A Primer on Command Line

4 BASH in LINUX/UNIX

Richard Van

The Go-To Guide for the Uninitiated

Yihan Shao Research Group (CC-ATS)
Department of Chemistry & Biochemistry
University of Oklahoma
USA
September 27th, 2022

Contents

1 Getting Started

2 Directories & Paths

2.1 pwd - print working directory

Pathnames help you figure out where you are in relation to the whole file system. Your current working directory (where you are) can be found with **pwd**. This command is useful when you need the full pathname to your current directory for copying/moving/downloading files and folders! Or sometimes you just forget where you are.

```
$ pwd
/home/<username>
```

2.2 ls - list

The **ls** command will list the contents of your working directory. If you are new on the supercomputer, you will not have any files/folders to list. If you're on your personal computer, you might see the following output (or something similar). Recall that directories are often blue unless modified.

```
$ ls
Desktop/ Downloads/
```

You can modify **ls** with *flags* to get more information about your contents. *Flags* are set by including the dash symbol (**-**) followed by a sort letter after the command.

For **ls**, some flags we use are:

```
$ ls -h # Human readable filesize (with -l returns 1K, 2G, etc.)
$ ls -l # Use a long listing format
$ ls -t # Sort by time, newest first
```

You can also combine flags to one dash (**-**), giving you the same output:

```
$ ls -hlt # Readable filesize, all listed, and sorted by time
```

2.3 mkdir - make directory

The **mkdir** command allows you to make a new directory in the current directory if no path is specified.

```
$ mkdir amino_acids
```

To see the directory you just created, type **ls**:

```
$ ls
Desktop/ Downloads/ amino_acids/
```

Subdirectories can be made if a path is specified, by adding the **-p** flag to **mkdir**. Multiple directories can be also be made if followed by a space (Check whether the directory is made by **ls**!!):

```
$ mkdir -p amino_acids/ala amino_acids/asn
$ ls
Desktop/ Downloads/ amino_acids/
$ ls amino_acids
ala asn
```

2.4 cd - changing directory

To change your current working directory and go in to another directy, you want the **cd** command. This command requires 1 argument which specifies the directory you want to change to. For example:

```
$ cd [path]
```

So if we wanted to **cd** into the **ala/** folder, we can type (Check whether you're in the directory you want with **pwd**!!):

```
$ cd amino_acids/ala
$ pwd
/home/<username>/amino_acids/ala
```

To go back a folder, follow the **cd** command by two periods (**..**). To go back 2 directories, just repeat the 2 periods (**..**) followed by a backslash (\). If you want to go all the way back to **/home/username** , just type **cd**. (Check with **pwd**!!)

```
$ cd ..
$ pwd
/home/<username>/amino_acids
```

or

```
$ cd ../../
$ pwd
/home/<username>
```

or

```
$ cd
$ pwd
/home/<username>
```

3 Managing Files

3.1 cp - copy

This command takes 2 arguments, for example:

```
$ cp [source1] [source2]
```

Where [**source1**] is the original file you want to copy, and [**source2**] is the new copied file. Note that [**source2**] can be a new file name or a pathname. We can copy an example script from my directory to your **/home/username** directory, if you **ls**, you should see the file:

```
$ cp /home/van/Scripts/bash_tutorial/example.txt /home/<username>
$ ls
Desktop Downloads amino_acids example.txt
```

The **cp** command can also copy directories if the flag **-r** is specified:

```
$ cp -r amino_acids aa_residues
$ ls
Desktop Downloads amino_acids aa_residues example.txt
```

A shorthand notation we often use is the single period symbol (**.**), which means the current working directory (or here).

```
$ cp /home/van/Scripts/bash_tutorial/aa.txt .
$ ls
Desktop Downloads amino_acids aa_residues example.txt aa.txt
```

3.2 mv - move

3.3 mv - rename

3.4 rm - remove

4 Display Files on Screen

4.1 clear - clear screen

4.2 cat - concatenate

4.3 less

4.4 head

4.5 tail