APPLIED OPERATING SYSTEM LABORATORY







LINUX DIRECTORIES AND FILES CREATION







OBJECTIVES

Upon completion of this module, the student will be able to:

 Perform file and directory creation, and move between directories using LINUX commands









TOPIC OUTLINE

Directory commands

- Display directory content
- Change directory
- Make directory
- Print working directory
- Remove directory

File creation

- Display contents of single or multiple files
- Create a file using cat command

Other commands

- clear command
- history command
- date command







Display directory content

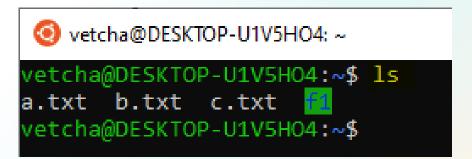
To display the contents of the current working directory

Syntax:

Is

Options that can be used with it.

- -a
- -A
- -d
- -
- -r
- -R



```
vetcha@DESKTOP-U1V5HO4: ~
```







Change directory

- To change directory Syntax:
 - cd <directory_name>
 - cd . means the current directory
 - cd .. means parent directory
 - cd means will take you to your home directory
 - cd will take you to your previous directory
 - cd ~ username will take you to the home directory of the user
 - cd <directorypath>

```
    vetcha@DESKTOP-U1V5HO4: ~
```

```
vetcha@DESKTOP-U1V5H04:~$ cd f1
vetcha@DESKTOP-U1V5H04:~/f1$ cd ..
vetcha@DESKTOP-U1V5H04:~$ cd ..
vetcha@DESKTOP-U1V5H04:~$ cd -
vetcha@DESKTOP-U1V5H04:~$ cd -
/home/vetcha
vetcha@DESKTOP-U1V5H04:~$ cd ~
vetcha@DESKTOP-U1V5H04:~$ cd ~
```

vetcha@DESKTOP-U1V5HO4: ~/f2/f3

vetcha@DESKTOP-U1V5H04:~**\$ cd f2/f3/** vetcha@DESKTOP-U1V5H04:~/**f2/f3\$**









Make directory

- To make directory/directories
 Syntax: mkdir <directoryname>
- To create multiple directories
- To create directory name with spaces
- To create a subdirectory /home/vetcha/folder1/folder2 (folder2 is not yet existing)
- To print or show the working directory

Syntax: pwd

```
vetcha@DESKTOP-U1V5HO4: ~

vetcha@DESKTOP-U1V5HO4: ~ $ mkdir folder1
vetcha@DESKTOP-U1V5HO4: ~ $ ls
a.txt b.txt c.txt folder1
vetcha@DESKTOP-U1V5HO4: ~ $

vetcha@DESKTOP-U1V5HO4: ~ $ mkdir dir1 dir2 dir3
vetcha@DESKTOP-U1V5HO4: ~ $ ls
a.txt b.txt c.txt dir1 dir2 dir3
vetcha@DESKTOP-U1V5HO4: ~ $ mkdir my\ folder
vetcha@DESKTOP-U1V5HO4: ~ $ mkdir my\ folder
vetcha@DESKTOP-U1V5HO4: ~ $ ls
a.txt b.txt c.txt dir1 dir2 dir3
vetcha@DESKTOP-U1V5HO4: ~ $ ls
a.txt b.txt c.txt dir1 dir2 dir3
vetcha@DESKTOP-U1V5HO4: ~ $ ls
a.txt b.txt c.txt dir1 dir2 dir3
vetcha@DESKTOP-U1V5HO4: ~ $ ls
a.txt b.txt c.txt dir1 dir2 dir3
```

vetcha@DESKTOP-U1V5HO4: ~

```
vetcha@DESKTOP-U1V5H04:~$ mkdir -p folder1/folder2
vetcha@DESKTOP-U1V5H04:~$ ls folder1/
folder2
vetcha@DESKTOP-U1V5H04:~$ cd folder1/folder2
vetcha@DESKTOP-U1V5H04:~/folder1/folder2$ pwd
/home/vetcha/folder1/folder2
vetcha@DESKTOP-U1V5H04:~/folder1/folder2$
```







