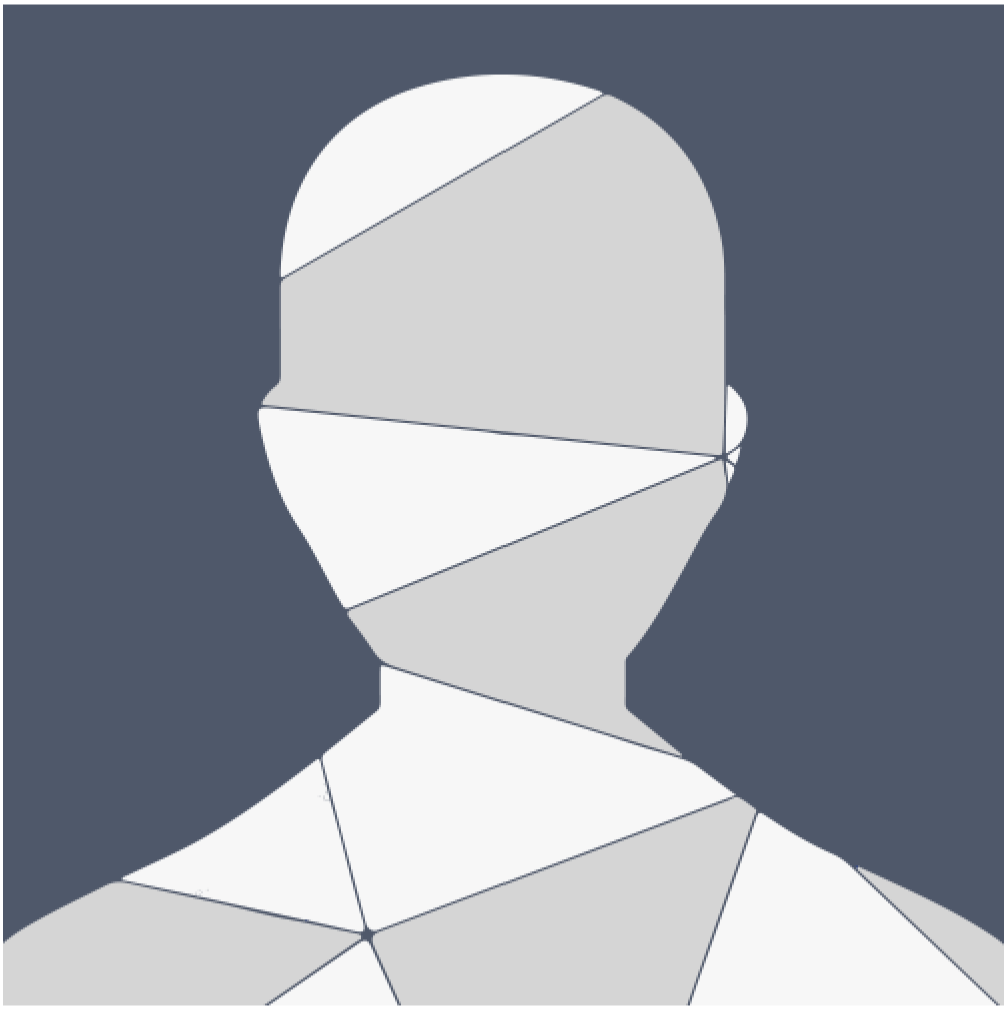
How to Set an Unlimited Bash History

Last updated: November 19, 2023



> Written by: Francis Ndungu (https://www.baeldung.com/linux/author/francisndungu)

Administration (https://www.baeldung.com/linux/category/administration)

bash (https://www.baeldung.com/linux/tag/bash)

export (https://www.baeldung.com/linux/tag/export)

source (https://www.baeldung.com/linux/tag/source)

1. Overview

Bash (/linux/sh-vs-bash#bash), the default shell in most Linux distributions, maintains a history of commands (/linux/bash-using-history-efficiently) entered in the command line. This feature allows us to reuse previously executed commands. By default, Bash limits the number of commands stored in history. However, we can modify it to maintain an unlimited history.

In this tutorial, we'll learn how to set an unlimited Bash history in Linux.

2. Setting an Unlimited Bash History

To set an unlimited Bash history, we need to modify two environment variables (/linux/list-env-variables), namely, *HISTSIZE and HISTFILESIZE (/linux/bash-histsize-vs-histfilesize)*. They're simply variables that control the command history behavior.

In terms of location, both these variables exist in *~/.bashrc* (/linux/bashrc-vs-bash-profile-vs-profile#2-significance-of-bashrc), a Bash configuration file located in a user's *home* directory:

```
$ cat ~/.bashrc
...

HISTSIZE=1000
HISTFILESIZE=2000

...
```

Above, the *cat (/linux/files-cat-more-less#cat)* command displays the default values for the *HISTSIZE* and *HISTFILESIZE* environment variables, respectively.

2.1. Exploring *HISTSIZE*

The *HISTSIZE* environment variable in Linux holds the maximum number of commands the shell remembers in its command history. Typically, when we execute a command in the terminal, Bash records it in a history list in memory. This history list allows us to access and run previously executed commands during a shell session.

For instance, *HISTSIZE=1000* indicates that the history list contains only *1000* of the most recent commands. Hence, when the number of commands exceeds *1000*, the oldest commands are removed to accommodate newer commands.

