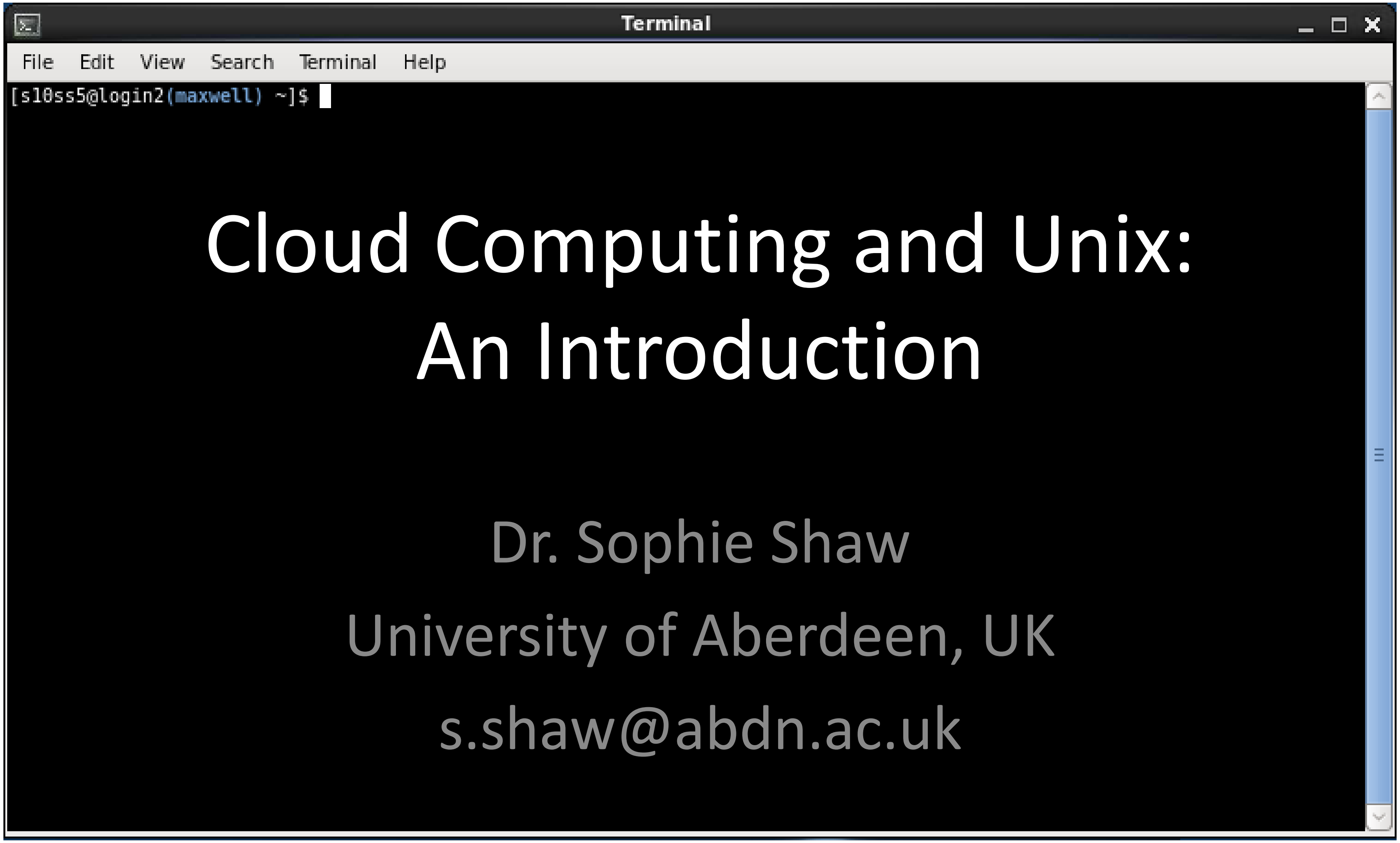


```
[sl0ss5@login2(maxwell) ~]$
```

Cloud Computing and Unix: An Introduction

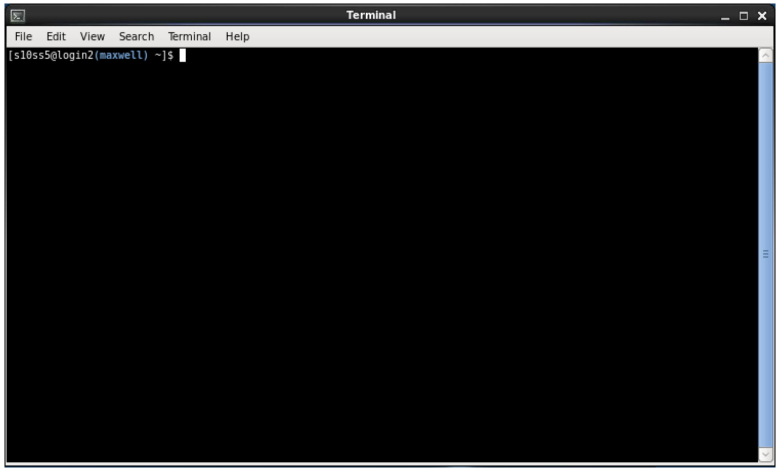
Dr. Sophie Shaw
University of Aberdeen, UK
s.shaw@abdn.ac.uk

Aberdeen
LONDON
London
Exeter

©Brian Doyle

What We're Going To Do

- Why Unix?
- Cloud Computing
- Connecting to AWS
- Introduction to Unix Commands



```
[s10ss5@login2(maxwell) ~]$
```

Etiquette

- PowerPoint interspersed with Challenges
- Ask me questions
- Ask demonstrators
- Work together
- Cheat!

Unix/Linux Command Reference

FOSSwire.com

File Commands

ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1 file2* – copy *file1* to *file2*
cp -r *dir1 dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1 file2* – rename or move *file1* to *file2* if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

Process Management

ps – display your currently active processes
top – display all running processes
kill *pid* – kill process id *pid*
killall *proc* – kill all processes named *proc* *
bg – lists stopped or background jobs; resume a stopped job in the background
fg – brings the most recent job to foreground
fg *n* – brings job *n* to the foreground

File Permissions

chmod *octal file* – change the permissions of *file* to *octal*, which can be found separately for user, group, and world by adding:

- 4 – read (r)
- 2 – write (w)
- 1 – execute (x)

Examples:
chmod 777 – read, write, execute for all
chmod 755 – rwx for owner, rx for group and world
For more options, see **man chmod**.

SSH

ssh *user@host* – connect to *host* as *user*
ssh -p *port user@host* – connect to *host* on port *port* as *user*
ssh-copy-id *user@host* – add your key to *host* for *user* to enable a keyed or passwordless login

Searching

grep *pattern files* – search for *pattern* in *files*
grep -r *pattern dir* – search recursively for *pattern* in *dir*
command* | grep *pattern – search for *pattern* in the output of *command*
locate *file* – find all instances of *file*

System Info

date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger *user* – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man *command* – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis *app* – show possible locations of *app*
which *app* – show which *app* will be run by default

Compression

tar cf *file.tar files* – create a tar named *file.tar* containing *files*
tar xf *file.tar* – extract the files from *file.tar*
tar czf *file.tar.gz files* – create a tar with Gzip compression
tar xzf *file.tar.gz* – extract a tar using Gzip
tar cjf *file.tar.bz2* – create a tar with Bzip2 compression
tar xjf *file.tar.bz2* – extract a tar using Bzip2
gzip *file* – compresses *file* and renames it to *file.gz*
gzip -d *file.gz* – decompresses *file.gz* back to *file*

Network

ping *host* – ping *host* and output results
whois *domain* – get whois information for *domain*
dig *domain* – get DNS information for *domain*
dig -x *host* – reverse lookup *host*
wget *file* – download *file*
wget -c *file* – continue a stopped download

Installation

Install from source:
```
./configure
make
make install
```
dpkg -i *pkg.deb* – install a package (Debian)
rpm -Uvh *pkg.rpm* – install a package (RPM)

