Basic Linux CLI Commands

Command	Description
ls	List the directory (folder) system.
cd pathname	Change directory (folder) in the file system.
cd ..	Move one level up (one folder) in the file system.
cp	Copy a file to another folder.
mv	Move a file to another folder.
mkdir	Creates a new directory (folder).
rmdir	Remove a directory (folder).
clear	Clears the CLI window.
exit	Closes the CLI window.
man command	Shows the manual for a given command.

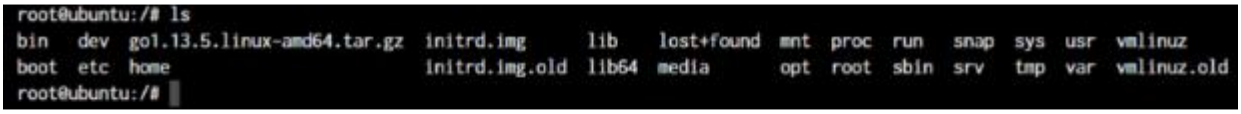
Basic Windows CLI Commands

Command	Description
dir	List the directory (folder) system.
cd pathname	Change directory (folder) in the file system.
cd \	Move to the root folder of the file system.
cd ..	Move one level up (one folder) in the file system.
copy	Copy a file to another folder.
move	Move a file to another folder.
type filename	Type a file.
mkdir or md	Creates a new directory (folder).
rmdir or rd	Removes a directory (folder).
cls	Clears the CLI window.
exit	Closes the CLI window.
help command	Shows the manual for a given command.

DOS Commands Help -You can display all available commands with the help command:

1. ls command

The ls command is commonly used to identify the files and directories in the working directory.

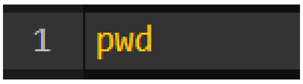


```
root@ubuntu:/# ls
bin   dev  go1.13.5.linux-amd64.tar.gz  initrd.img      lib    lost+found  mnt  proc  run   snap  sys  usr  vmlinuz
boot  etc  home                         initrd.img.old  lib64  media       opt  root  sbin  srv   tmp  var  vmlinuz.old
root@ubuntu:/#
```

2. pwd command

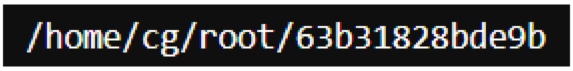
The pwd command is mostly used to print the current working directory on your terminal.

Command:



```
pwd
```

Output:



```
/home/cg/root/63b31828bde9b
```

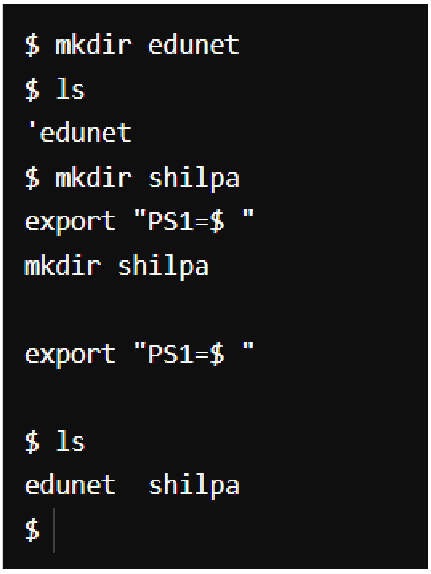
3. mkdir command

This mkdir command allows you to create fresh directories in the terminal itself. The default syntax is mkdir <directory name> and the new directory will be created.

Command:

mkdir Edunet

Output:

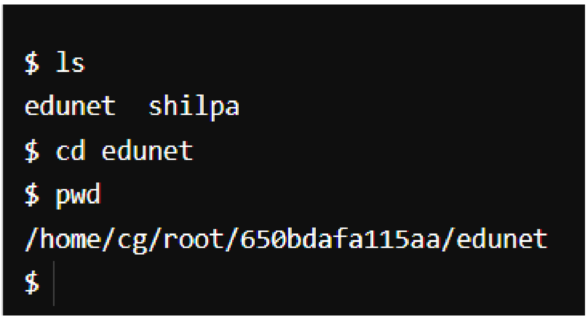


```
$ mkdir edunet
$ ls
'edunet
$ mkdir shilpa
export "PS1=$ "
mkdir shilpa

export "PS1=$ "

$ ls
edunet  shilpa
$
```

You can see we used ls first to see the directories present there and then **mkdir** to create another directory followed by **ls** to view the created directories.

