

# Introduction to UNIX command line

Serghei Mangul, Ph.D  
Assistant Professor of Clinical  
Pharmacy and Biological  
Sciences, University of  
Southern California



# What we will cover

- Unix - Learning the essentials
- Unix fundamentals , syntax and usage
- Navigate within the UNIX file system
- Identify the uses of the several UNIX commands

# Learning Objectives

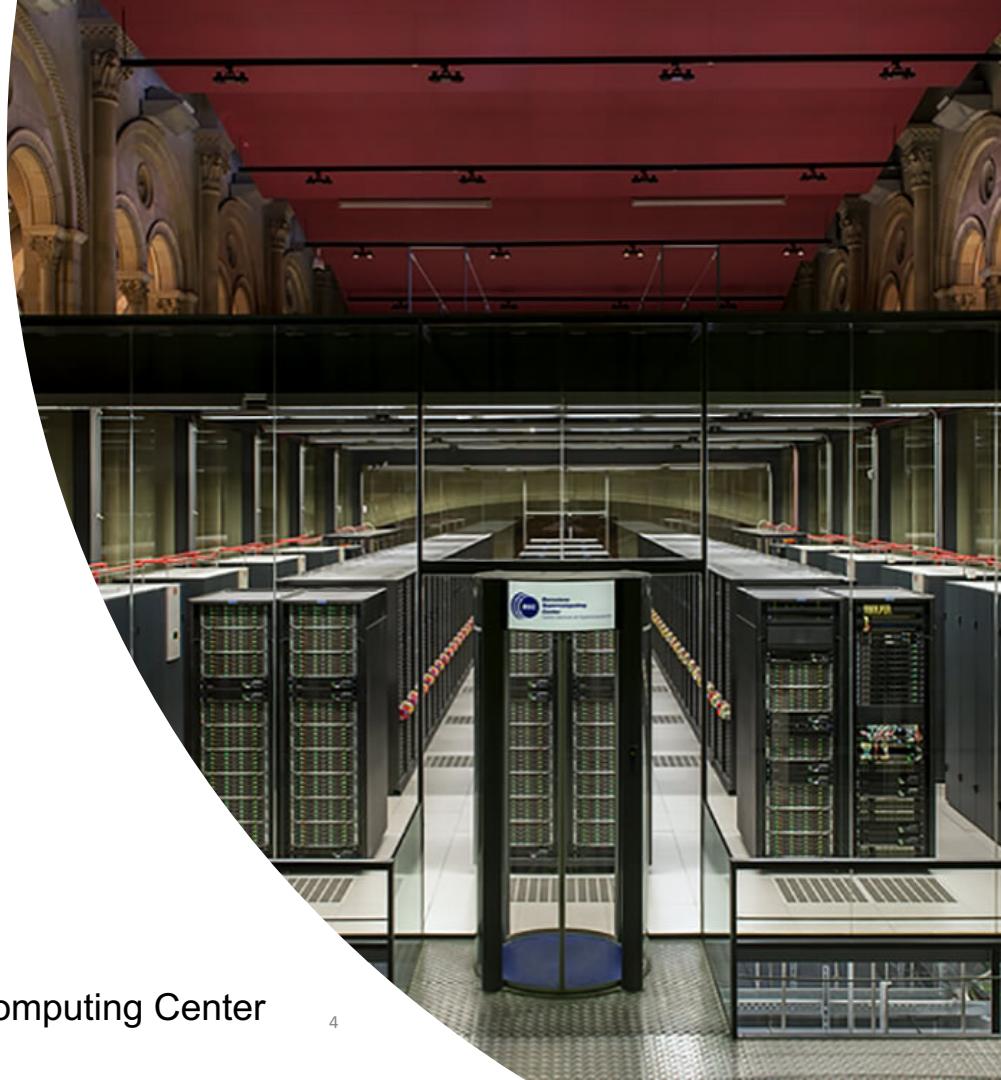
**By the end of this module, you will be able to:**

- Perform the following tasks within the UNIX environment:
  - Create a text file
  - List and reorder files
  - Display, copy, move, and delete directories and files

