

Linux Basics

Date : 08-Oct-2025

Wednesday

[LinkedIn](#)

[GitHub](#)

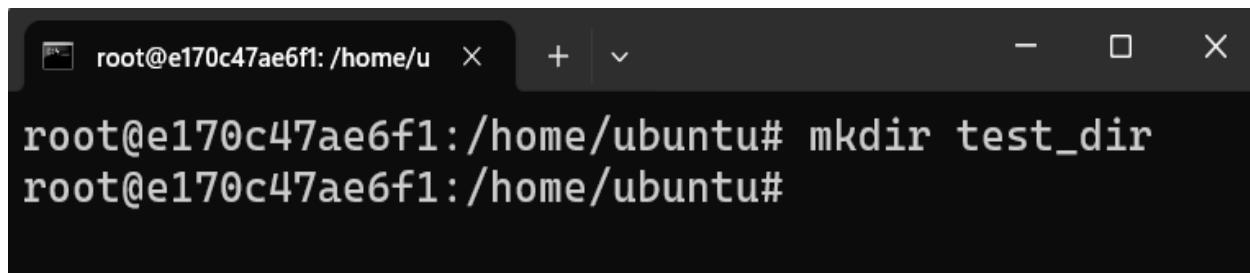
After Learning the Linux basics here is a small assignment to test myself.

It'll be going to be some commands from the class that I've taken .

1. Creating and renaming files/Directories

- Create a directory named `test_dir` using `mkdir`.

We are going to make a directory using the “**`mkdir`**” Command stand for Make Directory. If you are not familiar with the term Directory then the easy term for it is Folder  , that you have seen in your windows.



A terminal window showing a root shell on a Ubuntu system. The command `mkdir test_dir` is entered and executed successfully.

```
root@e170c47ae6f1:/home/ubuntu# mkdir test_dir
root@e170c47ae6f1:/home/ubuntu#
```

```
mkdir "dir_name"
mkdir test_dir
```

and using the command “**`mkdir test_dir`**” we made ourself a directory.

```
root@e170c47ae6f1:/home/u  × + | ▾  
root@e170c47ae6f1:/home/ubuntu# mkdir test_dir  
root@e170c47ae6f1:/home/ubuntu# ls  
test_dir  
root@e170c47ae6f1:/home/ubuntu#
```

By using the **ls** command we can see the list of files and directory.

Now we have made our “test_dir” directory

- Inside *test_dir*, create a empty file called *example.txt*

Now before make a empty file, we first have to go inside the *test_dir* directory.

Using the **cd** command: “**cd test_dir**”.

```
root@e170c47ae6f1:/home/ubuntu# cd test_dir  
root@e170c47ae6f1:/home/ubuntu/test_dir# touch exapmle.txt|
```

Now to make a text file we'll be using the **touch** command:

```
touch "file name"  
`touch example.txt`
```

Now we have a file name *example.txt*.

- Rename *example.txt* to *renamed_example.txt* using **mv** command.

First we check the file using the **ls** command.

Now that we confirmed the file *example.txt*, now we will be renaming it.

Mv command's full form is Move, It's used to move a file/directory to different places.

```
root@e170c47ae6f1:/home/ubuntu/test_dir# ls
root@e170c47ae6f1:/home/ubuntu/test_dir# touch example.txt
root@e170c47ae6f1:/home/ubuntu/test_dir# ls
example.txt
root@e170c47ae6f1:/home/ubuntu/test_dir# mv example.txt renamed_example.txt
root@e170c47ae6f1:/home/ubuntu/test_dir#
```

```
mv "file_name" "destination_name"
mv example.txt renamed_example.txt
```

If there is no file/or directory of that name that you have given then it'll make one and store it in.