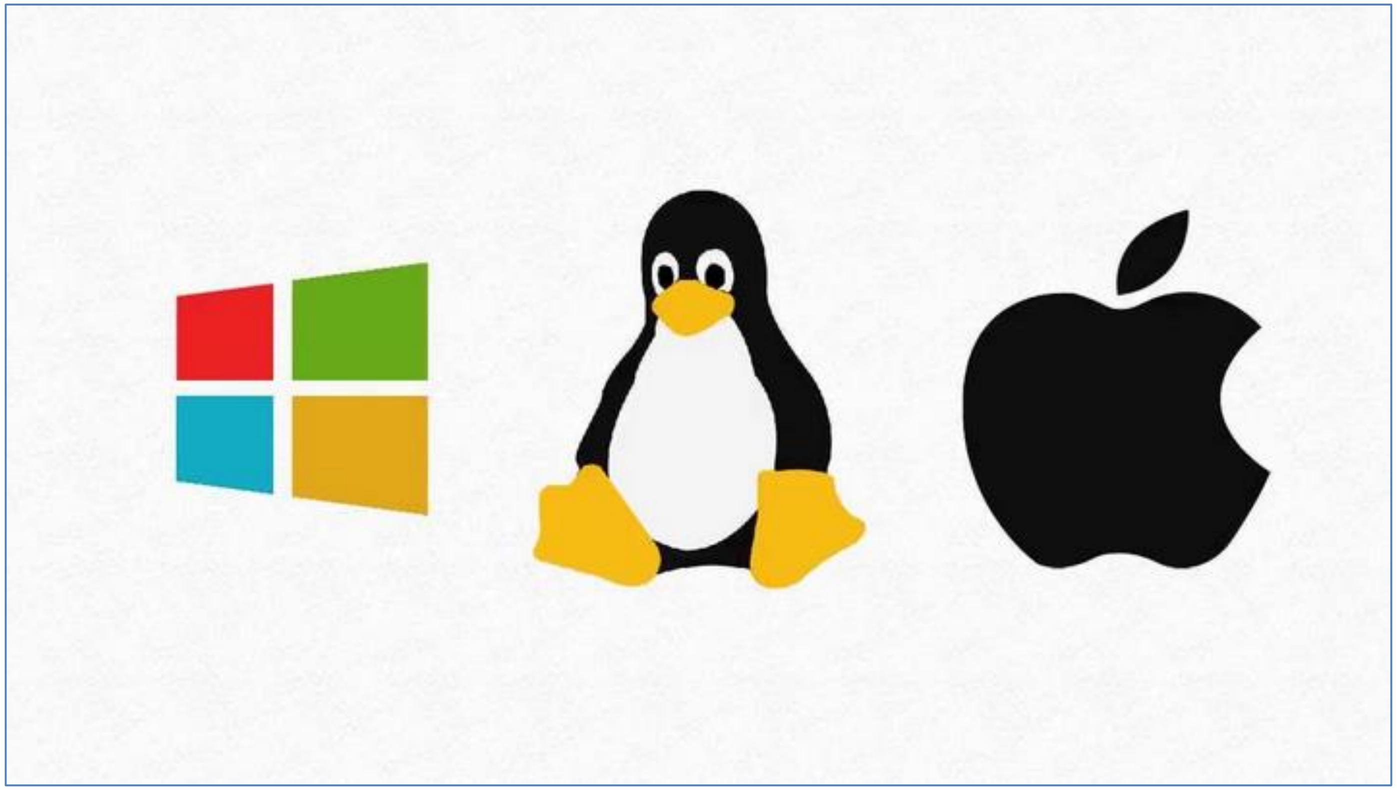
Shortcuts

Ctrl+C – halts the current command
Ctrl+Z – stops the current command, resume with **fg** in the foreground or **bg** in the background
Ctrl+D – log out of current session, similar to **exit**
Ctrl+W – erases one word in the current line
Ctrl+U – erases the whole line
Ctrl+R – type to bring up a recent command
!! - repeats the last command
exit – log out of current session

* use with extreme caution.

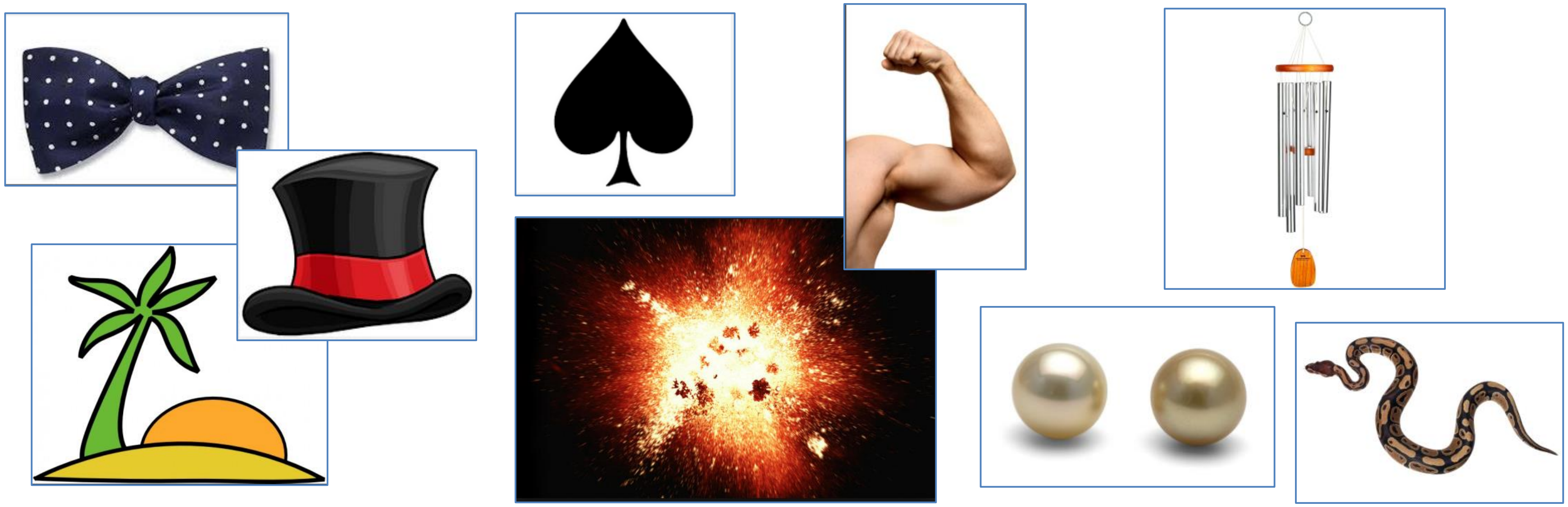
What is Unix?

- Operating System



Why Unix?

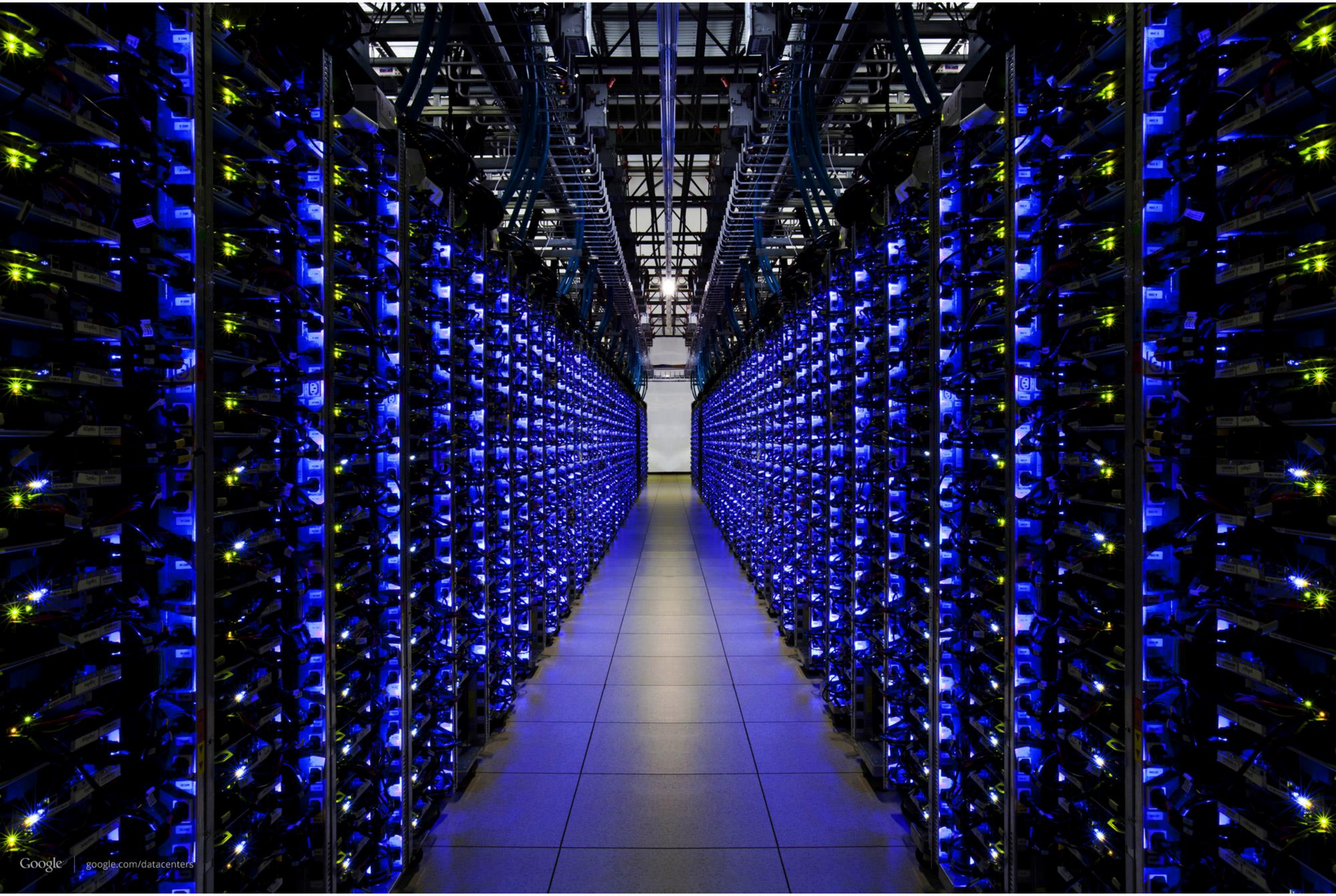
- Bioinformatics software designed to run on Unix platforms.
- Large amounts of data.
- Much faster than your Windows PC.



How Can We Use Unix?

- Linux computers or servers.
- Compute clusters.
- The cloud.
  - What we're going to use this week

So What is Cloud Computing?

