



MASSACHUSETTS
GENERAL HOSPITAL



HARVARD
MEDICAL SCHOOL



BROAD
INSTITUTE

GINGER On-Site Training Day 1: Intro to Linux

GINGER Program 2022
University of Cape Town

Teaching Fellows:

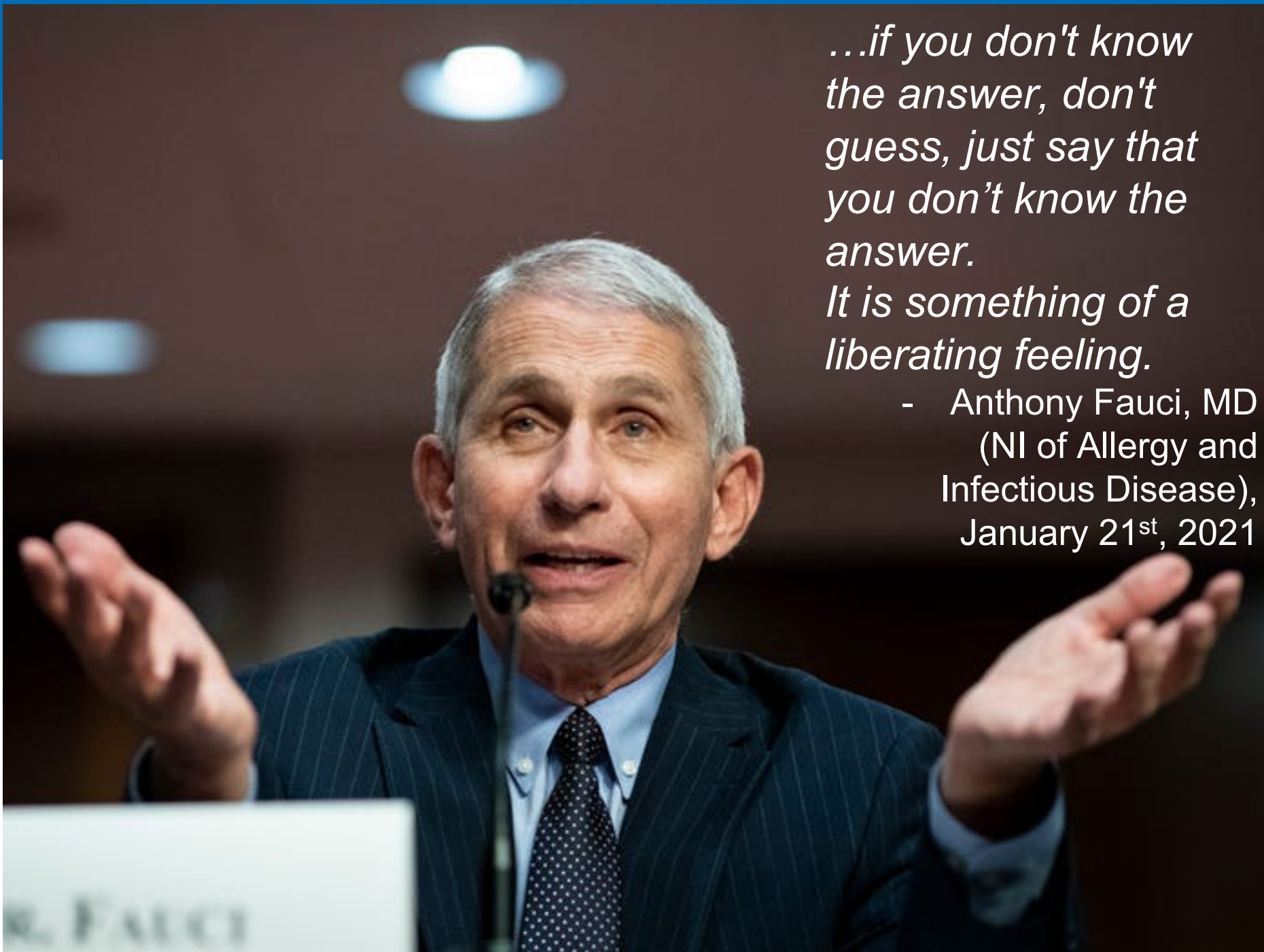
Kumar Veerapen, Ph.D.

Senior Expert I Data Science
Kumar.Veerapen@Novartis.com

Carla Marquez-Luna, Ph.D.

Postdoctoral Research Fellow
carlamarquezluna@gmail.com

	Monday, April 4	Tuesday, April 5	Wednesday, April 6	Thursday, April 7	Friday, April 8
9:00-10:30	Training Welcome and Introduction 9:00-9:30 - Training Overview 9:30-10:00 - Professor Dan Stein Welcome 10:00-11:00 - Begin Kampala Refresher	Plink Tutorial	Excursion to Robben Island (weather dependent)	9:00-10:00 am TBD 10:00 am NeuroGAP Site Visit	Step-by-Step GWAS
10:30-10:45	Tea Break	Tea Break			Tea Break
10:45-1:00	Kampala Refresher continued	11:00-12:00 - Professor Colett Dandara 12:00-1:00 - Intro to Plink			11:00 - Guest Lectures: Drs. Shareefa Dalvie and Nastassja Koen 12:00-1:00 - Step-by-Step GWAS
1:00-2:00	Lunch	Lunch	Lunch at the V&A Waterfront (weather dependent)	Lunch	Lunch
2:00-3:30	Intro to UNIX Fundamental Commands Genetic Data Formats and Conversion	Plink Tutorial	Group Project Work	Step-by-Step GWAS	Step-by-Step GWAS
3:30-3:45	Tea Break	Tea Break		Tea Break	Tea Break
3:45-5:00	GINGER group projects intro	Plink Tutorial Step-by-Step GWAS		Step-by-Step GWAS	4:00-4:30 - Step by Step GWAS 4:30-5:00 - Group Project Presentations



*...if you don't know
the answer, don't
guess, just say that
you don't know the
answer.*

*It is something of a
liberating feeling.*

- Anthony Fauci, MD
(NI of Allergy and
Infectious Disease),
January 21st, 2021

Outline

- What is Linux?
- Why Linux?
- Exercises

Word of Advice



How do you code?



