THE GEEK STUFF

Linux | DB | Open Source | Web

≡ MENU

15 Examples To Master Linux Command Line History

by RAMESH NATARAJAN *on* AUGUST 11, 2008

When you are using Linux command line frequently, using the history effectively can be a major productivity boost. In fact, once you have mastered the 15 examples that I've provided here, you'll find using command line more enjoyable and fun.

```
# ps aux
# cd ~
# ls -altr
# history
```

1. Display timestamp using HISTTIMEFORMAT

Typically when you type history from command line, it displays the command# and the command. For auditing purpose, it may be beneficial to display the timepstamp along with the command as shown below.

```
# export HISTTIMEFORMAT='%F %T '
# history | more
1  2008-08-05 19:02:39 service network restart
2  2008-08-05 19:02:39 exit
3  2008-08-05 19:02:39 id
4  2008-08-05 19:02:39 cat /etc/redhat-release
```

2. Search the history using Control+R

I strongly believe, this may be your most frequently used feature of history. When you've already executed a very long command, you can simply search history using a keyword and re-execute the same command without having to type it fully. **Press Control+R and type the keyword**. In the following example, I searched for **red**, which displayed the previous command "**cat /etc/redhat-release**" in the history that contained the word red.

```
# [Press Ctrl+R from the command prompt,
which will display the reverse-i-search prompt]
(reverse-i-search)`red': cat /etc/redhat-release
[Note: Press enter when you see your command,
which will execute the command from the history]
# cat /etc/redhat-release
Fedora release 9 (Sulphur)
```

Sometimes you want to edit a command from history before executing it. For e.g. you can search for **httpd**, which will display **service httpd stop** from the command history, select this command and **change the stop to start** and re-execute it again as shown below.

```
# [Press Ctrl+R from the command prompt,
which will display the reverse-i-search prompt]
(reverse-i-search)`httpd': service httpd stop
[Note: Press either left arrow or right arrow key when you see your
command, which will display the command for you to edit, before executing it]
# service httpd start
```

3. Repeat previous command quickly using 4 different methods

Sometime you may end up repeating the previous commands for various reasons. Following are the 4 different ways to repeat the last executed command.

1. Use the **up arrow** to view the previous command and press enter to execute it.
2. Type **!!** and press enter from the command line
3. Type **!-1** and press enter from the command line.
4. Press **Control+P** will display the previous command, press enter to execute it

4. Execute a specific command from history

In the following example, If you want to repeat the command #4, you can do **!4** as shown below.

```
# history | more
1  service network restart
2  exit
3  id
4  cat /etc/redhat-release

# !4
cat /etc/redhat-release
Fedora release 9 (Sulphur)
```

5. Execute previous command that starts with a specific word

Type ! followed by the starting few letters of the command that you would like to re-execute. In the following example, typing !ps and enter, executed the previous command starting with ps, which is 'ps aux | grep yp'.

```
# !ps
ps aux | grep yp
root     16947  0.0  0.1  36516  1264 ?        Sl   13:10   0:00 ypbind
root     17503  0.0  0.0   4124   740 pts/0    S+   19:19   0:00 grep yp
```

6. Control the total number of lines in the history using HISTSIZE

Append the following two lines to the .bash_profile and relogin to the bash shell again to see the change. In this example, only 450 command will be stored in the bash history.

```
# vi ~/.bash_profile
HISTSIZE=450
HISTFILESIZE=450
```

7. Change the history file name using HISTFILE

By default, history is stored in **~/.bash_history** file. Add the following line to the .bash_profile and relogin to the bash shell, to store the history command in .commandline_warrior file instead of .bash_history file. I'm yet to figure out a practical use for this. I can see this getting used when you want to track commands executed from different terminals using different history file name.

```
# vi ~/.bash_profile
HISTFILE=/root/.commandline_warrior
```

If you have a good reason to change the name of the history file, please share it with me, as I'm interested in finding out how you are using this feature.

8. Eliminate the continuous repeated entry from history using HISTCONTROL

In the following example pwd was typed three times, when you do history, you can see all the 3 continuous occurrences of it. To eliminate duplicates, set HISTCONTROL to ignoredups as shown below.

```
# pwd
# pwd
# pwd
# history | tail -4
44  pwd
45  pwd
46  pwd [Note that there are three pwd commands in history, after
executing pwd 3 times as shown above]
47  history | tail -4

# export HISTCONTROL=ignoredups
# pwd
# pwd
# pwd
# history | tail -3
56  export HISTCONTROL=ignoredups
```

```
57  pwd [Note that there is only one pwd command in the history, even after
executing pwd 3 times as shown above]
58  history | tail -4
```

9. Erase duplicates across the whole history using HISTCONTROL

The ignoredups shown above removes duplicates only if they are consecutive commands. To eliminate duplicates across the whole history, set the HISTCONTROL to erasedups as shown below.

```
# export HISTCONTROL=erasedups
# pwd
# service httpd stop
# history | tail -3
38  pwd
39  service httpd stop
40  history | tail -3

# ls -ltr
# service httpd stop
# history | tail -6
35  export HISTCONTROL=erasedups
36  pwd
37  history | tail -3
38  ls -ltr
39  service httpd stop
[Note that the previous service httpd stop after pwd got erased]
40  history | tail -6
```

10. Force history not to remember a particular command using HISTCONTROL

When you execute a command, you can instruct history to ignore the command by setting HISTCONTROL to ignorespace AND typing a space in front of the command as shown below. I can see lot of junior sysadmins getting excited about this, as they can hide a command from the history. It is good to understand how ignorespace works. But, as a best practice, don’t hide purposefully anything from history.

```
# export HISTCONTROL=ignorespace
# ls -ltr
# pwd
#  service httpd stop [Note that there is a space at the beginning of service,
to ignore this command from history]
# history | tail -3
67  ls -ltr
68  pwd
69  history | tail -3
```

11. Clear all the previous history using option -c

Sometime you may want to clear all the previous history, but want to keep the history moving forward.

```
# history -c
```

12. Subtitute words from history commands

When you are searching through history, you may want to execute a different command but use the same parameter from the command that you’ve just searched.

