

How to use Tree command to navigate file directory in Linux

The tree command is perfect for quickly viewing your entire directory structure. It displays your directories and files in a clean tree layout right in the terminal.

The tree command is perfect for quickly viewing your entire directory structure. It displays your directories and files in a clean tree layout right in your terminal. You can control the depth of the command, include hidden files, and even save the results to a file. This article will dive into how to get the most out of the tree command.

Note : Do not confuse the tree command with the pstree command . The tree command deals with file directories, while pstree deals with lists of running processes.

Basic syntax and settings

Tree isn't installed by default on every Linux distribution, but it's very small and easy to add. You just need to use your distribution's package manager. For example, on Debian/Ubuntu, use the following command:

```
sudo apt install tree
```

```
haroon@linux:~$ sudo apt install tree
[sudo] password for haroon:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 13 not upgraded.
Need to get 47.1 kB of archives.
After this operation, 111 kB of additional disk space will be used.
Get:1 http://pk.archive.ubuntu.com/ubuntu noble/universe amd64 tree amd64 2.1.1-2ubuntu3 [47.1 kB]
Fetched 47.1 kB in 2s (28.0 kB/s)
Selecting previously unselected package tree.
(Reading database ... 189471 files and directories currently installed.)
Preparing to unpack .../tree_2.1.1-2ubuntu3_amd64.deb ...
Unpacking tree (2.1.1-2ubuntu3) ...
Setting up tree (2.1.1-2ubuntu3) ...
Processing triggers for man-db (2.12.0-4build2) ...
```

For Fedora, CentOS or RHEL users, you should use the following command:

```
sudo dnf install tree
```

Arch Linux users can use the following command:

```
sudo pacman -S tree
```

Once installed, the simplest way to use it is:

```
tree
```

```
haroon@linux:~$ tree
.
├── Desktop
├── Documents
├── Downloads
├── Music
├── Pictures
├── Public
├── snap
│   ├── firefox
│   │   ├── 6565
│   │   ├── common
│   │   └── current -> 6565
│   ├── firmware-updater
│   │   ├── 210
│   │   ├── common
│   │   └── current -> 210
│   └── snapd-desktop-integration
│       ├── 315
│       │   ├── Desktop
│       │   ├── Documents
│       │   ├── Downloads
│       │   ├── Music
│       │   ├── Pictures
│       │   ├── Public
│       │   ├── Templates
│       │   └── Videos
│       └── common
└── common
```

This command will print the current directory tree, showing files and directories with their connections and a final summary, such as the number of directories and files. By default, this command lists everything recursively, showing files and directories in a hierarchical view.

Control directory depth

Sometimes you don't want full unlimited access to subdirectories. Maybe you're only interested in the top two levels, or you just want a clear overview rather than going too deep. To get a specific level, you can use the `-L` (level) flag of the `tree` command. For example, to show only two levels, you could use the following command:

```
tree -L 2
```

```
haroon@linux:~$ tree -L 2
.
├── Desktop
├── Documents
│   ├── Text Files
│   ├── TODO
│   └── Videos Doc
├── Downloads
├── Music
├── Pictures
├── Public
├── snap
│   ├── firefox
│   ├── firmware-updater
│   └── snapd-desktop-integration
├── Templates
├── TODO
├── TODO.txt
└── Videos

15 directories, 3 files
haroon@linux:~$
```

You can also combine it with other flags to be more effective. For example, you can combine the `-L` option with `-d` to show only directories and ignore files, but up to the third level:

```
tree -d -L 3
```

Unlimited, this option can be expanded infinitely, suitable for small setups but too much for large folders.

Include hidden files and folders

In the Linux world, any file or directory that begins with a period (.) is considered hidden. These files or directories are not typically used on a daily basis. They are often configuration files or metadata directories, such as `bashrc`, `profile`, or the `.git` directory common in Git repositories. The `tree` command will hide or ignore these hidden files and directories by default.

However, you can include them with the `-a` flag:

```
tree -a
```

Now you will get every directory, including all important hidden configuration files. If you want hidden files along with depth control, use this command:

```
tree -a -L 2
```

```
haroon@linux:~$ tree -a -L 2
.
├── .bash_history
├── .bash_logout
├── .bashrc
├── .cache
│   ├── event-sound-cache.tdb.6c98b09585a04753a9c79189374b46a5.x86_64-pc-linux-gnu
│   ├── evolution
│   ├── flatpak
│   ├── gstreamer-1.0
│   ├── ibus
│   ├── ibus-table
│   ├── mesa_shader_cache
│   ├── mesa_shader_cache_db
│   ├── tracker3
│   ├── ubuntu-report
│   └── update-manager-core
├── .config
│   ├── dconf
│   ├── evolution
│   └── gnome-initial-setup-done
```

Display files by matching pattern

Sometimes you don't want to see everything; you're looking for something specific. Maybe you're only interested in Python files, or you're looking for all your Markdown documents. The `-P` option lets you specify a pattern to match.