# Why Learn Unix?

---

- As biological data sets have grown **larger** and biological problems have become more complex, the requirements for computing power have also grown.
- Computers that can provide this power generally use the Unix operating system



# Why Learn Unix?

Most new  
bioinformatics  
software is created  
for Unix

Unix is very  
efficient in  
managing huge  
amounts of data

It is very popular,  
so it is easy to find  
information and get  
help

Unix is very stable -  
computers running  
Unix almost never  
crash

# What is Unix?

---

- UNIX-like system (e.g Linux, Ubuntu) is a type of operating system that lack a **graphical interface** and require inputting code.
- UNIX environment is often referred as command line interface opposed to graphical interface, which is common for Windows and MacOS.



# The Unix Shell

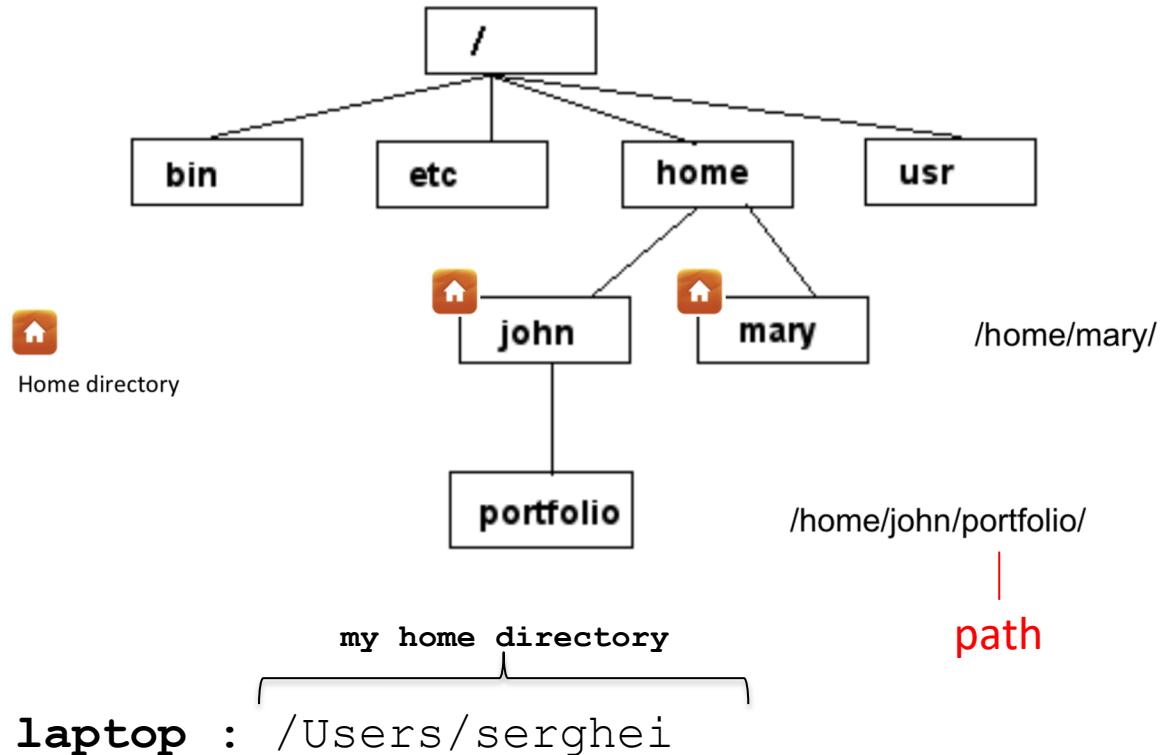
---



- A shell is a program that waits for you to type a command and then executes it.
  - type the command, then “return”

# Unix File System

Unix is cAsE sEnsItiVe !