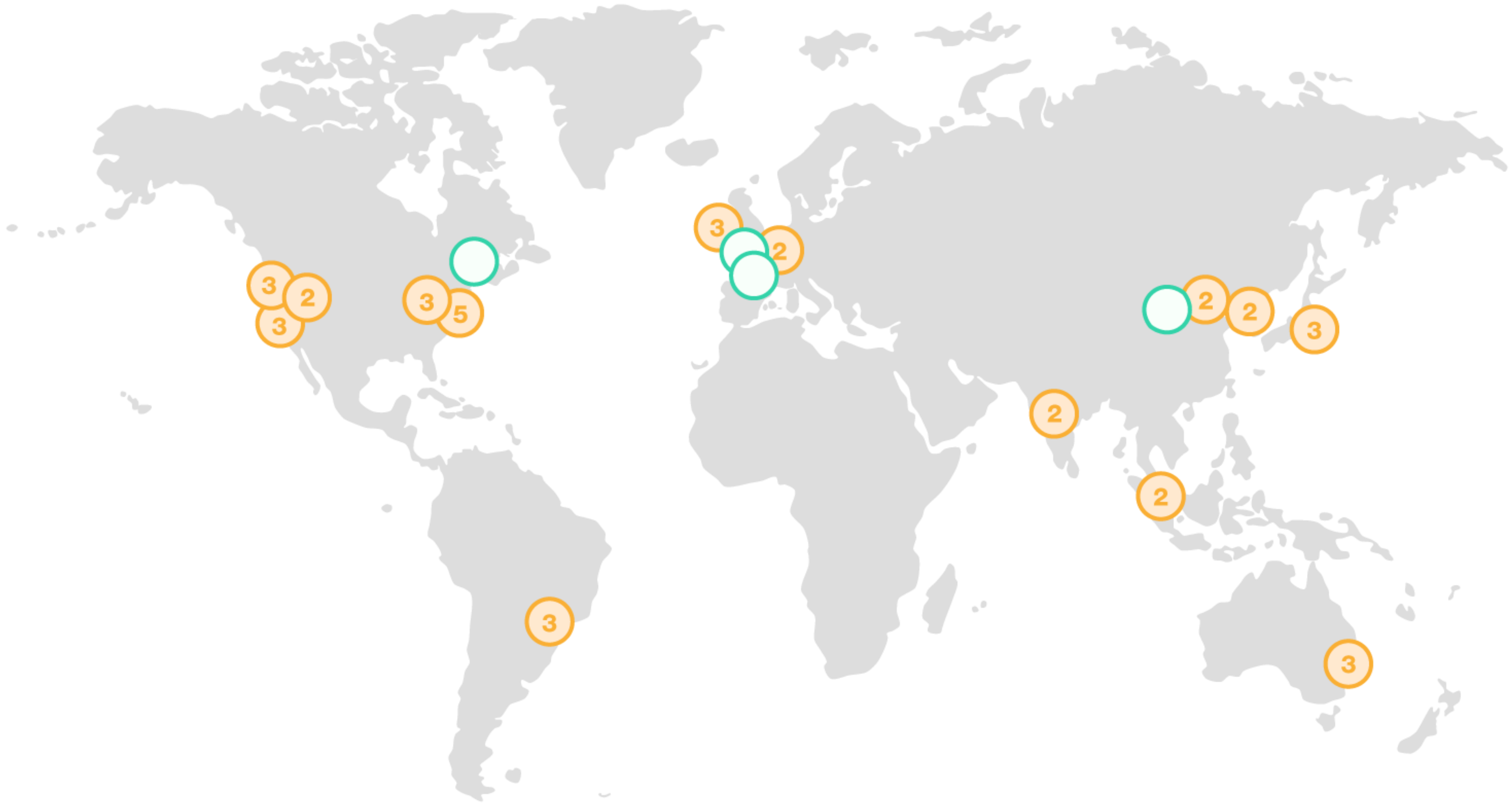

Google | google.com/datacenters

Cloud Computing Solutions

amazon web services™

Microsoft Azure

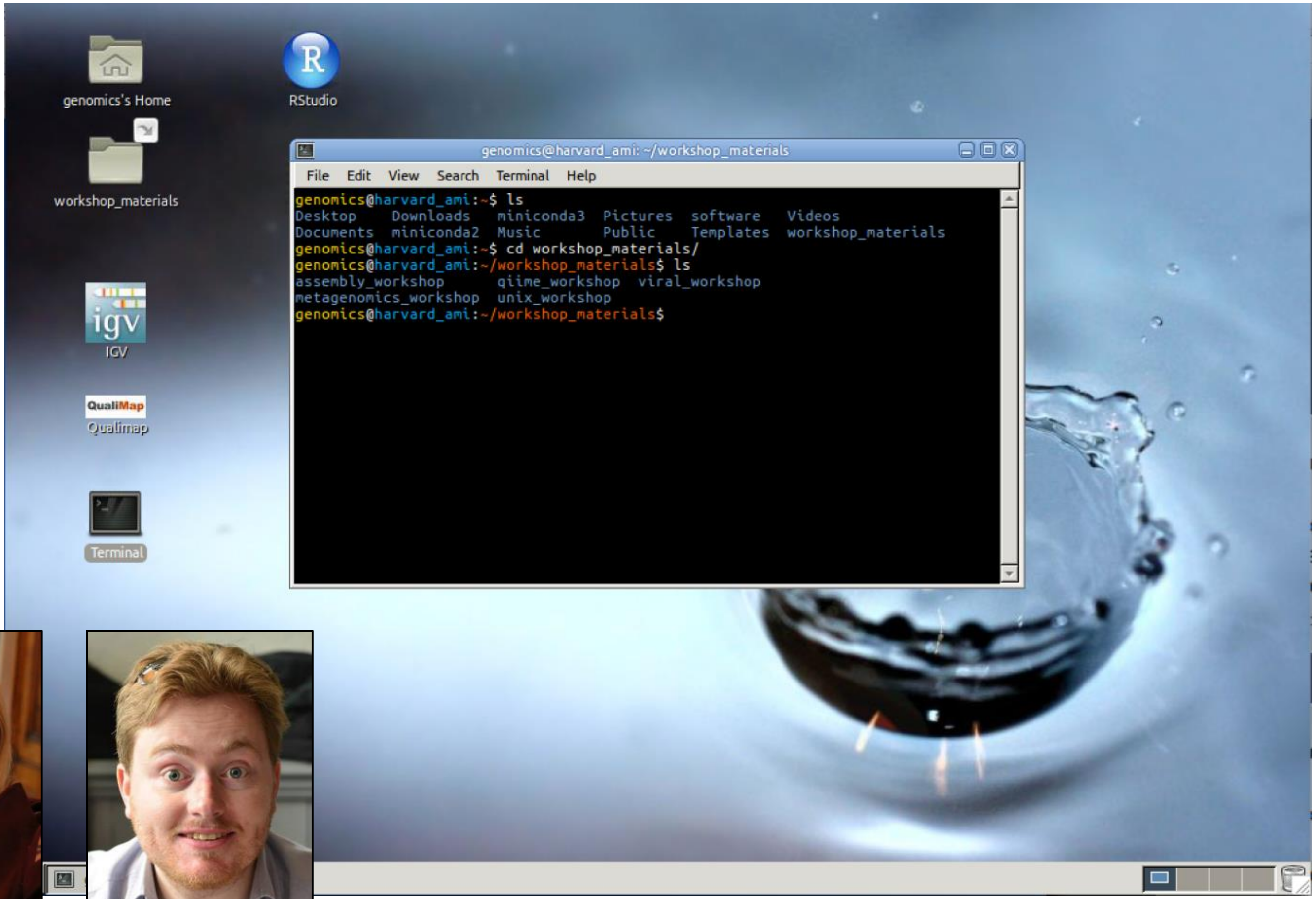
Google Compute Engine



AWS “Availability Zones” and Data Centres

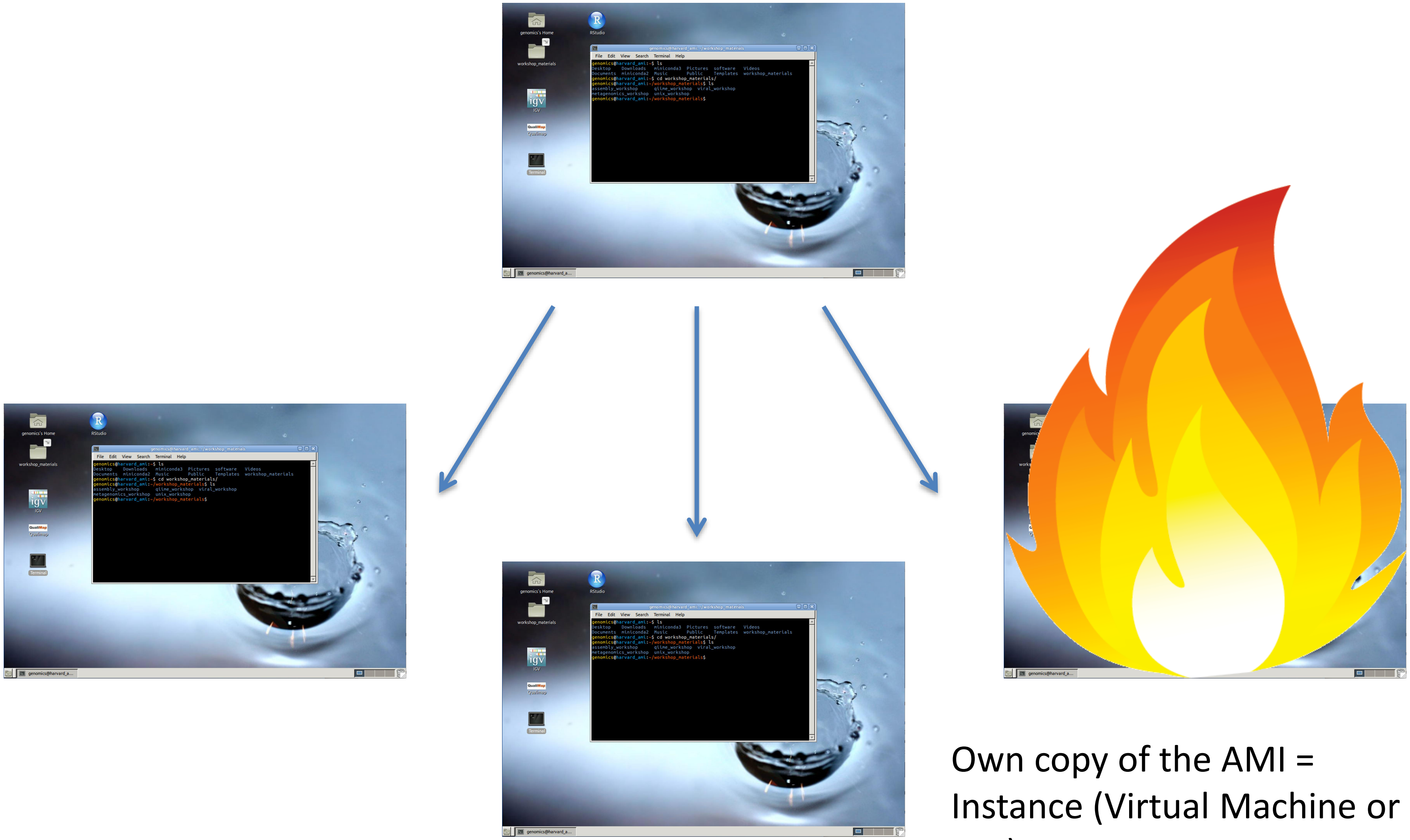
How it Works

AMI ("Amazon Machine Image")
Base computer with all data and software


genomics's Home
RStudio
workshop_materials
IGV
Qualimap
Terminal
genomics@harvard_ami: ~/workshop_materials
File Edit View Search Terminal Help

```
genomics@harvard_ami:~$ ls
Desktop    Downloads   miniconda3  Pictures  software   Videos
Documents  miniconda2  Music       Public    Templates  workshop_materials
genomics@harvard_ami:~$ cd workshop_materials/
genomics@harvard_ami:~/workshop_materials$ ls
assembly_workshop      qiime_workshop  viral_workshop
metagenomics_workshop  unix_workshop
genomics@harvard_ami:~/workshop_materials$
```