In the example below, the **!!:$** next to the vi command gets the argument from the previous command to the current command.

```
# ls anaconda-ks.cfg
anaconda-ks.cfg
# vi !!:$
vi anaconda-ks.cfg
```

In the example below, the **!^** next to the vi command gets the first argument from the previous command (i.e cp command) to the current command (i.e vi command).

```
# cp anaconda-ks.cfg anaconda-ks.cfg.bak
anaconda-ks.cfg
# vi  !^
vi anaconda-ks.cfg
```

13. Substitute a specific argument for a specific command.

In the example below, **!cp:2** searches for the previous command in history that starts with cp and takes the second argument of cp and substitutes it for the ls -l command as shown below.

```
# cp ~/longname.txt /really/a/very/long/path/long-filename.txt
# ls -l !cp:2
ls -l /really/a/very/long/path/long-filename.txt
```

In the example below, **!cp:$** searches for the previous command in history that starts with cp and takes the last argument (in this case, which is also the second argument as shown above) of cp and substitutes it for the ls -l command as shown below.

```
# ls -l !cp:$
ls -l /really/a/very/long/path/long-filename.txt
```

14. Disable the usage of history using HISTSIZE

If you want to disable history all together and don’t want bash shell to remember the commands you’ve typed, set the HISTSIZE to 0 as shown below.

```
# export HISTSIZE=0
# history
# [Note that history did not display anything]
```

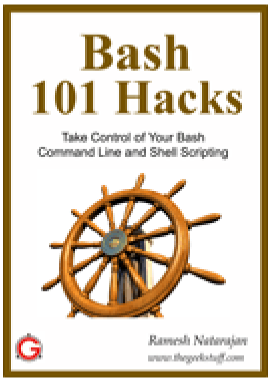
15. Ignore specific commands from the history using HISTIGNORE

Sometimes you may not want to clutter your history with basic commands such as pwd and ls. Use HISTIGNORE to specify all the commands that you want to ignore from the history. Please note that adding ls to the HISTIGNORE ignores only ls and not ls -l. So, you have to provide the exact command that you would like to ignore from the history.

```
# export HISTIGNORE="pwd:ls:ls -ltr:"
# pwd
# ls
# ls -ltr
# service httpd stop

# history | tail -3
79  export HISTIGNORE="pwd:ls:ls -ltr:"
80  service httpd stop
81  history
[Note that history did not record pwd, ls and ls -ltr]
```

Recommended Reading



Bash 101 Hacks, by Ramesh Natarajan. I spend most of my time on Linux environment. So, naturally I'm a huge fan of Bash command line and shell scripting. 15 years back, when I was working on different flavors of *nix, I used to write lot of code on C shell and Korn shell. Later years, when I started working on Linux as system administrator, I pretty much automated every possible task using Bash shell scripting. Based on my Bash experience, I've written Bash 101 Hacks eBook that contains 101 practical examples on both Bash command line and shell scripting. If you've been thinking about mastering Bash, do yourself a favor and read this book, which will help you take control of your Bash command line and shell scripting.

Awesome Linux Articles

Following are few awesome **15 examples** articles that you might find helpful.

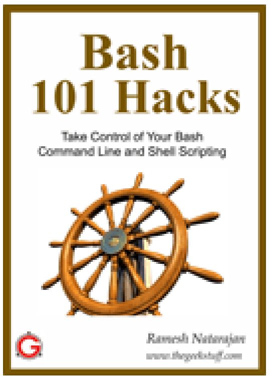
- Unix LS Command: 15 Practical Examples
- Get a Grip on the Grep! – 15 Practical Grep Command Examples
- Linux Crontab: 15 Awesome Cron Job Examples
- Mommy, I found it! — 15 Practical Linux Find Command Examples

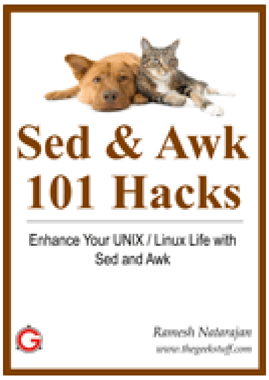
Like 12 Add your comment

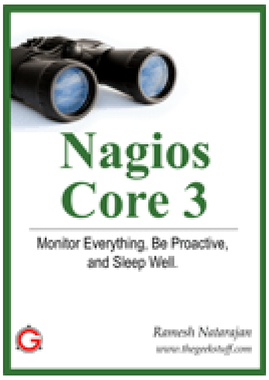
If you enjoyed this article, you might also like..

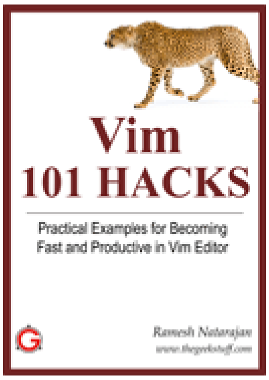
1. 50 Linux Sysadmin Tutorials
2. 50 Most Frequently Used Linux Commands (With Examples)
3. Top 25 Best Linux Performance Monitoring and Debugging Tools
4. Mommy, I found it! – 15 Practical Linux Find Command Examples
5. Linux 101 Hacks 2nd Edition eBook Free

- Awk Introduction – 7 Awk Print Examples
- Advanced Sed Substitution Examples
- 8 Essential Vim Editor Navigation Fundamentals
- 25 Most Frequently Used Linux IPTables Rules Examples
- Turbocharge PuTTY with 12 Powerful Add-Ons








Vim
101 HACKS
Practical Examples for Becoming Fast and Productive in Vim Editor
Ramesh Natarajan
www.thegeekstuff.com

Tagged as: ~/.bash_history, bash, bash control-R, bash shell, clear linux history, clearing command cache, clearing shell cache, commad history, command line, command line history, HISTCONTROL, HISTCONTROL=erasedups, HISTCONTROL=ignoredups, HISTFILE, HISTIGNORE, history, history command, HISTSIZE, HISTTIMEFORMAT, linux command, previous linux command

Comments on this entry are closed.

digifuzz

August 11, 2008, 8:05 am

hey, thanks. i actually learned a few things I didn't know I could do before.

