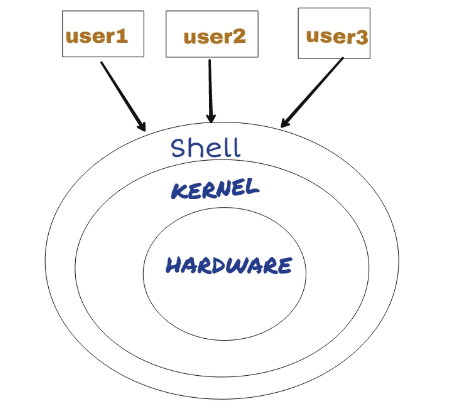
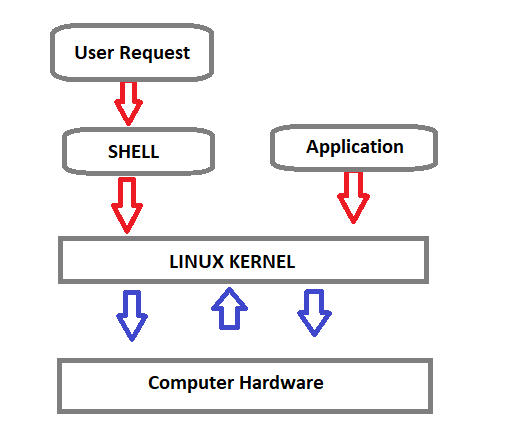
What is the "Shell"?

* Shell is an interface between an end-user and the Linux system.



* In other words, Shell is a program that receives commands from the user, relays them to the operating system to process and displays the output. Shell is one of the main parts of Linux OS. Each Linux distro comes with a GUI (Graphical User Interface), but essentially Linux has a CLI (Command-Line Interface).



Shell Types

**C Shell :** If you are using a C-type shell, the % character is the default prompt.

* C shell (csh)
* TENEX/TOPS C shell (tcsh)

**Bourne Shell :** If you are using a Bourne-type shell, the $ character is the default prompt.

* Bourne shell (sh)
* Korn shell (ksh)
* Bourne Again shell (bash)
* POSIX shell (sh)

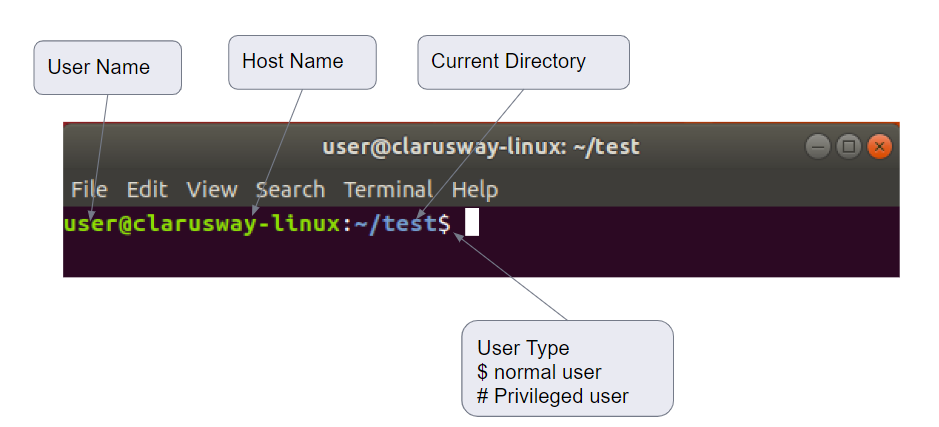
An enhanced version of SH is called BASH (which stands for **B**ourne **A**gain **SH**ell) and serves as the main shell program on the most Linux systems.

* The standard Linux shell (BASH) is both a command-line interpreter and a programming language.



The most common interpreter is BASH or the Bourne Again Shell, but there are others available as well and some of them does not use the dollar sign.

**Command Prompt**



The command prompt at the beginning of the command line is a short text string. The command prompt for Linux generally shows the current **user**, the current **host**, and the appropriate **directory**.

The command prompt is easily modified to display as desired with more or less information. At the end of the prompt list, the $(dollar sign) signifies the current user being unprivileged.

