Level Up Linux: 20 Advanced Commands for Mid-Level Users

Ravi Saive | Last Updated: February 27, 2024 | Read Time: 10 mins | Linux Commands | 56 Comments

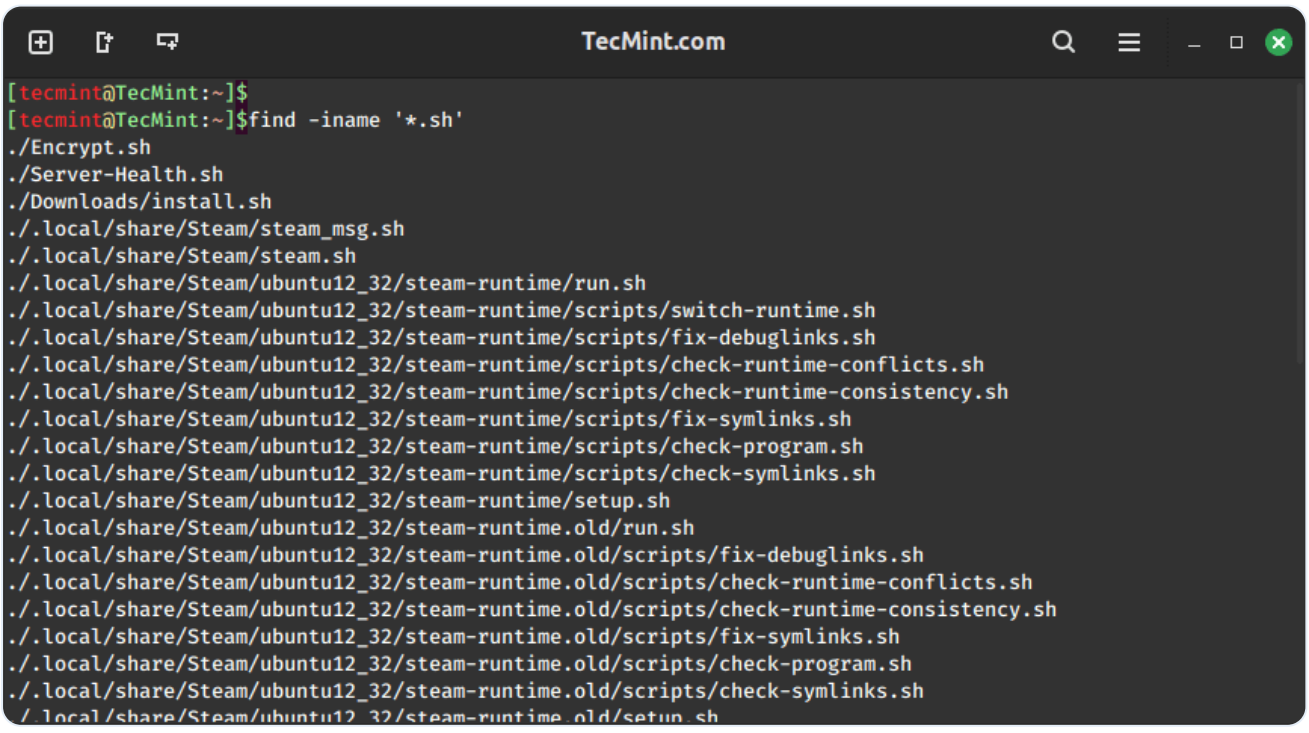
You may have found the first article, 'Useful Commands for Beginners' very helpful, as it was intended for newbies, this article is tailored for middle-level and advanced users.

It covers topics such as customizing search, understanding processes and how to terminate them, optimizing the Linux terminal for productivity, and compiling C, C++, and Java programs in a Unix-like environment.

21. find Command

The find command is used to search for files in the given directory, hierarchically starting at the parent directory and moving to sub-directories.

```
find -name *.sh
```



Find All Files with Extension

The `-name` option makes the search case sensitive. You can use the `-iname` option to find case-insensitive files with different capitalization patterns in the extension.

The `*` is a wildcard and searches all the files having an extension `.sh` you can use a filename or a part of the file name to customize the output.

```
find -iname *.SH
```

The following command is used to search for all files having extension `".tar.gz"` in the current directory and its subdirectories including mounted devices.

```
find -name *.tar.gz
```

22. grep Command

The grep command searches a specified file for lines that contain a match to provided strings or words.

In this case, it is used to search for the 'tecmint' user in the '/etc/passwd' file.

```
grep tecmint /etc/passwd
```

The `-i` option is used to search for the string "TECMINT" (case-insensitive) in the '/etc/passwd' file.

```
grep -i TECMINT /etc/passwd
```

The `-r` option is used to recursively search for the string "127.0.0.1" in the '/etc/hosts' file.

```
grep -r "127.0.0.1" /etc/hosts
```

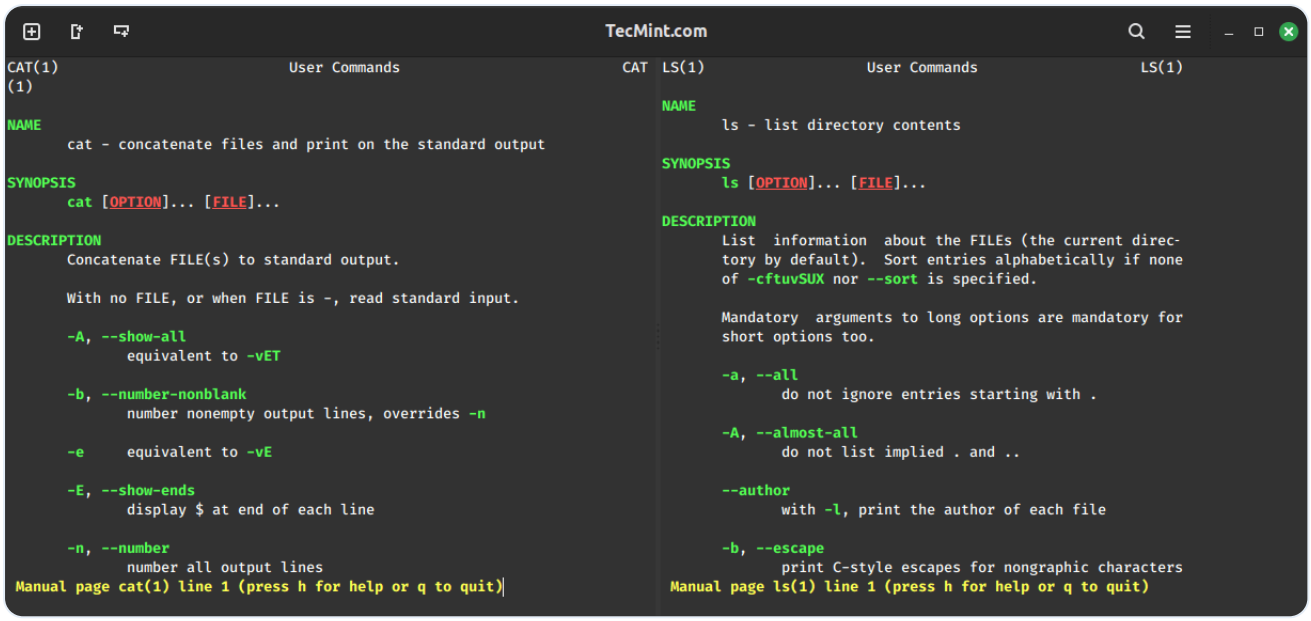
Grep Case-Insensitive String in File

23. man Command

The man command is the system's manual pager, which provides online documentation for all the possible options with a command and its usage.