∞

Webfreakz

August 11, 2008, 11:04 am

Whoaah! Very useful! Thanks =)

∞

Keith Dsouza

August 11, 2008, 9:14 pm

Cool tricks, will come definitely come in useful to me, bookmarked for reference.

∞

AurGlass

August 12, 2008, 3:36 pm

An observation:

It's not immediately clear from the title and introduction that this article applies to the bash shell. I realize that bash is the default shell and so practically synonymous with "the Linux command line", but there are other shells that a user could end up in, particularly if using a minimal rescue environment. It might be helpful to explicitly mention bash up front so people don't get confused if these tips don't work properly. Or perhaps show some of the differences for common shells if there are any for a given example.
Maybe even explaining briefly the difference between the 'command line' and a shell so people can learn what's really going on.

Great article though.

∞

Ramesh

August 12, 2008, 5:49 pm

@AurGlass,

Thanks for the feedback. Like you've mentioned, I left it out because the default shell in Linux is bash. You brought-up a very valid point and I'll keep that in mind for the future articles.

∞

frankb

August 12, 2008, 6:51 pm

The best reason I ever found to change the history file name using HISTFILE was described here. whereby each administrator of a box gets his own superuser history. Hope that's what you were looking for.

∞

miker

August 12, 2008, 9:17 pm

great article, very handy.

my new favourite feature of bash is using . to put the last argument of the previous command into your new command..something like this:

mkdir some_dir_name
cd .

some_dir_name will appear on your cd line.

∞

ta

August 12, 2008, 10:41 pm

Use histfile when you automount a common home directory and you want separate history files for each host that you log into.

export HISTFILE=~.bash_history.d/$HOSTNAME

∞

pwc101

August 13, 2008, 12:24 am

I find the !$ syntax for the last argument of the previous command quicker to type than !!:$ mentioned in 12. Likewise using !:1, !:2 for the previous arguments (starting with !:0 as the first).

∞

Martin

August 13, 2008, 2:39 am

very useful article, thank you very much

∞

Gordon Schulz

August 13, 2008, 2:44 am

Great Article, thank you. Lots of stuff I didn’t know before, very helpful!

∞

miker

August 13, 2008, 8:32 am

heh..my esc key was eaten in my above comment. that should read esc .

not just .

∞

Ramesh

August 13, 2008, 8:54 am

@frankb, @ta,

Thanks for sharing the reasons to use HISTFILE with examples.

@miker,

"esc ." is definitely a great use to substitute the argument from the previous command to the current command. Thanks for the tip.

@pwc101,

I agree, the argument number is probably easier to use. Thanks for the feedback.

∞

afox

August 13, 2008, 5:00 pm

I'm a huge fan of timestamped history. Comes in handy when you're trying to prove your client bricked his Linux server. 🙄

∞

Ajith Edassery

August 14, 2008, 2:51 am

Great tips Ramesh. I didn't find any excuse for not digging it 🙄

Cheers,
Ajith

∞

dethmetaljeff

August 15, 2008, 5:36 pm

you can also use !$ instead of !!:$ to retreive the last argument from the last command

```
mkdir -p really/deep/directory/tree
cd !$
```

will get you into really/deep/directory/tree

∞

Bash

August 15, 2008, 8:42 pm

For No 12, Alt+. (Alt+ dot) is much faster.

∞

Zaheer M K

August 16, 2008, 7:45 am

Hello Ramesh,

Excellent post!

I never knew about all this before.

Thank you so much for enlightening us.

Do keep writing.

Zaheer.

∞

anon

August 23, 2008, 11:11 am

#12 Can be shortened to !$ which is a bit faster.

∞

jeff

September 3, 2008, 9:10 pm

Ctrl-R. New favorite! Thanks. -j

∞

cholan dhamodaran

September 10, 2008, 6:26 am

very good…thanks a lot

∞

Ewoud Van Craeynest

September 11, 2008, 5:02 am

Very informative post!
Thanks

∞

Ramesh

September 13, 2008, 12:47 pm

@dethmetaljeff, bash, miker, anon,

Thanks for pointing out the alternative for **example#12**. i.e instead of **!!:$**, following can also be used to substitute the previous argument.

```
1. vi !!:$
2. vi !$
3. vi Alt-key + .
4. vi Esc-key + .
```

∞

Jean-Luc Lacroix

September 30, 2008, 1:11 am

The most comprehensive tutorial I have found on the history command I have found so far.

Nice work! Thank you.

∞

Roshan Bhattarai

October 2, 2008, 10:14 am

wow……..great list dude……..thanks for sharing ……..

∞

Christian

October 27, 2008, 2:16 pm

Hi, I like I lot but some how on my RedHat Linux kernel.com 2.4.20-8 #1 Thu Mar 13 17:54:28 EST 2003 i686 i686 i386 GNU/Linux
command given doesn't work:
export HISTTIMEFORMAT='%F %T '
I am running bash as following output said:
[root@kernel root]# echo $SHELL
/bin/bash
Any ideas?

Thx
Christian

∞

Nihar

October 29, 2008, 8:31 pm

Simply great. It will be of great help in my day- to – day development work at office.

∞

Muthu

November 3, 2008, 11:34 pm

It is an excellent article. These tricks helps me a lot in making commandline usage more interesting and smooth. Thanks

∞

DavidTan

December 8, 2008, 12:08 pm

What a nice list. Thanks!

∞

Velusamy

January 5, 2009, 1:32 am

Excellent Post

∞

sharmila

January 16, 2009, 10:44 pm

Hi

This article very is usefull, in fact more informative. I need a help from you all that I want the history command shoud shows the command with date:time:servername:userid:user's hostname or ip:command for monitoring purpose.

At the same time, I have to capture in a file what are the command entered by user then the file redirect to some where.

Your help is more appreciated in this regards. This is going to be implemented in red hat linux

regards
sharmila

∞

nagaraju.k
January 18, 2009, 1:44 pm

Hi,

Really great tips yaar….. I did not know in detail that history of commands work like this… Thanks alot and kindly post articles like above..which will educate geeks like me….

∞

nagaraju.k
January 18, 2009, 1:46 pm

Hi sharmila..

I am a system administrator in an insurance company working on RHEL. What bout you? I am presently trying Ubuntu8.1 desktop version..really its a great os.