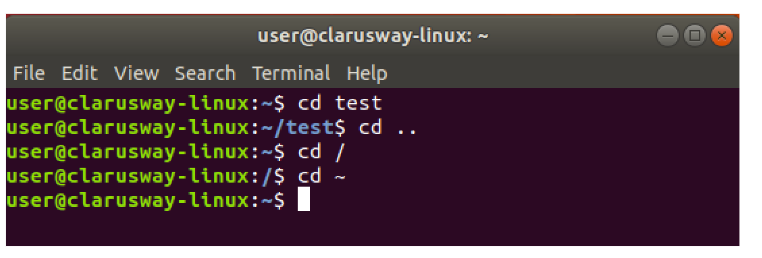
* Dollar sign ($) means you are a normal user and indicates you are logged in with the normal permissions.
* Hash (#) means you are the system administrator (root) and indicates you are logged in with root privileges.
* The "root" account on a Linux computer is the account with full privileges.
* Root access is often necessary for performing commands in Linux, especially commands that affect system files. Because root is so powerful, it's recommended to only request root access when necessary, as opposed to logging in as the root user.

## **View the Lesson (Command Line Basics)**

### Basic Shell Commands

This is a list of most frequently used Linux commands.

* The **~ (tilde)** symbol stands for your home directory.
* The **pwd** (stands for print working directory) command will allow you to know in which directory you're located.
* The **ls** command will show you the list of folders and files in your current directory.
* The **cp** command will make a copy of a file.
* The **cd** command will allow you to change directories.
* The **rm** command removes or deletes a file in your directory.
* The **rmdir** command will delete an empty directory.
* The **mkdir** command will allow you to create directories
* The **mv** command will move a file to a different location or will rename a file.
* cd - Navigate to the last directory you were working in.
* cd ~ or just cd Navigate to the current user's home directory.
* cd .. Go to the parent directory of current directory (mind the space between cd and ..)



## **View the Lesson (Command Line Basics)**

### Quoting

Quoting is used to disable special treatment of certain characters and words, as well as to prevent parameter expansion and preserve what is quoted.

The bash shell knows rare, special characters like $ (dollar sign), which is used to extend the value of the element. For example $PATH is used to extend the value of PATH element which is predefined variable in bash to hold system paths as shown below.



clarusway@f85a0c1549f4:~$ echo $PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin

    :/bin:/usr/games:/usr/local/games

The special character $ can also be used with the user's custom element. For example, in the bash shell, the user can define an element like greeting="Hello from Clarusway" and extend the value of this element as shown below.



clarusway@f85a0c1549f4:~$ greeting="Hello from Clarusway"

clarusway@f85a0c1549f4:~$ echo $greeting

Hello from Clarusway

There are three types of quotes:

1. **Double Quotes** : The double quote " preserve the literal value of most characters contained within the quotes, exceptions include $ (for variables), ' (for single quoting), \ (for escaping a character) .



clarusway@f85a0c1549f4:~$ echo $SHELL

/bin/bash

clarusway@f85a0c1549f4:~$ echo "$SHELL"

/bin/bash

clarusway@f85a0c1549f4:~$ echo "path to shell $SHELL"

path to shell /bin/bash

1. **Single Quotes**: The single quote ( 'quote' ) protects everything enclosed between single quotation marks.



clarusway@f85a0c1549f4:~$ echo $SHELL

/bin/bash

clarusway@f85a0c1549f4:~$ echo '$SHELL'

$SHELL

clarusway@f85a0c1549f4:~$ echo 'path to shell $SHELL'

path to shell $SHELL

1. **Backslash**: Use the backslash to change the special meaning of the characters or to escape special characters within the text such as quotation marks.

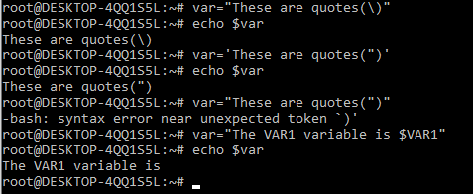


clarusway@f85a0c1549f4:~$ echo "\$SHELL"

$SHELL

clarusway@f85a0c1549f4:~$ echo "path to shell \$SHELL"

path to shell $SHELL



## **View the Lesson (Command Line Basics)**

### File Permission

### Linux File Ownership

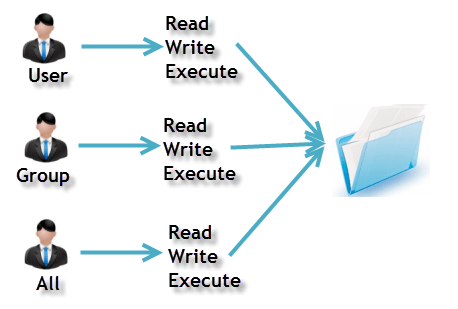
Each file and directory on Linux system has 3 types of owners assigned,

* **User :** A user is the owner of the file.
* **Group :** A user- group can contain multiple users.
* **Other/All :** Any other user who has access to a file.

