

# Linux Aliases and Hidden Files

This document explains aliases – keyboard shortcuts – and how to use them. (Naturally, my alias list includes Seismic Unix (SU) aliases.) Along the way I explain hidden files.

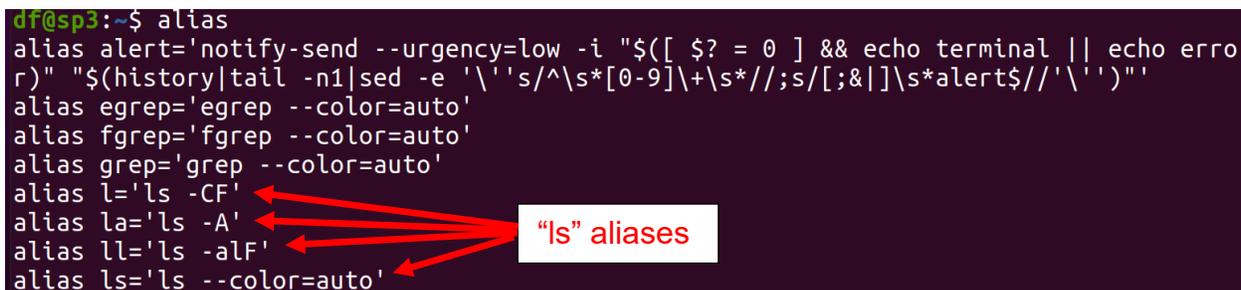
**Caution:** This discussion is about the Ubuntu Linux distribution. A different flavor of Linux might not use the `.bashrc` and `.bash_aliases` files as they are discussed here.

## 1. Starter aliases

Aliases are keyboard shortcuts; that is, compact commands. Most aliases people make are shortcuts of the “ls” (list files) and “cd” (change directory) commands but an alias can also run a program. Aliases can be many things.

I installed Ubuntu 20.04. Let us pretend I have not defined any aliases. If I enter the “alias” command with nothing following it, I will see all my available aliases:

```
df@sp3:~$ alias
alias alert='notify-send --urgency=low -i "${[ $? = 0 ]}&& echo terminal || echo error" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\''")'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
```



Although I did not define any aliases, I have eight aliases! The managers of Ubuntu put these in my `.bashrc` file. Above, the last four aliases are variations of the “ls” command; that is, they list my files in different ways. The first is a slightly more colorful version of the basic “ls” command. The second lists hidden files (described below) with other files. The third lists files, including hidden files, in “long” format – one row per entry with permissions, file size, and more. The fourth (bottom) adds color to any `ls` command.

## 2. Hidden files and “man” pages

Because the name of my bash resource file (`.bashrc`) starts with a period, it is a “hidden” file. The usual command to “list” files in a directory is:

```
df@sp3:~$ ls
Desktop  Downloads  MyAliases.txt  Pictures  snap  Videos  work2
Documents  Music      ozle01.su     Public   Templates  work
```

But this does not show hidden files, files that begin with a period. Hidden files are usually some sort of system file, which system people think we should not see. A simple way to see regular files *and* hidden files is:

```
df@sp3:~$ ls -A
.bash_aliases  .bashrc_backup Documents  .local      ozle01.su  snap      .viminfo
.bash_history  .cache        Downloads .mozilla    Pictures    .ssr      .wget-hsts
.bash_logout   .config      .fontconfig Music       .profile   Templates  work
.bashrc        Desktop      .gnupg    MyAliases.txt Public      Videos    work2
```

To see the many options of the “ls” command, open the help manual (“man”) page for “ls”:

```
$ man ls
```

“man” pages exist for many unix commands. Examples are `cp` (copy) and `df` (disk file report). An example of `df` is later in this document.

### 3. Example Ubuntu `.bash_aliases` file

I like creating/using aliases, but I think it is dangerous to edit my `.bashrc` file. If I make a mistake in that file, it could suddenly be a problem for me to use my account. **I write from experience.** Fortunately, there is an alternative to putting aliases in the bash resource file. The following lines are in my default `.bashrc` file:

```
99 # Alias definitions.
100 # You may want to put all your additions into a separate file like
101 # ~/.bash_aliases, instead of adding them here directly.
102 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
103
104 if [ -f ~/.bash_aliases ]; then
105     . ~/.bash_aliases
106 fi
```

The lines above mean, when I log in or when I “source” my `.bashrc` file, the `.bashrc` file will read (if it exists) another hidden file, the `.bash_aliases` file.