∞

Ramesh

January 21, 2009, 1:22 am

This article is very good,you are given very good explanation for History command in Linux, it’s very useful

∞

strang

January 25, 2009, 3:13 pm

Hi,
> # export HISTTIMEFORMAT=’%F %T ‘
> # history | more
> 1 2008-08-05 19:02:39 service network restart
> 2 2008-08-05 19:02:39 exit
Is there any possibility to see history in this format without “export HISTTIMEFORMAT=’%F %T “.
If no — How I can delete (clear) ’%F %T ‘ from HISTTIMEFORMAT ?
Thanks!

∞

Maxwell

April 2, 2009, 3:16 pm

Great website, great article, awesome comments section.

∞

Tom

May 20, 2009, 1:23 am

Hi!

This is a very extensive and useful description. I worked on AIX many years ago. There was a method, with I could handle the history in vi mode. I don't remember exactly. Maybe there was a prerequisite setting: the EDITOR environment variable?
And, I could enter in the vi mode with with ESC+K .

Now I'm using RHEL, and I'm searching about that, if it's working in Linux too.

regards

Tom

∞

Tom

May 25, 2009, 2:44 am

Hi!

I'am again. I've found it:

set -o vi

And entring in vi mode is just with the ESC.

Tom

∞

Parag Dixit

June 10, 2009, 12:25 pm

Great article!!!
This page went straight away into my favorites…
I was looking for a shortcut like Ctrl-R , but this did not work for me. one friend suggested used of Alt-P instead and I use it extensively…

∞

thiru

June 12, 2009, 5:14 am

Hi All,

Any one can help me on linux
1. How to check the 2 or 3 week log in log with time and date.
* i already try lot of command from this forum andmany other forum.

Please help me !!!!!!!!!

∞

srinivas antarvedi

June 17, 2009, 1:31 am

Thanks for the post

i do found
HISTTIMEFORMAT,
HISTIGNORE,
HISTSIZE as useful in multi user environment..

HISTTIMEFORMAT will be useful when there is disaster to trackdown
the time which can be comparable with the service down email notification

∞

Ramesh Natarajan

June 21, 2009, 12:12 am

@Jean-Luc Lacroix, Roshan, Nihar, Muthu, DavidTan, Velusamy, Ramesh, Nagaraju,

Thanks a lot for your comment. I'm very happy that you found this article useful.

@Maxwell,

Yeah. The comment section on this article is very engaging and I'm very greatful to all those who commented on this article.

I'm hoping that everybody who reads this comment section would take some time to leave a response.

@Christian,

I tried that on Red Hat Linux and it works without any problem. I tried on 2.6 kernel. But, I don't think that is the issue for you.
Can you make sure that you type the following command, instead of copy/paste. Sometimes, copy/paste may do some weird things.

```
$ export HISTTIMEFORMAT=' %F %T';
```

@Strang,
Why would you want to display the history in that format without using HISTTIMEFORMAT? Can you please provide little bit more details on what you are looking for?

@Tom,
Thanks for following up on this comment section with an answer to your own question.

@Parag,
mm.. I wonder why Ctrl-R didn't work for you. Is Ctrl-[character] combination somehow disabled on your system. Anyway, I'm glad you found the Alt-P helpful instead.

@Thiru,
What log are you trying to look for? Can you explain little bit more in detail on what exactly you are trying to achieve?

@Srinivas,
Yeah. Once you start using HISTTIMEFORMAT and HISTIGNORE, you will wonder, how you've used Linux command-line for so long without using this feature.

∞

Sourabh

June 23, 2009, 12:37 am

I may require the argument of nth cmd instead of previous command…
How can i get it ?

∞

yakupm

June 24, 2009, 1:57 pm

Excellent as you can tell from the ongoing positive feedback
-yak

∞

Ramesh Natarajan

June 24, 2009, 7:08 pm

@Sourabh,

```
3  ls /etc/sysctl.conf
4  export HISTTIMEFORMAT='%F %T';
5  uname -a
```

Let us say in the above example, you want to take the argument of history command#3 (which is ls /etc/sysctl.conf) and use it. You can do that as shown below.

```
$ cat !3:$
[Note: This is equivalent to "cat /etc/sysctl.conf" ]
```

@yakupm,
Thanks for your comment. I'm very glad that you found this article helpful.

∞

agn

July 6, 2009, 6:34 am

You can also use $_ to get the last arg of the previous command.

$ ls /etc/passwd
$ cat $_

Also, 'cd -' takes you to your previous location.

$ pwd
/etc/apache2/sites-enabled
$ cd /var/log/apache2
$ pwd
/var/log/apache2
$ cd –
$ pwd
/etc/apache2/sites-enabled

∞

Ramesh Natarajan

July 8, 2009, 10:20 pm

@Agn,

Thanks for explaining the meaning of $_ . It is definitely very helpful, when there are several arguments in the previous command and we want only the last one.

Regarding the "cd -", it is definitely an awesome cd hacks, that I've mentioned a while back in my 6 awesome cd command hacks article.

∞

Sebastian Kusnier

October 13, 2009, 1:55 am

Other way to get the last arg of the previous commmand is:

ESC-_

ls /etc/passwd
cat[ESC-_] -> cat /etc/passwd

∞

Keith Hearn

October 27, 2009, 4:42 pm

This is a great resource.

Another useful one is !* which give you all of the arguments from the previous command:
$ ls foo bar bing
foo bar bing
$ echo !*
echo foo bar bing
foo bar bing

You can also get the Nth argument with !:N
$ ls foo bar bing

```
foo bar bing
$ echo !:2
echo bar
bar
```

You can combine this with matching the most recent line that begins with a particular string:

```
$ echo !ls:1
echo foo
foo

$ echo !ls:*
echo foo bar bing
foo bar bing

$ echo !ls:0
echo ls
ls
```

Ok, the last one seems kinda silly, but it could be useful if you have several commands that start with the same string and you want to know which one you did most recently.

```
$ foocreate
$ foostart
$ foostop
$ foostart

$ echo !foo:0
echo foostart
foostart
```

Keith

∞

Nanda

March 15, 2010, 12:37 am

Hi,

We have one server and everybody using that server logs in using root and we do not normally logged out. I am just wondering if I can record all the commands type from all terminal including ssh terminals.
Is there any way ?