### Permission

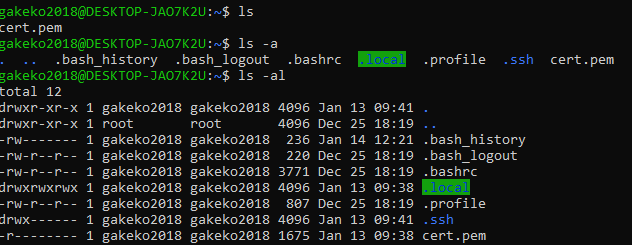
Each file and directory on your Linux system has 3 permissions defined for all the 3 owners.

* **Read :** The read permission gives you the authority to open and read a file.
* **Write :** The write permission gives you the authority to modify the contents of a file.
* **Execute :** In Linux, you cannot run a program unless the execute permission is set.

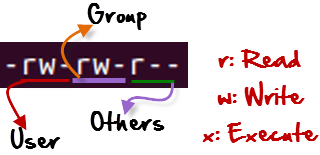


ls

ls -a







**r** = read permission

**w** = write permission

**x** = execute permission

**-** = no permission



### Changing Permission with chmod Command

We can use the chmod command which stands for **change mode**.

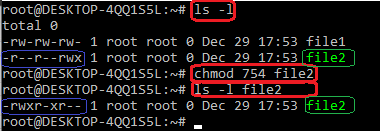
Using the command, we can set permissions (read, write, execute) on a file/directory for the owner, group and the world.

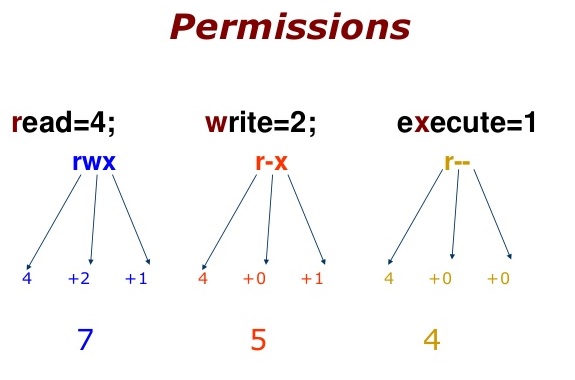
chmod permissions filename

The table below gives for all for permissions types.

| **Symbol** | **Permission Type** |
| --- | --- |
| --- | No Permission |
| --x | Execute |
| -w- | Write |
| -wx | Execute+Write |
| r-- | Read |
| r-x | Read+Execute |
| rw- | Read+Write |
| rwx | Read+Write+Execute |

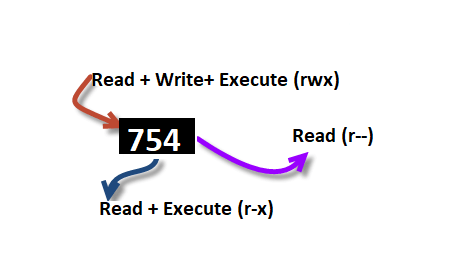
Example





754 code says;

* Owner can read, write and execute
* Usergroup can read and execute
* Other can only read



## **View the Lesson (Command Line Basics)**

### Ping & SSH Command

### Ping Command

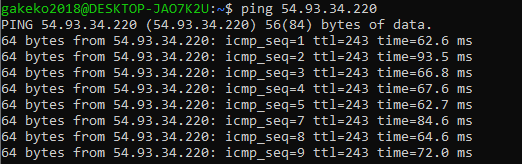
Ping or Packet Internet Groper is a network administration utility used to check the connectivity status between a source and a destination computer/device over an IP network. It also helps you assess the time it takes to send and receive a response from the network.

ping host-name/IP

Example

ping 54.93.34.220

Press CTRL+C (in MacOS => CMD+C) to exit.