How it Works


genomics's Home
RStudio
workshop_materials
IGV
Qualimap
Terminal
genomics@harvard_ami: ~/workshop_materials
File Edit View Search Terminal Help

```
genomics@harvard_ami:~$ ls
Desktop    Downloads   miniconda3  Pictures  software   Videos
Documents  miniconda2  Music       Public    Templates  workshop_materials
genomics@harvard_ami:~$ cd workshop_materials/
genomics@harvard_ami:~/workshop_materials$ ls
assembly_workshop      qiime_workshop  viral_workshop
metagenomics_workshop  unix_workshop
genomics@harvard_ami:~/workshop_materials$
```

Own copy of the AMI = Instance (Virtual Machine or VM)

Terminology

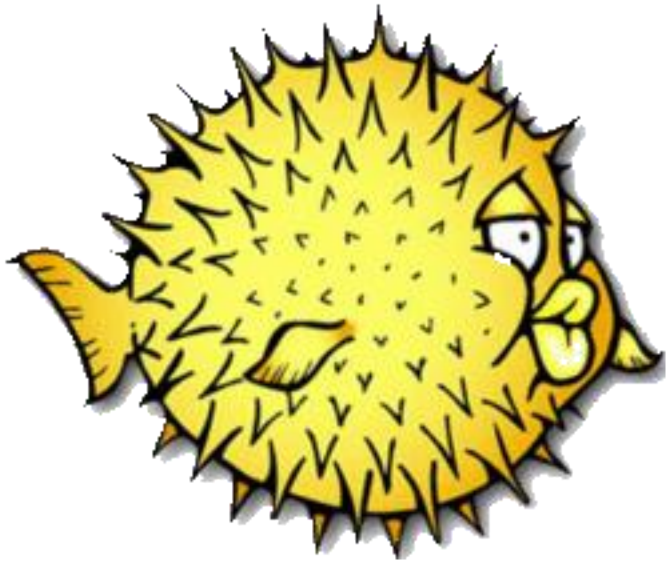
- Creating an instance – *buying a brand new computer with software already installed.*
- Starting an instance – *turning that computer on.*
- Stopping an instance – *turning that computer off.*
- Terminating an instance – *setting that computer on fire and throwing it out of the window.*

The Rules

- Only create one instance each.
- Stop your instance at the end of each day (unless you have software running).
- Name your instance (with YOUR name! No Bruce Waynes please)
- Only start or stop your own instance.
- Only terminate your own instance.

Connecting to Your Instance

Remote Desktop
Software
e.g. X2Go

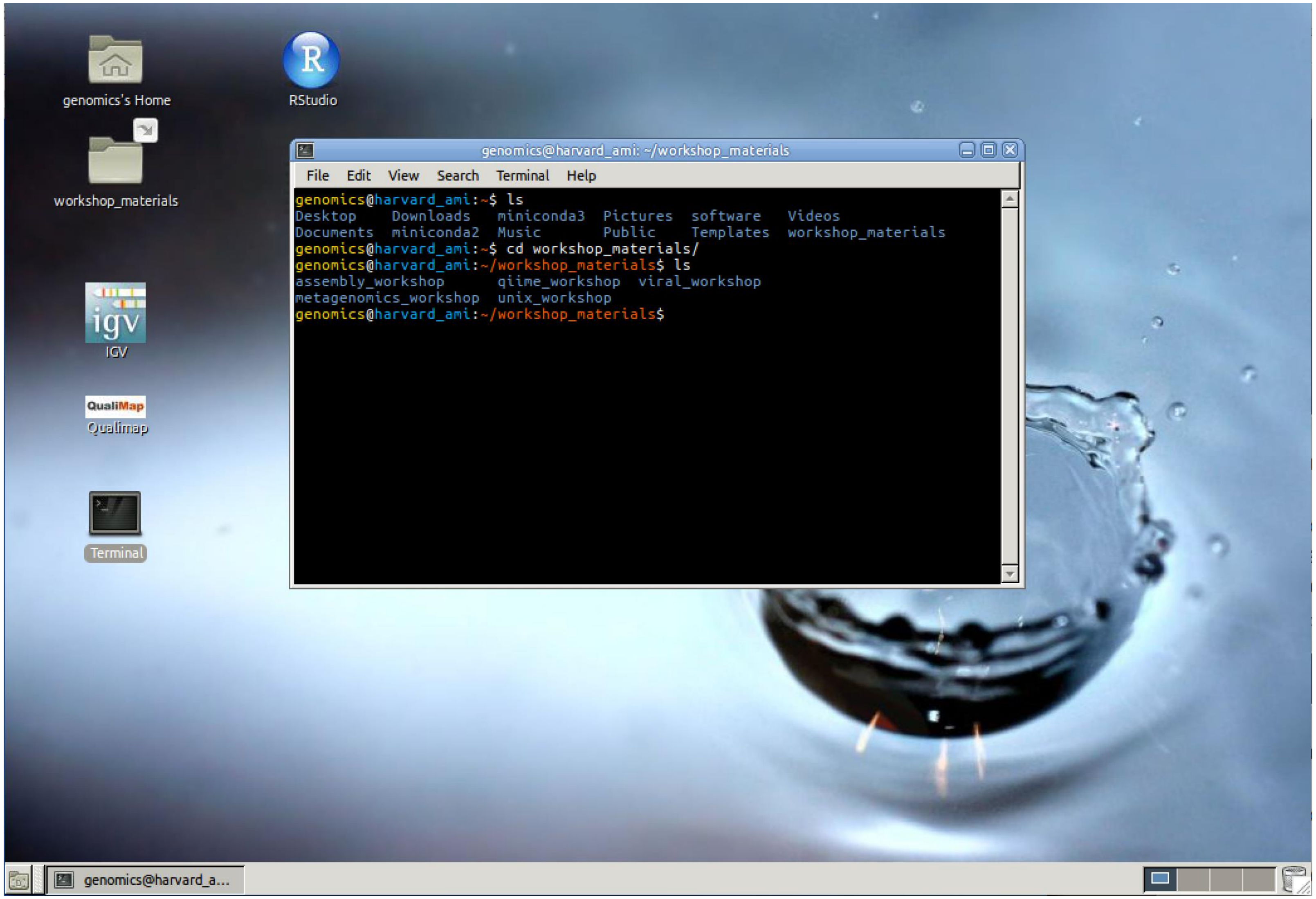


Secure Shell –
“SSH”
e.g. SSH or PuTTY

Now What?!

- You’re each going to create, start and connect to your own instance.

INSERT LIVE DEMO

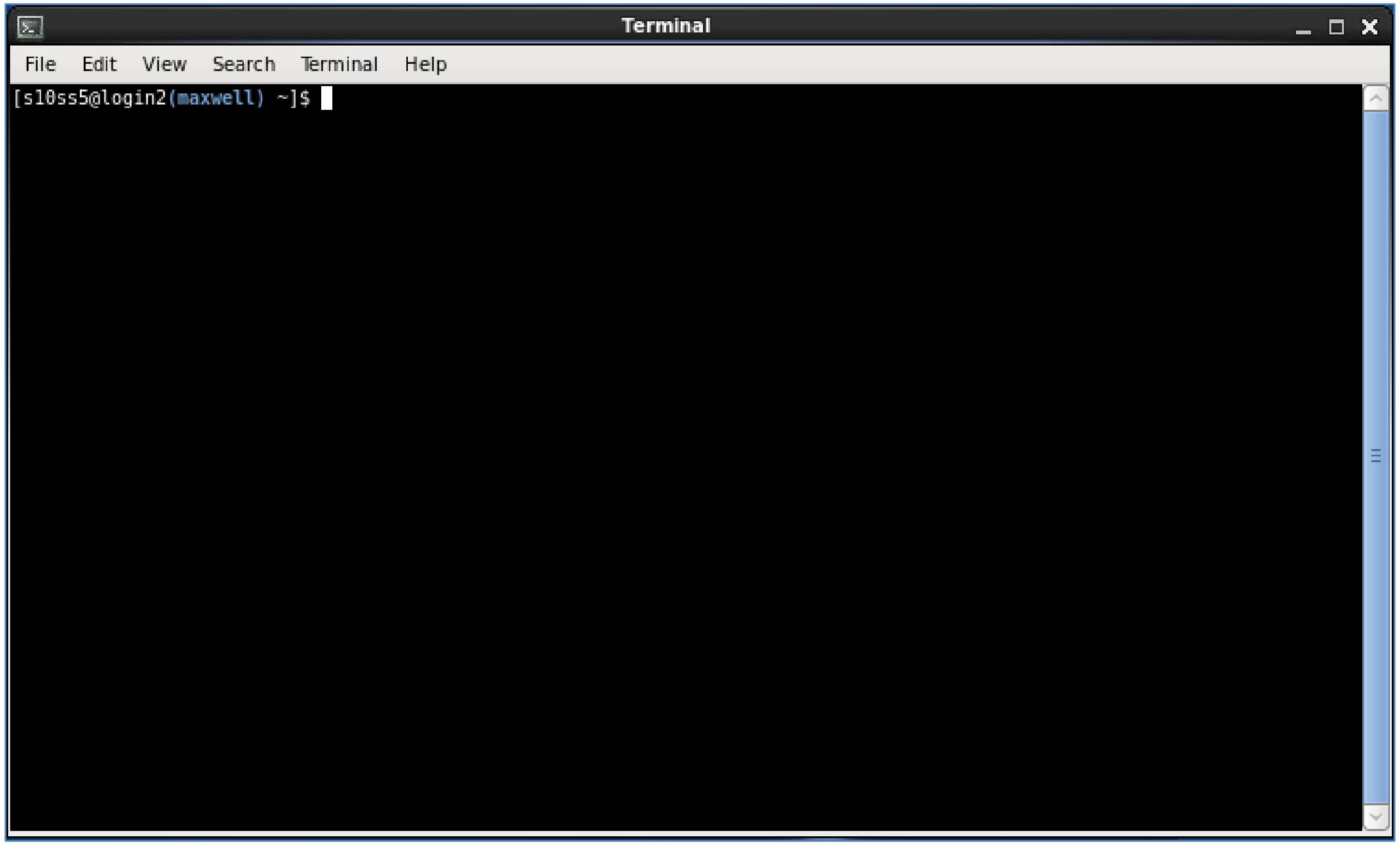


```
genomics@harvard_ami:~$ ls
Desktop    Downloads   miniconda3  Pictures  software   Videos
Documents  miniconda2  Music       Public    Templates  workshop_materials
genomics@harvard_ami:~$ cd workshop_materials/
genomics@harvard_ami:~/workshop_materials$ ls
assembly_workshop      qiime_workshop  viral_workshop
metagenomics_workshop  unix_workshop
genomics@harvard_ami:~/workshop_materials$
```

You’re now connected to your instance and you’re ready to learn some Unix!