- There are many ways to code.
- Beginners, don't worry about how short your code is. As long as IT WORKS!
- Later, get someone who is more experienced to do code review with you.



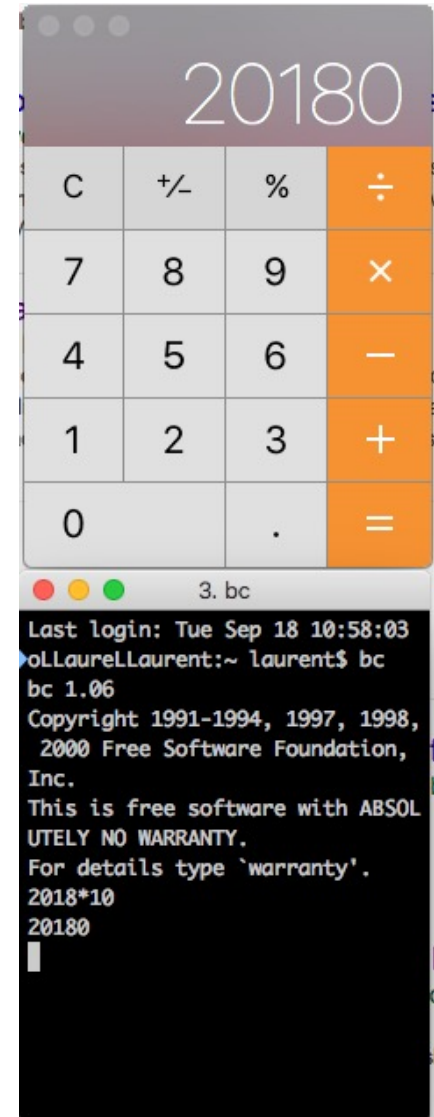
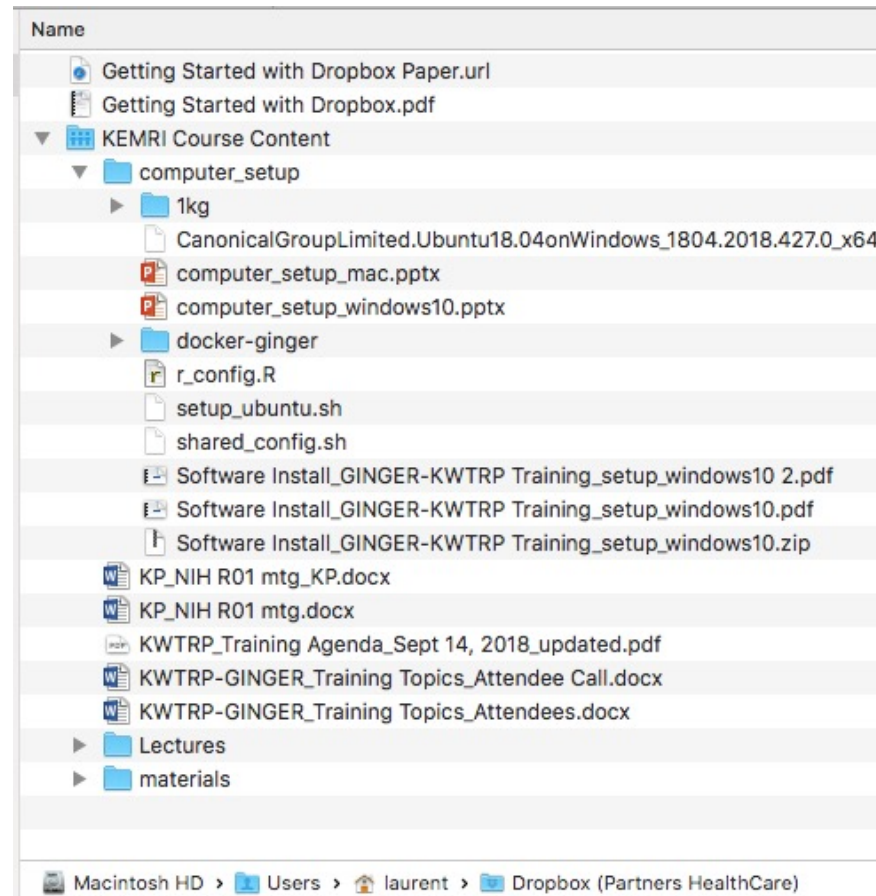
Computing basics