Almost all the Linux commands come with their corresponding manual pages. For example, the following '**man cat**' (Manual page for cat command) and '**man ls**' (Manual page for command ls) display the manual pages for a given command.

```
man cat
man ls
```



View Command Manual Pages

24. ps Command

The ps command gives the status of running processes with a unique ID called PID.

```
ps
```

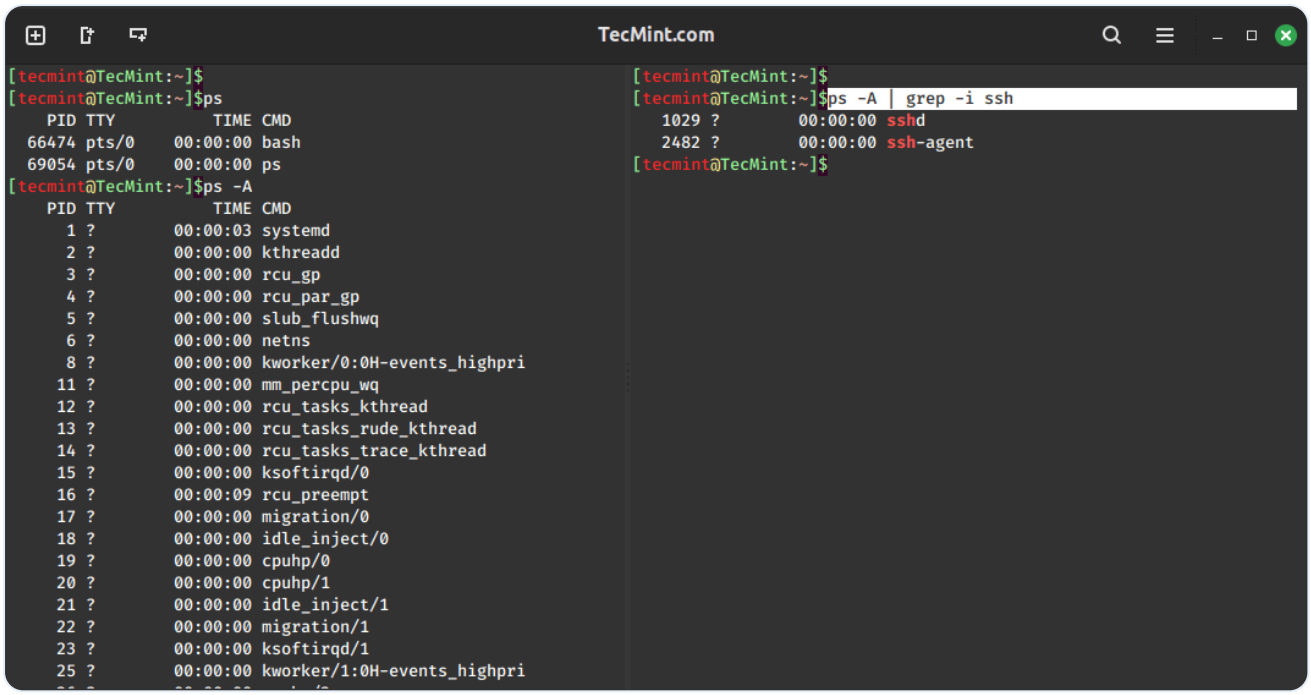
To list status of all the processes along with process ID and PID, use option `-A`.

```
ps -A
```

The ps command is very useful when you want to know which processes are running or may need PID sometimes, for a process to be killed. You can use it with the grep command to find customized output.

```
ps -A | grep -i ssh
```

Here ps is pipelined with grep command to find customised and relevant output of our need.



List Currently Running Processes

25. kill Command

The kill command in Linux is crucial for terminating unresponsive or irrelevant processes efficiently. Unlike Windows, where restarting is often required after killing a process, Linux allows you to kill and restart processes without rebooting the entire system.

For example, if you need to terminate the '**firefox**' program if it's not responding, you can use the **ps** command along with **grep** to find the process pid and then use the '**kill**' command to stop the process.

```
ps -A | grep -i firefox
kill 69881
```

Every time you re-run a process or start a system, a new **pid** is generated for each process and you can know about the currently running processes and their **pid** using the command 'ps'.