2.2. Exploring *HISTFILESIZE*

The *HISTFILESIZE* environment variable determines the number of commands in command history that can be stored in a user's history file.

To elaborate, when a user exits a Bash shell session, the history is written to the *~/.bash_history* file found in the user's *home* directory:

```
$ cat ~/.bash_history
git push
git status
exit
...
```

In the example above, we're able to see some of the commands available in the *~/.bash_history* file. Currently, the history file can only store up to *2000* lines of commands as indicated by *HISTFILESIZE=2000*. Importantly, we can navigate through these commands in the terminal, using the *up* and *down* arrow keys.

2.3. Modifying *HISTSIZE* and *HISTFILESIZE*

To modify both these variables, let's begin by opening the Bash configuration file *~/.bashrc* in the *home* directory. For this, we can use a text editor of our choice:

```
$ nano ~/.bashrc
```

In this example, we demonstrate using the *nano (/linux/files-vi-nano-emacs#nano)* text editor.

Once, the file is open, let's navigate to the *HISTSIZE* and *HISTFILESIZE* variables, modify their values to *-1*, and save the changes:

```
HISTSIZE=-1
HISTFILESIZE=-1
```

In Bash versions *4.3* and above, updating the *HISTSIZE* and *HISTFILESIZE* values to *-1* ensures that the history list size and history file size are unlimited respectively.

Finally, to apply the changes we can either close and reopen the terminal or execute the *source* (/linux/source-command) command:

```
$ source ~/.bashrc
```

This command reloads the Bash configuration. At this point, the updated value, *-1*, is always applied when we start a new shell session. Before we proceed, **it's important to note that editing the *~/.bashrc* file is only going to affect the current user**. Thus, it's not a system-wide change.

Alternatively, instead of editing the configuration file, we can use a different approach:

```
$ export HISTSIZE=-1; export HISTFILESIZE=-1
```

Here, we simply execute these two commands in our terminal. As a consequence, the environment variables only update for the current shell session. So, this method simply sets an unlimited Bash history for a specific shell session.

2.4. Verifying the Changes

In this section, we're confirming that our changes took effect. Here, we use the *echo* (/linux/printf-echo#echo) command to show the values of the *HISTSIZE* and *HISTFILESIZE* environment variables:

```
$ echo $HISTSIZE; echo $HISTFILESIZE
-1
-1
```

Above, both commands display *-1* indicating that we've been able to configure an unlimited Bash history.

2.5. Using Unlimited Bash History Efficiently

Next, let's discuss a few key points to consider when working with an unlimited Bash history.

First, we need to keep track of how the history file impacts disk space. Over time, this file increases in size, occupying a significant amount of disk space.

We can use existing commands to improve the command-line experience:

- *history* – displays a numbered list of the commands in the history list
- *history -c* – clears the history list
- *!!* – runs the last command in the history list
- *!n* – runs the *n*th command stored in history
- *CTRL + R* – searches through the history list interactively. To clarify, Bash starts displaying matches as soon as we begin typing what we're looking for.

Further, we can execute a command containing sensitive information without keeping it in history (/linux/execute-command-without-history). To do so, we simply precede the command with a space, and that command is excluded from the history list.

3. Conclusion

In this article, we explored setting an unlimited Bash history in Linux. In particular, we specifically modified the environment variables *HISTSIZE* and *HISTFILESIZE*.

To conclude, an unlimited Bash history ensures that we can retain all our previously executed commands. As
a result, both productivity and the command-line interface experience are enhanced. However, it's good security practice to monitor what sensitive information this unlimited Bash history may contain.

Comments are closed on this article!

(/linux/) (https://www.baeldung.com/linux/)

(/bael-search)

CATEGORIES

ADMINISTRATION (/LINUX/CATEGORY/ADMINISTRATION)
FILES (/LINUX/CATEGORY/FILES)
FILESYSTEMS (/LINUX/CATEGORY/FILESYSTEMS)
INSTALLATION (/LINUX/CATEGORY/INSTALLATION)
NETWORKING (/LINUX/CATEGORY/NETWORKING)
PROCESSES (/LINUX/CATEGORY/PROCESSES)
SCRIPTING (/LINUX/CATEGORY/SCRIPTING)
SEARCH (/LINUX/CATEGORY/SEARCH)
SECURITY (/LINUX/CATEGORY/SECURITY)
WEB (/LINUX/CATEGORY/WEB)

SERIES

LINUX ADMINISTRATION (/LINUX/LINUX-ADMINISTRATION-SERIES)
LINUX FILES (/LINUX/LINUX-FILES-SERIES)
LINUX PROCESSES (/LINUX/LINUX-PROCESSES-GUIDE)

ABOUT

ABOUT BAELDUNG (/ABOUT)
THE FULL ARCHIVE (HTTPS://WWW.BAELDUNG.COM/LINUX/FULL_ARCHIVE)
EDITORS (HTTPS://WWW.BAELDUNG.COM/EDITORS)

TERMS OF SERVICE (HTTPS://WWW.BAELDUNG.COM/TERMS-OF-SERVICE)
PRIVACY POLICY (HTTPS://WWW.BAELDUNG.COM/PRIVACY-POLICY)
COMPANY INFO (HTTPS://WWW.BAELDUNG.COM/BAELDUNG-COMPANY-INFO)
CONTACT (/CONTACT)