# Project directory

- Create sub-directories in your Drive to store specific projects or groups of information



Tips

Do **not** accumulate thousands of files with cryptic names in your Home directory

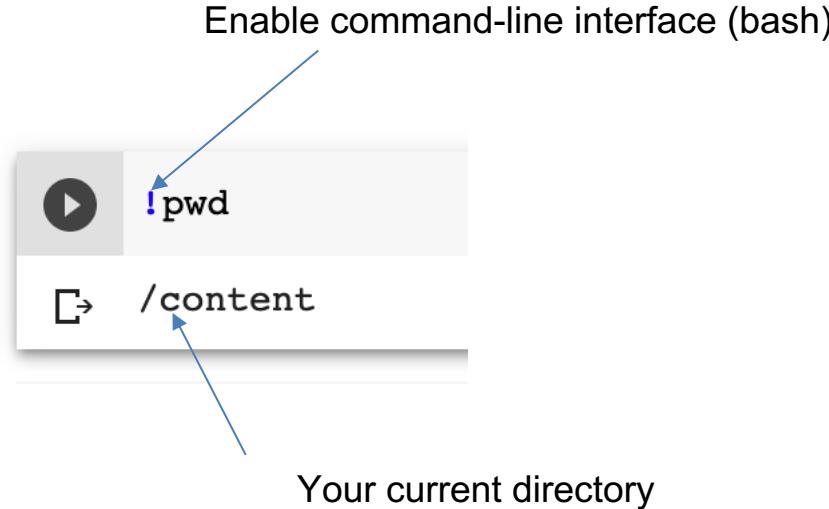
# How to run UNIX in Google Collab

- We need to mount Drive first
- We need to add !

```
[6] from google.colab import drive  
drive.mount('/content/drive')  
  
↳ Go to this URL in a browser: https://accounts.google.com  
  
Enter your authorization code:  
.....  
Mounted at /content/drive
```

# Command: pwd

- To display current directory

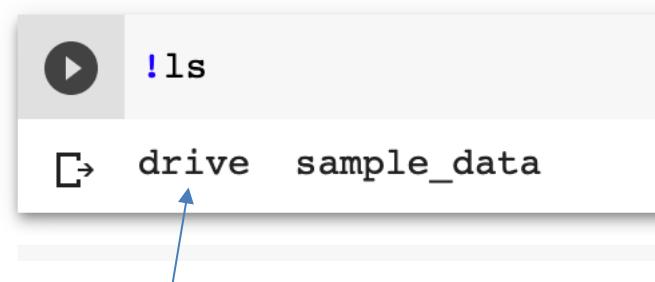


# Command : ls

list the files in the current directory

- **ls** has many options
  - -l long list (displays lots of info)
  - -t sort by modification time
  - -S sort by size
  - -h list file sizes in human readable format
  - -r reverse the order
  - Options can be combined: “ls -lh”

# Let's practice



```
!ls
└── drive  sample_data
```

A screenshot of a terminal window. The window has a light gray header bar with a play button icon and the text '!ls'. Below the header is a dark gray input field containing the command '!ls'. The main body of the terminal shows the output of the command: a single line with two entries: 'drive' and 'sample\_data'. A blue arrow points from the text 'This becomes available after Drive is mounted' down to the 'drive' entry in the terminal output.

This becomes available after Drive is mounted

# Command: cd

- cd changes your current working directory

The diagram illustrates the effect of the `cd` command. On the left, a terminal window shows the command `%cd drive` and the resulting output `C:\ /content/drive`. A blue arrow points to the right, where another terminal window shows the command `%cd My\ Drive/` and the resulting output `C:\ /content/drive/My Drive`. This visualizes how the command changes the current working directory.

This will make the location permanent when you move blocks

The diagram illustrates the permanence of a directory change. It shows three terminal windows. The first window shows the command `%cd data_science_workshop` and the output `C:\ /content/drive/My Drive/data_science_workshop`. A blue arrow points down to the second window, which shows the same command and output. A larger blue arrow points down to the third window, which shows the command `%cd data_science_workshop` and the output `C:\ /content/drive/My Drive/data_science_workshop`. This demonstrates that the initial directory change made it permanent, even after moving blocks.