Interface

Programs



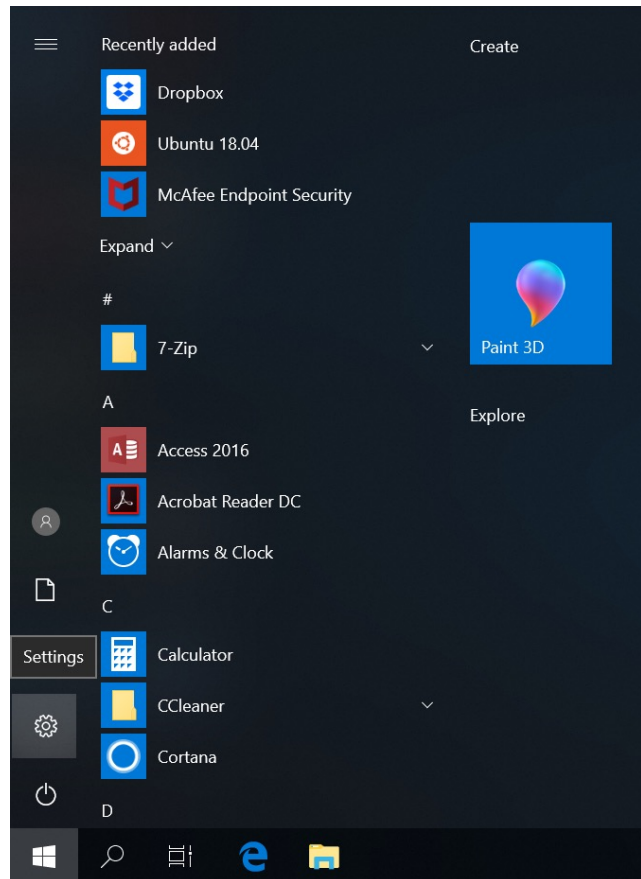
Data



Launching a Unix Shell

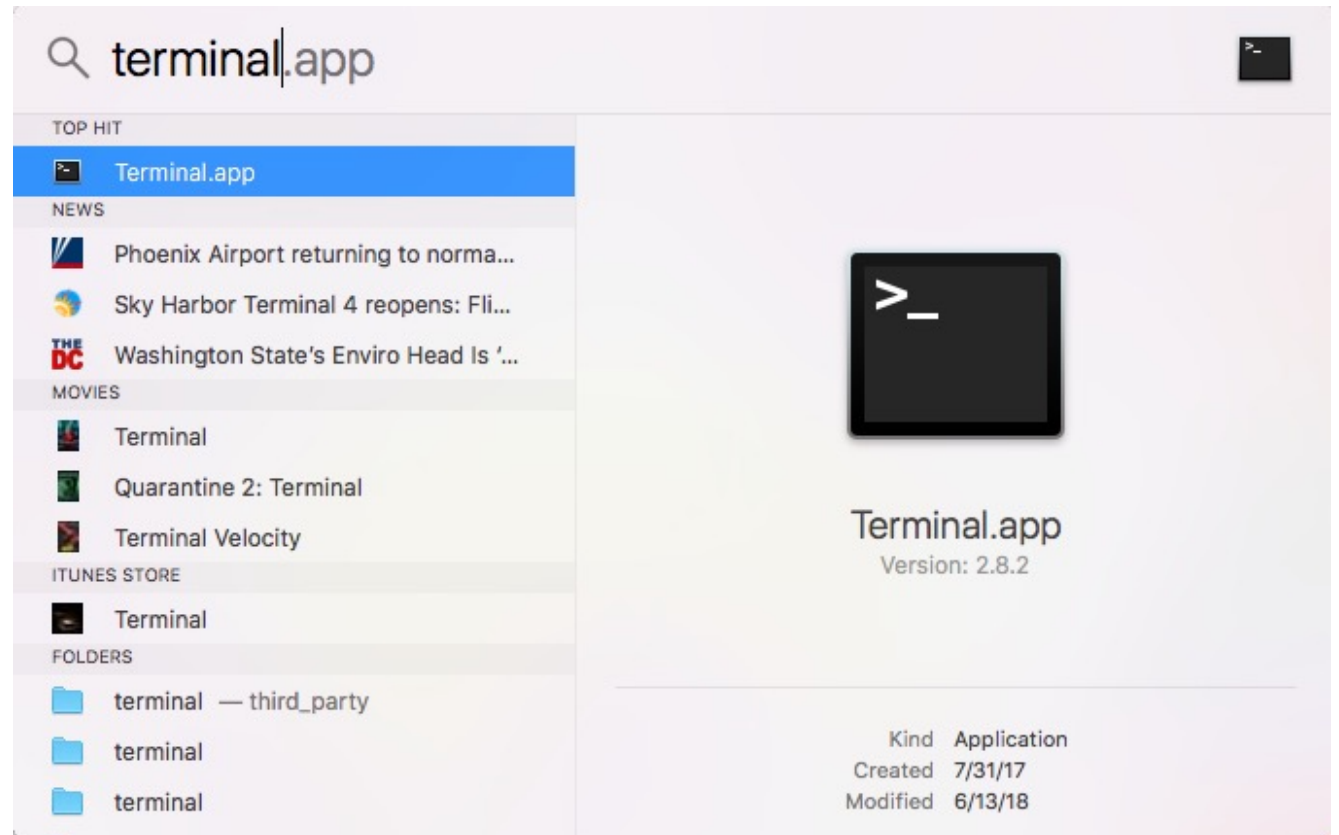
Windows 10

- Open the “Ubuntu” App

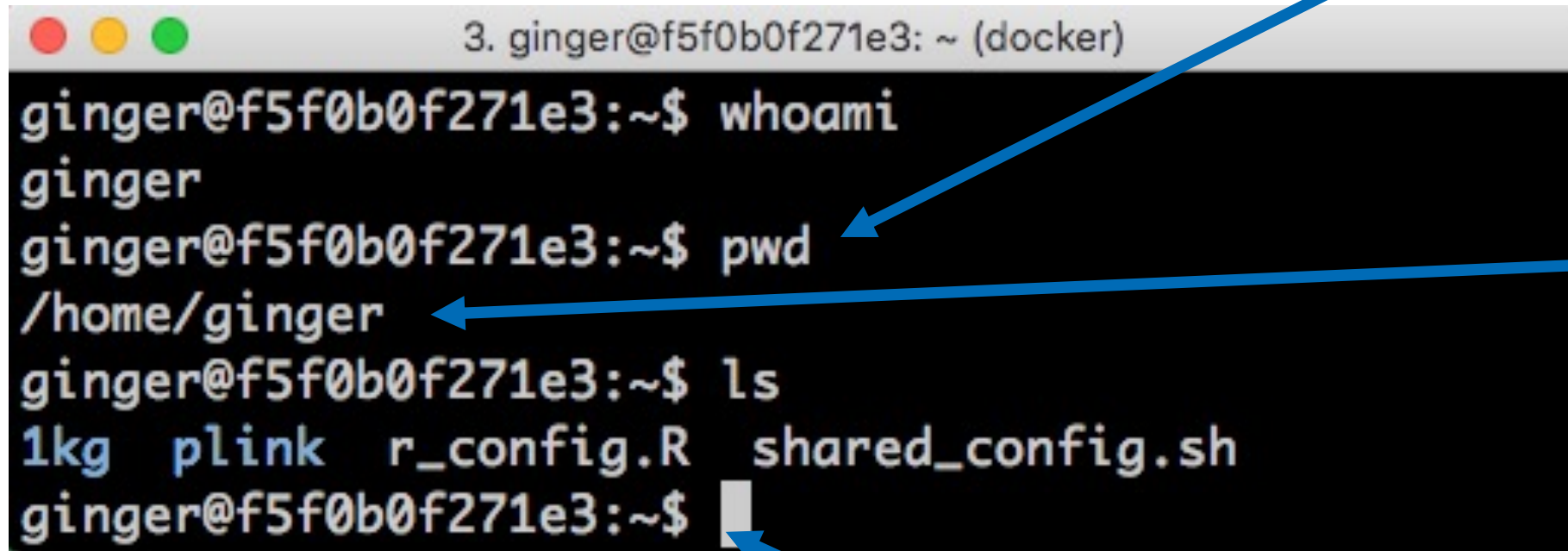


Mac OS

- Press $\text{⌘} + \text{Space}$ and then type “terminal”