4. cd command

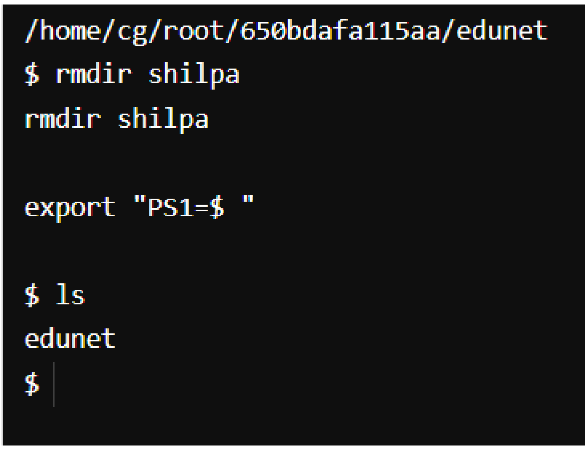
The cd command is used to navigate between directories. It requires either the full path or the directory name, depending on your current working directory. If you run this command without any options, it will take you to your home folder. Keep in mind that it can only be executed by users with sudo privileges.



```
$ ls
edunet  shilpa
$ cd edunet
$ pwd
/home/cg/root/650bdafa115aa/edunet
$
```

5. rmdir command

The rmdir command is used to delete permanently an empty directory. To perform this command the user running this command must be having **sudo** privileges in the parent directory.

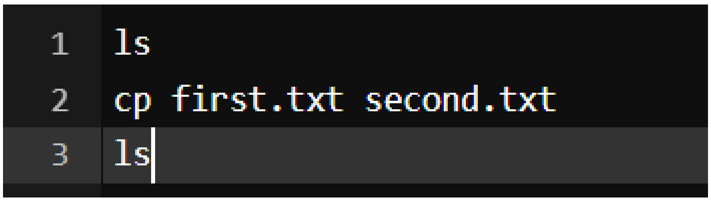


```
/home/cg/root/650bdafa115aa/edunet
$ rmdir shilpa
rmdir shilpa

export "PS1=$ "

$ ls
edunet
$
```

6. cp command

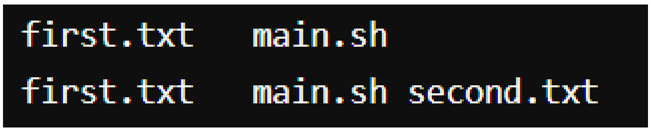
The cp command of Linux is equivalent to copy-paste and cut-paste in Windows.

Command:



```
ls
cp first.txt second.txt
ls
```

Output:



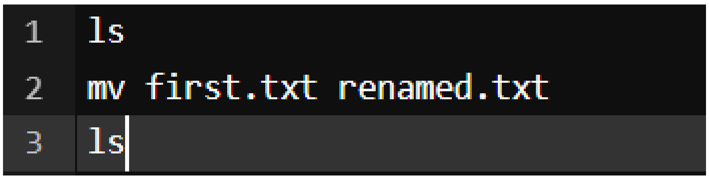
```
first.txt   main.sh
first.txt   main.sh second.txt
```

Here we used **ls** to view the files and then used **cp** to copy the files of ***first.txt*** to ***second.txt*** and again used **ls** command to view the updated files.

7. mv command

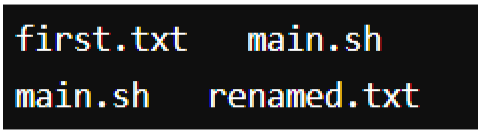
The mv command is generally used for renaming the files in Linux.

Command:



```
ls
mv first.txt renamed.txt
ls
```

Output:



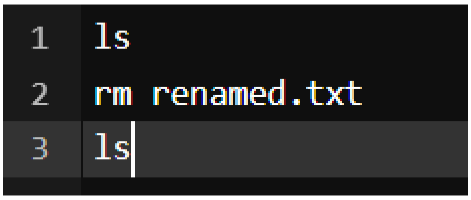
```
first.txt   main.sh
main.sh   renamed.txt
```

Here we used the **ls** command to check the directories and then used **mv <file name> <Renamed file name>** to rename the files, and then again we used the **ls** command to view the renamed file as you can see in the output screenshot.

8. rm command

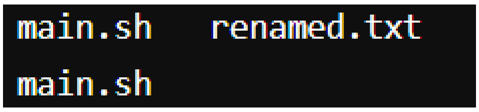
rm command in Linux is generally used to delete the files created in the directory.

