

By Jack Wallen

The Linux desktop has come a long, long way, but there are still times when I have to use the command line. (I am a hardcore user, after all.) But even though I'm used to typing, spending hours upon hours with my fingers at the keyboard, I still grow tired of typing the same commands over and over. To reduce that tedium, I always add aliases to my `.bashrc` file.

What is an alias?

An alias is basically a shortcut for a command you place in your `~/.bashrc` file. Aliases cut down on typing and can save you from having to look up a command. (If your memory is like mine, this can be a real boon!)

Aliases are set up near the bottom of the `.bashrc` file. You'll see a commented-out section that indicates where you should put them. The format of an alias is:

```
alias NICKNAME='full command here'
```

The keyword *alias* must be used. The nickname is what you will type at the command line. Make this nickname easy to remember. The `=` sign must also be used. After the `=` sign, you enter the full command, including flags and switches, enclosed in single quotes. Once you are done, save the `.bashrc` file and open up a new terminal. I always find it best to leave the original terminal window open in case there are problems. In the new terminal, type the alias nickname and the command will run.

To get you started, I've compiled the following list of aliases I have used over the years to help make my command-line experience a bit easier.

1 The ssh alias

This one should be a no-brainer for those of you who frequently secure shell into particular boxes. For this I add an alias like so:

```
alias server_name='ssh -v -l USERNAME IP ADDRESS'
```

Just change `server_name` to a memorable name for the server. Then, change `USERNAME` and `IP ADDRESS` to suit your needs.

2 The ls aliases

Some distributions don't include some of the handier `ls` commands. Generally, I like to see full listings instead of just filenames. For that I always include this alias:

```
alias ll='ls -l'
```

Another handy `ls` alias is this:

```
alias la='ls -a'
```

3 The rm safety net

I can't tell you how many times I have "rm'd" a file I shouldn't have "rm'd". To avoid this, I add this alias:

```
alias rm='rm -i'
```

Adding the `-i` flag it forces `rm` into interactive mode, which asks whether you're sure you want to remove a file.

4 A more useful df command

This handy tool tells you how much space you have left on a drive. Only thing is, if you run the command by itself it replies in 1K blocks. Most people would prefer to see this in terms of MB. To make that happen, add this alias:

```
alias df='df -h'
```

Now, every time you run the df command, the information will be returned in a human-readable format.

5 The nonstandard Firefox

Many times, I install Firefox in strange directories (or have more than one version of Firefox installed for testing purposes). For this, I will add an alias to start the correct Firefox. Say, for example, I have the beta of the newest, upcoming Firefox release installed, as well as the current stable Firefox. They are both installed in my home directory in different subdirectories. I will then add two aliases like so:

```
alias ff1='/home/jlwallen/firefox/firefox'  
alias ff2='/home/jlwallen/firefoxb3/firefox'
```

Now I can start the stable firefox with ff1 or the beta with ff2.

6 The bookmark alias

Speaking of Firefox, let's create an alias to open up it to a specific URL:

```
alias fftr='/home/jlwallen/firefox/firefox http://www.techrepublic.com'
```

This alias will open Firefox directly to the TechRepublic Web site.

7 The constant editing of a file

There are certain files that I am constantly editing. For instance, when I used Enlightenment E16 (I now use E17), I was frequently editing the menu file `~/e16/menus/user_apps`. Instead of constantly opening up a terminal and entering `nano ~/e16/menus/user_apps`, I used an alias that allowed me to type `emenu` and start editing. I used this alias:

```
alias emenu='aterm nano -e ~/.e16/menus/user_apps'
```

Now, I just enter the command `emenu` (or I can enter that in the run command dialog) to open up this file in an editor.

8 The apt-get update

There are numerous ways to use an alias to help you with apt-get. One of my favorite is to add this alias:

```
alias update='sudo apt-get update'
```

I only need to enter `update` and will be prompted for the sudo password. You can modify this to suit your frequent apt-get needs.

9 The rpm batch install

I like to do a lot of batch installing with rpm. I will typically dump a bunch of rpm files into an empty directory (created for this specific purpose) and run the command `rpm -ivh ~/RPM/*.rpm`. Of course, an alias makes this even easier:

```
alias brpm='rpm -ivh ~/RPM/*.rpm'
```

You have to create the `~/RPM` directory and enter the root password for this to work.

10 The long, arduous path

There are some paths that I often change to that seem to take eons to type. When I was working on the Afterstep window manager, I had to constantly change to the `~/GNUstep/Library/AfterStep/start` to edit menus. After a while, you get tired of typing `cd ~/GNUstep/Library/AfterStep/start` just to get to the directory. So I added an alias like so:

```
alias astart='cd ~/GNUstep/Library/AfterStep/start'
```

Naturally, you can change that to fit your needs. This will save you a lot of typing.

So there you have it: a few simple bash aliases that will ease the load on your fingers. You can modify them to suit you, and they'll give you a good start on creating your own handy bash aliases.

Additional resources

- TechRepublic's [Downloads RSS Feed](#) **XML**
- Sign up for the [Downloads at TechRepublic](#) newsletter
- Sign up for our [IT Leadership Newsletter](#)
- Check out all of TechRepublic's [free newsletters](#)
- [Get quick control over your Linux server with server commands](#)
- [How do I... Install plugins for Firefox in Linux?](#)
- [Customize Linux with these 10 boot prompt options](#)

Version history

Version: 1.0

Published: May 14, 2008