

A 15 Minute Introduction to the Linux Terminal

with Raspberry Pi

Data Services, University of Alabama Libraries

Vincent F. Scalfani

vfscalfani@ua.edu

Further Reading

Content in this presentation was adapted from some of our (much longer) previous Unix Shell Workshops. Check them out:

1. [Unix Shell - Fall 2020](#)
2. [Unix Shell for Data Transformation and Analysis - 2021](#)
3. [Unix Shell Introduction - Fall 2022](#)

Other key references:

1. [Software Carpentry: The Unix Shell](#)
2. [The Linux Command Line by William Shotts](#) (Book)

The author maintains a free PDF copy here: <http://linuxcommand.org/tlcl.php>

Raspberry Pi OS

Raspberry Pi OS is based on Debian Linux. There are many Linux distributions:

https://en.wikipedia.org/wiki/Linux_distribution

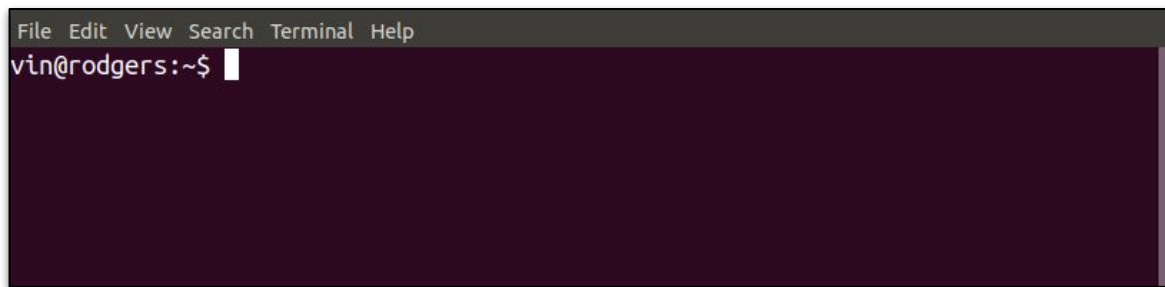
Other popular open-source distros are Ubuntu, Fedora, and Arch Linux.

There is a lot to like about Linux; here are a few of my favorites from using Linux full-time over the past 3 years:

1. It's free, open-source, and has a strong community for help
2. Programming environments are easy to set up
3. Software is installed through a package manager
4. Access to a terminal and many excellent command line programs and utilities

Unix Terminal

A Unix terminal is a text input/output environment [1]:



From the terminal input, a shell then interprets the commands (i.e., a command line interpreter).

Most Unix-like operating systems such as GNU/Linux distributions (e.g., Ubuntu) are using the GNU Bash Shell.

[1] Unix Stack Exchange Thread: [What is the exact difference between a 'terminal', a 'shell', a 'tty' and a 'console'?](#)

Unix/Linux Programs and Utilities

To run a program, you generally type the name of the program, followed by (optional) -arguments.

```
$ cut --help
```

```
Usage: cut OPTION... [FILE]...
```

```
Print selected parts of lines from each FILE to standard output.
```

```
With no FILE, or when FILE is -, read standard input.
```

```
Mandatory arguments to long options are mandatory for short options too.
```

```
-b, --bytes=LIST      select only these bytes
-c, --characters=LIST  select only these characters
-d, --delimiter=DELIM use DELIM instead of TAB for field delimiter
-f, --fields=LIST      select only these fields; also print any line
                        that contains no delimiter character, unless
                        the -s option is specified
```

Getting Help and Documentation in the Shell

A few different ways, first is manual pages:

```
# manual page for the manual pages
```

```
$ man man
```

```
# manual for bash shell
```

```
$ man bash
```

```
# manual for a specific program
```

```
$ man cut
```

```
# search the manual pages for the word
```

```
`zip`
```

```
$ man -k `zip`
```

```
# Browse manual pages (e.g., section 1) [1]
```

```
$ man -k . -s1
```

[1]

<https://superuser.com/questions/207450/list-of-all-available-man-pages>

Getting Help and Documentation in the Shell

Alternative ways are with help and info:

```
$ cut --help  
$ join --help
```

```
$ info cut  
$ info join
```

Finally, you can also browse docs online:

GNU Bash: <https://www.gnu.org/software/bash/manual/>

GNU CoreUtils: <https://www.gnu.org/software/coreutils/manual/coreutils.html>

Installing Programs in Debian Based Linux

Whenever possible, install from the package repository using apt or apt-get (may need sudo). You can search for packages here: https://www.debian.org/distrib/packages#search_packages.

```
# first, update your package lists
```

```
$ sudo apt-get update
```

```
# then install program
```

```
# (e.g., firefox-esr) on debian
```

```
$ sudo apt-get install firefox-esr
```

```
# update all packages
```

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

```
# If program is not in package repository, there is usually a .deb file provided on the software website (e.g., VS Code). If you trust the source, you can typically install as follows:
```

```
$ sudo apt install ./file.deb
```


Navigating Directories

print working directories

\$ pwd

list contents

\$ ls

Change directories

\$ cd Documents

Go up a level in directories

\$ cd ..

Making Directories and Creating Files

```
# First, CD into your Documents  
Folder
```

```
$ cd Documents  
$ pwd
```

```
# Then, make a couple folders with  
mkdir:
```

```
$ mkdir fold1  
$ mkdir fold2
```

```
# Check contents and then cd into  
fold1:
```

```
$ ls  
$ cd fold1
```

```
# Let's create a basic text file  
with nano (a text editor):
```

```
$ nano file1  
$ ls  
$ cat file1
```

```
1      John      32  
2      Beth      19  
3      Molly     23  
4      Cade      21
```

```
# move file1 to fold2 directory
```

```
$ mv file1 ~/Documents/fold2
```

Unix Programs and Utilities

There are many many useful core utilities, see

<https://www.gnu.org/software/coreutils/manual/coreutils.html>

Here is one example using cut:

```
$ cat file1
```

```
1      John      32
2      Beth      19
3      Molly     23
4      Cade      21
```

```
$ cut -f3 file1
```

```
32
19
23
21
```

Unix/Linux Shell Pipelines, Redirect, and Loops [1]

With the shell, we can use pipelines to create sequences of commands. Each command output is piped into the next command:

```
$ command1 | command2 | command3
```

We can redirect our output from a command or sequence of commands to a file:

```
$ command1 > myfile1.txt
```

```
$ command1 | command2 | command3 > myfile3.txt
```

Unix shell is also a programming language, and, for example, we can create loops to repeat tasks:

```
$ for item in list_of_items  
> do  
>     something_using $item  
> done
```

[1] See the [Software Carpentry Unix Shell](#) and [Bash Reference Manual](#)

Programs and Utilities

Here is an example:

```
$ cat file1
```

```
1  John  32  
2  Beth  19  
3  Molly 23  
4  Cade  21
```

sort the file by column 2, cut out columns 2 and 3, then display the first 2

```
$ sort -k2,2 file1 | cut -f2,3 | head -n2
```

End

That's it for this brief introduction, enjoy experimenting with the Raspberry Pis!