Command:



```
ls
rm renamed.txt
ls
```

Output:



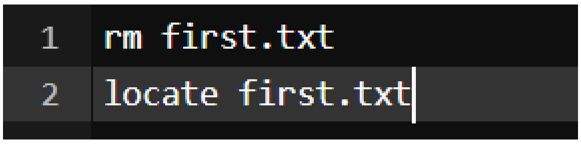
```
main.sh   renamed.txt
main.sh
```

You can see as we wrote the **ls** command to view the files in the terminal and then **rm <file name>** to delete the files and again we had the **ls** command to check the update.

9. locate command

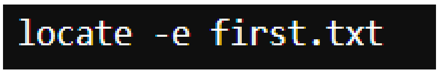
The locate command is generally used to locate the files in the database. Use an asterisk (*) to search for content that contains two or more words. As an example: **locate first*file.** This command will search the database for the files that contain these two names **first** and **file.**

Command:



```
rm first.txt
locate first.txt
```

Output:



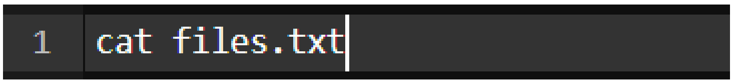
```
locate -e first.txt
```

We first used the **rm** command to delete the file and then used **locate** command to find the file in the database which in return has given the output with a **-e** as the file was removed.

10. cat command

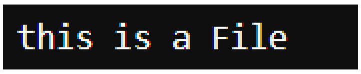
The cat command is the simplest command to use when you want to see the contents of a particular file. The only issue is that it simply unloads the entire file to your terminal. If you want to navigate around a huge file, should use **less** command alternatively.

Command:



```
cat files.txt
```

Output:

