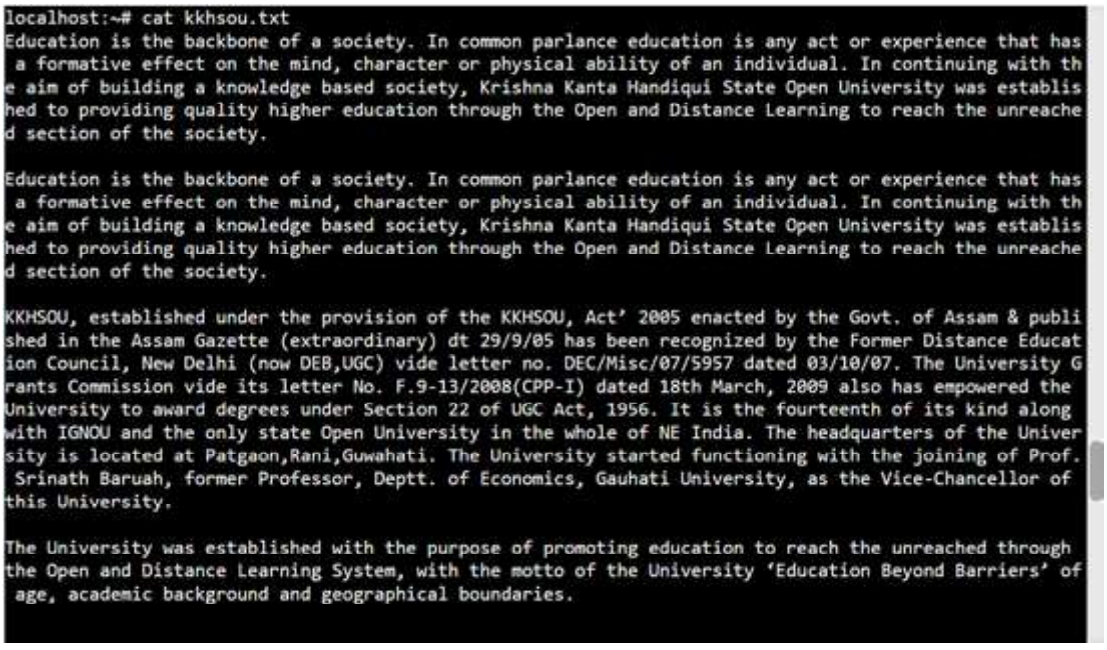
```
localhost:~# cat -b hello.c
     1  #include <stdio.h>
     2  int main(void)
     3  {
     4          printf("hello world\n");
     5          return 0;
     6  }

localhost:~#
```

7.4.2 more

We have already learnt that 'cat' command can be used to display the content of a file. Similarly we can also use the 'more' command to display the content of a file. Now, the question may arise that what are the differences between the 'cat' and 'more' command while displaying the content of a file? The answer is that, in case of larger files, 'cat' command output will scroll off your screen while 'more' command displays output one screenful at a time.

For example, suppose we have a file named 'kkhsou.txt' which contains some long texts. Now, to display the content of the file 'kkhsou.txt', we have to type the command cat kkhsou.txt as shown below:



You have seen that, we have to scroll to read the content of the file. Let us use the 'more' command to display the content of the same file 'kkhsou.txt' and see the differences.

- **more -num:** Limits the line displayed per page.(num is the total number of lines to be displayed)

```
localhost:~# more -10 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 building a knowledge based society, Krishna Kanta Handiqui State Open University was establis
hed to providing quality higher education through the Open and Distance Learning to reach the unreache
d section of the society.

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 building a knowledge based society, Krishna Kanta Handiqui State Open University was establis
hed to providing quality higher education through the Open and Distance Learning to reach the unreache
localhost:~#
```

It displays the first 10 lines of the file 'kkhsou.txt'.

7.4.3 less

The 'less' command is same as 'more' command but include some more features.

It automatically adjust with the width and height of the teminal window, while 'more' command cuts the content as the width of the terminal window get shorter.

For example, less kkhsou.txt

```
KKHSOU, established under the provision of the KKHSOU, Act' 2005 enacted by the Govt. of Assam & pub
lished in the Assam Gazette (extraordinary) dt 29/9/05 has been recognized by the Former Distance Educ
ation Council, New Delhi (now DEB,UGC) vide letter no. DEC/Misc/07/5957 dated 03/10/07. The University
 Grants Commission vide its letter No. F.9-13/2008(CPP-I) dated 18th March, 2009 also has empowered th
e University to award degrees under Section 22 of UGC Act, 1956. It is the fourteenth of its kind alon
g with IGNOU and the only state Open University in the whole of NE India. The headquarters of the Univ
ersity is located at Patgaon,Rani,Guwahati. The University started functioning with the joining of Pro
f. Srinath Baruah, former Professor, Deptt. of Economics, Gauhati University, as the Vice-Chancellor o
f this University.

The University was established with the purpose of promoting education to reach the unreached through
the Open and Distance Learning System, with the motto of the University 'Education Beyond Barriers
 of age, academic background and geographical boundaries.

The main aim of the university is to develop and provide easily accessible modes of quality higher edu
cation and training with the use of latest educational inputs and technology. Because of the inherent
flexibility in terms of pace and place of learning, methods of evaluation, etc., The University holds
the promise of providing equality of opportunities for higher education and bringing into its fold the
 deprived and denied sections along with the fresh learners.

The programmes of the University are so designed that it can spread education to all the learners of t
he region.The course structures of the university have been designed at par with the national curricul
um. Along with the traditional programmes, the University offers various professional Certificate and
Diploma programmes to enable the already enrolled students in various colleges and educational institu
tions of the state to pursue add-on studies simultaneously with their regular courses.

:
```



7.5 LET US SUM UP

- ls command shows the full list or content of your directory.
- ls -l command displays the list in reverse order.
- ls -lX command groups the files with same extensions together in the

list.

- cp command is used to copy a file or a directory.
- cp -r command is used to copy a directory along with its sub directories.
- To rename a directory mv command is used.
- rm command permanently removes a file.
- You can delete a directory having sub directories inside it with the rm -r command.
- To create a new directory, mkdir command is used.



7.6 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



7.7 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: ls -a

Ans to Q No 2: ls -R

Ans to Q No 3: ls ../

Ans to Q No 4: cp -r

Ans to Q No 5: pwd



7.8 MODEL QUESTIONS

Q 1: Create a directory named EMPLOYEE and DEPARTMENT.

Q 2: Copy a file to the directory EMPLOYEE.

Q 3: Rename the directory DEPARTMENT to DEPT.

Q 4: Change your current directory to a different directory and then display your current working directory.

*** ***** ***