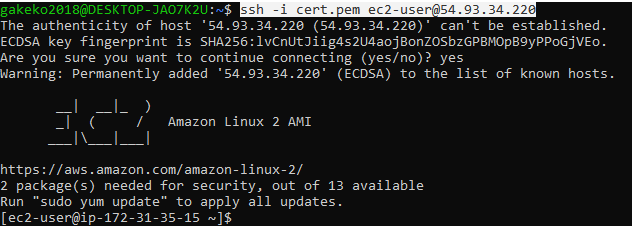
### SSH

* ssh stands for “Secure Shell”.
* It is a protocol used to securely connect to a remote server/system.
* ssh is secure in the sense that it transfers the data in an encrypted form between the host and the client

ssh user@host(IP/Domain\_name)

Example

ssh -i cert.pem ec2-user@54.93.34.220



## **View the Lesson (Command Line Basics)**

### whoami Command

Displays user, group and privileges information for the user who is currently logged on to the local system.

* whoami command is used both in Lunix Operating System and as well as in Windows Operating System.
* It is basically the concatenation of the strings “who”,”am”,”i” as whoami.

whoami

Example:



clarusway@f85a0c1549f4:~$ whoami

clarusway

Formun Üstü

Previous

Formun Altı

Formun Üstü

End of lesson

## **View the Lesson (Creating, Moving, and Deleting Files)**

### Creating, Moving, Deleting

Creating, moving, and deleting files and directories.