Any Questions So Far?

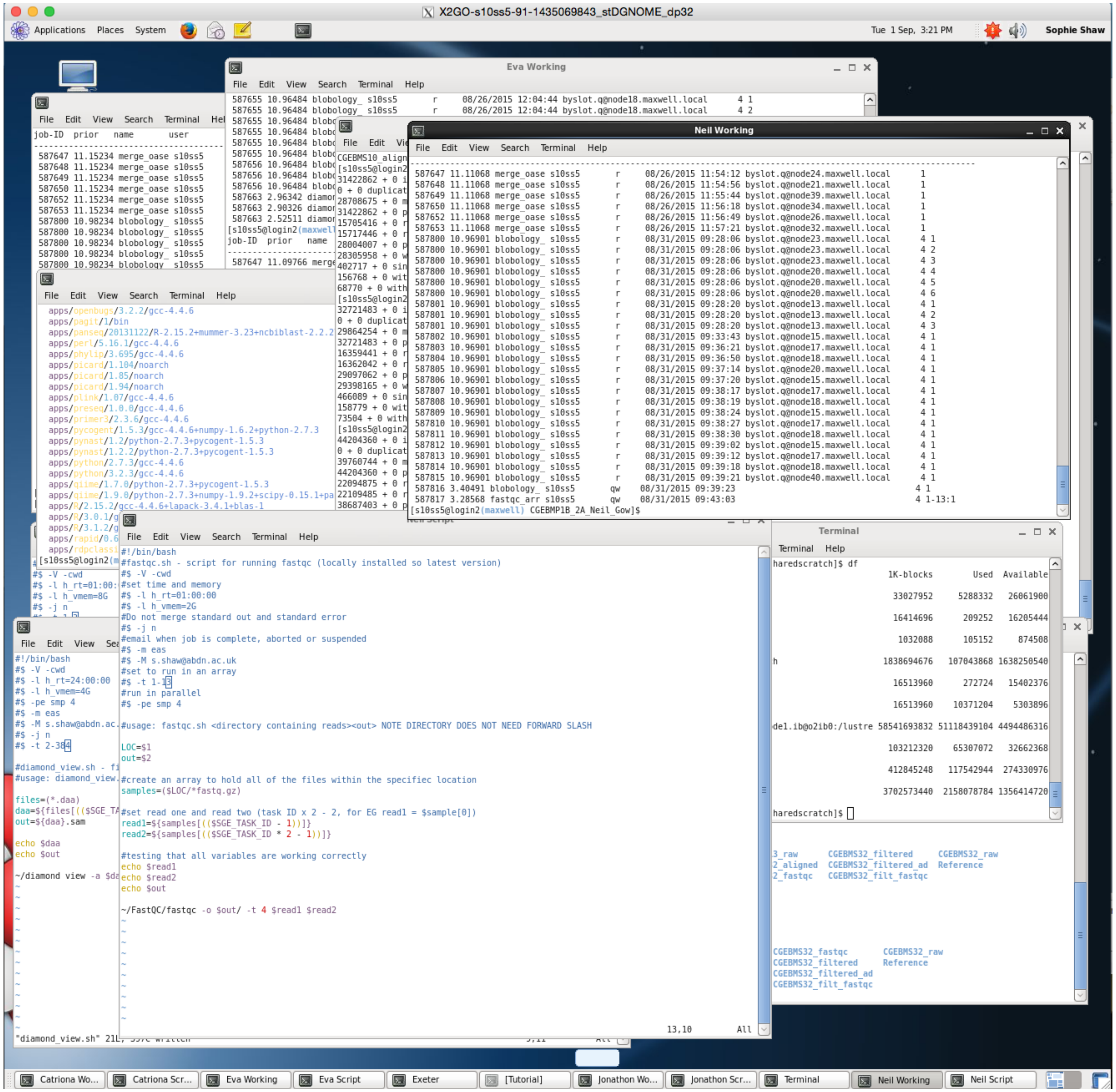
The Terminal Window



```
[sl0ss5@login2(maxwell) ~]$ 
```

The Command Line, The Shell, The Prompt

Where you see this “$” followed by text, I want you to type the text on your command line