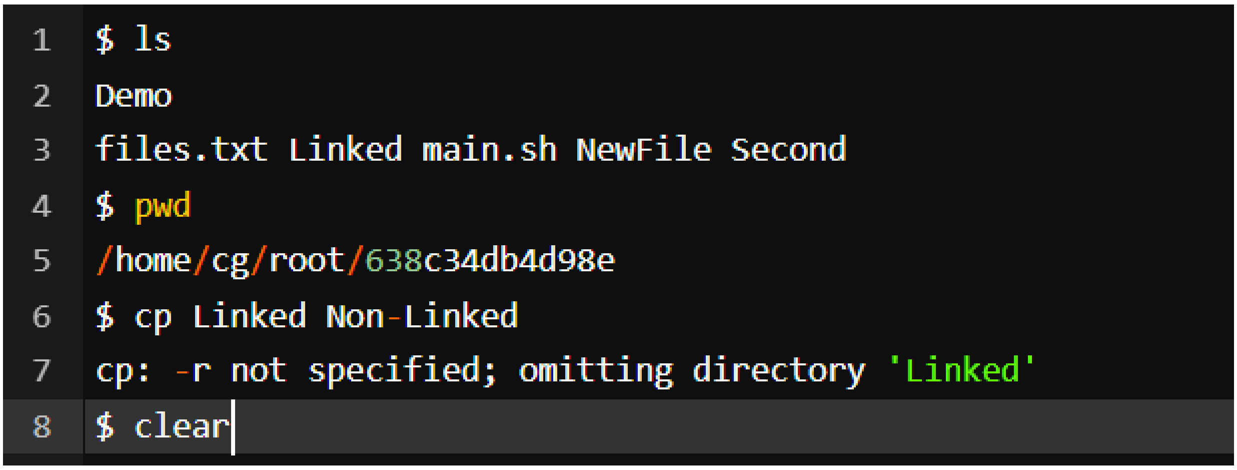


```
this is a File
```

11. clear command

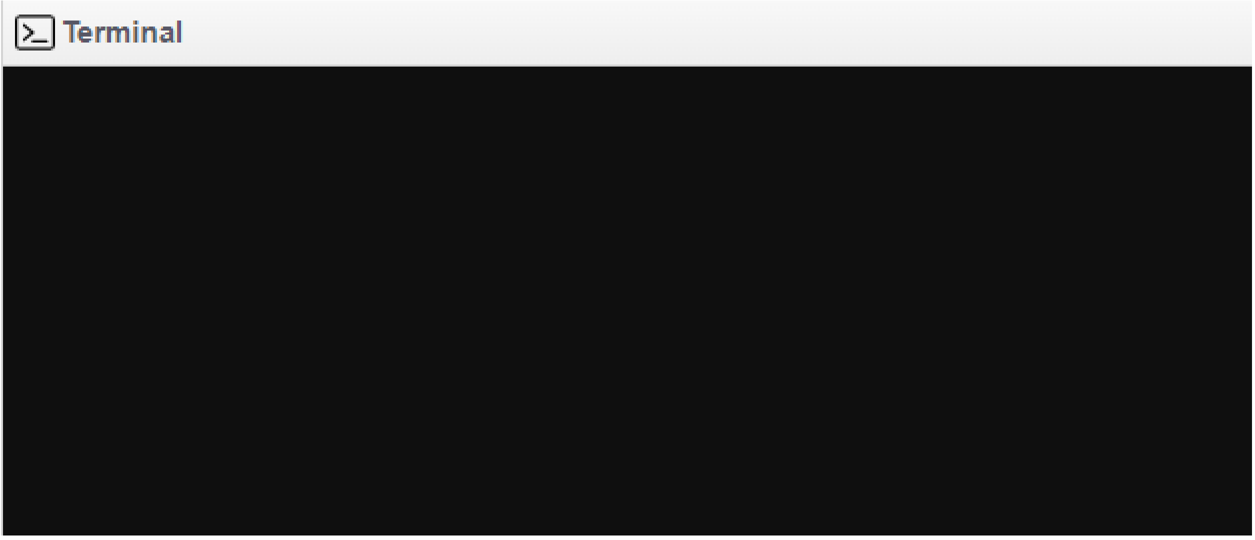
The clear command is a standard command to clear the terminal screen.

Command: *This was the terminal before the command.



```
$ ls
Demo
files.txt Linked main.sh NewFile Second
$ pwd
/home/cg/root/638c34db4d98e
$ cp Linked Non-Linked
cp: -r not specified; omitting directory 'Linked'
$ clear
```

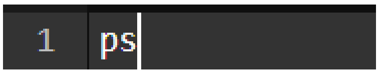
Output:



12. ps command

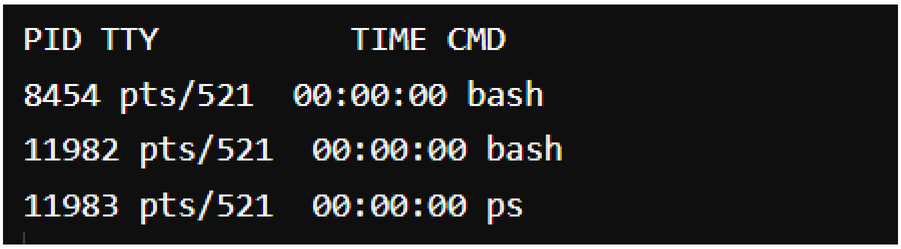
ps command in Linux is used to check the active processes in the terminal.

Command:



```
ps
```

Output:



```
PID TTY          TIME CMD
8454 pts/521  00:00:00 bash
11982 pts/521  00:00:00 bash
11983 pts/521  00:00:00 ps
```

13. man command

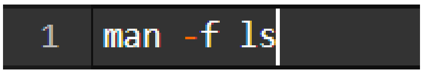
The man command displays a user manual for any commands or utilities available in the Terminal, including their name, description, and options.

Command to view the full manual:

man <command name>

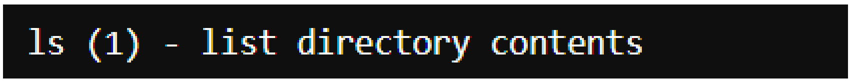
For example, suppose you want to look up the manual for the ls command: **man ls**

Command:



```
man -f ls
```

Output:



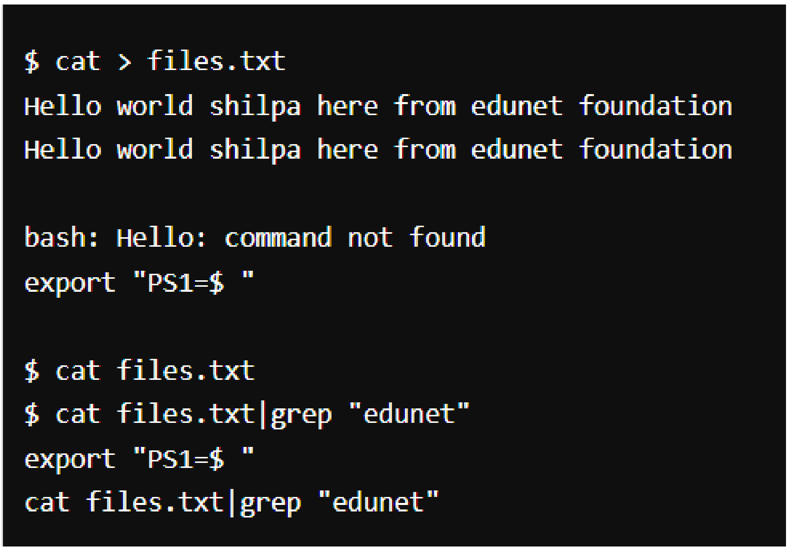
```
ls (1) - list directory contents
```

