

Useful Linux Tips & Tricks



Minimizing Keystrokes for linux terminal	3
Turning Off Your System	5
Multiple Commands: When First One Fails	6
Search Command	6
Unfreeze Your Linux terminal from accidental Ctrl+S	7
Displaying an Output As a Table	7
Running Programs After Killing a Session	8
Recording Your Command Line Session	8
Replacing Spaces with Tabs	8
Shutdown Command	9

Minimizing Keystrokes for Linux terminal

Every time a finger touches a computer keyboard and mouse it costs **time** and **money**, so the best way to increase productivity is to reduce the amount of time **spent on typing**.

Example -:-)

1. After making some changes in **github** repo we do(**alias**)

```
git add .  
git commit -m "long method"
```

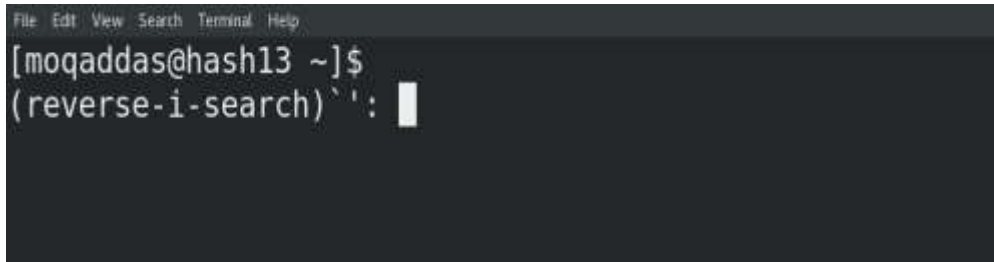
So we can reduce above commands using **alias**

```
alias gc='git add . && git commit -m'
```

So now we can use only **gc** to do the above stuff.

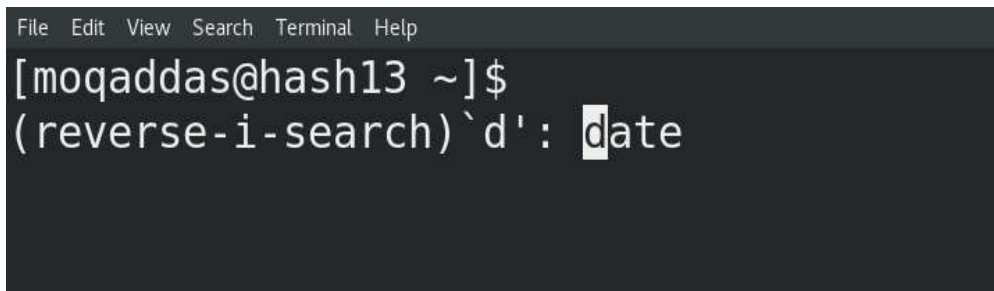
2. Search terminal history with **Ctrl + R** command

In the terminal we write lots of commands with options, so **rather than writing** the same commands again and again we can use some shortcut that will help to **find** the same command **speedily**.




After pressing **Ctrl + R** prompt look like above figure

Now let me press only “d” and see what happened



Here if hit “enter” then see what happened



It run history “date” commands which we run early

So using **ctrl + R** we can use it to find terminal history commands.

3. **Tab -:**) It autocompletes command
4. **Ctrl + C -:**) It aborts program by sending SIGINT signal
5. **Ctrl + L -:**) It work like “clear” command in linux
6. **Ctrl + A and Ctrl + E -:**) Move the cursor to beginning of line and end

```
[moqaddas@hash13 ~]$ ps -aux | grep chrome
```

Here cursor is last so if we want to move cursor to first then we can use option **Ctrl + A**.

```
[moqaddas@hash13 ~]$ ps -aux | grep chrome
```

But I want to move the cursor at last then we can use the option **Ctrl + E**.

```
[moqaddas@hash13 ~]$ ps -aux | grep chrome
```

Turning Off Your System

Did you know you can use certain commands to turn off your system at a specific time? You may or may not be physically present to shut down your computer but you can choose when to. Configure the shutdown time using this command:

```
# shutdown 17:00
```

Your system will shut down exactly at 17:00! Instead of hours, you can also choose minutes.

To cancel shutdown job

```
# shutdown -c
```

Multiple Commands: When First One Fails

In the previous section, we talked about how to use a single command for running several commands. But what should you do in case the first command does not run successfully? You want to run the subsequent command only if the previous one was successful.

For this use “||” separator like shown below:

```
# Command_1 || command_2
```

After this, command 2 will run only after the command 1 when you use the above single-line command.