## If you know the full path



```
%cd /content/drive/My Drive/data_science_workshop
```

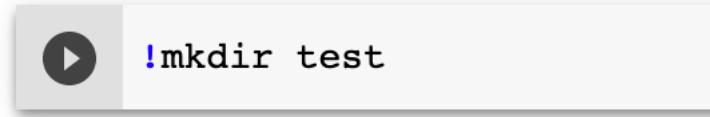


```
↳ /content/drive/My Drive/data_science_workshop
```

# Command: mkdir

- To create a new directory use “mkdir”

`mkdir test`



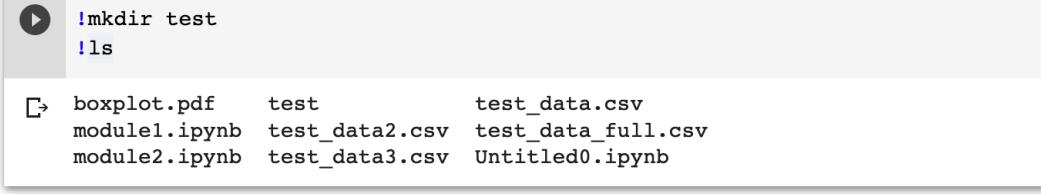
Tips

If no error message is displayed  
means the command was run  
successfully

Use ls  
command to  
check if  
directory  
was created

```
!mkdir test
!ls

boxplot.pdf
test test_data.csv
module1.ipynb
test_data2.csv
test_data_full.csv
module2.ipynb
test_data3.csv
Untitled0.ipynb
```



A terminal window showing the execution of the 'mkdir' and 'ls' commands. The terminal prompt is visible at the top left. The user enters 'mkdir test' followed by 'ls'. The output shows the creation of a 'test' directory and its contents: 'boxplot.pdf', 'test test\_data.csv', 'module1.ipynb', 'test\_data2.csv', 'test\_data\_full.csv', 'module2.ipynb', 'test\_data3.csv', and 'Untitled0.ipynb'.

```
▶ mkdir test
▶ ls

boxplot.pdf      test          test_data.csv
module1.ipynb   test_data2.csv  test_data_full.csv
module2.ipynb   test_data3.csv  Untitled0.ipynb
```

# Command: cd

- “~” is the location of your home directory
- “..” is the location of the directory above the current one

Your project directory



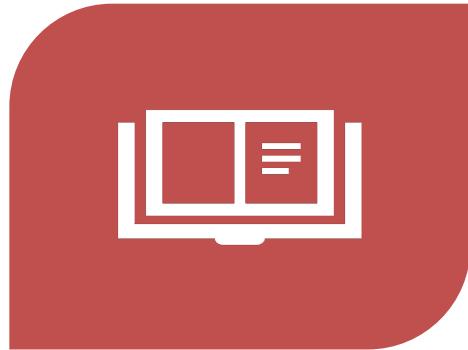
```
/content/drive/My Drive/data_science_workshop
```

# Let's practice

```
%cd /content/drive/My\ Drive/data_science_workshop/test  
%cd ..
```



# How to know more?



MANUAL

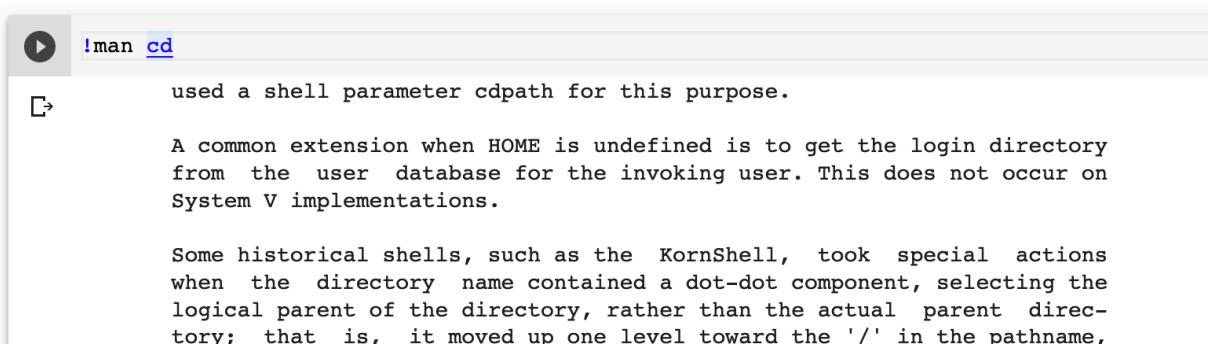


GOOGLE

# Command : man

- displays manual pages

```
!man cd
```



The screenshot shows a terminal window with the command `!man cd` entered in the input field. The output is displayed in three sections:

- A header line: `used a shell parameter cdpath for this purpose.`
- A detailed explanation: `A common extension when HOME is undefined is to get the login directory from the user database for the invoking user. This does not occur on System V implementations.`
- A historical note: `Some historical shells, such as the KornShell, took special actions when the directory name contained a dot-dot component, selecting the logical parent of the directory, rather than the actual parent directory; that is, it moved up one level toward the '/' in the pathname.`

ls sort by date

All News Books Shopping Videos More Settings Tools

About 123,000,000 results (0.48 seconds)

## Ls command Sort by date

**The -t flag will sort the ls command output by last date and time modified:**

1. Open the Terminal if you have not done so already (/Applications/Utilities/ in mac OS) and navigate to the directory you wish to sort by date with ls.
2. Issue the following command syntax:

[More items...](#) • Feb 2, 2017

osxdaily.com › 2017/02/02 › ls-sort-by-date ▾

### How to Sort ls Command by Date | OSXDaily

Search for: Ls command Sort by date

About Featured Snippets Feedback

superuser.com › questions › how-can-i-sort-the-output-... ▾

## How can I sort the output of 'ls' by last modified date? - Super ...

Nov 7, 2011 - ls -t. or (for reverse, most recent at bottom): ls -tr. The ls man page describes this in more details, and lists other options.

<a href="#">ls: show date of last updated file in directory tree, for each ...</a>	Mar 21, 2016
<a href="#">Unix/Linux find and sort by date modified - Super User</a>	Nov 6, 2011
<a href="#">Get Last Modified Date of File in Linux - Super User</a>	Sep 21, 2015
<a href="#">How to sort the output of find? - Super User</a>	Sep 19, 2015
<a href="#">More results from superuser.com</a>	

10 answers

## How can I sort the output of 'ls' by last modified date?

Asked 11 years, 2 months ago Active 3 months ago Viewed 1.2m times

How can I sort the output of ls by last modified date?

1458

linux unix shell ls

259

share improve this question follow

edited Jul 28 '14 at 9:23



Garrett  
247 3 6

asked Apr 9 '09 at 13:17



1 Related (not necessarily a duplicate): [Unix/Linux find and sort by date modified](#) – Peter Mortensen '16 at 14:15

add a comment

10 Answers

Active Oldest

1729

ls -t

or (for reverse, most recent at bottom):

ls -tr

# General Syntax: \*

- “\*” can be used as a wildcard in Unix

```
ls *csv
```

```
ls t*
```

```
ls *txt
```

```
ls: cannot access '*txt': No such file  
or directory
```

# Displaying a file

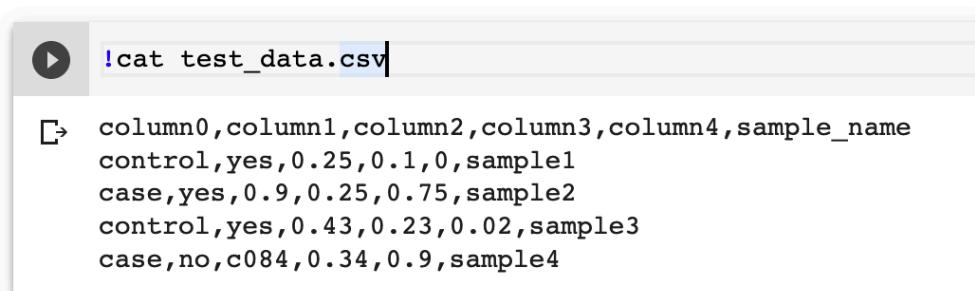
Various ways to display a file in Unix

- cat
- head
- tail

# Command: cat

- dumps an entire file to standard output
- good for displaying short, simple files

```
!cat test_data.csv
```



