

Linux Command Structure

In this narrative, consistent with Unix/Linux, all commands (and file names) are case sensitive. You will likely need to return to this page to review this material until it is mastered.

commands and utilities:

A quick note on the difference between commands and utilities.

A utility is a program. Examples of utility programs include who, date, and ls. Utilities are stored on hard drives and are only loaded into main memory when used.

A complete command is all of the text typed at the command line consisting of the utility name and all of the options/flags/switches and arguments

This subtle distinction between utilities and commands is generally ignored and the two terms are often used interchangeably. With this it is not wrong to call who, date or ls commands.

Single command

The most simple form of a UNIX command is just a single command by itself. Many UNIX/Linux commands will do useful work with just the command by itself. The next examples show two commands (who and date) by themselves.

The output generated by a single command by itself is called the default behavior of that command.

who

The who command will tell you all of the users who are currently logged into a computer. This is not particularly informative on a personal computer where you are the only person using the computer, but it can be useful on a server or a large computing system.

Failed command

The shell will let you know when you have typed something it doesn't understand.

Type "asdf" and then the ENTER or RETURN key. You should see a message similar to the following:

```
$ asdf
-bash: asdf: command not found
$
```

Options, switches, or flags

A command may optionally be followed by one or more options. The options are also commonly called flags or switches.

Options are usually a single character preceded by a minus sign or hyphen character.

As an example:

Adding the -u flag to the command date will cause it to report the time and date in UTC (Coordinated Universal) time (also known as Zulu Time and formerly known as Greenwich

Mean Time or GMT). The seconds are slightly higher in the second example because of the passage of time.

```
$ date
Sat Aug 25 19:09:19 PDT 2012
$
$ date -u
Sun Aug 26 02:09:27 UTC 2012
$
```

Option grouping

You can combine multiple options for a single command.

The following example shows the ls command by itself and with two options, first individually, and then combined.

```
$ ls
Desktop Downloads Movies Pictures scripts
Documents Library Music Public

$ ls -F
Desktop/ Downloads/ Movies/ Pictures/ scripts/
Documents/ Library/ Music/ Public/
$ ls -a
.          .cups    Library  scripts
..         .vminfo  Movies
.DS_Store Desktop  Music
.Trash     Documents Pictures
-bash_history Downloads Public

$ ls -a -F
./          .cups/   Library/  scripts
../         .vminfo  Movies/
.DS_Store  Desktop/ Music/
.Trash/    Documents/ Pictures/
-bash_history Downloads/ Public/
$
```

You can group option by listing them in any order (with a few rare exceptions) and either with their own minus sign or after a single minus sign. All of the following versions of the ls command work the same way:

```
$ ls -a -F
$ ls -F -a
$ ls -aF
$ ls -Fa
$
```

Arguments

Commands may also optionally have arguments. The most common arguments are the names of files.

Technically, the options just mentioned are also arguments, but in common practice options are separated from other arguments in discussions. Also, technically, the operators mentioned below are also arguments, but again it is useful to separate operators from other arguments in discussions.