THE GEEK STUFF
Linux | DB | Open Source | Web

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
```

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
# 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

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
```

```
# 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
```

```
# pwd
# history | tail -3
56  export HISTCONTROL=ignoredups
57  pwd [Note that there is only one pwd command in the history, even aft
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

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 ser
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
```

```
# 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]
```

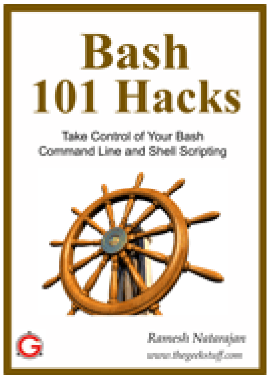
15. Ignore specific commands from the history using HISTIGNORE

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