What does a Shell look like?



```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ whoami
ginger
ginger@f5f0b0f271e3:~$ pwd
/home/ginger
ginger@f5f0b0f271e3:~$ ls
1kg  plink  r_config.R  shared_config.sh
ginger@f5f0b0f271e3:~$
```

A command previously executed
“pwd”

Result of
“pwd”
command

Prompt – waiting for the next command

How do you know what to type in the Shell?

Unix/Linux Command Reference

FOSSwire.com

File Commands	System Info
ls - directory listing	date - show the current date and time
ls -al - formatted listing with hidden files	cal - show this month's calendar
cd <i>dir</i> - change directory to <i>dir</i>	uptime - show current uptime
cd - change to home	w - display who is online
pwd - show current directory	whoami - who you are logged in as
mkdir <i>dir</i> - create a directory <i>dir</i>	finger <i>user</i> - display information about <i>user</i>
rm <i>file</i> - delete <i>file</i>	uname -a - show kernel information
rm -r <i>dir</i> - delete directory <i>dir</i>	cat /proc/cpuinfo - cpu information
rm -f <i>file</i> - force remove <i>file</i>	cat /proc/meminfo - memory information
rm -rf <i>dir</i> - force remove directory <i>dir</i> *	man <i>command</i> - show the manual for <i>command</i>
cp <i>file1 file2</i> - copy <i>file1</i> to <i>file2</i>	df - show disk usage
cp -r <i>dir1 dir2</i> - copy <i>dir1</i> to <i>dir2</i> ; create <i>dir2</i> if it doesn't exist	du - show directory space usage
mv <i>file1 file2</i> - rename or move <i>file1</i> to <i>file2</i> if <i>file2</i> is an existing directory, moves <i>file1</i> into directory <i>file2</i>	free - show memory and swap usage
ln -s <i>file link</i> - create symbolic link <i>link</i> to <i>file</i>	whereis <i>app</i> - show possible locations of <i>app</i>
touch <i>file</i> - create or update <i>file</i>	which <i>app</i> - show which <i>app</i> will be run by default
cat > <i>file</i> - places standard input into <i>file</i>	
more <i>file</i> - output the contents of <i>file</i>	Compression
head <i>file</i> - output the first 10 lines of <i>file</i>	tar cf <i>file.tar files</i> - create a tar named <i>file.tar</i> containing <i>files</i>
tail <i>file</i> - output the last 10 lines of <i>file</i>	tar xf <i>file.tar</i> - extract the files from <i>file.tar</i>
tail -f <i>file</i> - output the contents of <i>file</i> as it grows, starting with the last 10 lines	tar czf <i>file.tar.gz files</i> - create a tar with Gzip compression
	tar xzf <i>file.tar.gz</i> - extract a tar using Gzip
	tar cjf <i>file.tar.bz2</i> - create a tar with Bzip2 compression

Anatomy of a Shell command

“ls” command

```
3. ginger@f5f0b0f271e3: ~ (docker)
-rw-r--r-- 1 ginger ginger 58 Jul 5 2016 toy.ped
ginger@f5f0b0f271e3:~$ ls
1kg plink r_config.R shared_config.sh
ginger@f5f0b0f271e3:~$ ls -l
total 16
drwxr-xr-x 2 ginger ginger 4096 Sep 18 02:39 1kg
drwxr-xr-x 2 ginger ginger 4096 Sep 18 02:39 plink
-rw-r--r-- 1 ginger ginger 30 Sep 18 02:16 r_config.R
-rw-r--r-- 1 ginger ginger 1032 Sep 18 02:16 shared_config.sh
ginger@f5f0b0f271e3:~$ ls -l plink
total 6216
-rw-r--r-- 1 ginger ginger 35147 Jul 5 2016 LICENSE
-rwxr-xr-x 1 ginger ginger 6297321 Sep 13 23:26 plink
-rwxr-xr-x 1 ginger ginger 16755 Sep 13 23:26 prettify
-rw-r--r-- 1 ginger ginger 27 Jul 5 2016 toy.map
-rw-r--r-- 1 ginger ginger 58 Jul 5 2016 toy.ped
ginger@f5f0b0f271e3:~$
```

Result:

List files and directories in current directory

Anatomy of a Shell command

“ls” command

```
3. ginger@f5f0b0f271e3: ~ (docker)
-rw-r--r-- 1 ginger ginger 58 Jul 5 2016 toy.ped
ginger@f5f0b0f271e3:~$ ls
1kg plink r_config.R shared_config.sh
ginger@f5f0b0f271e3:~$ ls -l
total 16
drwxr-xr-x 2 ginger ginger 4096 Sep 18 02:39 1kg
drwxr-xr-x 2 ginger ginger 4096 Sep 18 02:39 plink
-rw-r--r-- 1 ginger ginger 30 Sep 18 02:16 r_config.R
-rw-r--r-- 1 ginger ginger 1032 Sep 18 02:16 shared_config.sh
ginger@f5f0b0f271e3:~$ ls -l plink
total 6216
-rw-r--r-- 1 ginger ginger 35147 Jul 5 2016 LICENSE
-rwxr-xr-x 1 ginger ginger 6297321 Sep 13 23:26 plink
-rwxr-xr-x 1 ginger ginger 16755 Sep 13 23:26 prettify
-rw-r--r-- 1 ginger ginger 27 Jul 5 2016 toy.map
-rw-r--r-- 1 ginger ginger 58 Jul 5 2016 toy.ped
ginger@f5f0b0f271e3:~$
```

“-l” flag

Result:
List files and directories in current directory. Output is a list with details about each file/directory.