∞

dojuba

May 5, 2010, 2:18 am

A very useful article and comments. In case anybody is interested, in HISTIGNORE wildcard * works. Like
HISTIGNORE="ls:ls *:"
will ignore ls, ls -ltr, ls -a etc but not lsattr, or other commands that start with 'ls'
(which is obviously the case if you put ls*)

∞

adam

May 23, 2010, 10:24 pm

hello there!
thank you,

I have to use "history -w" after "history -c" to Clear all the previous history.
but why ?

∞

Hello

July 12, 2010, 10:21 pm

It is very good that your document it will very effective for me to develop my knowledge

thank you very very much so what i can do for you

∞

steeven paul

September 14, 2010, 3:56 am

Wonderful articles. thank u very much.

∞

Sameed

September 22, 2010, 2:40 am

Nice articles !!!! 😜

I was trying to Block only " history -c" command. But history command came under bash. Is there any way to prevent the user to delete there history. ?
User should not allow to delete history ... 🙁

∞

stanley

November 18, 2010, 12:39 pm

friends,
is they any way to complete the recent commands, by typing just a few letters of it?
and pressing ALT+P, which gives the history of the commands?
thanks

∞

Maha

November 23, 2010, 1:40 pm

Best description read so far about History ... fast and clear to understand 🙂
Thanks!

∞

Paul Makepeace

February 2, 2011, 10:08 am

You can use !$ for the last arg of the last command (little shorter than !!:$) I used to use this so much back in the day before up & down arrows working on all terminals it's wired into my muscle memory. V useful!

∞

jigs

March 2, 2011, 12:46 pm

does "history -c" command leaves no trail? are the sessions or transactions also deleted in that "history -c" command?

∞

santhosh

March 18, 2011, 4:53 am

super

∞

Vishwajeet Kumar

April 21, 2011, 8:24 am

The tutorial is excellent

∞

Alex

June 7, 2011, 1:36 am

Thanks for the great tips!

I have another question, how to make the history to show the time stamp (you already did), userID, ip address and the command?

∞

Andy Lee Robinson

June 29, 2011, 3:57 am

Some nice tips there
“cd -” is one command I’ve been looking for since 4dos did the same using “cdd”

What I really really missed was the forward version of Ctrl-R
When searching back for a command, I frequently overshoot and can’t go back.
(histsize=1000000 🙄
The man page says it is Ctrl-S, but this clashes on the command line and suspends output until Ctrl-Q is pressed.

My solution is to add this line to ~/.inputrc in your home dir:
Control-t: forward-search-history

(Ctrl-T transposes two chars and not as useful as searching forward, so can be dumped…)

Et voila!

∞

Andy Lee Robinson

July 8, 2011, 11:44 am

Actually, add
Control-t: forward-search-history
to /etc/inputrc
or copy /etc/inputrc to your home dir and modify.

The method I described stopped the HOME and END keys working, because a user’s ~/.input.rc seems to replace and not append to the server default in /etc/input.rc !

∞

Shivaprakash

September 7, 2011, 7:48 am

Nice tutorial

∞

Mike

September 29, 2011, 9:32 am

As far as your request to notify you of a use for an alternate history file, I have one:

We track various users' commands to determine what the commands were that produced a bad server/result/etc. Users don;t always provide honest answers. So I hide the history file in another portion of the file system and sort them out by user.

However, I do have one question that I have not been able to figure out a solution for.

When I set the HISTTIMEFORMAT to show date and time (%F %T) and also set the HISTFILE to my custom location, the time format is *always* a UNIX timestamp, and never in a user-readable format.

Does anyone have a solution for this?

∞

VJ

October 5, 2011, 7:31 am

Thanks a lot , really helpful in finding something that I was looking for.

∞

Arun Saha

October 16, 2011, 6:57 pm

Very helpful indeed!

We saw three uses of HISTCONTROL

export HISTCONTROL=ignoredups
export HISTCONTROL=erasedups
export HISTCONTROL=ignorespace

Are they mutually exclusive? What if someone wants both "ignoredups" and "ignorespace"?

∞

Jan Goeteyn

October 18, 2011, 8:04 am

Please note that the F8 behavior (= cycling only that part of the history that begins with the characters you typed so far) and is known from the Windows cmd.exe shell, has a linux equivalent by pressing Ctrl r again to find the next previous instance of that word or letter. Subsequently pressing backspace cycles in the oposite direction until you are again at the most recent instance still matching that word or letter.

This also answers the issue from Andy Lee Robinson, who frequently got an overshoot due to his entousiasm to repeatedly enter the Ctrl r command and can't go back: Use back space to go back.

∞

Seb

October 22, 2011, 6:20 am

@ Arun :

export HISTCONTROL=ignorespace:erasedups

By the way, for me on Ubuntu 10.10 with
export HISTIGNORE=”ls:ls -ltr”

ls -ltr still appears in history, I need to set “ls *”

∞

Abdullah nawash

December 28, 2011, 2:58 am

Thank you so much for enlightening us.
use history -d (to delete specific line from history)

∞

Abhijit

January 3, 2012, 1:35 pm

Awesome work Ramesh !!! Kudos to your efforts.

∞

BUKUMI OLUYEDI

January 5, 2012, 3:22 pm

Kindly apprise more on the LINUX COMMANDS. I am a undergraduate student of computer engineering from Nigeria.

∞

acebee

January 24, 2012, 2:18 am

excellent – reminds me of time spent distant past in bengaluru on a certain floor sitting on a old as400..

∞

Uma

March 2, 2012, 1:21 pm

I have a great use for redirecting HISTFILE to a desired location. I have a quota limit on my users directory and recently my .bash_history got erased because I was close to the limit! I am hoping by redirecting the history file to a different location I can prevent this fiasco!

∞

Madhav

March 17, 2012, 11:01 pm

It is a good one , great , more clear

∞

sriram

March 24, 2012, 1:47 am

Thanx Very useful,

Regards,
sriram.a

∞

Deepak

April 13, 2012, 7:21 am

Thanks for the book it has enhanced my speed of work using those short cuts, i even recommended it to my other mates. please do publish such books even on networking will be helpful

∞

ratnakarreddy

April 24, 2012, 1:28 am

Hi,
I have been using CentOS Linux release 6.0 and i am unable to get the history of all the users.

