TECMINT
#1 LINUX BLOG

Menu

5 Command Line Tools to Find Files Quickly in Linux

Ravi Saive │ Last Updated: July 13, 2023 │ Read Time: 3 mins │ Linux Commands │ 6 Comments

Searching or finding files on a Linux system from the terminal can be a little of a challenge especially for newbies. However, there are several command line tools/utilities for locating files in Linux.

In this article, we will review 5 command line tools to find, locate and search files quickly on Linux systems.

1. Find Command

find command is a powerful, widely used CLI tool for searching and locating files whose names match simple patterns, in a directory hierarchy. Using find is simple, all you need to do is provide a starting point (top of the directory heirarchy) where the search beings. This can be the current directory or any other directory where you suspect the file you are looking for is stored.

After the starting point, you can specify an expression (composed of test, actions, options and operators) which describes how to match files and what to do with the files that were matched.

It supports multiple options to locate files using attributes such as permissions, users, groups, file type, date, size and other possible criteria. You can learn some useful find command usage examples in the following articles:

1. 35 Practical Examples of Linux Find Command
2. Ways to Use 'find' Command to Search Directories More Efficiently
3. How to Find Files With SUID and SGID Permissions in Linux
4. How to Use 'find' Command to Search for Multiple Filenames (Extensions) in Linux
5. How to Find and Sort Files Based on Modification Date and Time in Linux

2. Locate Command

locate command is another commonly used CLI utility for searching files quickly by name, just like find command. However, it is practically more efficient and faster compared to its counterpart because, instead of searching through the file system when a user initiates a file search operation (the way find works), locate queries a database which contains bits and parts of files and their corresponding paths on the file system.

This database can be prepared and updated using the **updatedb** command. Note that **locate** will not report files created after the most recent update of the relevant database.

Read Also: How to Install 'locate Command' to Find Files in Linux

3. Grep Command

Although grep command is not a tool for directly searching files (its instead used to print lines matching a pattern from one or more files), you can employ it to locate files. Assuming you know a phrase in the file(s) you are looking for or you are looking for a file that contains a particular string of characters, **grep** can help you list all files that contain a particular phrase.

For example, if you are looking for a **README.md** file which contains the phrase "**An assortment**", which you suspect should be somewhere in your home directory, possibly in **~/bin**, you can locate it as shown.

```
$ grep -Ri ~/bin -e "An assortment"
OR
$ grep -Ri ~/bin/ -e "An assortment" | cut -d: -f1
```

Where the grep flag:

- `-R` – means search the specified directory recursively
- `-i` – means ignore case distinctions

- `-e` – specifies the phrase to be used as a pattern for searching

- `-d` – specifies the delimter

- `-f` – sets the field to be printed

You can learn some useful grep command usage examples in the following articles:

1. [12 Practical Examples of Linux Grep Command](#)
2. [11 Advance Linux Grep Commands Usage and Examples](#)
3. [How to Find a Specific String or Word in Files and Directories](#)

4. Which Command

which command is a tiny and straightforward utility for locating the binary of a command; it outputs the absolute path of a command. For example:

```
$ which find
$ which locate
$ which which
```

5. Whereis Command

whereis command is also used to locate a command and it additionally shows the absolute path of the source, and manual page files for the command.

```
$ whereis find
$ whereis locate
$ whereis which
$ whereis whereis
```

Read Also: [5 Ways to Find a 'Binary Command' Description and Location on File System](#)

That's all for now! If we have missed any Commandline tools/utilities for quickly locating files on a Linux system, let us know via the comment form below. You can ask any questions concerning this topic as well.

🏷 [commandline tools](#), [Linux Tricks](#)

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:

Install and Configure ConfigServer Security & Firewall (CSF) in Linux

NEXT ARTICLE:

2 Ways to Re-run Last Executed Commands in Linux

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.

Join the TecMint Weekly Newsletter (More Than 156,129 Linux Enthusiasts Have Subscribed)

Was this article helpful? Please add a comment or buy me a coffee to show your appreciation.

