



Linux Essentials Workshop-1

Clarusway



Subject: Linux Basic Shell Commands

Learning Goals

- Practice using the linux basic shell commands.

Introduction

We've covered a lot of Linux concepts, but now it's time to put the concepts in to practice. We'll start with basic shell commands.

Practice Using the **Bash Shell** in Lesson

You might notice a few new things in this lesson that you haven't encountered before. We'll walk you through them.

Code Along

Part 1 - Basic Commands 1/2

1. Open the terminal

- When you first open the terminal, you are in the home directory of your user.

```
open the terminal (Ubuntu/WSL/Try it/Git Bash/VS Code or whatever)
```

2. To know which directory you are in, you can use the "pwd" command.

- It gives us the absolute path, which means the path that starts from the root. The root is the base of the Linux file system. It is denoted by a forward slash(/).
- The user directory is usually something like "/home/username".

```
pwd
/home/username

ls
```

3. Go to desktop

- cd desktop
- Use the "cd" command to go to a directory. For example, if you are in the home folder, and you want to go to the downloads folder, then you can type in "cd Downloads".
- Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is.
- But there is a problem with these commands. Imagine you have a folder named "Raspberry Pi". In this case, when you type in "cd Raspberry Pi", the shell will take the second argument of the command as a different one, so you will get an error saying that the directory does not exist. Here, you can use a backward slash. That is, you can use "cd Raspberry\ Pi" in this case.
- Spaces are denoted like this: If you just type "cd" and press enter, it takes you to the home directory.
- To go upper folder before that, you can type "cd .." . The two dots represent one folder up.

```
cd Desktop or cd Downloads

cd
cd ..
```

4. Create a directory named "clarusway"

- mkdir clarusway
- Use the mkdir command when you need to create a folder or a directory. For example, if you want to make a directory called "DIY", then you can type "mkdir DIY". Remember, as told before, if you want to create a directory named "DIY Hacking", then you can type "mkdir DIY\ Hacking".
- go to clarusway directory cd clarusway

```
mkdir clarusway
cd clarusway
```

5. List all the files/folders with details

- `ls -al`
- "ls" command is used to know what files/folders are in the directory you are in. You can see all the hidden files/folders by using the command "`ls -a`".
- As we created the folder yet, there is no files in it.

```
ls -al  
ls -a
```

6. Create a file named test.txt

- The touch command is used to create a file. It can be anything, from an empty txt file to an empty zip file.

```
touch test.txt
```

7. Write "hello" to test.txt

- The "echo" command helps us move some data, usually text into a file.

```
echo "hello" > test.txt
```

8. Add "My name is Robert" to test.txt

```
echo "My name is Robert" >> test.txt
```

9. Print the content of test.txt

- Use the cat command to display the contents of a file. It is usually used to easily view programs.

```
cat test.txt
```

10. Create a file named test1.txt and write "hello, this is second text file" while creating.

```
cat > test1.txt
hello
this is second text file
(Contol C to exit)
```

11. Print the contents of test1.txt

- As you see this is the second way of creating a file. We created this file with cat command and wrote the content while creating it.

```
cat test1.txt
```

12. Display the contents of the file test.txt and test1.txt at the same time.

```
cat test.txt test1.txt
```

13. Create text files with the name multiple1.txt through multiple10.txt in current directory.

- Also you can use "touch multiple1.txt multiple2.txt ... multiple10.txt" command.

```
touch multiple{1..10}.txt
touch multiple1.txt multiple2.txt ... multiple10.txt
```

14. Go to upper directory

```
cd ..
```

15. Create a directory named "europe"

```
mkdir europe
cd europe
```

16. Create two files named lab.txt and lab1.txt at the same time

```
touch lab.txt lab1.txt
```

17. Write "This is the second directories first file" in lab.txt

```
echo "This is the second directories first file" >> lab.txt
```