When i run history command i am getting only current user's history (HISTSIZE=1000)

I want to get all the history of each and every user in my server, if i run #history (may be other) as root user.

Please help me…

∞

Uma

May 21, 2012, 9:13 am

I am interested in saving my history file on a daily basis using a cronjob, but have been unsuccessful so far! Do you have any suggestions on how to do this?

Uma

∞

kishore

June 15, 2012, 7:45 am

How to view the history of a file

ie) what are all the modification happended wrt a file

say if it is /etc/passwd – last 5 modifications

∞

Paul King

August 6, 2012, 7:22 am

I think the reason for the HISTFILE feature would be for a large network where the history file name has been standardized for all of the users. A user could also chang the name to preserve a session where he knows that the commands will be unique, intending to change it back later on.

∞

Randy S.

August 11, 2012, 5:21 pm

I wanted for some time to understand command history in a more usable way.
This looked like a good post to spend the effort with.
Man, was it ever!
PER Ramesh Natarajan ..
“you’ll find using command line more enjoyable and fun.”
Definitely true. Thank you Mr. Natarajan.

∞

J.D. Laub

September 14, 2012, 5:13 pm

I set HISTFILE=/dev/null . Concurrent shells don’t share their history (which used to drive me batty), and the history is forgotten when the shell dies.

∞

Paul

September 18, 2012, 8:10 pm

Very handy — thanks!

∞

Anonymous

October 18, 2012, 8:26 am

When multiple sysadmins su to the same user such as 'root' or 'oracle' it is helpful to separate out the history for each user. This is easily done by adding the following to the profile. This is especially useful on HP-UX.

export HISTFILE=${HISTFILE:?}.$(logname)

∞

Uriel

October 28, 2012, 3:26 am

I have several (around 5) bash sessions opened simultaneously. When I want to re-execute a certain command, history would not be helpful, because it stores commands only from the *current* shell, and I don't remember from which shell I run each command. How can I overcome this?

∞

John R

November 11, 2012, 9:46 am

Uma,

You could use the HISTFILE in your bash settings to include the date of the session – that way a new history file would be created on each day.

Automatic backup. Tada!

∞

Mike

November 14, 2012, 2:22 pm

Here's a script I put in /etc/profile.d that takes effect upon login:

```
if [ $UID -eq 0 ] ; then
shopt -s histappend
mkdir -p /.histories
```

```
export HISTTIMEFORMAT=”%F %T ”
HISTFILE=/.histories/.”$(who am i | awk ‘{print $1}’)_`date +%Y%m`_history”
touch /.histories/.”$(who am i | awk ‘{print $1}’)_`date +%Y%m`_history”
chattr +a /.histories/.”$(who am i | awk ‘{print $1}’)_`date +%Y%m`_history”
HISTIGNORE=””
HISTCONTROL=””
HISTSIZE=1000
readonly HISTFILE
readonly HISTIGNORE
readonly HISTCONTROL
readonly HISTSIZE
readonly HISTFILESIZE
readonly PASSWORD_COMMAND=”history -a”
export HISTFILE HISTIGNORE HISTCONTROL HISTSIZE HISTFILESIZE
echo “———New Session `date +%Y%m%d`———” >> $HISTFILE
fi
```

It will sort out history files based on who su’d to root. Normal user history files are in the /home as usual. This allows us to detect who may execute commands as root since we have multiple people with root access on several systems.

Obviously, I have not publicized the existence of this script so users will not know to search for these hidden history files.

∞

antonior

December 14, 2012, 5:31 am

Hi! I have to get more information from history command, so the remote host and the pts is possibile??

∞

Command Line Warrior

December 15, 2012, 10:11 pm

I have /home shared across multiple servers with NFS and I've found it very useful to adjust your .bashrc and other .dot.files to reflect the server host name etc..

HISTFILE=~/.bash_history-$HOSTNAME
or something like this one,
HISTFILE=~/.bash_history-$HOSTNAME-$USER

∞

hi

March 18, 2013, 6:35 am

you as for resons to rename history file….
what i like to do it keep the file in an seperate log folder and everyday start new log with current date as file name…. any idee if this is posible?

∞

AlexZ

April 3, 2013, 10:39 pm

Ctr-R – i'm happy. Thanks!

∞

Jym

April 4, 2013, 1:50 pm

≖ Excellent uses for $HISTFILE have already been enumerated. Here's another one: thwarting script kiddies who may have compromised your host's system, or your media. Having your history off in some idiosyncratic place will slow down anyone trying to paw through $HOME/.bash_history for information.

(Not a perfect solution, of course; a script could glean $HISTFILE from the appropriate dotfiles in $HOME — though it's also possible to put that in, say, an export.rc file in yet another idiosyncratic place.)

∞

Srinvasu

April 11, 2013, 11:27 pm

Nice..

∞

Agent Marty

June 19, 2013, 2:31 am

When you write a command put a comment after it.
The comment will be included in history and later you can grep for the comment.

∞

Elias Cantos

July 18, 2013, 8:31 am

Great article. Well written. Good examples. Succinct.

∞

Jesse

August 7, 2013, 8:01 am

Thanks for your article, I’ve only read down to tip #7 but I’ll be sure to read the rest when it’s not so late at night and I don’t have to work in the morning 😉

Re changing the name of the history file:
I’ve recently improved the passwords I use for various things from being character mnemonics such as 2BEis4me! to being 50 character phrases. However, I’d been using the bash shell to type them out where I could see them to check they were right before copy and pasting them into my browser. I then realised they were being SAVED in the bash history!!

I’ve deleted them from the history now but changing the file name/location of the history file would appear to make it more difficult for a malicious script to access. It wouldn’t stop someone who knew what they were doing because they could just read it pragmatically but… I don’t know, if someone managed to inject code into something capable of reading files, like a panel app that appeared to do something legitimate, having the history called something different would make it less compatible. That’s the only reason to change the location that I can think of, but one of the things I really like about Linux is that rather than hardcode things that no one can think of a reason to change they’re left open so that should one want to (or find themselves in a situation where it is useful) the stuff no one can think of a reason to change still can be changed if needed.