Anatomy of a Shell command

“ls” command

```
3. ginger@f5f0b0f271e3: ~ (docker)
-rw-r--r-- 1 ginger ginger 58 Jul 5 2016 toy.ped
ginger@f5f0b0f271e3:~$ ls
1kg plink r_config.R shared_config.sh
ginger@f5f0b0f271e3:~$ ls -l
total 16
drwxr-xr-x 2 ginger ginger 4096 Sep 18 02:39 1kg
drwxr-xr-x 2 ginger ginger 4096 Sep 18 02:39 plink
-rw-r--r-- 1 ginger ginger 30 Sep 18 02:16 r_config.R
-rw-r--r-- 1 ginger ginger 1032 Sep 18 02:16 shared_config.sh
ginger@f5f0b0f271e3:~$ ls -l plink
total 6216
-rw-r--r-- 1 ginger ginger 35147 Jul 5 2016 LICENSE
-rwxr-xr-x 1 ginger ginger 6297321 Sep 13 23:26 plink
-rwxr-xr-x 1 ginger ginger 16755 Sep 13 23:26 prettify
-rw-r--r-- 1 ginger ginger 27 Jul 5 2016 toy.map
-rw-r--r-- 1 ginger ginger 58 Jul 5 2016 toy.ped
ginger@f5f0b0f271e3:~$
```

“-l” flag

Pass “plink” as argument to “ls”

Result:
Same as “ls -l”, but now shows the content of the “plink” directory.

Getting help

```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                do not ignore entries starting with .
  -A, --almost-all        do not list implied . and ..
      --author              with -l, print the author of each file
  -b, --escape             print C-style escapes for nongraphic characters
      --block-size=SIZE    scale sizes by SIZE before printing them; e.g.,
                          '--block-size=M' prints sizes in units of
                          1,048,576 bytes; see SIZE format below
  -B, --ignore-backups     do not list implied entries ending with ~
  -c                       with -lt: sort by, and show, ctime (time of last
                          modification of file status information);
                          with -l: show ctime and sort by name;
                          otherwise: sort by ctime, newest first
  -C                       list entries by columns
      --color[=WHEN]       colorize the output; WHEN can be 'always' (default
                          if omitted), 'auto', or 'never'; more info below
```

Interacting with the file system

Moving around the filesystem

3. ginger@f5f0b0f271e3: ~ (docker)

```
ginger@f5f0b0f271e3:~$  
ginger@f5f0b0f271e3:~$ pwd  
/home/ginger  
ginger@f5f0b0f271e3:~$ ls  
1kg  plink  r_config.R  shared_config.sh  
ginger@f5f0b0f271e3:~$ cd plink  
ginger@f5f0b0f271e3:~/plink$ pwd  
/home/ginger/plink  
ginger@f5f0b0f271e3:~/plink$ cd ..  
ginger@f5f0b0f271e3:~$ pwd  
/home/ginger  
ginger@f5f0b0f271e3:~$ cd /home  
ginger@f5f0b0f271e3:/home$ ls  
ginger  rstudio  
ginger@f5f0b0f271e3:/home$ cd ~  
ginger@f5f0b0f271e3:~$ pwd  
/home/ginger  
ginger@f5f0b0f271e3:~$ cd .  
ginger@f5f0b0f271e3:~$ pwd  
/home/ginger  
ginger@f5f0b0f271e3:~$
```

Name	
▼	home
▼	ginger
▶	1kg
▼	plink
	LICENSE
	plink
	plink_mac.zip
	prettify
	toy.map
	toy.ped
	r_config.R
	shared_config.sh
▶	rstudio

More about the filesystem and ls -l

- = file
d = directory

```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ ls -l
total 20
-rw-r--r-- 1 ginger ginger 21 Sep 18 17:44 123.txt
drwxr-xr-x 2 ginger ginger 4096 Sep 18 02:39 1kg
drwxr-xr-x 2 ginger ginger 4096 Sep 18 02:39 plink
-rw-r--r-- 1 ginger ginger 30 Sep 18 02:16 r_config.R
-rw-r--r-- 1 ginger ginger 1032 Sep 18 02:16 shared_config.sh
ginger@f5f0b0f271e3:~$
```