Like we have given the destination name renamed_example.txt that file is there so it just made it right there.

```
root@e170c47ae6f1:/home/ubuntu/test_dir# ls
root@e170c47ae6f1:/home/ubuntu/test_dir# touch example.txt
root@e170c47ae6f1:/home/ubuntu/test_dir# ls
example.txt
root@e170c47ae6f1:/home/ubuntu/test_dir# mv example.txt renamed_example.txt
root@e170c47ae6f1:/home/ubuntu/test_dir#
root@e170c47ae6f1:/home/ubuntu/test_dir# ls
renamed_example.txt
root@e170c47ae6f1:/home/ubuntu/test_dir# |
```

Now we check with ls for the file name.

As you can see the result, we've attained our goal.

2. Viewing file content

- Use cat to display the content of /etc/passwd.

Cat command is used to view the content of a text file.

First we have to come to the '.', we can come to it by just running cd command.

```
root@e170c47ae6f1:/# cat etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
root@e170c47ae6f1:/#
```

```
cat file_location/name  
cat etc/passwd  
etc : location & passwd : file name
```

The content written below ["cat etc/passwd"](#) is the content of the file “Passwd”.

- Display only the first 5 lines of /etc/passwd using head.

The command used to show the starting line is [head command](#).

```
root@e170c47ae6f1:/# head -5 etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
root@e170c47ae6f1:/#
```

As you can see the head command shows the first 5 lines off the content.

```
head -"n" "location/name"  
head -5 etc/passwd  
etc/passwd : location and name of file  
n : 5 {no. of lines}
```

-
- Display only the last 5 lines of /etc/passwd using tail.

The command used to show the ending line is **tail command**.

```
root@e170c47ae6f1:/# tail -5 etc/passwd  
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin  
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin  
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin  
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash  
root@e170c47ae6f1:/#
```

Just like the head command tail command is used to find the last line of any file.

```
tail -5 etc/passwd
```

3. Searching for patterns

- Use grep to find all lines containing the word "root" in /etc/passwd.

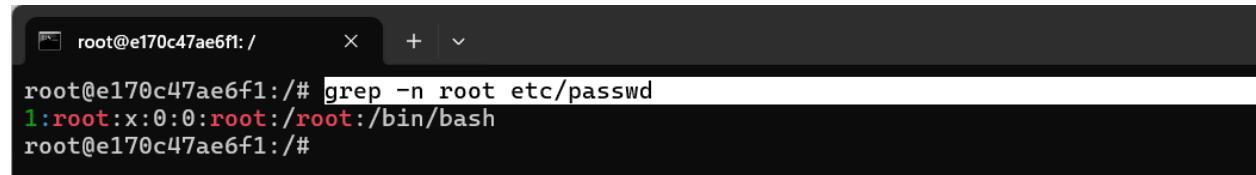
Grep command is used to Find specific patterns, words, or lines in a file or output.

```
root@e170c47ae6f1:/# grep root etc/passwd  
root:x:0:0:root:/root:/bin/bash  
root@e170c47ae6f1:/#
```

```
grep "word" "location/name"  
grep "root" etc/passwd  
word : root {the word to find}
```

By this we can find any word in the files, Generally used to find the words from the log file, for example "grep error logfile.txt" by this command the engineer can usually find the place of error.

And there is also a command for finding the exact number of the word is in.



A terminal window showing the command `grep -n root etc/passwd`. The output shows the line number 1 followed by the word "root".

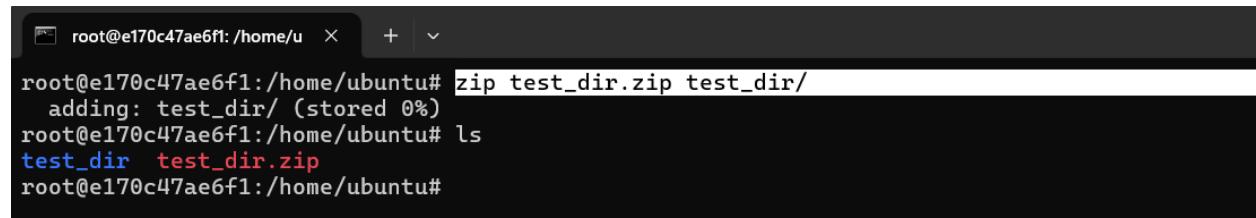
```
root@e170c47ae6f1:/# grep -n root etc/passwd  
1:root:x:0:0:root:/root:/bin/bash  
root@e170c47ae6f1:/#
```

```
grep -n root etc/passwd  
n : for showing the line number the  
word is in, for now it's line no. 1
```

