



# How to use the history command in Linux

Become more efficient at the command prompt with the powerful history command.

By [Steve Morris](#)

April 1, 2020 | [4 Comments](#) | 4 min read

463 readers like this.



**Image by:** Internet Archive Book Images. Modified by Opensource.com. CC BY-SA 4.0

As I spend more and more time in terminal sessions, it feels like I'm continually finding new commands that make my daily tasks more efficient. The GNU `history` command is one that really changed my work day.

The GNU `history` command keeps a list of all the other commands that have been run from that terminal session, then allows you to replay or reuse those commands instead of retyping them. If you are an experienced terminal user, you know about the power of `history`, but for us dabblers or new sysadmin folks, `history` is an immediate productivity gain.

First of all, the `history` command isn't actually a command. You can see this for yourself by looking for the command on your system:

```
$ which history
which: no history in (/usr/local/bin:/usr/bin:/bin:/usr/games:/usr/
```

Your computer can't find the `history` command because it's a built-in keyword of your shell. Because it's written into the shell you're using, there can be some variation in how history behaves depending on whether you're using [Bash](#), `tcsh`, [Zsh](#), `dash`, [fish](#), `ksh`, and so on. This article is based upon the Bash implementation of history, so some functions may not work in other shells. However, most of the basic functions are the same.

## History 101

To see `history` in action, open a terminal program on your Linux installation and type:

```
$ history
```

Here's the response I got:

```
1  clear
2  ls -al
3  sudo dnf update -y
4  history
```

The `history` command shows a list of the commands entered since you started the session. The joy of `history` is that now you can replay any of them by using a command such as:

```
$ !3
```

The `!3` command at the prompt tells the shell to rerun the command on line 3 of the history list. I could also access that command by entering:

```
$ !sudo dnf
```

This prompts `history` to search for the last command that matches the pattern you provided (in this case, that pattern is **dnf**) and run it.

## Searching history

You can also use `history` to rerun the last command you entered by typing `!!`. By pairing it with `grep`, you can search for commands that match a text pattern or, by using it with `tail`, you can find the last few commands you executed. For example:

```
$ history | grep dnf
3  sudo dnf update -y
5  history | grep dnf

$ history | tail -n 3
4  history
5  history | grep dnf
6  history | tail -n 3
```

Another way to get to this search functionality is by typing `Ctrl-R` to invoke a recursive search of your command history. After typing this, the prompt changes to:

```
(reverse-i-search)`':
```

Now you can start typing a command, and matching commands will be displayed for you to execute by pressing **Return** or **Enter**.

## Changing an executed command

You can also use `history` to rerun a command with different syntax. You can revise history with `history`. For example, if I want to change my previous command `history | grep dnf` to `history | grep ssh`, I can execute the following at the prompt:

```
$ ^dnf^ssh^
```

The command is rerun, but with `dnf` replaced by `ssh`. In other words, this command is run:

```
$ history | grep ssh
```

# Removing history

There may come a time that you want to remove some or all the commands in your history file. If you want to delete a particular command, enter `history -d <line number>`. To clear the entire contents of the history file, execute `history -c`.

The history file is stored in a file that you can modify, as well. Bash shell users find it in their home directory as `.bash_history`.

## Next steps

There are a number of other things that you can do with `history`:

- Set the size of your history buffer to a certain number of commands

- Record the date and time for each line in history

- Prevent certain commands from being recorded in history

For more information about the `history` command and other interesting things you can do with it, take a look at Seth Kenlon's articles about [parsing history](#), [history search modifiers](#), and the [GNU Bash Manual](#).

*This article was originally published in June 2018 and has been updated with additional information by the editor.*

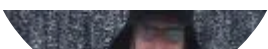
Tags:

LINUX

COMMAND LINE

## Steve Morris

Serves as the Director of Information Technology at Ambrose University in Calgary, Alberta, Canada. Cut his Linux teeth on Slackware, but spends most of his days with either Ubuntu or Fedora. Free time is spent running in an effort to complete that first half-marathon.



## 4 Comments

These comments are closed.



[Athul Joy](#) | June 5, 2018      No readers like this yet.

What are the other use of grep and tail commands ?



[Steve Morris](#) | June 5, 2018      No readers like this yet.

Athul, for grep and other general "find" type functions here's some resources:

<https://opensource.com/article/18/4/how-find-files-linux>

<https://www.tecmint.com/12-practical-examples-of-linux-grep-command/>

For the tail command, I'd recommend the following links:

<https://www.linux.com/blog/14-tail-and-head-commands-linuxunix>

<https://linuxide.com/linux-command/linux-tail-command/>

Hope this helps!



[Athul Joy](#) | June 8, 2018      No readers like this yet.

Thanks, Steve Morris



Tom Carlson | July 10, 2018      No readers like this yet.

Did not know of the Ctrl-R method. Many thanks for that!

I alias "wdil" to "history | grep" to search for past commands, but the Ctrl-R method is even slicker. ("wdil" stands for "When Did I Last?")

## Related Content



[What's new in GNOME 44?](#)



[5 reasons virtual machines still matter](#)



[Remove the background from an image with this Linux command](#)



This work is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.

## ABOUT THIS SITE

The opinions expressed on this website are those of each author, not of the author's employer or of Red Hat.

**Opensource.com** aspires to publish all content under a **Creative Commons license** but may not be able to do so in all cases. You are responsible for ensuring that you have the necessary permission to reuse any work on this site. Red Hat and the Red Hat logo are trademarks of Red Hat, Inc., registered in the United States and other countries.

A note on advertising: Opensource.com does not sell advertising on the site or in

any of its newsletters.