Related Posts

How to Sync New and Changed Files Using 'rsync' Command

How to Search Files by Name or Extension Using find Command

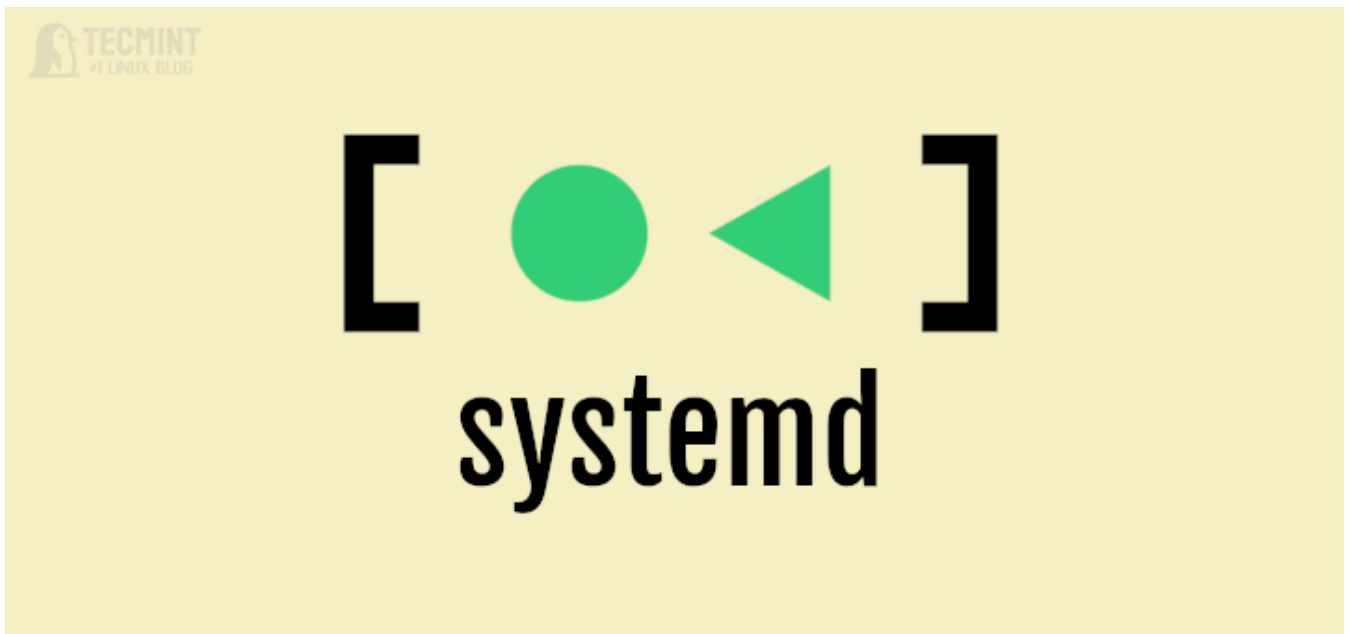## 10 Lesser Known Linux Commands – Part 2

11 Lesser Known Useful Linux Commands

26 Security Hardening Tips for Modern Linux Servers

How to Remove Systemd Services on Linux

💬 **6 Comments**

Leave a Reply

Kryztoval

September 10, 2018 at 12:47 am

Why are you piping cut when you could have used `"-l"` or – `-files-with-matches` directly and avoid using a second command.

Reply

Aaron Kili
September 12, 2018 at 3:38 pm

@Kryztoval

Ok, many thanks for the head up.

Reply

Andrew McDermott
September 9, 2018 at 12:27 pm

If you are using bash you should use the built-in type instead of the external which.

Reply

Aaron Kili
September 12, 2018 at 3:39 pm

@Andrew

True, good addition, many thanks.

Reply

Gray Arswright

September 9, 2018 at 2:40 am

`which` is great, however, I've found that it can lull its users into a false sense of security – the results of `which` aren't necessarily going to be the executable that handles your call.

If you want to know for certain what's running when you type `foo` at the command line, here are a few additional commands to keep in mind:

- `alias | grep "foo"` – Check for declared alias.
- `typeset –f "foo"` – Check for declared function (useful within shell scripts, particularly if you're using someone else's libraries).
- `type –f "foo"` – Check for shell builtin, alias, or executable, or symlinked executable.
- `file –dereference –E "/path/to/foo"` – Find out what's on the receiving end of that symlink.

Reply

Aaron Kili

September 12, 2018 at 3:37 pm

@Gray

Oh, this is good, many thanks for sharing this useful tip.

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...

Do You Enjoy My Blog?

Support from readers like YOU keeps this blog running. Buying me a cup of coffee is a simple and affordable way to show your appreciation and help keep the posts coming!

Buy Me a Coffee

Linux Commands and Tools

A Bash Script to Create a Bootable USB from ISO in Linux

How to Run Multiple Commands on Multiple Linux Servers

5 Useful Ways to Do Arithmetic in Linux Terminal

How to Enable, Disable and Install Yum Plug-ins

10 Lesser Known Linux Commands – Part 2

Linux 'tree Command' Usage Examples for Beginners

Linux Server Monitoring Tools

Setting Up Real-Time Monitoring with 'Ganglia' for Grids and Clusters of Linux Servers

How to Add Hosts in OpenNMS Monitoring Server

iftop – A Real Time Linux Network Bandwidth Monitoring Tool

GoAccess (A Real-Time Apache and Nginx) Web Server Log Analyzer

How to Install Zabbix on RHEL 8

Guider – A System Wide Linux Performance Analyzer

Learn Linux Tricks & Tips

How to Block or Disable Normal User Logins in Linux

How to Check Timezone in Linux

How to Add or Remove Linux User From Group

How to Keep 'sudo' Password Timeout Session Longer in Linux

YTP-DL: The Easiest Way to Download Songs from YouTube Videos

How to Run MySQL/MariaDB Queries Directly from the Linux Command Line

Best Linux Tools

11 Best Open Source Note-Taking Apps for Linux

5 Online Tools for Generating and Testing Cron Jobs for Linux

5 Best Command Line HTTP Clients for Linux

11 Best Free and Low-Cost SSL Certificate Authorities

8 Best RDP (Remote Desktop) Clients for Linux in 2024

13 Most Used Microsoft Office Alternatives for Linux

Hosting Sponsored by : **Linode Cloud Hosting**