4. Zipping and Unzipping

- Compress the `test_dir` directory into a file named `test_dir.zip` using `zip`.

Zip command in Linux used to compress the file and directories into the zip archive(with `.zip` extension).



A terminal window showing the command `zip test_dir.zip test_dir/`. The output shows the file `test_dir/` being added to the zip archive with a 0% compression rate.

```
root@e170c47ae6f1:/home/ubuntu# zip test_dir.zip test_dir/  
adding: test_dir/ (stored 0%)  
root@e170c47ae6f1:/home/ubuntu# ls  
test_dir test_dir.zip  
root@e170c47ae6f1:/home/ubuntu#
```

```
zip "new_file.zip" "file_name"  
zip test_dir.zip test_dir  
creates test_dir.zip containing test_dir
```

- Unzip test_dir.zip into a new directory named unzipped_dir.

Unzip is command used to unzip the .zip file.

Now we first have to make an *unzipped_dir* directory using `mkdir` command and then unzip the file inside the directory.

```
[root@e170c47ae6f1: /home/u] x + | ~  
root@e170c47ae6f1:/home/ubuntu# mkdir unzipped_dir  
root@e170c47ae6f1:/home/ubuntu# ls  
test_dir test_dir.zip unzipped_dir  
root@e170c47ae6f1:/home/ubuntu# unzip test_dir.zip -d unzipped_dir  
Archive: test_dir.zip  
  creating: unzipped_dir/test_dir/  
root@e170c47ae6f1:/home/ubuntu# ls unzipped_dir/  
test_dir  
root@e170c47ae6f1:/home/ubuntu# |
```

```
mkdir unzipped_dir  
unzip "file" -d "location"  
unzip test_dir.zip -d unzipped_dir  
-d : extract files into external dir.  
first file name to be unzipped then -d to extract and then  
location of directory in which it'll be extracted
```

If “-d” is not given then it will not unzip the file.

“-d” is used to extract in the external directory, if you want to extract in the same directory then you can just :

```
unzip test_dir.zip
```