Wow that took much longer to say than I thought it would.

All the best, and thanks again for your tips.
🙂
Jesse

∞

---

asdf@asdf.tld

August 29, 2013, 12:01 pm

I love you. Thanks for the history howto.

∞

Harish Puvvula

October 11, 2013, 3:58 am

You rock! Haven’t found a better tutorial than this! Thanks a ton

∞

---

ksw

November 11, 2013, 7:49 pm

Definitely, the most useful tips here

∞

Masoud kh

February 22, 2014, 8:00 am

very good examples….thanks Ramesh Natarajan

∞

s sudhakar

June 3, 2014, 1:13 am

how to share a common partition between linux and windows..

∞

David Ongaro

June 8, 2014, 7:15 pm

Andy Lee Robinson@: Actually Ctrl-t is very useful, because transposition of two characters is one of the most common typing mistakes. You should get used to using it!

Use

stty stop ^

to disabling control flow mapping of Ctrl-s for the terminal then search forward can work with the standard key.

∞

Unknown

July 2, 2014, 6:37 am

To track command executed by all user, i'm using .bach_logout to send via mail all command executed on history. To get each histfile separated by user and by

session, i'm using this:
HISTFILE=~/.shHistory_${DATEX}_$(echo $(tty)|cut -d "/" -f4)

To be clear, the history date format was changed like below:
HISTTIMEFORMAT="%Y-%m-%d %T "

The issue is the Time format sended is still in Unix timestamp. By using history command, it's OK.

```
$ history
1 2014-07-02 15:31:50 Login cmd started date 2014-07-02 15:31:50
2 2014-07-02 15:31:53 echo $HISTLOG
3 2014-07-02 15:31:59 vim .bash_profile
```

Does anyone have an idea to resolve this?
Thanks

∞

TJGeezer

July 20, 2014, 10:27 am

To leave no permanent history of a session, set the history file in /dev/shm/ which is actually RAM space. The history will be available during the session but will disappear in a reboot.

∞

phatrick

September 1, 2014, 5:02 am

By far the best way to enable history use is to use the incremental "prefix" search made available through the .inputrc file.

This way you can search the current text by pressing the up and down arrows:

$ foo

press the arrow key once
foo xxx
press the arrow key again
foo yyy

etc..

Note: you still need Ctrl+R if you want to search for text not at the start of the line..

Incremental searching with Up and Down is configured in ~/.inputrc add these lines:

```
### For Sane incremental history search, simply type the first few letters of the
### command and the up arrow/down arrow to search the subset of the history
“\e[A”: history-search-backward
“\e[B”: history-search-forward
## Old behaviour still available with Ctrl+P and Ctrl+N
## If that prevents Left and Right from working, fix them like this:
#”\e[C”: forward-char
#”\e[D”: backward-char

#fix ctrl+arrow word jumping in ubuntu
#existing ~/.inputrc file:
“\e[1;5C”: forward-word
“\e[1;5D”: backward-word
“\e[5C”: forward-word
“\e[5D”: backward-word
“\e\e[C”: forward-word
“\e\e[D”: backward-word
```

∞

DroBuddy

October 12, 2014, 5:34 am

OMFG, I never knew about Ctrl+R and that is so powerful! Awesome guide, broha….

∞

honey

March 6, 2015, 9:05 am

is there any simple command to directly check previous day history and remove it for space?

∞

Edgar Gregori

March 14, 2015, 12:01 pm

Tks for No. 4.

∞

Richard Black

May 6, 2015, 6:40 pm

“history -r” for when you accidentally type your password at the prompt. It will “revert” history back to when you started this session:
“-r Read the contents of the history file and append them to the current history list.”

∞

Claudio

May 8, 2015, 2:26 am

Hello, I am curious if this is possible…

On my systems I set HOSTFILE variable in .bashrc file like this

HISTFILE=$HOME/hist/`uname -n`.`date +%y%m%d`.$$

This has the advantage to keep under “$HOME/hist/” directory all the history files by user sessions. So if an account is shared between more than one user they can keep their history sessions separated and stored.
The big disadvantage of this approach is that I cannot use the history command (or history reverse search Ctrl+R) against previous sessions because basically HISFILE is always different and unique at each new session.

I am wondering if is it possible to say to the bash to run history commands against a number of different text history files contained in to a directory instead of a single file. Some sort of HISTDIR variable??
Does it exist? Could it be a bash new releases feature request?

Thanks,
Claudio

∞

Anonymous
July 20, 2015, 9:08 pm

Hi,
Nice article. However, unless I missed something in your text, this tutorial is for Redhat based distros only? Right?

∞

---

tomas
July 29, 2015, 7:07 am

Hi, with alt + . you saved about 2 years of life ! thanks..
I recommend you using:
ctrl + u … delete whole line
ctrl + w … delete word
ctrl + k … delete from cursor position to end

∞

AMit

December 17, 2015, 10:57 pm

Thanks.

I want to do, “Execute command without keeping history” . I don’t want to show in terminal when run the History command.

∞

CatWoman

February 4, 2016, 2:30 am

14. Disable the usage of history using HISTSIZE
# export HISTSIZE=0
# history
# [Note that history did not display anything]

∞

CatWoman

February 4, 2016, 2:32 am

Hi Amit

It’s already there in Number 14.

14. Disable the usage of history using HISTSIZE
# export HISTSIZE=0
# history
# [Note that history did not display anything]

∞

Taimur

June 8, 2016, 12:30 am

i want to know where does the history or logs of user modification / group modification????

∞

David

December 8, 2016, 3:37 am

For point 7 : want to use it to identify on which server i use this command (having a common history for 100 servers is very confusing and near useless at the end).

Problem after is : how to check the history from a file. i don't find a param like history -f . Set HISTFILE dynamically not easy and will write all search in the bad file then 🙄

i try look for a tag to put in history log to get a line like :

but found nothing at this time.

∞

Monil

March 9, 2017, 1:53 am

We have a common Linux test env where multiple people work(connect through ssh using Putty), We have a requirement to store commands fired from all terminals in such a way that we can track each and every command, fired by which IP/user and from which TTY so that we can verify it using 'last' command.