X2GO-s10ss5-91-1435069843_stDGNOME_dp32
Applications Places System
Tue 1 Sep, 3:21 PM
Sophie Shaw
Eva Working
Neil Working
File Edit View Search Terminal Help
Terminal
Catriona Wo... Catriona Scr... Eva Working Eva Script Exeter [Tutorial] Jonathon Wo... Jonathon Scr... Terminal Neil Working Neil Script

```
587655 10.96484 blobology_ s10ss5       r     08/26/2015 12:04:44 byslot.q@node18.maxwell.local      4 1
587655 10.96484 blobology_ s10ss5       r     08/26/2015 12:04:44 byslot.q@node18.maxwell.local      4 2
```

```
--------------------------------------------------------------------------------------------------------------
587647 11.11068 merge_oase s10ss5       r     08/26/2015 11:54:12 byslot.q@node24.maxwell.local      1
587648 11.11068 merge_oase s10ss5       r     08/26/2015 11:54:56 byslot.q@node21.maxwell.local      1
587649 11.11068 merge_oase s10ss5       r     08/26/2015 11:55:44 byslot.q@node39.maxwell.local      1
587650 11.11068 merge_oase s10ss5       r     08/26/2015 11:56:18 byslot.q@node34.maxwell.local      1
587652 11.11068 merge_oase s10ss5       r     08/26/2015 11:56:49 byslot.q@node26.maxwell.local      1
587653 11.11068 merge_oase s10ss5       r     08/26/2015 11:57:21 byslot.q@node32.maxwell.local      1
587800 10.96901 blobology_ s10ss5       r     08/31/2015 09:28:06 byslot.q@node23.maxwell.local      4 1
587800 10.96901 blobology_ s10ss5       r     08/31/2015 09:28:06 byslot.q@node23.maxwell.local      4 2
587800 10.96901 blobology_ s10ss5       r     08/31/2015 09:28:06 byslot.q@node23.maxwell.local      4 3
587800 10.96901 blobology_ s10ss5       r     08/31/2015 09:28:06 byslot.q@node20.maxwell.local      4 4
587800 10.96901 blobology_ s10ss5       r     08/31/2015 09:28:06 byslot.q@node20.maxwell.local      4 5
587800 10.96901 blobology_ s10ss5       r     08/31/2015 09:28:06 byslot.q@node20.maxwell.local      4 6
587801 10.96901 blobology_ s10ss5       r     08/31/2015 09:28:20 byslot.q@node13.maxwell.local      4 1
587801 10.96901 blobology_ s10ss5       r     08/31/2015 09:28:20 byslot.q@node13.maxwell.local      4 2
587801 10.96901 blobology_ s10ss5       r     08/31/2015 09:28:20 byslot.q@node13.maxwell.local      4 3
587802 10.96901 blobology_ s10ss5       r     08/31/2015 09:33:43 byslot.q@node15.maxwell.local      4 1
587803 10.96901 blobology_ s10ss5       r     08/31/2015 09:36:21 byslot.q@node17.maxwell.local      4 1
587804 10.96901 blobology_ s10ss5       r     08/31/2015 09:36:50 byslot.q@node18.maxwell.local      4 1
587805 10.96901 blobology_ s10ss5       r     08/31/2015 09:37:14 byslot.q@node20.maxwell.local      4 1
587806 10.96901 blobology_ s10ss5       r     08/31/2015 09:37:20 byslot.q@node15.maxwell.local      4 1
587807 10.96901 blobology_ s10ss5       r     08/31/2015 09:38:17 byslot.q@node17.maxwell.local      4 1
587808 10.96901 blobology_ s10ss5       r     08/31/2015 09:38:19 byslot.q@node18.maxwell.local      4 1
587809 10.96901 blobology_ s10ss5       r     08/31/2015 09:38:24 byslot.q@node15.maxwell.local      4 1
587810 10.96901 blobology_ s10ss5       r     08/31/2015 09:38:27 byslot.q@node17.maxwell.local      4 1
587811 10.96901 blobology_ s10ss5       r     08/31/2015 09:38:30 byslot.q@node18.maxwell.local      4 1
587812 10.96901 blobology_ s10ss5       r     08/31/2015 09:39:02 byslot.q@node15.maxwell.local      4 1
587813 10.96901 blobology_ s10ss5       r     08/31/2015 09:39:12 byslot.q@node17.maxwell.local      4 1
587814 10.96901 blobology_ s10ss5       r     08/31/2015 09:39:18 byslot.q@node18.maxwell.local      4 1
587815 10.96901 blobology_ s10ss5       r     08/31/2015 09:39:21 byslot.q@node40.maxwell.local      4 1
587816 3.40491 blobology_ s10ss5       qw    08/31/2015 09:39:23                                    4 1
587817 3.28568 fastqc_arr s10ss5       qw    08/31/2015 09:43:03                                    4 1-13:1
[s10ss5@login2(maxwell) CGEBMP1B_2A_Neil_Gow]$
```

```
#!/bin/bash
#fastqc.sh - script for running fastqc (locally installed so latest version)
#$ -V -cwd
#set time and memory
#$ -l h_rt=01:00:00
#$ -l h_vmem=2G
#Do not merge standard out and standard error
#$ -j n
#email when job is complete, aborted or suspended
#$ -m eas
#$ -M s.shaw@abdn.ac.uk
#set to run in an array
#$ -t 1-13
#run in parallel
#$ -pe smp 4

#usage: fastqc.sh <directory containing reads><out> NOTE DIRECTORY DOES NOT NEED FORWARD SLASH

LOC=$1
out=$2

#create an array to hold all of the files within the specifiec location
samples=($LOC/*fastq.gz)

#set read one and read two (task ID x 2 - 2, for EG read1 = $sample[0])
read1=${samples[(($SGE_TASK_ID - 1))]}
read2=${samples[(($SGE_TASK_ID * 2 - 1))]}

#testing that all variables are working correctly
echo $read1
echo $read2
echo $out

~/FastQC/fastqc -o $out/ -t 4 $read1 $read2
~
~
~
~
~
~
~
~
~
~
                                                                            13,10         All
```

```
haredscratch]$ df
             1K-blocks        Used   Available
              33027952     5288332    26061900
              16414696      209252    16205444
               1032088      105152      874508
h           1838694676   107043868  1638250540
              16513960      272724    15402376
              16513960    10371204     5303896
del.ib@o2ib0:/lustre 58541693832 51118439104 4494486316
             103212320    65307072    32662368
             412845248   117542944   274330976
            3702573440  2158078784  1356414720
haredscratch]$
```

```
3_raw      CGEBMS32_filtered    CGEBMS32_raw
2_aligned  CGEBMS32_filtered_ad Reference
2_fastqc   CGEBMS32_filt_fastqc

CGEBMS32_fastqc       CGEBMS32_raw
CGEBMS32_filtered     Reference
CGEBMS32_filtered_ad
CGEBMS32_filt_fastqc
```

```
job-ID  prior   name       user
-----------------------------------
587647 11.15234 merge_oase s10ss5
587648 11.15234 merge_oase s10ss5
587649 11.15234 merge_oase s10ss5
587650 11.15234 merge_oase s10ss5
587652 11.15234 merge_oase s10ss5
587653 11.15234 merge_oase s10ss5
587800 10.98234 blobology_ s10ss5
587800 10.98234 blobology_ s10ss5
587800 10.98234 blobology_ s10ss5
587800 10.98234 blobology_ s10ss5
587800 10.98234 blobology_ s10ss5
```

```
587655 10.96484 blobo
587655 10.96484 blobo
587655 10.96484 blobo
587655 10.96484 blobo
587656 10.96484 blobo
587656 10.96484 blobo
587656 10.96484 blobo
587663 2.96342 diamon
587663 2.90326 diamon
587663 2.52511 diamon
[s10ss5@login2(maxwell
job-ID  prior   name
-----------------------
587647 11.09766 merge
```

```
CGEBMS10_align
[s10ss5@login2
31422862 + 0 i
0 + 0 duplicat
28708675 + 0 m
31422862 + 0 p
15705416 + 0 r
15717446 + 0 r
28004007 + 0 p
28305958 + 0 w
402717 + 0 sin
156768 + 0 wit
68770 + 0 with
[s10ss5@login2
32721483 + 0 i
0 + 0 duplicat
29864254 + 0 m
32721483 + 0 p
16359441 + 0 r
16362042 + 0 r
29097062 + 0 p
29398165 + 0 w
466089 + 0 sin
158779 + 0 wit
73504 + 0 with
[s10ss5@login2
44204360 + 0 i
0 + 0 duplicat
39760744 + 0 m
44204360 + 0 p
22094875 + 0 r
22109485 + 0 r
38687403 + 0 p
```

```
apps/openbugs/3.2.2/gcc-4.4.6
apps/pagit/1/bin
apps/pansea/20131122/R-2.15.2+mummer-3.23+ncbiblast-2.2.2
apps/perl/5.16.1/gcc-4.4.6
apps/phylip/3.695/gcc-4.4.6
apps/picard/1.104/noarch
apps/picard/1.85/noarch
apps/picard/1.94/noarch
apps/plink/1.07/gcc-4.4.6
apps/preseq/1.0.0/gcc-4.4.6
apps/primer3/2.3.6/gcc-4.4.6
apps/pycogent/1.5.3/gcc-4.4.6+numpy-1.6.2+python-2.7.3
apps/pynast/1.2/python-2.7.3+pycogent-1.5.3
apps/pynast/1.2.2/python-2.7.3+pycogent-1.5.3
apps/python/2.7.3/gcc-4.4.6
apps/python/3.2.3/gcc-4.4.6
apps/qiime/1.7.0/python-2.7.3+pycogent-1.5.3
apps/qiime/1.9.0/python-2.7.3+numpy-1.9.2+scipy-0.15.1+pa
apps/R/2.15.2/gcc-4.4.6+lapack-3.4.1+blas-1
apps/R/3.0.1/g
apps/R/3.1.2/g
apps/rapid/0.6
apps/rdpclassi
[s10ss5@login2(m
```

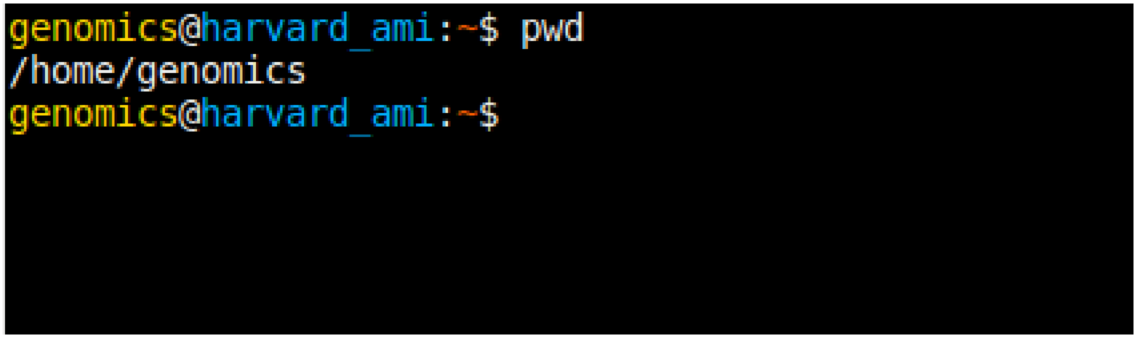
```
#$ -V -cwd
#$ -l h_rt=01:00:
#$ -l h_vmem=8G
#$ -j n
```

```
#!/bin/bash
#$ -V -cwd
#$ -l h_rt=24:00:00
#$ -l h_vmem=4G
#$ -pe smp 4
#$ -m eas
#$ -M s.shaw@abdn.ac
#$ -j n
#$ -t 2-384

#diamond_view.sh - fi
#usage: diamond_view.

files=(*.daa)
daa=${files[(($SGE_TA
out=${daa}.sam

echo $daa
echo $out

~/diamond view -a $da
~
~
~
~
~
~
~
~
~
~
~
~
~
"diamond_view.sh" 21L,
```

Location is Important

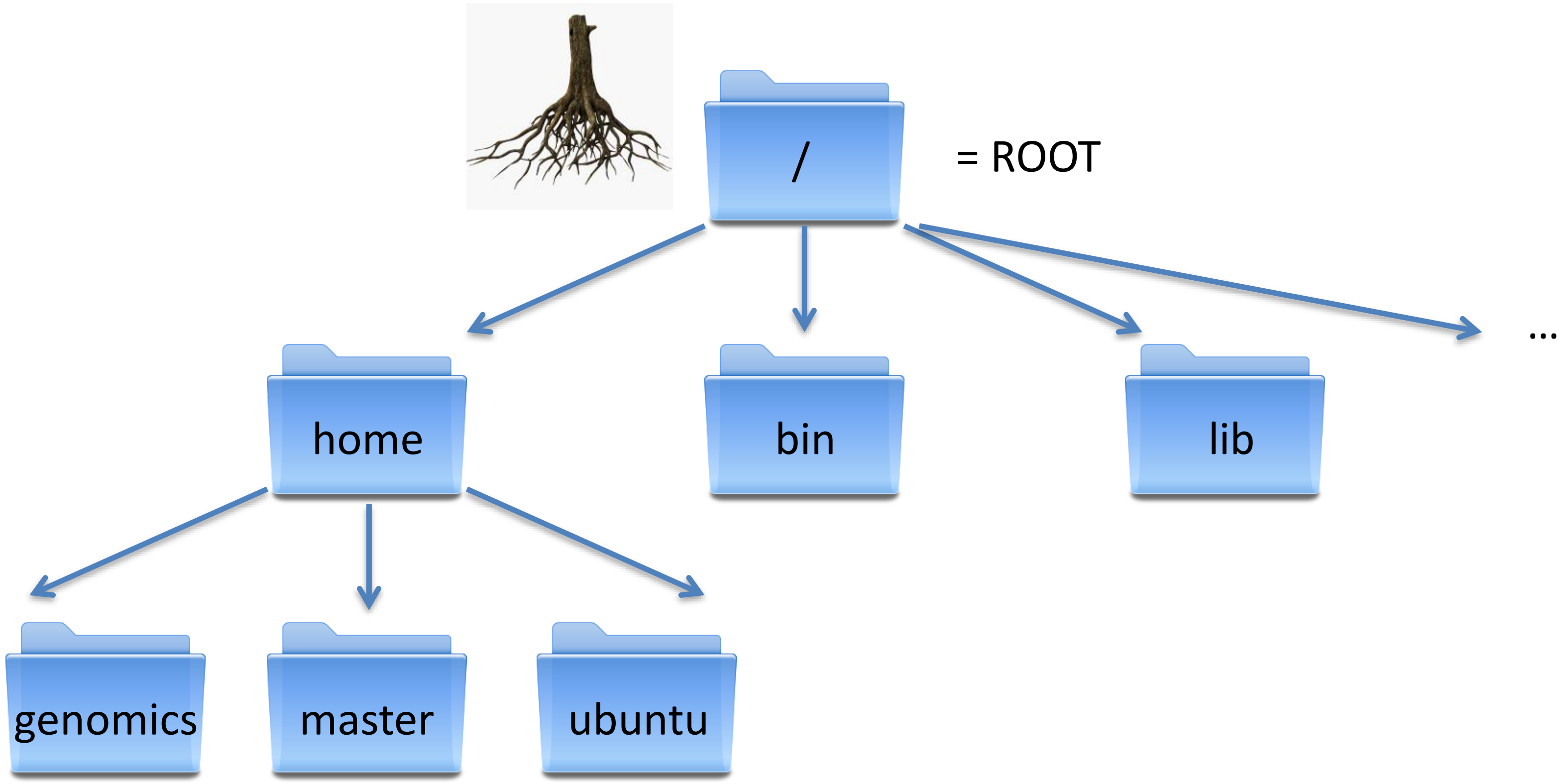
First Task – Where am I?