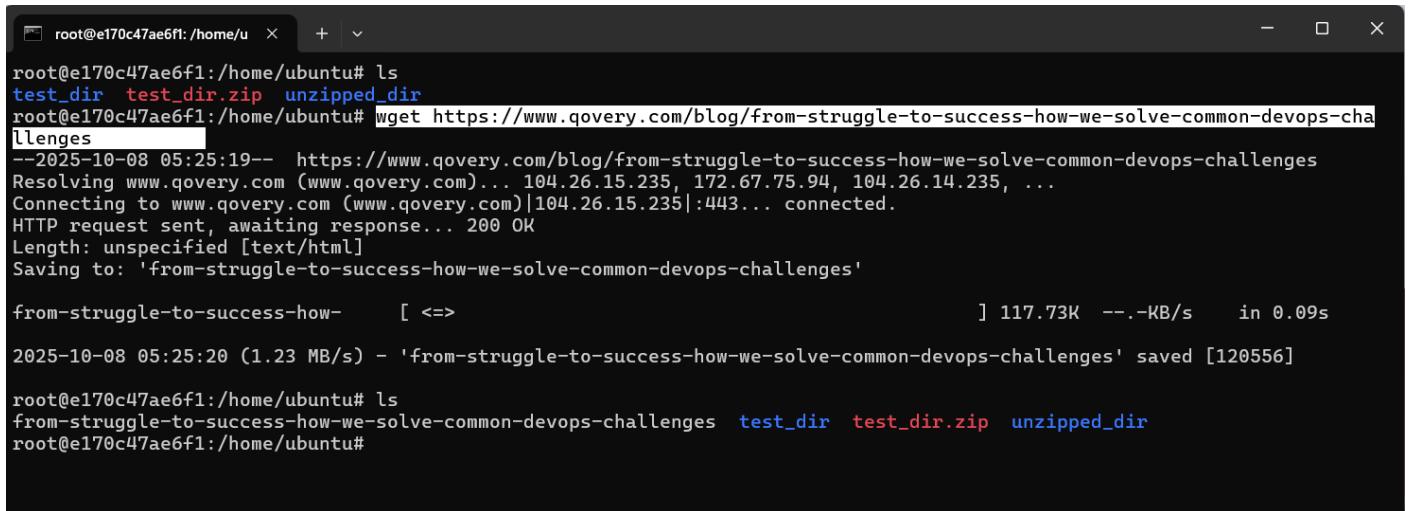
5. Downloading Files

- Use wget to download a file from a URL

wget is used to get the file from the internet.

It downloads the file, and there is a similar command that is “curl”, it’s fetches the text from the URL.

In this we are going to use the link { <https://www.qovery.com/blog/from-struggle-to-success-how-we-solve-common-devops-challenges> }



A terminal window showing the execution of the wget command. The terminal is running on a root shell on an Ubuntu system. The user runs 'ls' to show files 'test_dir', 'test_dir.zip', and 'unzipped_dir'. Then they run 'wget https://www.qovery.com/blog/from-struggle-to-success-how-we-solve-common-devops-challenges'. The output shows the progress of the download, including the URL, IP addresses, and a 200 OK response. Finally, 'ls' is run again to show the newly created file 'from-struggle-to-success-how-we-solve-common-devops-challenges'.

```
root@e170c47ae6f1:/home/ubuntu# ls
test_dir test_dir.zip unzipped_dir
root@e170c47ae6f1:/home/ubuntu# wget https://www.qovery.com/blog/from-struggle-to-success-how-we-solve-common-devops-challenges
--2025-10-08 05:25:19--  https://www.qovery.com/blog/from-struggle-to-success-how-we-solve-common-devops-challenges
Resolving www.qovery.com (www.qovery.com)... 104.26.15.235, 172.67.75.94, 104.26.14.235, ...
Connecting to www.qovery.com (www.qovery.com)|104.26.15.235|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'from-struggle-to-success-how-we-solve-common-devops-challenges'

from-struggle-to-success-how-      [ <=>                               ] 117.73K  --.-KB/s  in 0.09s
2025-10-08 05:25:20 (1.23 MB/s) - 'from-struggle-to-success-how-we-solve-common-devops-challenges' saved [120556]

root@e170c47ae6f1:/home/ubuntu# ls
from-struggle-to-success-how-we-solve-common-devops-challenges  test_dir  test_dir.zip  unzipped_dir
root@e170c47ae6f1:/home/ubuntu#
```

```
wget "link"
wget https://www.qovery.com/blog/from-struggle-to-
success-how-we-solve-common-devops-challenges
```

In this wget command make get the details from the link and make a text file and stored the output in it.

6. Changing Permission

- Create a file named secure.txt and change it's permission to read-only for everyone using chmod.

First we have to make a file named secure.txt, then we'll change its permission to read only.