touch - create a file

rm - delete the file

cp - used to copy file or folder

mv - used to move file or folder

mkdir - create a folder

rmdir - delete folder

### Working with directories

Create a new directory mkdir <NAME>

Copy a directory cp -r <SOURCE> <DESTINATION>

Move a directory mv <SOURCE> <DESTINATION>

Delete a directory rm -r <DIRECTORY>

Delete an empty directory rmdir <DIRECTORY>

### Working with files

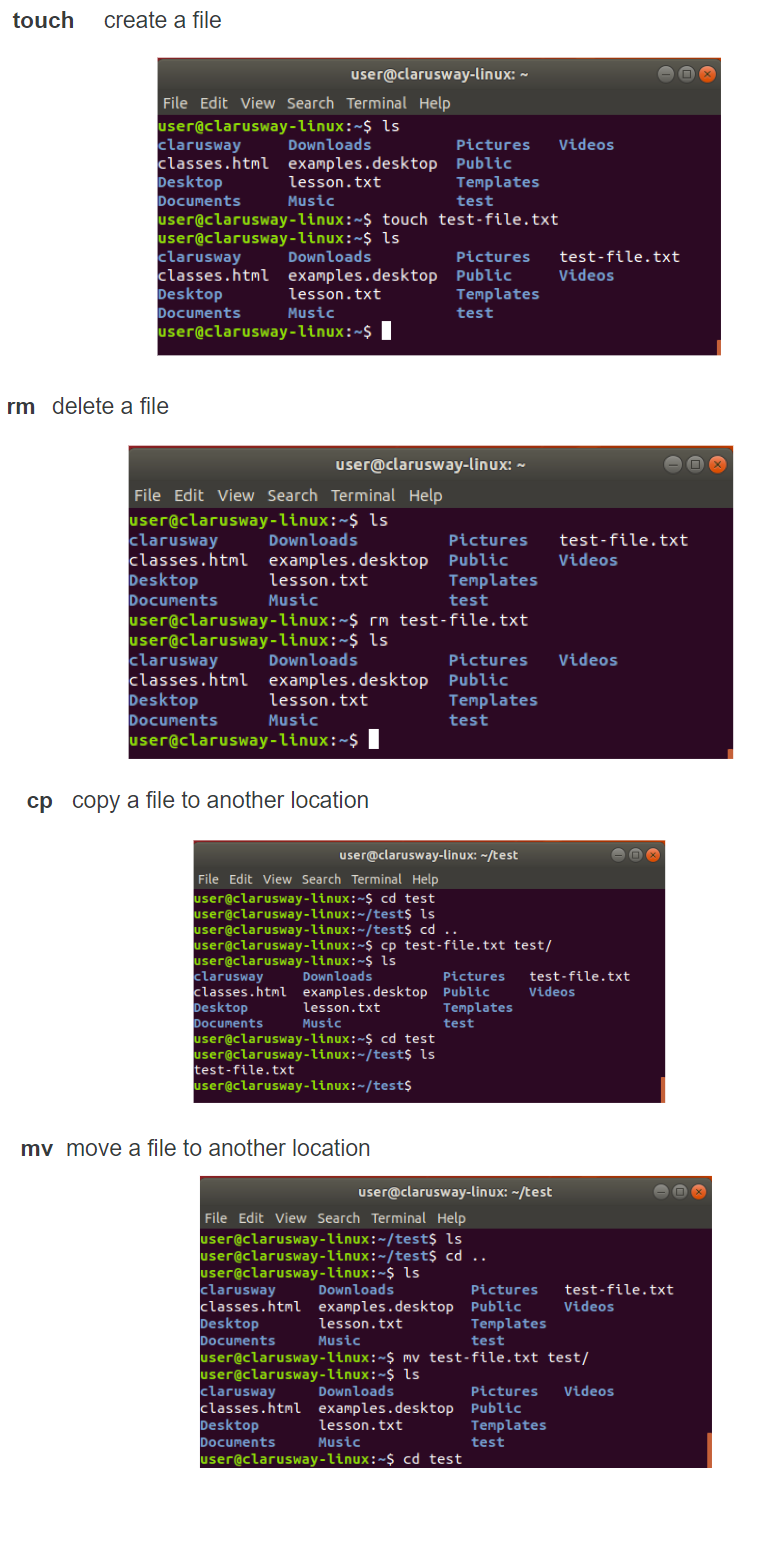
Create a new file touch filename

Removing Files rm -option filename

**Option** -r includes the contents of a directory and the contents of all subdirectories when you remove a directory.

**Option** -i prevents the accidental removal of existing files or directories.

**Example:**



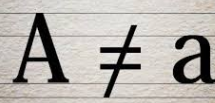
## **View the Lesson (Creating, Moving, and Deleting Files)**

### Case Sensitivity

Most of the common Linux file systems are case sensitive; this is something to keep in mind when creating and moving directories and files.

**Using Case Sensitivity**

Lower-case and upper-case letters have different ASCII representation.



Example;

touch newfile

touch Newfile

The commands above will create two different files.

## **View the Lesson (Creating, Moving, and Deleting Files)**

### Simple Globbing

Globbing is primarily used to match patterns in filenames or text by using a wildcard character to create the pattern.

? (Question mark) : Match any single character

\* (Asterisk) : Match any number of character(s)

[] (Brackets) : Match character from a range

^ (Caret) : Used to match starting character

$ (Dollar sign) : Used to match ending character

{} (Curly brace) : Used to match more than one pattern

| Pipe : Used for applying more than one condition

Example :



clarusway@f85a0c1549f4:~$ ls

all file1 file2 file3 file4 happiness loneliness

    reverse

clarusway@f85a0c1549f4:~$ ls file?

file1 file2 file3 file4

clarusway@f85a0c1549f4:~$ ls ?????

file1 file2 file3 file4

clarusway@f85a0c1549f4:~$ ls ????1

file1

clarusway@f85a0c1549f4:~$ ls file\*

file1 file2 file3 file4

clarusway@f85a0c1549f4:~$ ls \*

all file1 file2 file3 file4 happiness loneliness

    reverse

clarusway@f85a0c1549f4:~$ ls \*ness

happiness loneliness

clarusway@f85a0c1549f4:~$ ls \*[1-4]

file1 file2 file3 file4

clarusway@f85a0c1549f4:~$ ls \*[2-3]

file2 file3

clarusway@f85a0c1549f4:~$ ls \*[[:digit:]]

file1 file2 file3 file4

* [[:upper:]] or [[A-Z]] - match upper-case character
* [[:lower:]] or [[a-z]] - match lower-case character
* [[:digit:]] or [[0-9]] - match digits
* [[:alpha:]] or [[a-zA-Z]] - match either case character
* [[:alphanum:]] or [[a-zA-Z0-9]] - match alphanumeric

Formun Altı