Kill Running Processes

Another way to kill the same process is.

```
pkill apache2
```

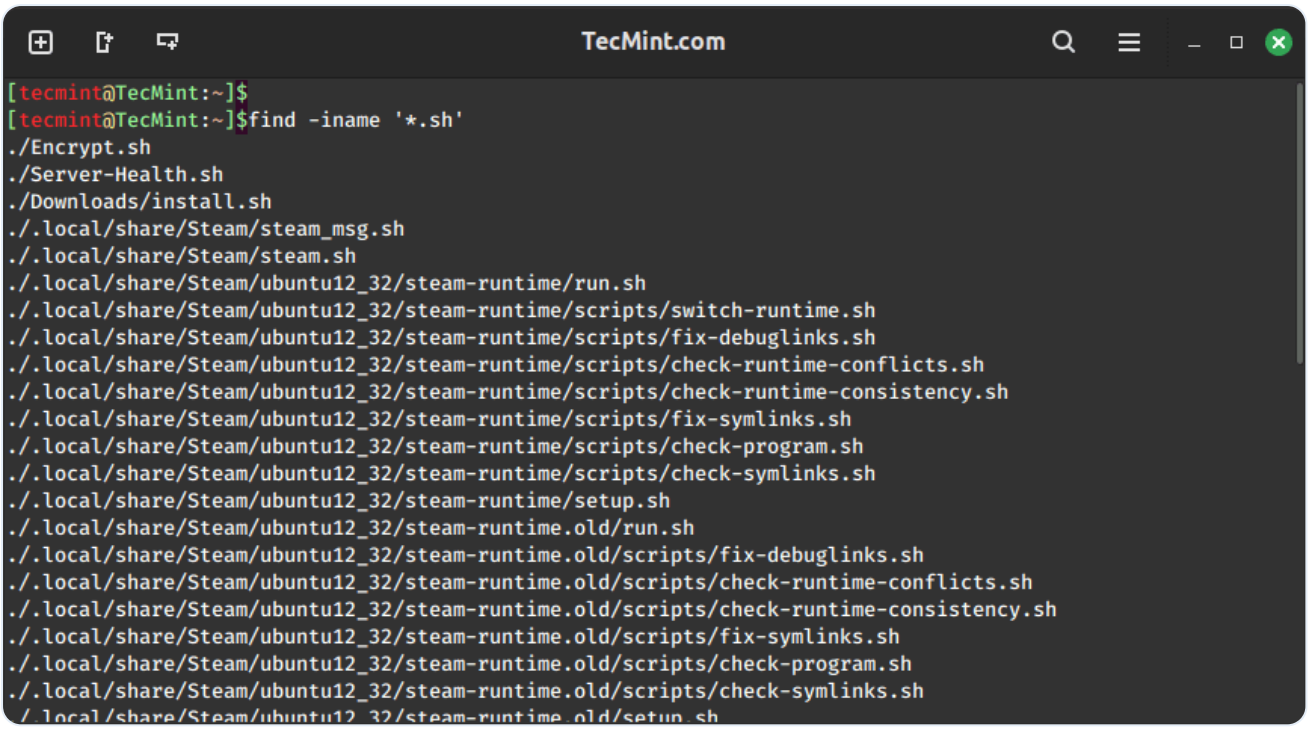
The **kill** command requires **job id/process id** for sending signals, whereas, in **pkill**, you have an option of using a pattern, specifying process owner, etc.

26. whereis Command

The whereis command is used to locate the **Binary**, **Sources**, and **Manual Pages** of the command.

For example, to locate the **Binary**, **Sources**, and **Manual Pages** of the command '**ls**' and '**kill**'.

```
whereis ls
whereis kill
```



Find Command Binary Location

The **whereis** command is useful to know where the binaries are installed for manual editing sometimes.

27. systemctl Command

The systemctl command controls the starting, stopping, restarting, enabling, disabling, and checking of the status of a service or program.

```
sudo systemctl start sshd
sudo systemctl stop sshd
sudo systemctl restart sshd
sudo systemctl enable sshd
sudo systemctl disable sshd
sudo systemctl status sshd
```

28. alias Command

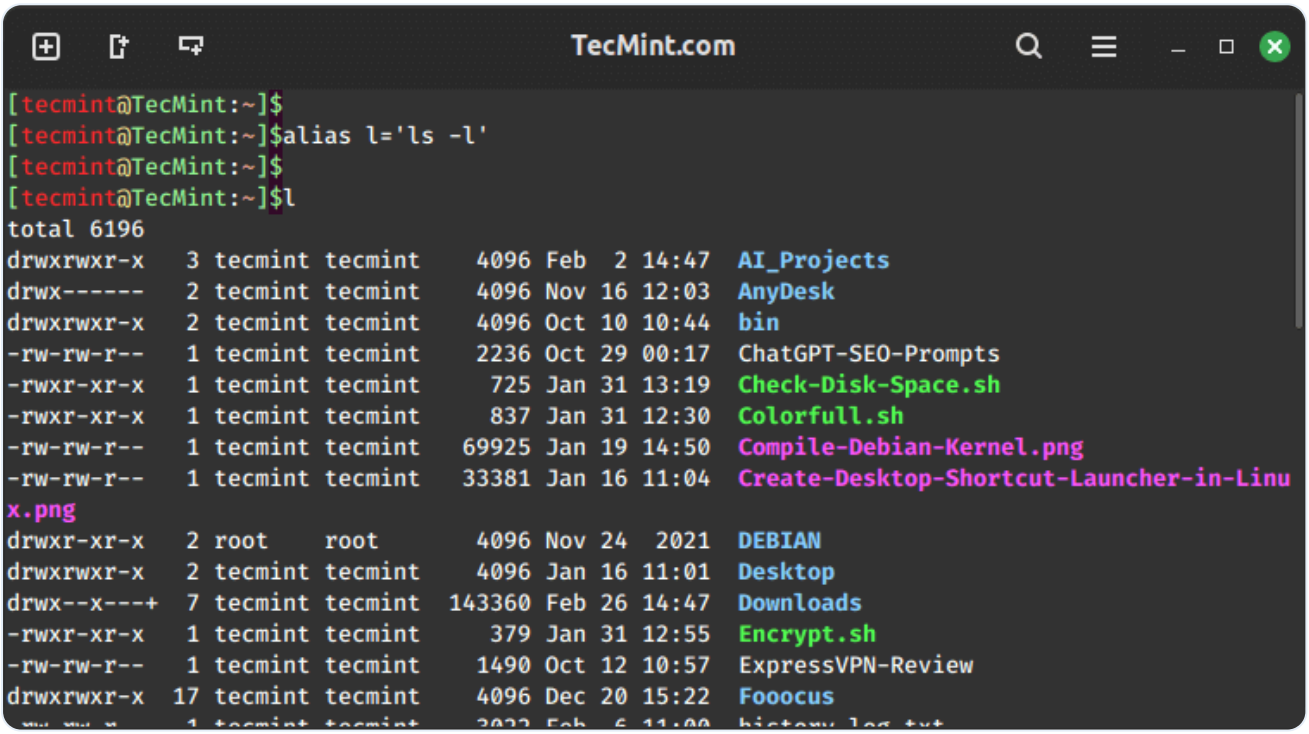
The alias command is a built-in shell command that lets you assign a name for a long command or frequently used command.

I frequently use the 'ls -l' command, which consists of 5 characters, including spaces. Therefore, I created an alias for it as `'l'`.

```
alias l='ls -l'
```

check if it works or not.

```
l
```



Create Command Alias in Linux

To remove alias `'l'`, use the following 'unalias' command.

```
unalias l
```

check, if 'l' still is an alias or not.

```
l
```

```
l: command not found
```

Adding a bit of fun to Linux commands by creating aliases for specific important commands to other important commands.

```
alias cd='ls -l' (set alias of ls -l to cd)
alias su='pwd'   (set alias of pwd to su)
```

Now, imagine the humor when your friend types the cd command, expecting to change directories but instead gets a directory listing. Similarly, if he attempts 'su', all he sees is the location of the working directory.