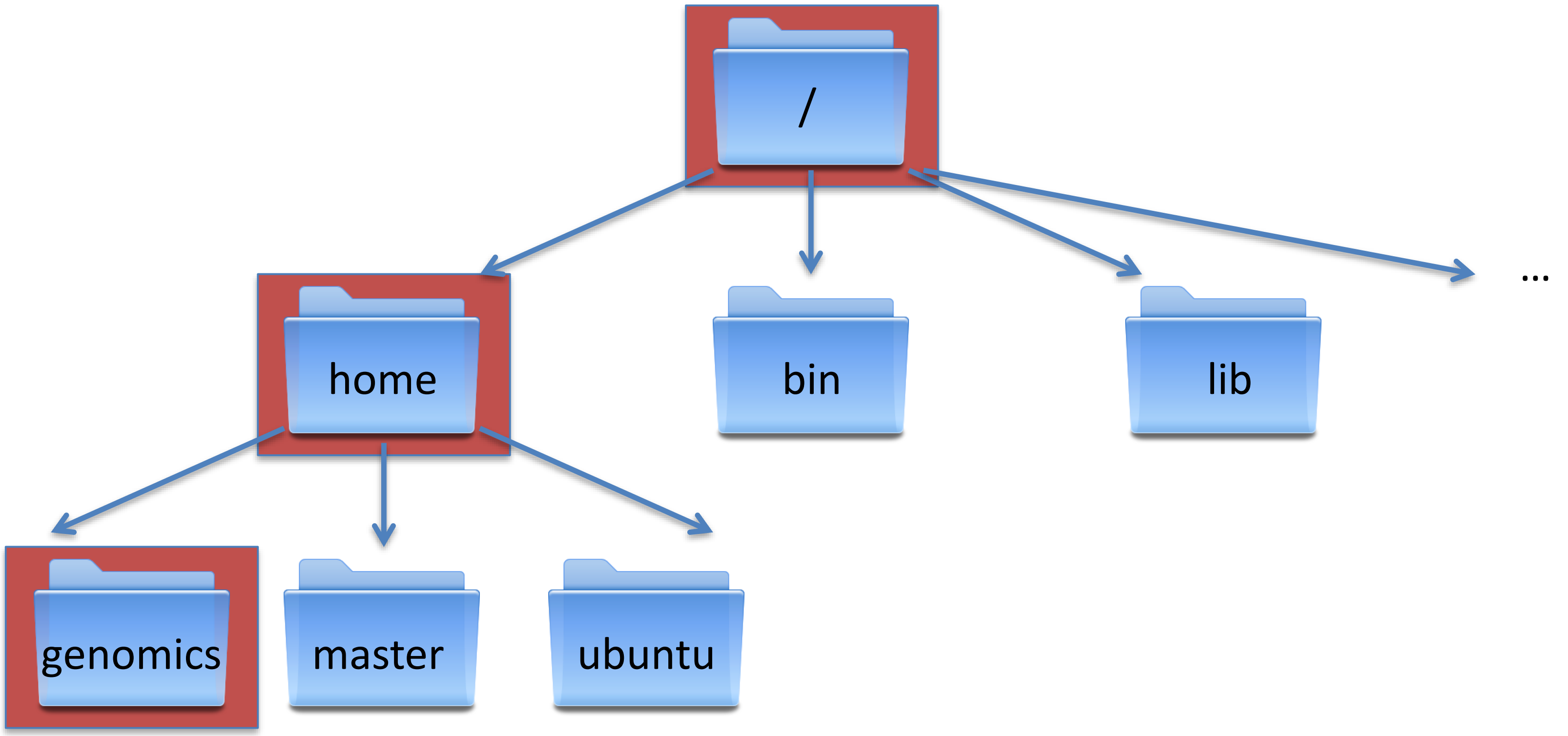
```
$ pwd
```

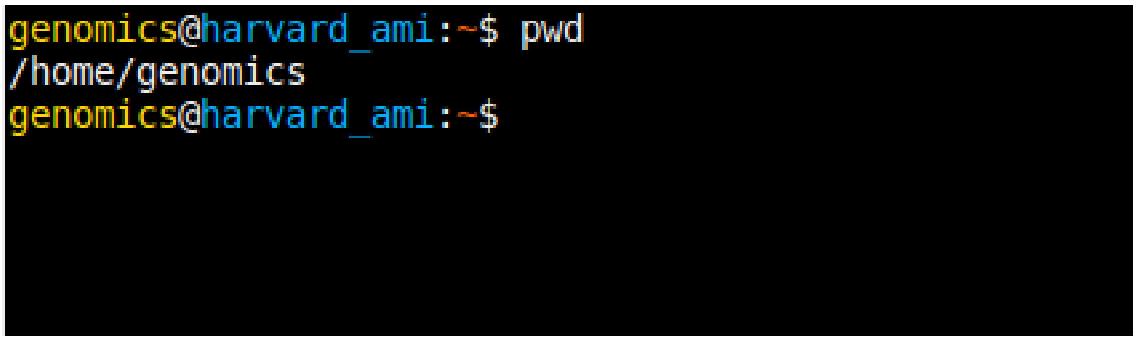


```
genomics@harvard_ami:~$ pwd
/home/genomics
genomics@harvard_ami:~$
```

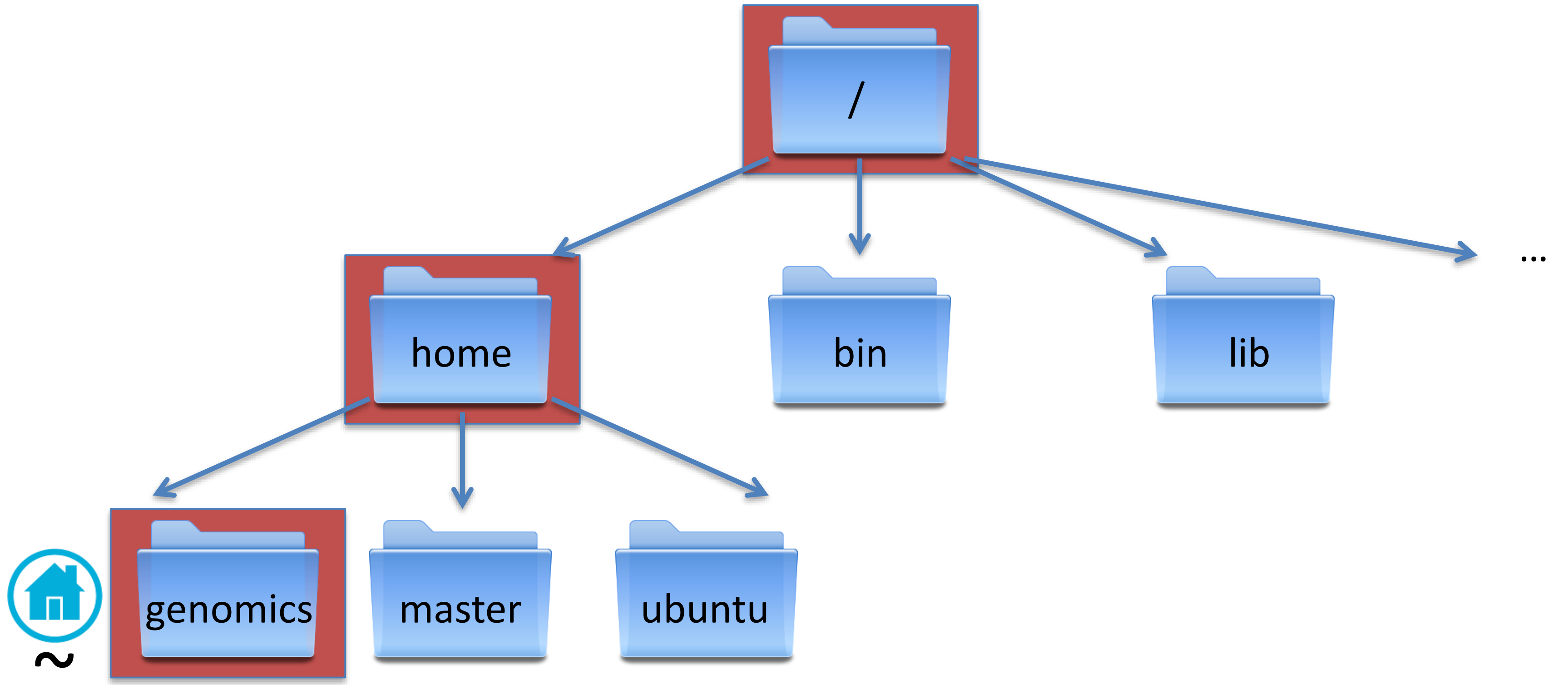
This is your “present working directory”