Permissions:
rwx r-x r-x

owner group size modification date

Sudo: becoming root (admin) for a command

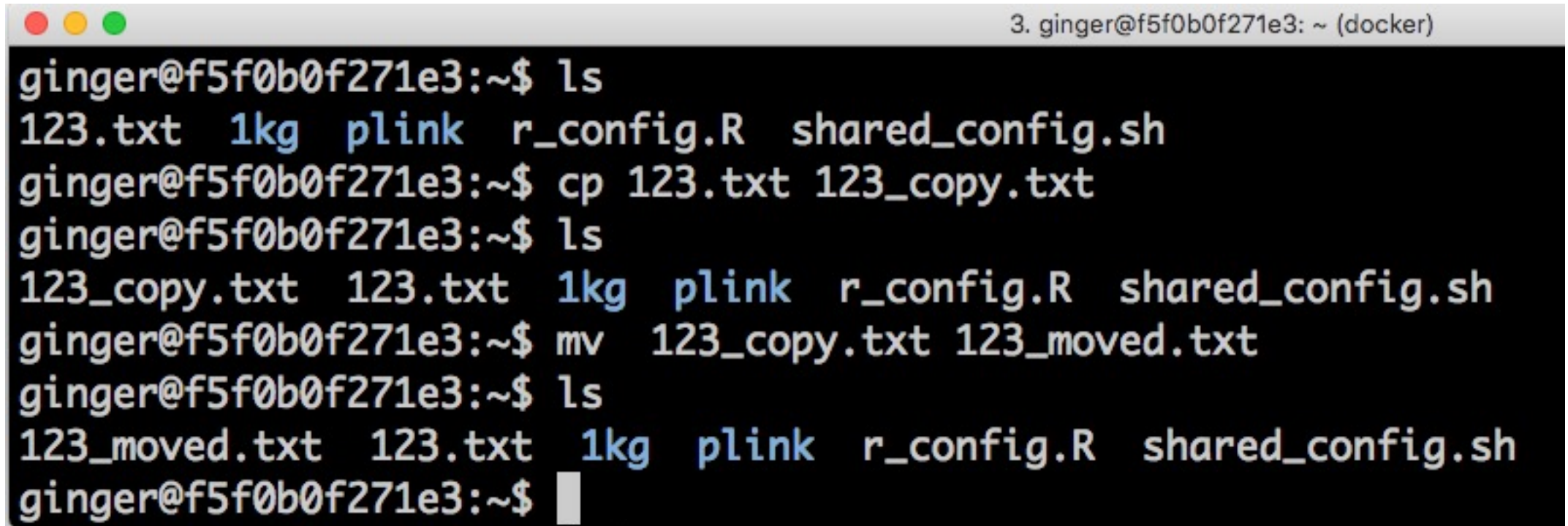
```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ ls -l
total 28
-rw-r--r-- 1 ginger ginger 21 Sep 18 17:44 123.txt
drwxr-xr-x 2 ginger ginger 4096 Sep 18 02:39 1kg
-rw----- 1 root root 38 Sep 18 18:32 admin_file.txt
drwxr-xr-x 2 ginger ginger 4096 Sep 18 02:39 plink
-rw-r--r-- 1 ginger ginger 30 Sep 18 02:16 r_config.R
-rw-r--r-- 1 ginger ginger 1032 Sep 18 02:16 shared_config.sh
-rw-r--r-- 1 ginger ginger 9 Sep 18 18:25 test.txt
ginger@f5f0b0f271e3:~$ cat admin_file.txt
cat: admin_file.txt: Permission denied
ginger@f5f0b0f271e3:~$ sudo cat admin_file.txt
This is the content of admin_file.txt
ginger@f5f0b0f271e3:~$
```


Creating / deleting files and folders

```
3. ginger@f5f0b0f271e3: ~ (docker)

ginger@f5f0b0f271e3:~$
ginger@f5f0b0f271e3:~$ pwd
/home/ginger
ginger@f5f0b0f271e3:~$ ls
1kg  plink  r_config.R  shared_config.sh
ginger@f5f0b0f271e3:~$ touch test.txt
ginger@f5f0b0f271e3:~$ ls
1kg  plink  r_config.R  shared_config.sh  test.txt
ginger@f5f0b0f271e3:~$ rm test.txt
ginger@f5f0b0f271e3:~$ ls
1kg  plink  r_config.R  shared_config.sh
ginger@f5f0b0f271e3:~$ mkdir test
ginger@f5f0b0f271e3:~$ ls
1kg  plink  r_config.R  shared_config.sh  test
ginger@f5f0b0f271e3:~$ rm test
rm: cannot remove 'test': Is a directory
ginger@f5f0b0f271e3:~$ rm -r test
ginger@f5f0b0f271e3:~$
```

Copying and moving files

A terminal window with a title bar containing three colored circles (red, yellow, green) on the left and the text "3. ginger@f5f0b0f271e3: ~ (docker)" on the right. The terminal has a black background with white text. It shows a sequence of commands and their outputs: 1. Command: `ls`. Output: `123.txt 1kg plink r_config.R shared_config.sh`. 2. Command: `cp 123.txt 123_copy.txt`. 3. Command: `ls`. Output: `123_copy.txt 123.txt 1kg plink r_config.R shared_config.sh`. 4. Command: `mv 123_copy.txt 123_moved.txt`. 5. Command: `ls`. Output: `123_moved.txt 123.txt 1kg plink r_config.R shared_config.sh`. The prompt `ginger@f5f0b0f271e3:~$` is visible at the start of each line.

```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ ls
123.txt 1kg plink r_config.R shared_config.sh
ginger@f5f0b0f271e3:~$ cp 123.txt 123_copy.txt
ginger@f5f0b0f271e3:~$ ls
123_copy.txt 123.txt 1kg plink r_config.R shared_config.sh
ginger@f5f0b0f271e3:~$ mv 123_copy.txt 123_moved.txt
ginger@f5f0b0f271e3:~$ ls
123_moved.txt 123.txt 1kg plink r_config.R shared_config.sh
ginger@f5f0b0f271e3:~$
```

Wildcards (*)

```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ ls
123_moved.txt  123.txt  1kg  plink  r_config.R  shared_config.sh
ginger@f5f0b0f271e3:~$ ls *.txt
123_moved.txt  123.txt
ginger@f5f0b0f271e3:~$ rm *.txt
ginger@f5f0b0f271e3:~$ ls
1kg  plink  r_config.R  shared_config.sh
ginger@f5f0b0f271e3:~$ ls *r*
r_config.R  shared_config.sh
ginger@f5f0b0f271e3:~$
```