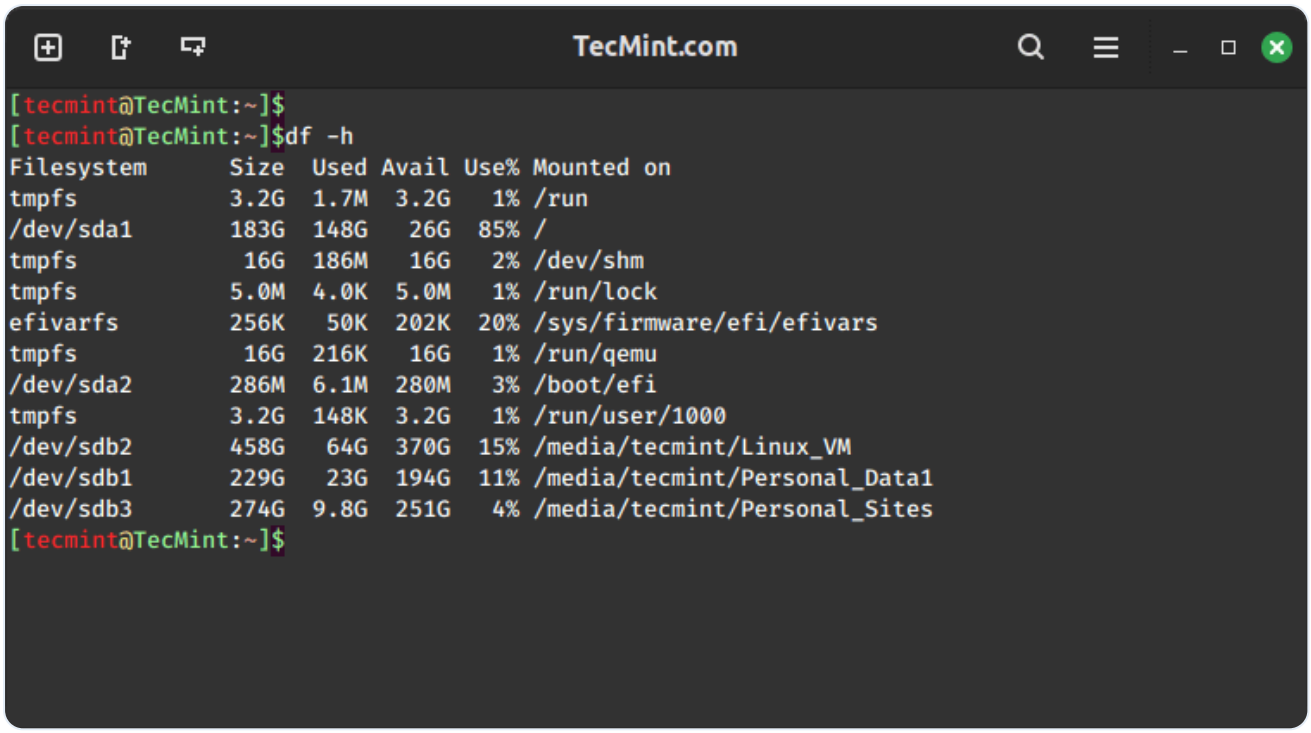
You can remove the alias later using the 'unalias' command, as explained above.

29. df Command

The df command is used to show the information about disk space usage on the file system. It shows the total, used, and available space on each mounted file system.

```
df -h
```

The `-h` option is used to print the disk space usage in a human-readable format, showing sizes in gigabytes (GB) and megabytes (MB) for each mounted file system on your system.



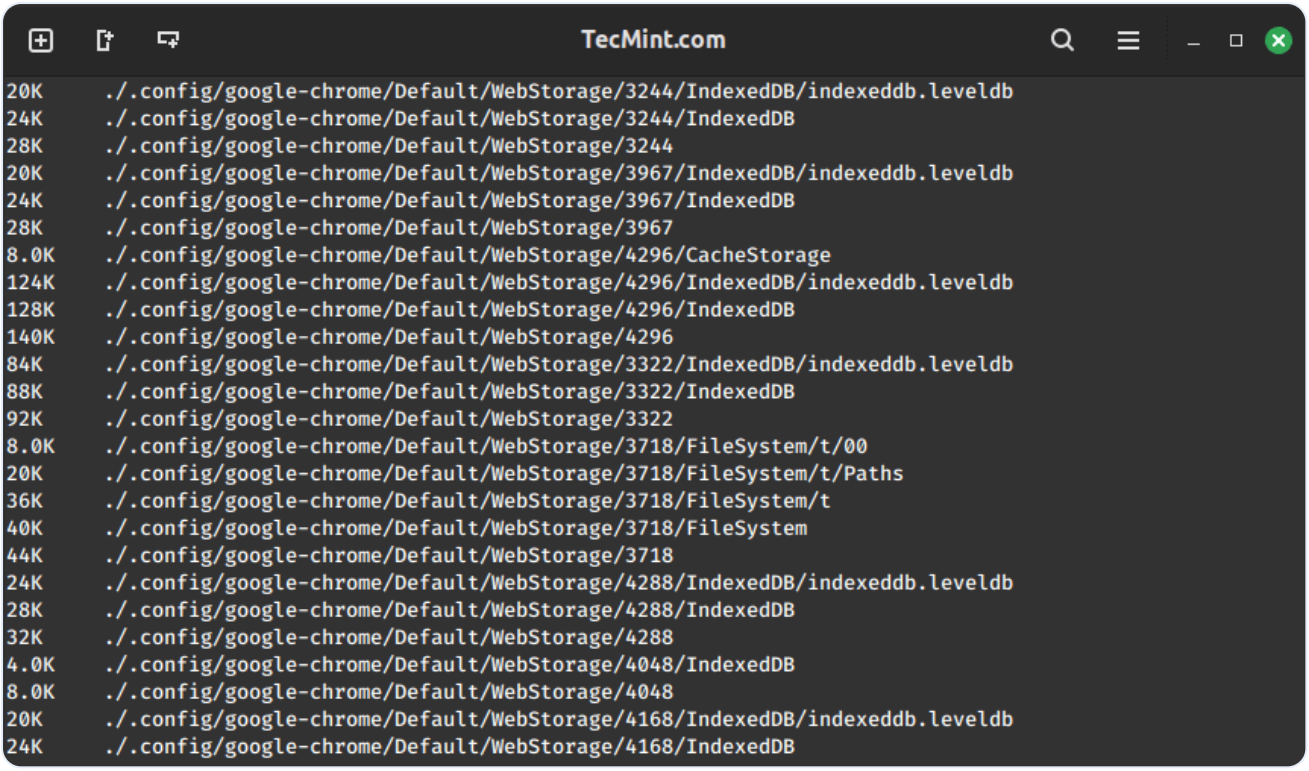
Show Linux Disk Usage

30. du Command

The du command is used to show the disk space usage of files and directories, which includes the total disk space occupied by a specific file or directory, including the space used by its subdirectories.

```
du -h
```

The `-h` option is used to print the file usage in a human-readable format, showing sizes in gigabytes (GB) and megabytes (MB).