Remove empty directory

To remove empty directory/directories
 Syntax:
 rmdir [-option] < directoryname>

 To remove directory and its subdirectory

```
vetcha@DESKTOP-U1V5HO4: ~

vetcha@DESKTOP-U1V5HO4: ~ $ ls dir2

dir3

vetcha@DESKTOP-U1V5HO4: ~ $ rmdir -p dir2/dir3

vetcha@DESKTOP-U1V5HO4: ~ $ ls
   a.txt b.txt c.txt dir3 folder!

vetcha@DESKTOP-U1V5HO4: ~ $
```







Remove directory that is not empty

 To remove directory that is not empty Syntax:

rm -r <directoryname>

-r means remove directories and their contents recursively

```
vetcha@DESKTOP-U1V5HO4: ~
/etcha@DESKTOP-U1V5HO4:~$ ls dir2
 etcha@DESKTOP-U1V5HO4:~$ rmdir -p dir2/dir3
/etcha@DESKTOP-U1V5HO4:~$ ls
a.txt b.txt c.txt
/etcha@DESKTOP-U1V5HO4:~$
vetcha@DESKTOP-U1V5HO4: ~
        b.txt c.txt
etcha@DESKTOP-U1V5HO4:~$ ls folder1
 .txt b.txt c.txt
etcha@DESKTOP-U1V5HO4:~$ rmdir folder1
rmdir: failed to remove 'folder1': Directory not empty
etcha@DESKTOP-U1V5HO4:~$ rm -r folder1
/etcha@DESKTOP-U1V5HO4:~$ ls
```

b.txt c.txt

etcha@DESKTOP-U1V5HO4:~\$







FILE CREATION COMMAND

Create a file or files

 To create a file or files using cat command

```
Syntax:

cat > <filename>

type text

CTRL+D (to save and exit)
```

- cat means concatenate
- To view the content of file/files

```
Syntax:
```

cat <filename>

```
vetcha@DESKTOP-U1V5HO4: ~

vetcha@DESKTOP-U1V5HO4: ~ $ cat > file1.txt

First file
vetcha@DESKTOP-U1V5HO4: ~ $ cat file1.txt

First file
vetcha@DESKTOP-U1V5HO4: ~ $

vetcha@DESKTOP-U1V5HO4: ~ $
```









OTHER LINUX COMMANDS

clear command

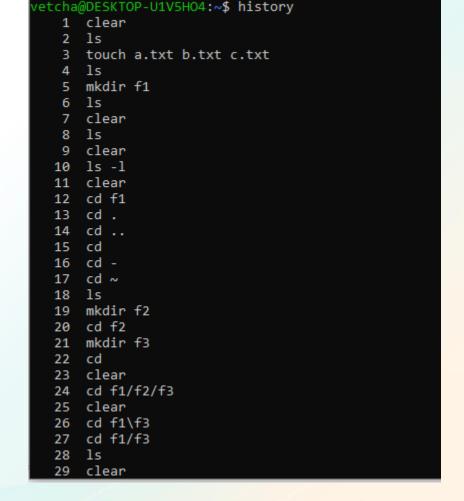
To clear the screen.

Syntax: clear

history command

 To display previously entered commands. This information is stored the ~/.bash_history file located at the home directory of each user.

Syntax: history



vetcha@DESKTOP-U1V5HO4: ~







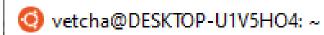


OTHER LINUX COMMANDS

date command

To display the date and time.

Syntax: date



```
vetcha@DESKTOP-U1V5HO4:~$ date
Tue Jul 28 20:03:02 PST 2020
vetcha@DESKTOP-U1V5HO4:~$ date +%d/%m/%Y
28/07/2020
vetcha@DESKTOP-U1V5HO4:~$
```







REFERENCES

- Sobell, M., et al. (2017). A Practical Guide to Linux Commands, Editors, and Shell Programming, 4th Ed. Addison-Wesley Professional
- Cobbaut, P. (2016). Mastering Linux- Networking
- Blum, R., (2015). Linux Command Line and Shell Scripting Bible