18. cat lab.txt

- Print the content of lab.txt

```
cat lab.txt
```

19. Make a copy of lab.txt in the current directory as newlab.txt

- We use this command when we are copying the file in the current directory.
- If the destination file exists, it will be overwritten. (To get rid of this use -i. This will give a warning.)

```
cp lab.txt newlab.txt  
cp -i lab.txt newlab.txt ((cp: overwrite 'newlab.txt'?))
```

20. List all the files in the current directory.

```
ls -a
```

21. Print the contents of newlab.txt

```
cat newlab.txt
```

22. List any file in current directory with the extension of ".txt"

```
ls *.txt
```

23. List any file which begins with "n" and ends with ".txt"

- List any file begins with lab and we dont know the end of it

```
ls n*.txt
ls lab?.txt
```

Part 2 - Basic Commands 2/2

24. Copy newlab.txt to the clarusway directory.

- As you remember newlab.txt is in the europe directory. We will copy it in a different directory named clarusway.
- To copy a file to a directory, specify the absolute or the relative path to the directory. If the destination directory is omitted the file is copied to the current directory.
- An absolute or full path points to the same location in a file system, regardless of the current working directory. To do that, it must include the root directory.
- By contrast, a relative path starts from some given working directory, avoiding the need to provide the full absolute path.

```
cp newlab.txt /home/san/clarusway (absolute path)
cp newlab.txt ../clarusway/ (relative path)
```

25. Copy the files beginnig with "m*{6..10}" and ending with ".txt" from clarusway directory to europe directory.

```
cd ..
cd clarusway
cp m*{6..10}.txt ../europe/
```

26. Move the files beginnig with "m*{1..3}" and ending with ".txt" from clarusway directory to europe directory.

- The move command is a command line utility that moves files or directories from one place to another .
- It supports moving single files, multiple files and directories. It can prompt before overwriting and has an option to only move files that are new than the destination.

```
cd ..  
cd clarusway  
mv m*{1..3}.txt ../europe/
```

27. List all the files on Clarusway directory.

```
ls -al
```

28. Rename the newlab.txt file as linuxlab.txt.

```
cd ../europe  
mv newlab.txt linuxlab.txt
```

29. List the files with details on clarusway directory and see the renamed linuxlab.txt file.

```
ls -l
```

30. Print the permission details of linuxlab.txt.

```
ls -l linuxlab.txt
```

31. Change file permissions of the linuxlab.txt as everybody can read, write and execute.

```
chmod -v ugo=rwx linuxlab.txt
```

32. Change execute permissions of everybody to no execute permission on linuxlab.txt

```
chmod -v 666 linuxlab.txt
```

33. Change the file permissions one level up on linuxlab.txt

```
chmod -v +x linuxlab.txt
```

34. Delete the linuxlab.txt file

- to delete this file you should go to the directory of the file or use the absolute path of file

```
rm linuxlab.txt or rm -rf linuxlab.txt
```

35. Create "america" directory

```
mkdir america
```

36. Delete america directory

```
rmdir america
```

37. Delete clarusway directory

- You can not delete this directory because it is not empty.

```
cd ..  
rmdir clarusway  
rm -r clarusway
```

38. Go to the europe directory

```
cd europe
```


39. List the contents of europe directory

```
ls
```

40. Create a hidden linuxworkshop.txt file

```
touch .linuxworkshop.txt
```

41. Use ls command to list the contents of the europe directory

- As you see we can not see the linuxworkshop.txt file
- To list all the files, we use "ls -a" command.

```
ls  
ls -a
```

42. Make the linuxworkshop.txt visible

```
mv .linuxworkshop.txt linuxworkshop.txt
```

43. Exit from the terminal

- Tips and Tricks for Using Linux Command Line
- You can use the clear command to clear the terminal if it gets filled up with too many commands.
- TAB can be used to fill up in terminal. For example, You just need to type "cd Doc" and then TAB and the terminal fills the rest up and makes it "cd Documents".
- Ctrl+C can be used to stop any command in terminal safely. If it doesn't stop with that, then Ctrl+Z can be used to force stop it.

```
exit
```

😊 **Thanks for Attending** ✍️

Clarusway