Search Command

Linux/Unix comes with a huge number of commands and thus it become quite difficult sometimes to remember each and every command. *apropos* command becomes useful in such cases. *apropos* command helps the user when they don't remember the exact command but knows a few keywords related to the command that define its uses or functionality. It searches the Linux man page with the help of the keyword provided by the user to find the command and its functions.

Run once below command to create database

```
# mandb -c
```

Then search by keyword like mail keyword

```
# apropos mail
```

Unfreeze Your Linux terminal from accidental Ctrl+S

You probably are habitual of using Ctrl+S for saving. But if you use that in Linux terminal, you'll have a frozen terminal.

Don't worry, you don't have to close the terminal, not anymore. Just use Ctrl+Q and you can use the terminal again.

ctrl+Q

1. Empty a file without deleting it

If you just want to empty the contents of a text file without deleting the file itself, you can use a command similar to this:

```
# > filename
```

Displaying an Output As a Table

When running commands in the Linux terminal, sometimes their outputs can be quite hard to follow. Overcrowded strings and cluttered information are the reasons for this.

Working through this can prove to be quite challenging, especially when you're pressed for time.

These messy outputs particularly affect system administrators, as they need to efficiently execute operations in the Linux terminal and extract the information swiftly.

We can facilitate this best by utilizing an alternate separator such as colons with the -s option.

```
# cat /etc/passwd | column -t -s :
```

Running Programs After Killing a Session

Closing your shell after running a program will immediately kill it. This limits the amount of multitasking capable on your system. Luckily, there's a way around it. We can use the `nohup` command to achieve this.

This allows for maximum convenience and an optimal environment for all your system administrating and multitasking purposes. The `nohup` stands for "no hangup". So, the function of the command is all in the name.

```
# nohup wget somesite.com/my.png
```

Recording Your Command Line Session

If you want to record what you've typed in your shell screen, you can use the `script` command, which will save all of your typing to a specified file or a file named `typescript` by default.

```
$ script [FILENAME]
```

Replacing Spaces with Tabs

You can use the `tr` command to replace any character with another character. This command is convenient for automating processes like the one in the example below.

Having to replace characters manually can take up a significant amount of time that one could spend working on other things. Use the syntax below for replacing spaces with tabs with the `tr` command.


```
$ cat file.txt | tr ':[space]:' 't' > out.txt
```

Shutdown Command

The shutdown command brings the system down in a **secure way**. When the shutdown is initiated, **all logged-in users and processes are notified** that the system is going down, and **no further logins are allowed**.

Example

Let's see some basic syntax of shutdown command

shutdown [OPTIONS] [TIME] [MESSAGE]

1. We can shutdown os using command

```
[moqaddas@hash13 ~]$ shutdown now
```

Or

```
[moqaddas@hash13 ~]$ shutdown +0
```

2. At specific time how to shutdown os

```
[moqaddas@hash13 ~]$ shutdown 10:20  
Shutdown scheduled for Thu 2022-01-06 10:20:00 IST, use 'shutdown -c' to cancel.  
[moqaddas@hash13 ~]$
```

Syntax : **shutdown hours:minutes** (here hours in **24** hour format)

3. We can also shutdown os only passing **minutes**

```
[moqaddas@hash13 ~]$ shutdown +3  
Shutdown scheduled for Wed 2022-01-05 10:25:39 IST, use 'shutdown -c' to cancel.  
[moqaddas@hash13 ~]$
```

Here shutdown schedule after 3 minutes from current time

4. Halt the system using **shutdown**

```
[moqaddas@hash13 ~]$ sudo shutdown -H  
[sudo] password for moqaddas:  
Shutdown scheduled for Wed 2022-01-05 10:27:51 IST, use 'shutdown -c' to cancel.  
[moqaddas@hash13 ~]$
```

halt instructs the hardware to stop all CPU functions, but leaves it powered on

5. Reboot the system

```
[moqaddas@hash13 ~]$ shutdown -r  
Shutdown scheduled for Wed 2022-01-05 10:29:56 IST, use 'shutdown -c' to cancel.  
[moqaddas@hash13 ~]$
```

6. Power off the system

```
[moqaddas@hash13 ~]$ shutdown -P  
Shutdown scheduled for Wed 2022-01-05 10:30:38 IST, use 'shutdown -c' to cancel.  
[moqaddas@hash13 ~]$
```

7. Cancel shutdown operation

```
[moqaddas@hash13 ~]$ shutdown -P  
Shutdown scheduled for Wed 2022-01-05 18:05:40 IST, use 'shutdown -c' to cancel.  
[moqaddas@hash13 ~]$  
[moqaddas@hash13 ~]$ shutdown -c
```