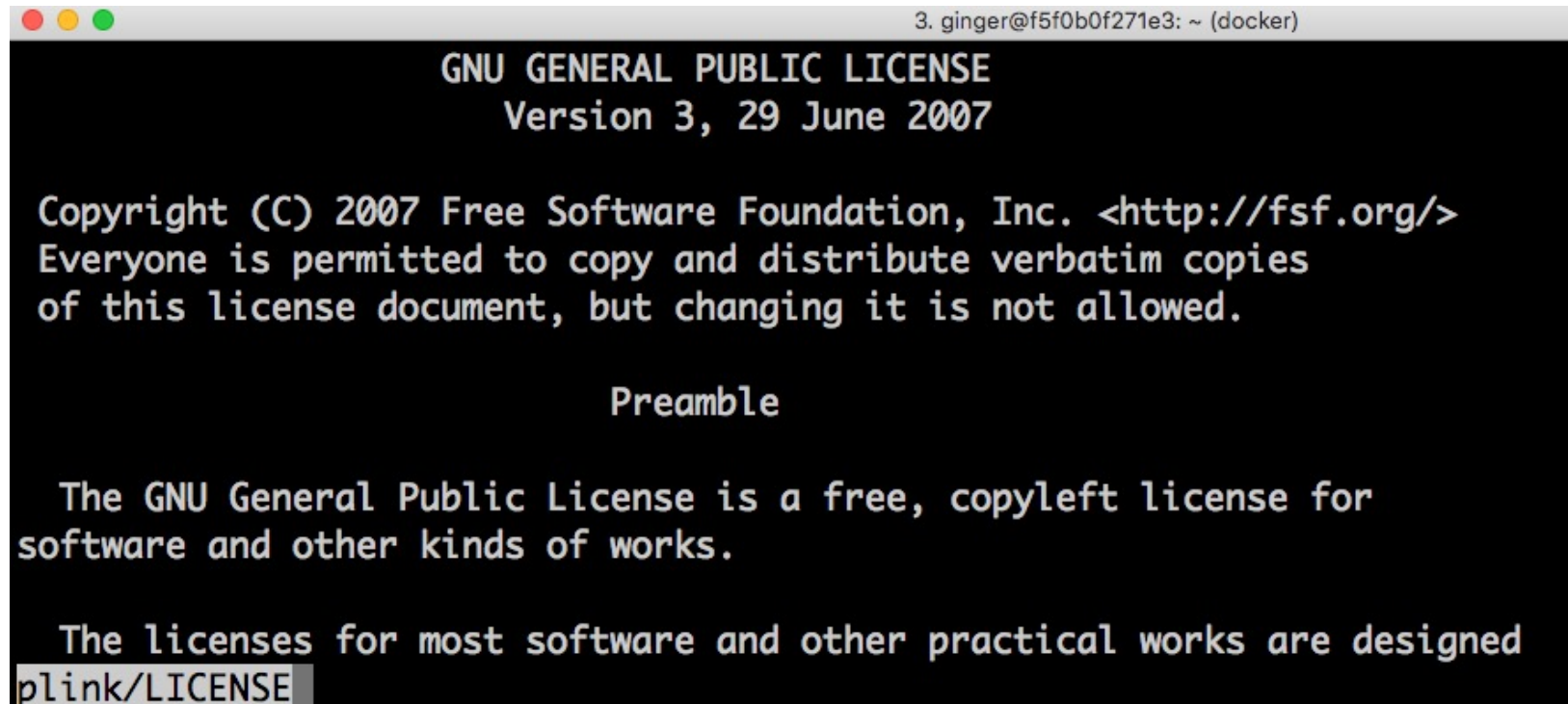
Interacting with text files

Viewing the content of a text file

```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ ls
123.txt 1kg plink r_config.R shared_config.sh
ginger@f5f0b0f271e3:~$ cat 123.txt
1
2
3
4
5
6
7
8
9
10
ginger@f5f0b0f271e3:~$ head -3 123.txt
1
2
3
ginger@f5f0b0f271e3:~$ tail -3 123.txt
8
9
10
ginger@f5f0b0f271e3:~$
```


Reading longer text files: less

```
ginger@f5f0b0f271e3:~$ ls plink/  
LICENSE plink prettify toy.map toy.ped  
ginger@f5f0b0f271e3:~$ less plink/LICENSE
```



The screenshot shows a terminal window with a title bar that reads "3. ginger@f5f0b0f271e3: ~ (docker)". The terminal content displays the GNU General Public License, Version 3, dated 29 June 2007. It includes the copyright notice for the Free Software Foundation, Inc., and the preamble explaining that the license is free and copyleft. The text is displayed in a monospaced font, and the cursor is visible at the end of the last line shown.

```
GNU GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007  
  
Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.  
  
Preamble  
  
The GNU General Public License is a free, copyleft license for  
software and other kinds of works.  
  
The licenses for most software and other practical works are designed  
plink/LICENSE
```

Commands:

- q: quit
- f: forward 1 page
- b back 1 page
- ↑↓→←: move by 1 character in the direction of the arrow
- /text: search for "text"
- S <enter>: chop long lines
- 100g: Go to line 100

Sorting text files: sort

```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ cat not_sorted.txt
a      20
b       8
a       3
ginger@f5f0b0f271e3:~$ sort not_sorted.txt
a      20
a       3
b       8
ginger@f5f0b0f271e3:~$ sort -k2 not_sorted.txt
a      20
a       3
b       8
ginger@f5f0b0f271e3:~$ sort -k2n not_sorted.txt
a       3
b       8
a      20
ginger@f5f0b0f271e3:~$ sort -k1 -k2n not_sorted.txt
a      20
a       3
b       8
ginger@f5f0b0f271e3:~$
```

More text utilities: wc, uniq and diff

```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ cat test.txt
one
one
two
one
three
ginger@f5f0b0f271e3:~$ wc test.txt
 5  5 22 test.txt
ginger@f5f0b0f271e3:~$ wc -l test.txt
5 test.txt
ginger@f5f0b0f271e3:~$ uniq test.txt
one
two
one
three
ginger@f5f0b0f271e3:~$ uniq test.txt > test.uniq.txt
ginger@f5f0b0f271e3:~$ diff test.txt test.uniq.txt
2d1
< one
ginger@f5f0b0f271e3:~$
```

Searching a text file: grep (basic)

```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ cat toy.map
1      rs0      0      1000
1      rs10     0      1001
2      rs10     0      1001
10     rs10     0      1002
ginger@f5f0b0f271e3:~$ grep 1001 toy.map
1      rs10     0      1001
2      rs10     0      1001
ginger@f5f0b0f271e3:~$ grep ^1 toy.map
1      rs0      0      1000
1      rs10     0      1001
10     rs10     0      1002
ginger@f5f0b0f271e3:~$ grep -w ^1 toy.map
1      rs0      0      1000
1      rs10     0      1001
ginger@f5f0b0f271e3:~$ grep -c -w ^1 toy.map
2
ginger@f5f0b0f271e3:~$
```

Looking for a file within a subdirectory: find

- Go into a directory and try:
 - find .
- Maybe you know what the name is:
 - find . -name "myfile.txt"
- Maybe you just know the extension:
 - find . -name "*.R"

Redirecting output to a text file

```
3. ginger@f5f0b0f271e3: ~ (docker)
ginger@f5f0b0f271e3:~$ ls
123.txt  1kg  plink  r_config.R  shared_config.sh
ginger@f5f0b0f271e3:~$ head -2 123.txt
1
2
ginger@f5f0b0f271e3:~$ head -2 123.txt > test.txt
ginger@f5f0b0f271e3:~$ cat test.txt
1
2
ginger@f5f0b0f271e3:~$ tail -2 123.txt >> test.txt
ginger@f5f0b0f271e3:~$ cat test.txt
1
2
9
10
ginger@f5f0b0f271e3:~$
```

Looking back at your old commands

- Run the command: history

Compressing text data

gzip everything!