Chmod values are as follow :

Read {r}: 4

Write {w}: 2

Execute {x}: 1

3: execute + write || 5 : read + execute || 6:
read + write

```
root@e170c47ae6f1:/home/u ~ + | ~
root@e170c47ae6f1:/home/ubuntu/test_dir# touch secure.txt
root@e170c47ae6f1:/home/ubuntu/test_dir# ls -l
total 0
-rw-r--r-- 1 root root 0 Oct  8 02:11 renamed_example.txt
-rw-r--r-- 1 root root 0 Oct  8 05:45 secure.txt
root@e170c47ae6f1:/home/ubuntu/test_dir# chmod 444 secure.txt
root@e170c47ae6f1:/home/ubuntu/test_dir# ls -l
total 0
-rw-r--r-- 1 root root 0 Oct  8 02:11 renamed_example.txt
-r--r--r-- 1 root root 0 Oct  8 05:45 secure.txt
root@e170c47ae6f1:/home/ubuntu/test_dir#
```

chmod "value" "file/dir_name"

chmod 444 secure.txt

After this the secure.txt permission is changed to
read only

7. Working with Environment Variables

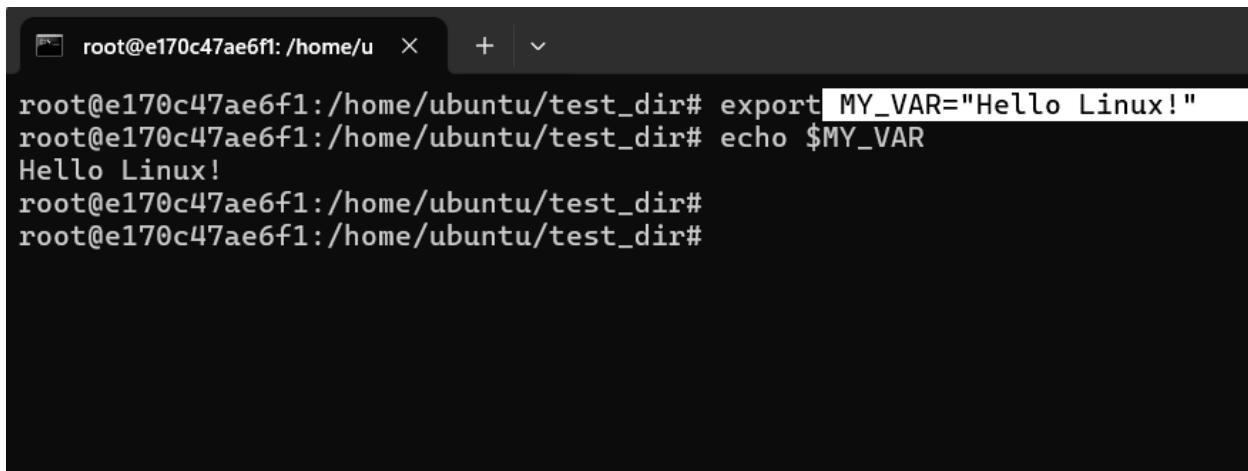
- Use `export` to set a new environment variable called `my_var` with the value “Hello Linux!”

“`export`” command is used to set the environment variables.

In easy word you can say that you are allowing value to the variable, often used for the frequently used commands.

Let's assume I use `man{manual}` command often then what can I do is, I can set the variable `mn=man`, by this I can use the `man` command with the word of my preference.

Let's give you variable a value:



A terminal window showing a root shell on an Ubuntu system. The user runs the command `export MY_VAR="Hello Linux!"`. Then they run `echo $MY_VAR`, which outputs "Hello Linux!". The terminal window has a dark background and light-colored text. The title bar shows the path `/home/u`.

```
root@e170c47ae6f1:/home/ubuntu/test_dir# export MY_VAR="Hello Linux!"
root@e170c47ae6f1:/home/ubuntu/test_dir# echo $MY_VAR
Hello Linux!
root@e170c47ae6f1:/home/ubuntu/test_dir#
root@e170c47ae6f1:/home/ubuntu/test_dir#
```

```
export "variable"="value"
export MY_VAR="Hello Linux!"
Here variable is MY_VAR
And it's value is "Hello Linux"
```