14. grep command

The grep command is used to find a specific string in a series of outputs. For example, if you want to find a string in a file, you can use the syntax: **<Any command with output> | grep “<string to find> “**

For Example:

cat Files.txt | grep “edunet”

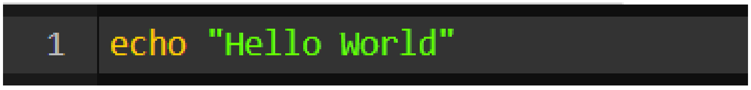


```
$ cat > files.txt
Hello world shilpa here from edunet foundation
Hello world shilpa here from edunet foundation

bash: Hello: command not found
export "PS1=$ "

$ cat files.txt
$ cat files.txt|grep "edunet"
export "PS1=$ "
cat files.txt|grep "edunet"
```

15. echo command

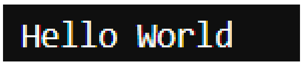
echo command in Linux is specially used to print something in the terminal

Command:



```
echo "Hello World"
```

Output:



```
Hello World
```

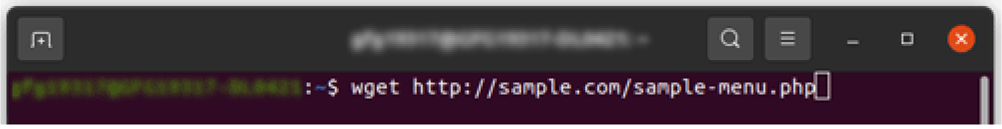
16. wget command

The wget command in the Linux command line allows you to download files from the internet. It runs in the background and does not interfere with other processes.

Here is the basic syntax: **wget [option] [url]**

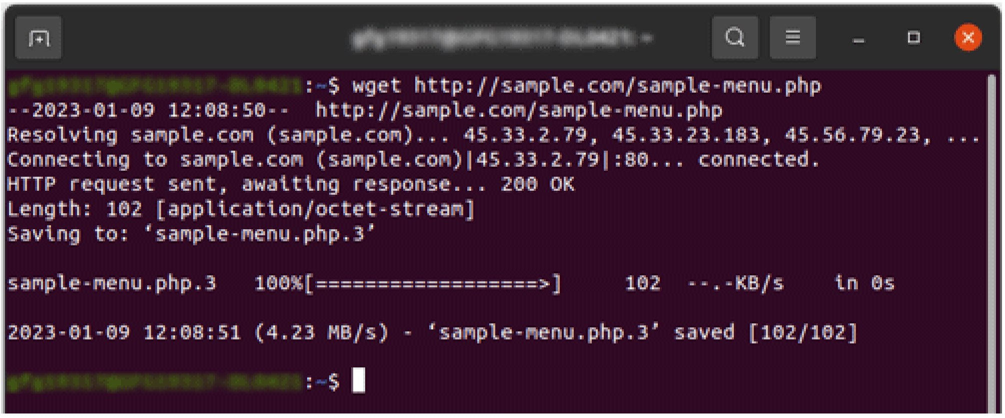
Command:

wget http://sample.com/sample-menu.php



```
[redacted]:~$ wget http://sample.com/sample-menu.php
```

Output:



```
[redacted]:~$ wget http://sample.com/sample-menu.php
--2023-01-09 12:08:50--  http://sample.com/sample-menu.php
Resolving sample.com (sample.com)... 45.33.2.79, 45.33.23.183, 45.56.79.23, ...
Connecting to sample.com (sample.com)|45.33.2.79|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 102 [application/octet-stream]
Saving to: ‘sample-menu.php.3’

sample-menu.php.3   100%[===================>]     102  --.-KB/s    in 0s

2023-01-09 12:08:51 (4.23 MB/s) - ‘sample-menu.php.3’ saved [102/102]

[redacted]:~$ 
```