Show File Disk Usage

31. rm Command

The rm command stands for remove, which is used to remove or delete files and directories permanently from the file system.

The basic syntax for removing a file is:

```
rm file
```

The basic syntax for removing a directory is:

```
rm -rf directory
```

The `-r` (recursive, removes directories and their contents) and `-f` (force remove files without prompts for confirmation).

The `"rm -rf"` command is a destructive command. If you accidentally execute it in the wrong directory, all files and the directory itself are permanently lost.

32. echo Command

The echo command as the name suggests echoes a text on the standard output. It has nothing to do with the shell, nor does the shell read the output of the echo command.

However, in an interactive script, an echo passes the message to the user through the terminal. It is one of the commands that is commonly used in scripting, interactive scripting.

```
echo "Tecmint.com is a very good website"

Tecmint.com is a very good website
```

Let's create a small interactive bash script that will display a personalized welcome message on the terminal.

```
#!/bin/bash

echo "Welcome to the Interactive Welcome Script!"
echo "----------------------------------------"

# Prompt the user to enter their name
echo "Please enter your name:"
read name

# Display a personalized welcome message
echo "Hello, $name! Welcome to the interactive script. Have a great day!"
```

Save this script in a file, for example, **welcome_script.sh**, and make the script executable using the command.

```
chmod +x welcome_script.sh
```

Then, you can run it by typing in the terminal.

```
./welcome_script.sh
```

Interactive Bash Script

33. passwd Command

The **passwd** command is used to change own password or another user's password when executed by the sudo privileges.

For example, to change the password for the current user, simply type:

```
passwd
```

If you have the sudo privileges, you can change another user's password by specifying the username:

```
sudo passwd username
```

34. lpr Command

The **lpr** command is used for submitting print jobs to a printer. It sends files to a printer's print queue, allowing users to print documents from the command line.

```
lpr document.txt
```

The 'lpq' command lets you view the status of a printer (whether it's up or not), and the jobs (files) waiting to be printed.

35. cmp Command

The **cmp** command compares two files of any type and writes the results to the standard output. By default, '**cmp**' returns 0 if the files are the same; if they differ, the byte and line number at which the first difference occurred is reported.

To provide examples for the **cmp** command, let's consider two files:

```
cat file1.txt

Hi My name is Tecmint
```

```
cat file2.txt

Hi My name is tecmint [dot] com
```

Now, let's compare two files and see the output of the command.

```
cmp file1.txt file2.txt

file1.txt file2.txt differ: byte 15, line 1
```

36. wget Command

The wget command is a free utility for non-interactive (i.e., can work in the background) download of files from the web. It supports HTTP, HTTPS, FTP protocols, and HTTP proxies.

For example, to download a file named "Server-Health.sh" from a website, you would use:

```
wget https://www.tecmint.com/wp-content/scripts/Server-Health.sh
```

37. mount Command

The **mount** command is used to mount a filesystem that doesn't mount itself. You need root permission to mount a device.

First, run '**lsblk**' after plugging in your filesystem and identify your device, and note down your device's assigned name.

```
lsblk

NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0 931.5G  0 disk
├─sda1   8:1    0 923.6G  0 part /
├─sda2   8:2    0     1K  0 part
└─sda5   8:5    0   7.9G  0 part [SWAP]
sr0     11:0    1  1024M  0 rom
sdb      8:16   1   3.7G  0 disk
└─sdb1   8:17   1   3.7G  0 part
```

From this screen it was clear that I plugged in a 4 GB pendrive thus '**sdb1**' is my filesystem to be mounted. Become a **root** to perform this operation and change to the **/dev** directory where all the file system is mounted.

```
su
cd /dev
```

Create a directory named anything that should be relevant for reference.

```
mkdir usb
```

Now mount filesystem '**sdb1**' to directory '**usb**'.

```
mount /dev/sdb1 /dev/usb
```

Now you can navigate to **/dev/usb** from the terminal or **X-windows** system and access files from the mounted directory.

38. gcc Command

The gcc is the in-built compiler for the 'c' language in the linux environment. A simple c program, save it on your desktop as Hello.c (remember the '.c' extension is a must).

```
#include <stdio.h>
int main()
{
  printf("Hello world\n");
  return 0;
}
```

Next, compile and run it.

```
gcc Hello.c
./a.out

Hello world
```

On compiling a c program the output is automatically generated to a new file "a.out" and every time you compile a c program same file "a.out" gets modified.

Hence it is good advice to define an output file during compilation and thus there is no risk of overwriting to output file.

```
gcc -o Hello Hello.c
```

Here '-o' sends the output to the 'Hello' file and not 'a.out'.

39. g++ Command

The g++ is the in-built compiler for 'C++' , the first object-oriented programming language. A simple C++ program, save it on your desktop as Add.cpp (remember the '.cpp' extension is a must).

```
#include <iostream>

using namespace std;

int main()
    {
          int a;
          int b;
          cout<<"Enter first number:\n";
          cin >> a;
          cout <<"Enter the second number:\n";
          cin>> b;
          cin.ignore();
          int result = a + b;
          cout<<"Result is"<<"  "<<result<<endl;
          cin.get();
          return 0;
     }
```

Next, compile and run it.

```
g++ Add.cpp
./a.out

Enter the first number:
...
...
```

On compiling a C++ program the output is automatically generated to a new file "a.out" and every time you compile a C++ program same file "a.out" gets modified.

Hence it is good advice to define an output file during compilation and thus there is no risk of overwriting to output file.

```
g++ -o Add Add.cpp
./Add
```

```
Enter the first number:
...
...
```

40. java Command

Java is one of the world's highly used programming languages and is considered fast, secure, and reliable. Most of the web-based service of today runs on Java.

Create a simple Java program by pasting the below test to a file, named **tecmint.java** (remember the '**.java**' extension is a must).

```
class tecmint {
  public static void main(String[] arguments) {
    System.out.println("Tecmint ");
  }
}
```

Next, compile and run it.

```
javac tecmint.java
java tecmint
```

Almost every distribution comes packed with a **gcc compiler**, major number of distros have inbuilt **G++** and **Java compilers**, while some may not. You can apt or yum the required package.

Don't forget to mention your valuable comment and the type of article you want to see here. I will soon be back with an interesting topic about the lesser-known facts about Linux.

> Hey TecMint readers,

Exciting news! Every month, our top blog commenters will have the chance to win fantastic rewards, like free Linux eBooks such as RHCE, RHCSA, LFCS, Learn Linux, and Awk, each worth $20!

Learn more about the contest and stand a chance to win by sharing your thoughts below!