A screenshot of a terminal window. The command `!cat test_data.csv` is entered in the input field. The output shows five rows of CSV data:

	column0	column1	column2	column3	column4	sample_name
1	control	yes	0.25	0.1	0	sample1
2	case	yes	0.9	0.25	0.75	sample2
3	control	yes	0.43	0.23	0.02	sample3
4	case	no	c084	0.34	0.9	sample4

# Command: head

- displays the top part of a file
  - by default it shows the first 10 lines
    - **-n** option allows you to change that

# Command: tail

- Same as head, but shows the last lines

# Let's practice!

```
!head -n 2 test_data.csv  
!tail -n 1 test_data.csv
```



```
!head -n 2 test_data.csv
```

```
↳ column0,column1,column2,column3,column4,sample_name  
control,yes,0.25,0.1,0,sample1
```



# File Commands

- Copying a file: **cp**



- Move or rename a file: **mv**



- Remove a file: **rm**

# Copy

```
cp <source> <destination>
```

- to copy a file use **cp**
- to copy a directory use **cp -r**

# Let's practice



```
!cp test_data.csv test_data_new.csv
```

```
!cp test_data.csv test/
```

```
!cp -r test test2
```

```
!cp test_data.csv test/test_data_new.csv
```

# Command: mv

```
mv <source> <destination>
```

- moves a file/directory to a different location
- renames a file/directory



# Let's practice

```
!mv test_data_new.csv test2/
```



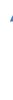
```
%cd test2/  
ls  
/content/drive/My Drive/data_science_workshop/test2  
test_data_new.csv
```

```
!mv test_data.csv test_DATA.csv
```



Rename file

```
!mv test_DATA.csv test/TEST_DATA.csv
```



Move and rename file

# Command: rm

- to remove a file use **rm**
- to remove a directory use **rm -r**

```
!rm test.log  
!rm -r new2  
!ls
```



Files and directories deleted with **rm** are gone forever and cannot be recovered!!!

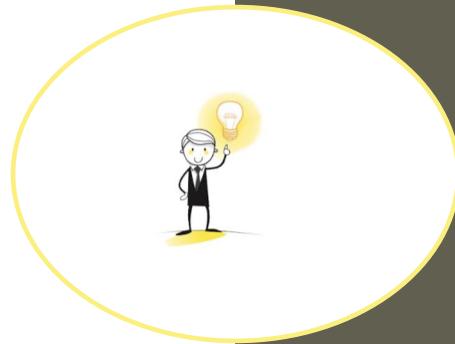
# Good to know

- **cp/mv/rm** can work on many files at once:

```
cp file1 file2 new/  
rm file1 file2 file27
```

- **cp/mv/rm** can work with \*:

```
mv f* new/  
rm f*  
rm l*s  
rm *txt
```



# Summary Commands UNIX

---

Directory commandline	
pwd	display current directory
mkdir	create a new directory
cd	change current working directory
cp-r	copy a directory
File commandline	
ls	list files in the current directory
cat	display the entire file
less	display part of the file
tail	display the end of the file
head	display the head of the file
cp	copy a file
mv	move a file
rm	remove a file

# More about UNIX

9 hours of detailed tutorial

*Intro to Unix Workshop taught by Serghei Mangul at UCLA's Collaboratory; parts 1, 2, and 3:*



<http://www.sergheimangul.com/video/>

## Trends in Biotechnology

Volume 35, Issue 10, October 2017, Pages 901-903



Scientific Life

Addressing the Digital Divide in Contemporary Biology: Lessons from Teaching UNIX

Serghei Mangul <sup>1, 2, 9, 10</sup>, Lana S. Martin <sup>1, 1</sup>, Alexander Hoffmann <sup>3</sup>, Matteo Pellegrini <sup>4</sup>, Eleazar Eskin <sup>1, 5</sup>

<https://www.sciencedirect.com/science/article/abs/pii/S0167779917301567>