For example, if you just want to see Python files, run:

```
tree -P "*.py"
```

If you are looking for all text files, use:

```
tree -P "*.txt"
```

You can also use wildcards and get creative with your patterns. Let's say you need to find all files that start with config; you can do that with:

```
tree -P "config*"
```

When you use `-P`, the tree will still show the directory structure, but it will only show files that match your pattern. Empty directories, that is, directories that do not contain matching files, will still appear in the output. If you find this confusing, you can use the `--prune` option :

```
tree -P "*.py" --prune
```

This option will remove any empty directories from the output, giving you a cleaner view that only shows the paths to the target files. One more thing: If you want to exclude names that match a pattern, use `-I` instead of `-P`.

Display file sizes in an easy-to-read format

Honestly, seeing file sizes in bytes isn't really that helpful. When a file is 524288 bytes, you have to stop and do some mental math to come up with about 512 KB. Fortunately, the `-h` option solves this problem beautifully. It displays the file size in a readable format, exactly how we think about file sizes. For example:

```
tree -h
```

```
haroon@linux:~$ tree -h
[4.0K] .
├── [4.0K] Desktop
├── [4.0K] Documents
│   ├── [4.0K] Text Files
│   │   └── [ 2] Hello File
│   ├── [ 2] TODO
│   └── [4.0K] Videos Doc
├── [4.0K] Downloads
├── [4.0K] Music
├── [4.0K] Pictures
├── [4.0K] Public
├── [4.0K] snap
│   ├── [4.0K] firefox
│   │   ├── [4.0K] 6565
│   │   ├── [4.0K] 7177
│   │   ├── [4.0K] common
│   │   └── [ 4] current -> 6565
│   ├── [4.0K] firmware-updater
│   │   ├── [4.0K] 210
│   │   ├── [4.0K] common
│   │   └── [ 3] current -> 210
│   └── [4.0K] snapd-desktop-integration
│       └── [4.0K] 315
│           ├── [4.0K] Desktop
│           ├── [4.0K] Documents
│           ├── [4.0K] Downloads
│           └── [4.0K] Music
```

Now instead of 524288, you'll see 512K. A file that's 1048576 bytes will become 1.0M. This is useful when you want to see where all your disk space has gone. You can quickly scan through the directory tree and spot suspiciously large files.

Here is another command combination that many people use often:

```
tree -hL 2
```

This command shows two levels of depth, with human-readable file sizes.

Combining tree with other commands

In Linux, the real power isn't in any single command, but in how you combine them. The tree command just outputs text. And we can pass it to other commands and perform various operations on it. The `()` takes the output from the command on the left and sends it as input to the command on the right.

Let's say you want to run a directory tree on a large directory, but you don't want it to flash across the screen in 30 seconds. You want to be able to scroll, search, and read it at your own pace. You can do this by combining tree with the less command:

```
tree /usr/lib | less
```

Run this command, and your screen will lock to the first page of the tree output.

```
/usr/lib
├── apg
│   └── apg
├── apparmor
│   ├── apparmor.systemd
│   ├── profile-load
│   └── rc.apparmor.functions
├── apt
│   ├── apt-helper
│   ├── apt.systemd.daily
│   └── methods
│       ├── cdrom
│       ├── copy
│       ├── file
│       ├── ftp
│       ├── gpgv
│       ├── http
│       ├── https -> http
│       ├── mirror
│       ├── mirror+copy -> mirror
│       ├── mirror+file -> mirror
│       └── mirror+ftp -> mirror
/ftp
```

You can now use the arrow keys (Up/Down), spacebar to move down a full page, / followed by a word to search for that word in the entire tree, q to exit and return to the command prompt.

Save or export the results of the tree

Sometimes you need to save that tree structure for later use. Maybe you're documenting a project, creating a report, or just want to take a snapshot of the directory structure before making changes. The simplest way to do this is to use a redirect command like this:

```
tree > directory_structure.txt
```

This command will save the entire tree output to a text file. Simple, efficient, and you can open it in any text editor.

You can also create an HTML page with clickable links and collapsible sections using the built-in -H option:

```
tree -H . > structure.html
```

Open it in a browser and you'll get an interactive view of your folder structure.

To archive the document, you can also save the output in a specific format:

```
tree -L 3 -a --dirsfirst > project_docs.txt
```

The --dirsfirst option lists directories before files, making the output more organized. This is perfect for README files or project documentation.

Finally, you can also add to existing files if you're building a larger document:

```
tree -L 2 >> documentation.txt
```

This command will append the tree output to the end of your existing document file without overwriting what is already there.

You finished reading the article "**How to use Tree command to navigate file directory in Linux**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