PREVIOUS ARTICLE:

Top 5 Command Line Browsers for Linux

NEXT ARTICLE:

6 Most Notable Open Source Centralized Log Management Tools

Ravi Saive

I am an experienced GNU/Linux expert and a full-stack software developer with over a decade in the field of Linux and Open Source technologies

Each tutorial at TecMint is created by a team of experienced Linux system administrators so that it meets our high-quality *standards*.

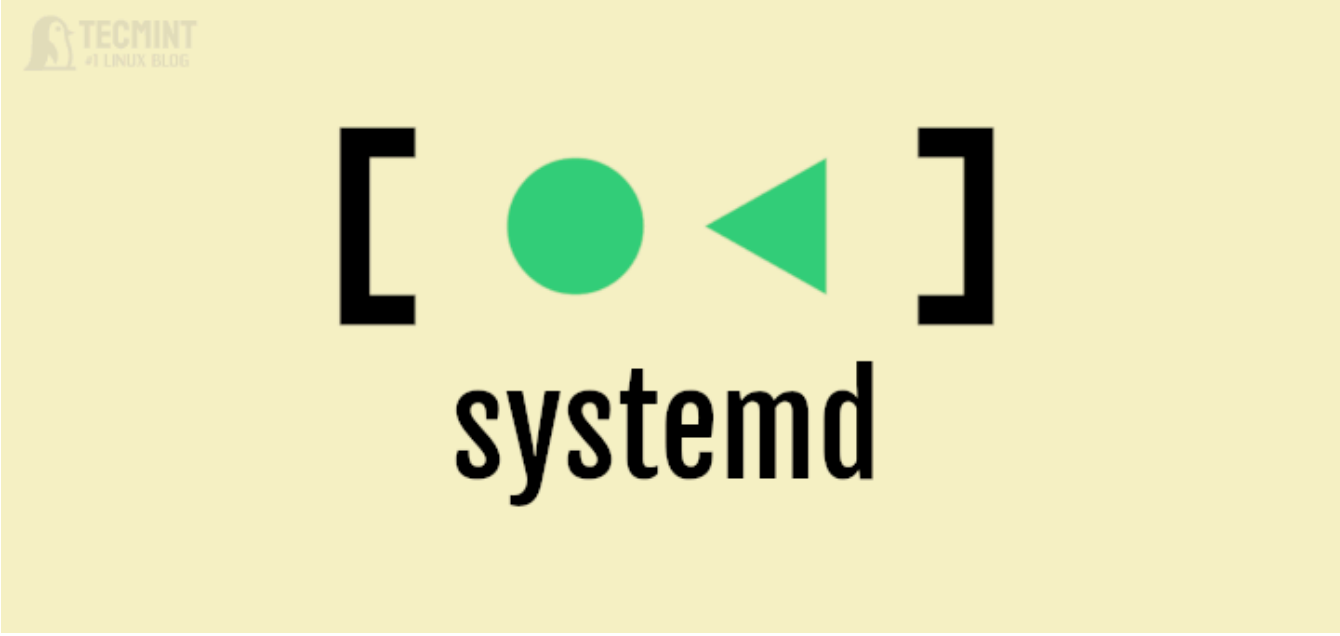
Related Posts

How to Search Files by Name or Extension Using find Command

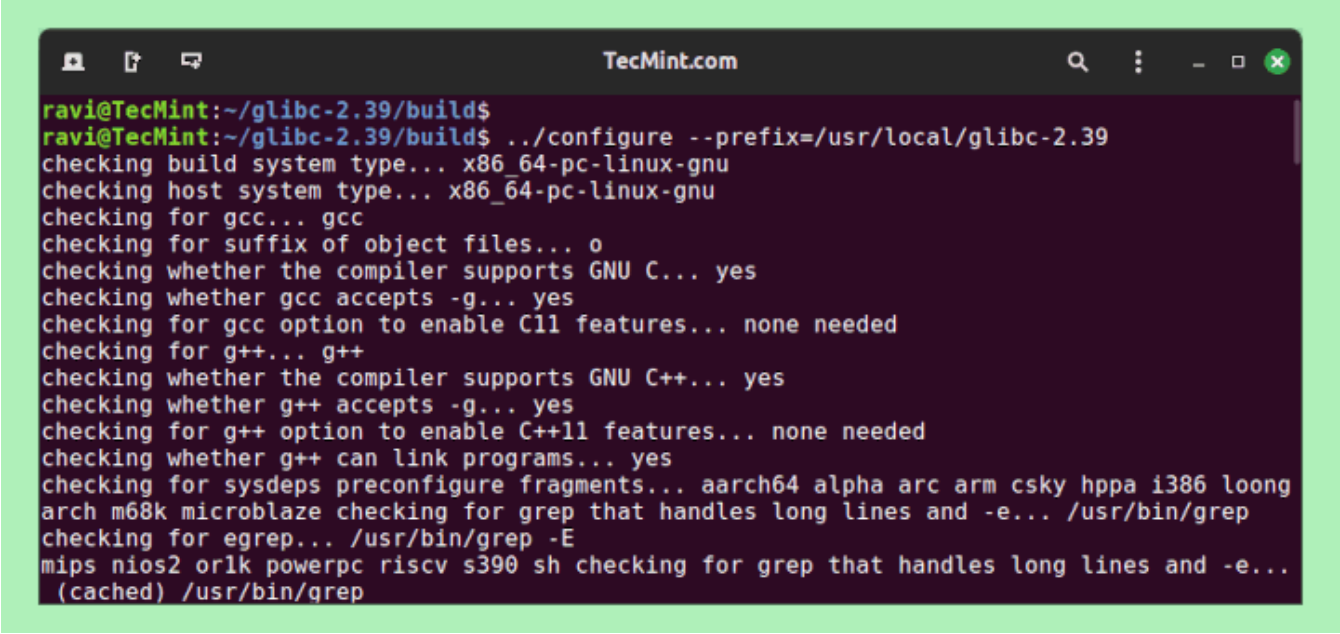
10 Lesser Known Linux Commands – Part 2

11 Lesser Known Useful Linux Commands

26 Security Hardening Tips for Modern Linux Servers



How to Remove Systemd Services on Linux



How to Install and Run Multiple glibc Libraries in Linux

56 Comments

Leave a Reply

juan

June 1, 2017 at 9:49 pm

Good tutorial about Linux commands, but you cannot say something like for ‘C++‘, the first object oriented programming language” check the information before, please.

Reply

namecentos

December 8, 2016 at 1:46 pm

Great tutorial, useful commands and clear explanation. Benefited a lot. Thanks for this article.