- There are many algorithms for file compression. gzip is the most popular in the unix world
- Achieves compression of 1/10th to 1/2 file size
- Many languages can read gzip files natively – and often faster!
 - Python: `gzip.open("file.gz")`
 - R: `read.table('file.gz')`

Compressing text files with gzip

```
3. ginger@f5f0b0f271e3: ~/gzip (docker)
ginger@f5f0b0f271e3:~/gzip$ ls -l -h
total 4.0K
-rw-r--r-- 1 ginger ginger 1.1K Sep 18 19:27 shared_config.sh
ginger@f5f0b0f271e3:~/gzip$ gzip shared_config.sh
ginger@f5f0b0f271e3:~/gzip$ ls -l -h
total 4.0K
-rw-r--r-- 1 ginger ginger 549 Sep 18 19:27 shared_config.sh.gz
ginger@f5f0b0f271e3:~/gzip$ gzip -d shared_config.sh.gz
ginger@f5f0b0f271e3:~/gzip$ ls -l -h
total 4.0K
-rw-r--r-- 1 ginger ginger 1.1K Sep 18 19:27 shared_config.sh
ginger@f5f0b0f271e3:~/gzip$ gzip -c shared_config.sh > shared_config.sh.gz
ginger@f5f0b0f271e3:~/gzip$ ls -l -h
total 8.0K
-rw-r--r-- 1 ginger ginger 1.1K Sep 18 19:27 shared_config.sh
-rw-r--r-- 1 ginger ginger 549 Sep 18 19:29 shared_config.sh.gz
ginger@f5f0b0f271e3:~/gzip$
```

Working with gzipped files

```
3. ginger@f5f0b0f271e3: ~/gzip (docker)
ginger@f5f0b0f271e3:~/gzip$ cat 123.txt
1
2
3
4
5
6
ginger@f5f0b0f271e3:~/gzip$ gzip 123.txt
ginger@f5f0b0f271e3:~/gzip$ cat 123.txt.gz
R[123.txt322222226
ginger@f5f0b0f271e3:~/gzip$
ginger@f5f0b0f271e3:~/gzip$ zcat 123.txt.gz
1
2
3
4
5
6
ginger@f5f0b0f271e3:~/gzip$
```

- cat -> zcat (gzcat on MacOS)
- grep -> zgrep
- less -> zless
- For other tools, we need pipes!

Pipes

Pipe example: display 3 first lines of gzipped

```
ginger@f5f0b0f271e3:~/gzip$ ls
123.txt.gz
ginger@f5f0b0f271e3:~/gzip$ zcat 123.txt.gz
1
2
3
4
5
6
ginger@f5f0b0f271e3:~/gzip$ zcat 123.txt.gz | head -3
1
2
3
ginger@f5f0b0f271e3:~/gzip$
```

A pipe (“|”) joins two commands together by taking the output (stdout) of the left command and sending it as input (stdin) to the right command.

You can chain as many commands as you like with pipes

```
3. ginger@f5f0b0f271e3: ~ (docker)

ginger@f5f0b0f271e3:~$ cat toy.map
1      rs0      0      1000
1      rs0      0      1000
1      rs10     0      1001
2      rs10     0      1001
10     rs10     0      1002
1      rs0      0      1000

ginger@f5f0b0f271e3:~$ sort -k1n toy.map | uniq | wc -l
4

ginger@f5f0b0f271e3:~$ grep -w rs10 toy.map | sort -k1n | uniq | gzip -c > result.txt.gz
ginger@f5f0b0f271e3:~$ zcat result.txt.gz
1      rs10     0      1001
2      rs10     0      1001
10     rs10     0      1002

ginger@f5f0b0f271e3:~$
```

Pipes to find old commands

- Look through your old commands
 - `history | less`
- Find an old command you wrote that you knew had the word “find” in it
 - `history | grep find`

Disconnecting from a session to reconnect later

- The most common tools are: screen and tmux
- screen starts a new screen
- Detach the window by holding control, and typing: ad
- Then you can safely log off
- Re-attach the window using: screen -r

Running large computations

- On a cluster:
 - qsub: Sun Grid Engine
 - bsub: LSF
- On the cloud:
 - dsub: Closest to a grid system like q/bsub. Some light setup required
 - Cromwell/WDL: requires a server to set up and bulky json/wdl files to run
 - Hail Batch (coming soon!): will be the easiest but not yet available

Tasks for today wrt Linux

1. Ensure [gcloud CLI](#) is installed
2. Check your cloud access
 1. using `gcloud auth list`
 1. Then config accordingly
 2. 1.1) fails, try out `gcloud auth login`
3. What if you are lazy to point and click? We could use `gsutil`. Let's find out what is in this bucket: `gs://neurogap_phenos_genos/` (hint something to do with `ls` or listing files)
4. Remember that VM we created? How do you, using your own command line? Ask Kumar
 1. `ssh` into it
 2. pausing vs stopping
 3. upload vs download files to VM
5. [Download the exercise in pdf](#) and do the exercises on your VM

Going further

- <https://swcarpentry.github.io/shell-novice/>
- <http://swcarpentry.github.io/shell-extras/>



MASSACHUSETTS
GENERAL HOSPITAL



HARVARD
MEDICAL SCHOOL



BROAD
INSTITUTE

GINGER On-Site Training Day 1: Intro to Linux QUESTIONS? 😊

GINGER Program 2022
University of Cape Town

Teaching Fellows:

Kumar Veerapen, Ph.D.

Senior Expert I Data Science
Kumar.Veerapen@Novartis.com

Carla Marquez-Luna, Ph.D.

Postdoctoral Research Fellow
carlamarquezluna@gmail.com