I have changed my .bash_profile accordingly

```
# Will Give me ip of person logged in
WHOAMI=`who -m | cut -d '(' -f 2| cut -d ')' -f1`
# Will give me tty ID
MYTTY=`who -m | awk '{print $2;}' | cut -d '/' -f2`
DATE=`date +"%Y_%m_%d_%H%M%S"`
DAY=`date +"%Y_%m_%d"`
```

```
shopt -s histappend
mkdir -p $HOME/HISTORY/${WHOAMI}/${DAY}
touch $HOME/HISTORY/${WHOAMI}/${DAY}/.HIST_${MYTTY}_${DATE}
export HISTTIMEFORMAT='%F %T '
export HISTFILESIZE=100
export HISTSIZE=100
# stores history file per terminal
export HISTFILE=$HOME/HISTORY/${WHOAMI}/${DAY}/.HIST_${MYTTY}_${DATE}
export PS1='[\[\e[4;32m\]\u@\h\[\e[0m\] \[\e[1;36m\]$PWD\[\e[0m\]]\! $'
# Updates the HISTFILE at real time i.e. when user presses enter
export PROMPT_COMMAND="history -a; history -c; history -r; ${PROMPT_COMMA
history -r $HISTFILE
```

After changing the .bash_profile, the history command stopped showing previous entries.

When I changed PROMPT_COMMAND to "history -a; history -r; ${PROMPT_COMMAND}" it started working but
The HISTFILE is not updated at real time; it's only updated when exit command is fired.
If user disconnects the putty session by right click and using disconnect option, HISTFILE is not updated at all. 😟

P.S:- If i comment export HISTFILE it stores to .bash_history file and everything works smoothly and .bash_history is updated in real time, but I do not get the tty id or IP from which command was fired

O.S:- Red Hat Enterprise Linux Server release 6.8 (Santiago)

∞

Kavitha H R

March 15, 2017, 11:39 pm

always nice stuff:)

∞

pankaj srivastava

May 1, 2017, 11:00 pm

Hi,
your history command uses are very useful for me. I want to know that can we store command name and its output in history or any other way ? If yes please help me.
pankaj srivastava

∞

Next post: mysqlshow – Get Quick Info On MySQL DB, Table, Column and Index
Previous post: 6 Steps to Secure Your Home Wireless Network

RSS | Email | Twitter | Facebook

EBOOKS

Free
Linux 101 Hacks 2nd Edition eBook - Practical Examples to Build a Strong Foundation in Linux

Bash 101 Hacks eBook - Take Control of Your Bash Command Line and Shell Scripting

Sed and Awk 101 Hacks eBook - Enhance Your UNIX / Linux Life with Sed and Awk

Vim 101 Hacks eBook - Practical Examples for Becoming Fast and Productive in Vim Editor

Nagios Core 3 eBook - Monitor Everything, Be Proactive, and Sleep Well

The Geek Stuff

16,196 followers

Follow Page

Share

POPULAR POSTS

- 15 Essential Accessories for Your Nikon or Canon DSLR Camera
- 12 Amazing and Essential Linux Books To Enrich Your Brain and Library
- 50 UNIX / Linux Sysadmin Tutorials
- 50 Most Frequently Used UNIX / Linux Commands (With Examples)
- How To Be Productive and Get Things Done Using GTD
- 30 Things To Do When you are Bored and have a Computer
- Linux Directory Structure (File System Structure) Explained with Examples
- Linux Crontab: 15 Awesome Cron Job Examples
- Get a Grip on the Grep! – 15 Practical Grep Command Examples
- Unix LS Command: 15 Practical Examples
- 15 Examples To Master Linux Command Line History
- Top 10 Open Source Bug Tracking System
- Vi and Vim Macro Tutorial: How To Record and Play
- Mommy, I found it! -- 15 Practical Linux Find Command Examples
- 15 Awesome Gmail Tips and Tricks
- 15 Awesome Google Search Tips and Tricks
- RAID 0, RAID 1, RAID 5, RAID 10 Explained with Diagrams
- Can You Top This? 15 Practical Linux Top Command Examples
- Top 5 Best System Monitoring Tools
- Top 5 Best Linux OS Distributions
- How To Monitor Remote Linux Host using Nagios 3.0
- Awk Introduction Tutorial – 7 Awk Print Examples

How to Backup Linux? 15 rsync Command Examples

The Ultimate Wget Download Guide With 15 Awesome Examples

Top 5 Best Linux Text Editors

Packet Analyzer: 15 TCPDUMP Command Examples

The Ultimate Bash Array Tutorial with 15 Examples

3 Steps to Perform SSH Login Without Password Using ssh-keygen & ssh-copy-id

Unix Sed Tutorial: Advanced Sed Substitution Examples

UNIX / Linux: 10 Netstat Command Examples

The Ultimate Guide for Creating Strong Passwords

6 Steps to Secure Your Home Wireless Network

Turbocharge PuTTY with 12 Powerful Add-Ons

CATEGORIES

Linux Tutorials

Vim Editor

Sed Scripting

Awk Scripting

Bash Shell Scripting

Nagios Monitoring

OpenSSH

IPTables Firewall

Apache Web Server

MySQL Database

Perl Programming

Google Tutorials

Ubuntu Tutorials

PostgreSQL DB

Hello World Examples

C Programming

C++ Programming

DELL Server Tutorials

Oracle Database

VMware Tutorials

About The Geek Stuff

My name is **Ramesh Natarajan**. I will be posting instruction guides, how-to, troubleshooting tips and tricks on Linux, database, hardware, security and web. My focus is to write articles that will either teach you or help you resolve a problem. Read more about Ramesh Natarajan and the blog.

Contact Us

Email Me : Use this Contact Form to get in touch me with your comments, questions or suggestions about this site. You can also simply drop me a line to say hello!.

Follow us on Twitter

Become a fan on Facebook

Support Us

Support this blog by purchasing one of my ebooks.

Bash 101 Hacks eBook

Sed and Awk 101 Hacks eBook

Vim 101 Hacks eBook

Nagios Core 3 eBook