When I write, “source” my `.bashrc` file, I mean that I enter the following command:

```
$ source .bashrc
```

Interesting fact: a new Ubuntu installation does not have a `.bash_aliases` file. So, I created file `.bash_aliases` in my home directory. Below are the contents of my `.bash_aliases` file.

```
df@sp3:~$ cat .bash_aliases
# "ls" aliases
alias l='ls -CF'
alias la='ls -a'
alias ll='ls -lF'
alias lla='ls -AlF'
alias llrt='ls -lFrt'

# more aliases
alias rm='rm -i'
alias zo='~/work/zorro.sh'

# aliases for "cd"
alias sudemos='cd $CWPROOT/src/demos'
alias susrc='cd $CWPROOT/src/su/main'
alias sugraph1='cd $CWPROOT/src/su/graphics'
alias sugraph2='cd $CWPROOT/src/xplot/main'
alias supar='cd $CWPROOT/src/par/main'
#
```

Most of the bottom “cd” aliases let me quickly move into directories that are source code (C code) for Seismic Unix. The exception is “sudemos,” my quick change to the very rich folders of SU demonstration scripts.

**Important:** The command to create an alias has no spaces around the equals (“=”) sign.

## 4. An alias to run a program

Suppose I have the text below in file `zorro.sh` in my “work” directory, a directory that is just under my home directory.

```
df@sp3:~/work$ cat zorro.sh
#!/bin/bash
echo " --> whoami = $(whoami)"
echo " --> CWPROOT = $CWPROOT"
echo ""
df -H
exit 0
```

I realize saying “home” can be confusing. Know that when people say “my home directory,” they almost always do NOT mean

```
/home
```

They almost always mean a top level *personal* directory, like

```
/home/david
```

or

```
/home/sara
```

I could have the following alias to execute the `zorro.sh` script:

```
alias zo='~/work/zorro.sh'
```

This example, the `zorro.sh` script, is not interesting, but if I create a script that I would run frequently, for example a diagnostic program, I might find it useful to create a shortcut.

After adding an alias (or after editing an alias) in the `.bash_aliases` file, I have to “source” my `.bashrc` file again to make the `.bashrc` file re-read the `.bash_aliases` file:

```
$ source .bashrc           if I am in my home directory
```

```
$ source ~/.bashrc        or, supply the path
```

If I have more than one terminal open when I “source” my `.bashrc` file, the other open terminals will not be able to use any new or edited aliases until I also “source” the `.bashrc` file in the other terminals.

## 5. Temporary alias

I do not find the following useful, but experts will laugh at me if I do not include this.

I can create a temporary alias; that is, an alias that only lasts as long as my current terminal is open. To do so, I create an alias on the command line, not in the `.bash_aliases` file.

```
$ alias zo='~/work/zorro.sh'
```

Now this alias is available until I close this terminal.

## 6. Naming an alias

Before I name an alias, I test that name on the command line to be sure the operating system (OS) is not using it. For example, I entered “flow”:

```
df@sp3:~$ flow
Command 'flow' not found, but can be installed with:
sudo snap install flow
```

This tells me “flow” can be a command, so, I would avoid “flow” for an alias name.

Another example: I typed “cc”:

```
df@sp3:~$ cc
cc: fatal error: no input files
compilation terminated.
```

Oops! That looks serious. To see the system “man” page for “cc” I entered:

```
$ man cc
```

and learned that “cc” is a command to compile C and C++ code. I would definitely avoid using “cc” as an alias name.

My point is, try to remember to test an alias name before using it. You could be overwriting a valuable system command.

## 7. Alias hierarchy

If I have the same alias defined more than once, the last alias is the one the system will use. My aliases are read, top-to-bottom in the `.bashrc` file, then my `.bash_aliases` aliases are read top-to-bottom. Again, if there are duplicate aliases, the last one read is the one the system retains for use.