Reply

Kristen Nygaard

November 27, 2015 at 1:42 am

C++ was not the first OO programming language
https://en.wikipedia.org/wiki/Simula

Reply

srinivas

October 9, 2015 at 2:55 pm

i have created 2 files named file1 & file2
in file 1 i had given : abcd
1234
in file2 i had given : abc
123
when i tried to execute cmp file1 file2
it is showing file1 file2 differ line 1, byte 4
why it is not showing difference in line 2 ???

Reply

lw_leecher
November 9, 2016 at 11:00 pm

You can read the manual “man cmp” if you like, I don’t really know how does this command work, maybe and i say maybe it shows you just the first different byte but you can use this command “diff” or “sdiff”. have a nice day.

Reply

Anns
May 13, 2018 at 3:25 pm

Because it shows first difference and stops comparing. First difference occurs at first line = abcd

Reply

Louis Dominguez

September 21, 2015 at 10:49 am

Im glad I found this, most of the articles I read did not help but this got me there and made me learn new commands as well.
thanks for your effort and i hope you keep giving back the IT community!

Reply

Dragos

September 4, 2015 at 6:35 pm

Systemd new system and service manager with systemctl a new way to manipulate services on Centos7/Redhat7.
ex: systemctl enable/disable/start/restart, etc….network.sevice.
Great work. Learning a lot from you.
Thanks!!!!!!!!!!!!!!!!

Reply

Rakesh Gosai

January 17, 2015 at 3:46 pm

indeed helpfull,
i must appreciate your effort.

Reply

Karthik

December 12, 2014 at 6:43 am

Thank you for this article. It really helps us newbies! :)
Avishek, could you please suggest good books/learning material for beginners?

Reply

Prado

October 9, 2014 at 1:18 pm

Thanks Avishek,

Great work! I somehow lost touch of some of these commands. Your tutorial is a refresher to me. Keep up the good work

Reply

Avishek Kumar

October 9, 2014 at 10:18 pm

Welcome Prado

Reply

Avishek Kumar

October 2, 2014 at 3:06 pm

Thanks

Reply

Frank

September 29, 2014 at 6:22 pm

Thanks for the useful commands, I had forgotten some and your article helped me relearn the basics again. Cheers

Reply

Avishek Kumar

October 2, 2014 at 3:06 pm

Tecmint is pleased to know this.

Reply

Dipak

September 12, 2014 at 3:37 pm

awesome material.

Reply

Avishek Kumar

October 2, 2014 at 3:07 pm

Thanks

Reply

Shamjith

September 12, 2014 at 4:51 am

This is really helpful, Helps a lot to refresh my knowledge

Reply

Avishek Kumar

October 2, 2014 at 3:08 pm

Thanks

Reply

Srinivas

August 21, 2014 at 3:28 pm

Really Useful stuff :) Good job :)

Reply

Avishek Kumar

August 23, 2014 at 3:28 pm

Thanks @ Srinivas for your feedback.

Reply

Alok

July 25, 2014 at 2:58 am

Nice and Very useful commands!

Reply

Avishek Kumar

July 25, 2014 at 3:32 pm

Our Pleasure @ Alok

Reply

arunprasad

July 5, 2014 at 2:42 pm

thank you very much for ur wonderful lecture and I want to know more abt Linux and ethical hacking could you refer the best books and ideas

Reply

Avishek Kumar

July 9, 2014 at 12:12 pm

Dear arunprasad,
We work to bring to you Linux How-TO, Linux/Package/Tool Installation, Shell Scripting, and every other Genre of Linux but we at this point of time and in recent future have no plan to deal in Ethical Hacking.

Reply

Tapan Kumar Samal

May 21, 2014 at 7:15 pm

Its Very useful commands For Linux User!!!!

Reply

Avishek Kumar

July 9, 2014 at 12:13 pm

Dear Tapan,
Thanks for the recognition.

Reply

sushant
April 28, 2014 at 10:58 pm

Hi Avi,

thanks so much for all ur last Q&A interview series and bash learning series.would u please extend ur series with new how configure DNS & DHCP servers in deep level, i believe we will get so many questions and answers and as of like me new sysadmins have so much issues , trouble to configure this servers :) !!

Reply

Avishek Kumar
July 9, 2014 at 12:14 pm

Dear sushant,
keep connected we will be coming up with your recommendation very soon and don't forget to remind us, if we forget.

Reply

James
February 24, 2014 at 8:30 am

very aswsome! thanx for providing lots of informations about linux always!

Reply

Avishek Kumar

July 9, 2014 at 12:15 pm

Dear James,

Thanks for such a WOW-comment :)

Reply

Mark Dean

December 30, 2013 at 10:37 pm

Good deal and good stuff (although I'd agree that a lot of this is more beginner than advanced but all good). Regarding file name extensions as in your example of the .sh, that requirement is a Windows/DOS construct and not a *nix one. There is no requirement that a shell script have a .sh at the end. It is a convention that some/many distros recognize and a good practice but it is not required. A script is executable by setting the execute bit and not by its extension. So myscript.sh and myscript both are valid as is myscript.bob or myscript.great-it doesn't matter what the extension is or even if it has one.

Also, in order for it to work, the first line *must* have the so-called she-bang and path to the shell/binary you want to handle the script as in #!/bin/bash or #!/bin/ksh etc. or other binary if using perl, #!/usr/bin/perl or python and so on. Without this it will not work.

Now, maybe distros are getting smarter and assuming a .sh means use the default shell or X is assuming that but it is a bad practice to use assumptions like that.

Also, using chmod 777 is a dangerous thing unless you are sure that you want *everyone* and *everything* on the system to be able to read/write/execute the file/script. A better way is to use chmod 755 or 775 or 764 — really it depends on what you want you and others on the system to be able to do. Or simply make it executable via the alpha way–chmod +x and leave other permissions as they are. Make sure that you do a ls -l and review what the permissions are.

Reply

Mark Dean

December 30, 2013 at 10:49 pm

Oops, I meant chmod a+x to make it executable and leave the other permissions as they are...

Reply

Avishek Kumar

January 2, 2014 at 3:03 pm

@ Mark Dean, Your concern is very important, and we welcome your views. we did above 777, just to ensure that a newbie dont get trapped into any kind of permission issue.

Reply

ikarus

December 23, 2013 at 4:05 pm

Thanks Avishek

This one actually works for me. Very useful. Looking forward to the rest.

Regards
IK

(beginner for 10 years)

Reply

Avishek Kumar

December 25, 2013 at 1:37 pm

Welcome @ ikarus, Thanks for your feedback. All of articles of this series has already been published, and is highly appreciated by our reader. Please find the links below, to navigate to other articles of this series.

https://www.tecmint.com/useful-linux-commands-for-newbies/
https://www.tecmint.com/20-advanced-commands-for-linux-experts/

other suggested readings are:
https://www.tecmint.com/20-funny-commands-of-linux-or-linux-is-fun-in-terminal/
https://www.tecmint.com/chaining-operators-in-linux-with-practical-examples/

Reply

Yaro

October 31, 2013 at 12:31 pm

Not a perfect list, some of these I'd actually consider beginner-level stuff, but I wanna go over a few things.

"sudo" is not guaranteed to be on a Linux system. It's one of my personal essentials but most Linux distributions outside of those centered almost exclusively on desktops actually don't install it by default. They figure any administrator knows how to become root without sudo. In my opinion its still a good idea to have sudo since it can protect you from yourself. (Heck, you could just do a su root -c " for roughly the same effect as sudo.)

gcc and g++ are not built in commands. In fact, a lot of distributions don't install them by default and many others use alternatives like egcc or even llvm (Somehow.). Like sudo, they are on my essentials list.

A shame your daemon commands here are Ubuntu-centric. Upstart is considered almost universally in the *nix sphere to be an abysmal init replacement compared to systemd or openrc. Largely because it has a completely backwards unit dependency system (It makes no sense to launch EVERY unit possible that depends on a unit you launch, instead it makes more sense to NOT launch a particular unit unless explicitly asked for or if required by another unit.), but either way,for a general "Linux" command listing it's bad form to suggest anything more than perhaps "/etc/init.d/ " with no command to invoke the service.

Reply

Ilyas Sharif

October 29, 2013 at 2:10 am

Useful commands for beginners, you have written “service /etc/init.d/apache2 start”. while it should be “/etc/init.d/apache2 start” you can’t use service while using /etc/init.d/

Reply

Author

Ravi Saive

October 29, 2013 at 4:56 pm

Dear Muzhda,

Thanks corrected in write-up.

Reply

Hitesh Patel

August 28, 2013 at 7:24 am

Very useful commands!

Reply

Avishek Kumar

January 2, 2014 at 3:01 pm

thanks @ Hitesh Patel, for Your wonderful Feedback.

Reply

Sabounchi

August 20, 2013 at 10:25 pm

How can I run and compile visual c on Ubuntu
System

Reply

Avishek Kumar

July 9, 2014 at 12:19 pm

Dear Sabounchi, I am not sure if it is going to work or not but you may try wine. Although i don't think it is going to work as i have an instinct you might face library issue.

The only option you are left with is to get Virtualbox.

Reply

Rick Stanley

August 8, 2013 at 11:07 pm

@Anonymous

All the commands are available in all Linux Distros, with one exception. sudo is not turned on in all Distros as it is for Ubuntu. Please keep in mind that Ubuntu is just

another Linux Distro, and not some unique O/S by itself!

Reply

Anonymous

August 8, 2013 at 10:49 am

There are good explanations, but it's sad that it is so much Ubuntu-centric.

Reply

Avishek Kumar

January 2, 2014 at 3:00 pm

@ Anonymous, it is non-centric. All the command runs on all the machine.

Reply

kenneth karlsson

August 8, 2013 at 9:09 am

Your example of using find will not work. You have to quote the text if you use a meta character.

find . -name abc will work
find . -name *abc will not work

find . -name "*abc" will work

Reply

tim

August 8, 2013 at 6:48 am

removing dir should be rmdir not rm -rf imo, or the title should be changed

Reply

Michael Belisle

June 4, 2013 at 9:45 am

I tried those example programs. They worked except for the g++ program.
The errors I received:
Add.cpp:1:1: error: ‘include’ does not name a type
include
^
Add.cpp: In function ‘int main()’:
Add.cpp:9:11: error: ‘cout’ was not declared in this scope
cout<> a;
^
Add.cpp:15:44: error: ‘endl’ was not declared in this scope
cout<<"Result is"<<" "<<result<<endl;

Reply

Avishek

June 4, 2013 at 3:42 pm

@ Michael Belisle, sorry! that was an error on our part, we forgot to put pre-processor directive (#).

Please paste the above code again and then compile it and run. Let us know if the problem is solved or still persisting.

Reply

Mahesh

June 3, 2013 at 7:18 pm

Explanation is too simple to understand. I learnt few more things here. Would love to know more :)
Please keep posting.

Reply

Avishek Kumar

January 2, 2014 at 2:59 pm

Thanks @ Mahesh, for such a wonderful feedback.

Reply

ilaiyaraja

May 28, 2013 at 5:46 pm

it's very useful commands for linux users

Reply

Avishek Kumar

January 2, 2014 at 2:57 pm

Thanks @ ilaiyaraja, for the recognition.

Reply

Avishek

May 28, 2013 at 3:13 pm

Thanks Mahabir, Your response to our hard-work means a lot and encourages us to write more.

Reply

Mahabir

May 28, 2013 at 9:32 am

This is one of the best Article on linux commands with very simple examples easily understandable

Reply

Got Something to Say? Join the Discussion...

Thank you for taking the time to share your thoughts with us. We appreciate your decision to leave a comment and value your contribution to the discussion. It's important to note that we moderate all comments in accordance with our comment policy to ensure a respectful and constructive conversation.

Rest assured that your email address will remain private and will not be published or shared with anyone. We prioritize the privacy and security of our users.

Name *

Email *

Save my name, email, and website in this browser for the next time I comment.

Post Comment

Search...

Linux Commands and Tools

Ntfy – Get Desktop or Phone Alerts When Long Running Command Finishes

15 Useful 'Sockstat Command Examples' to Find Open Ports in FreeBSD

How To Create a Linux Swap File

How to Create and Execute a .Jar File in Linux Terminal

16 Top Command Examples in Linux [Monitor Linux Processes]

How to Find and Remove Duplicate/Unwanted Files in Linux Using 'FSlint' Tool

Linux Server Monitoring Tools

How to Install Tripwire IDS (Intrusion Detection System) on Linux

A Shell Script to Send Email Alert When Memory Gets Low

Trickle – Control Network Traffic Bandwidth Of Applications in a Linux

How to Add Hosts in OpenNMS Monitoring Server

rtop – An Interactive Tool to Monitor Remote Linux Server Over SSH

ngrep – A Network Packet Analyzer for Linux

Learn Linux Tricks & Tips

4 Ways to Send Email Attachment from Linux Command Line

How to Search and Remove Directories Recursively on Linux

How to Find Number of Files in a Directory and Subdirectories

How to Configure Custom SSH Connections to Simplify Remote Access

Learn How to Set Your $PATH Variables Permanently in Linux

5 Command Line Tools to Find Files Quickly in Linux

Best Linux Tools

10 Most Popular Download Managers for Linux in 2023

8 Best SSH Clients for Linux in 2024

25 Outstanding Backup Utilities for Linux Systems in 2024

17 Best RSS Feed Readers for Linux in 2024

18 Best NodeJS Frameworks for App Development in 2023

11 Best IP Address Management Tools for Linux Network