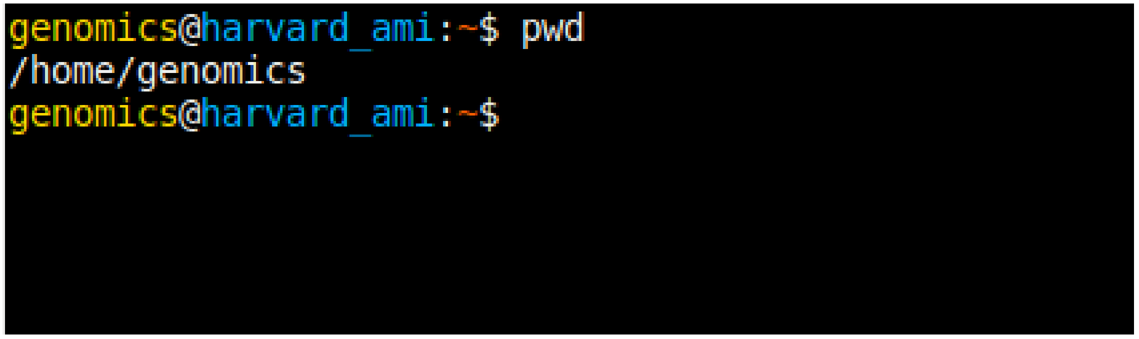

/
= ROOT
home
bin
lib
...
genomics
master
ubuntu





```
genomics@harvard_ami:~$ pwd
/home/genomics
genomics@harvard_ami:~$
```





```
genomics@harvard_ami:~$ pwd
/home/genomics
genomics@harvard_ami:~$
```

This location is also known as your Home Directory

Tilde is shorthand for Home:

~

Now let’s create some directories and files

Make a directory

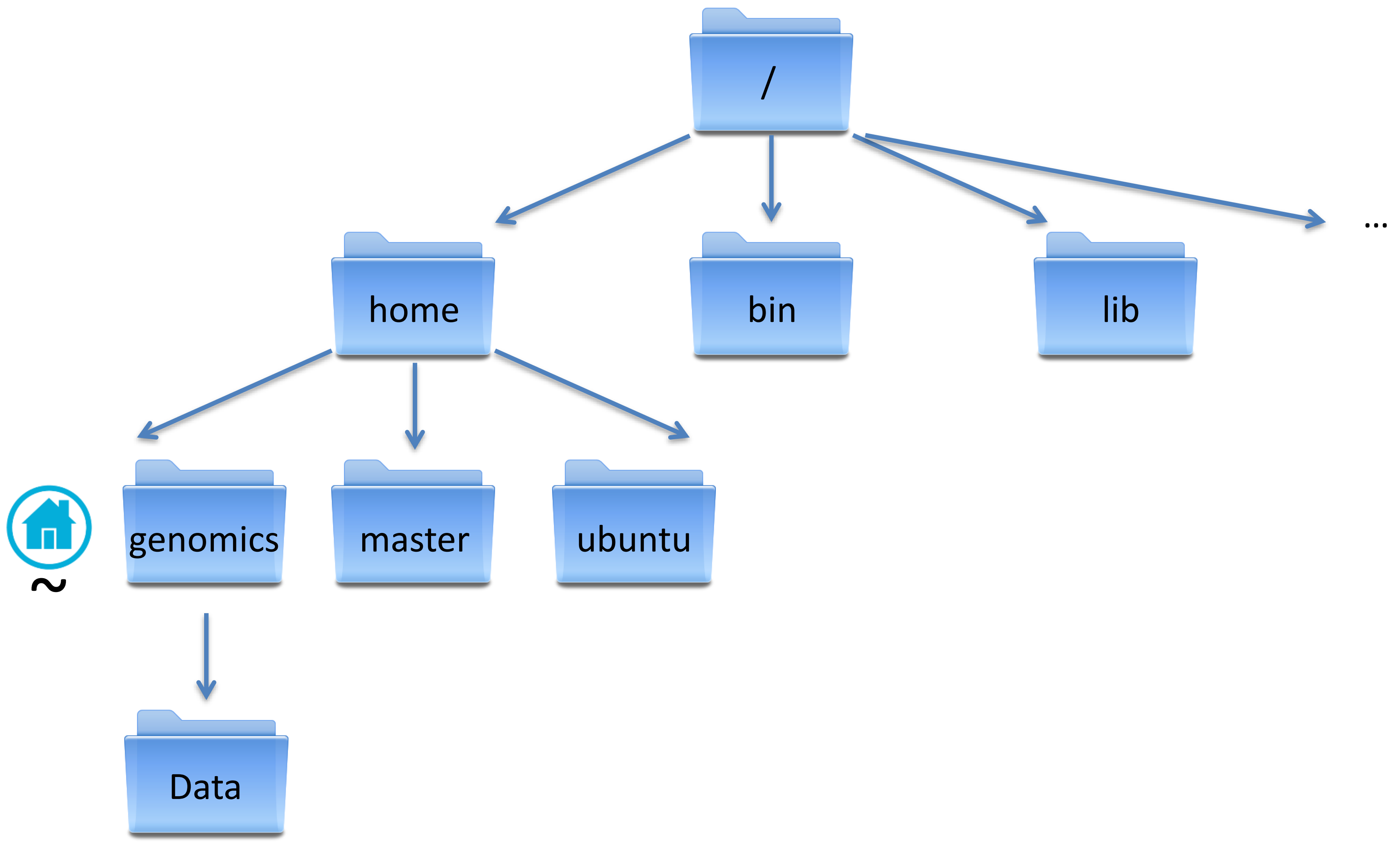
```
$ mkdir Data
```

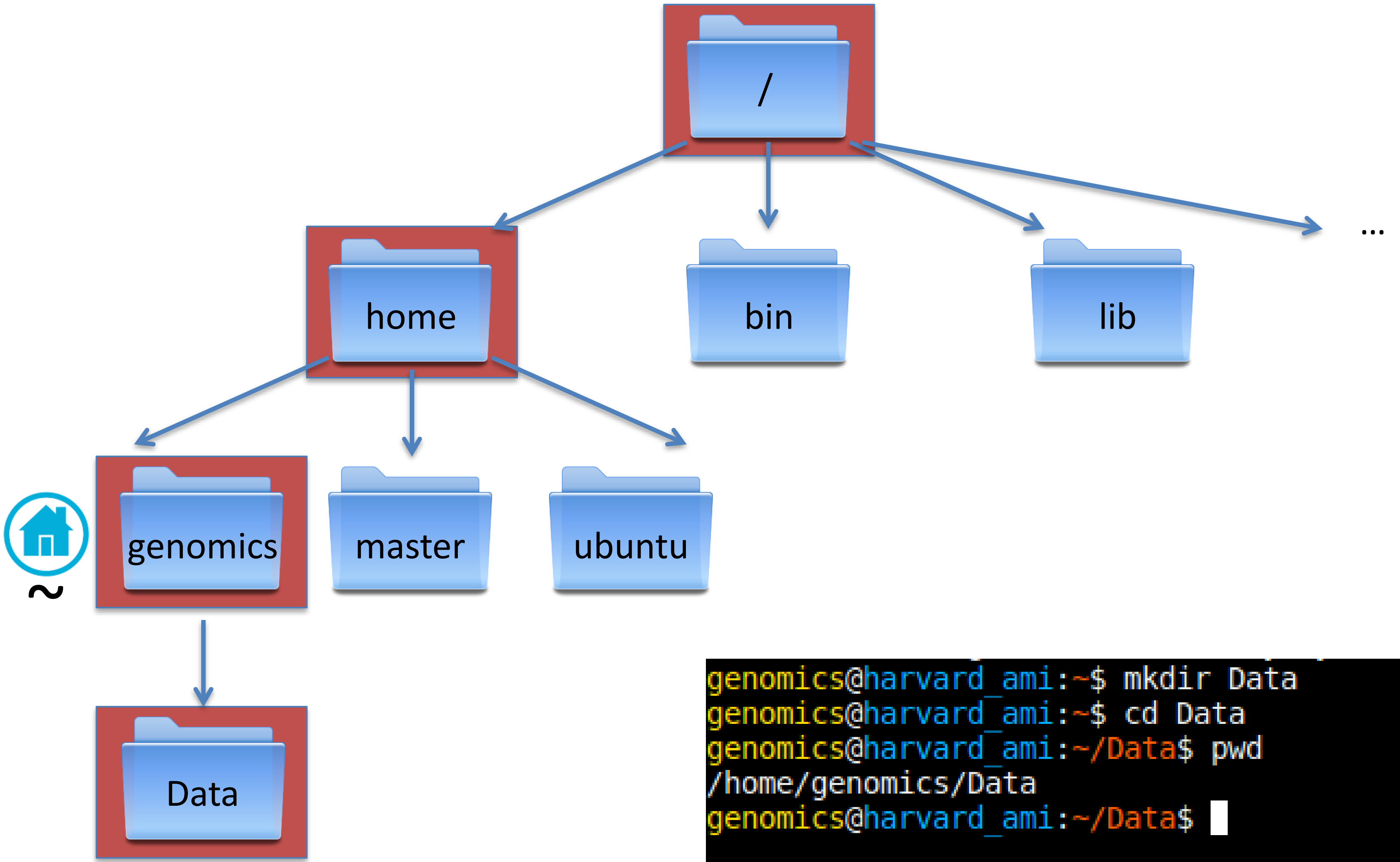
Change into this directory

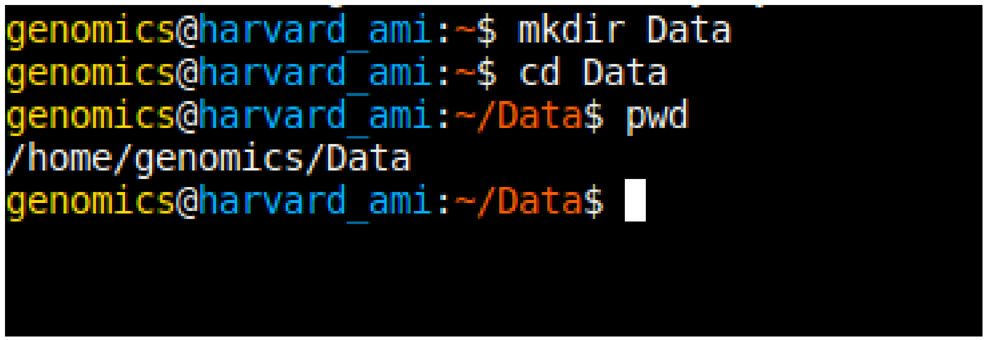
```
$ cd Data
```

Now what is your present working directory?

NOTE! Directory names (and file names for the matter) can not contain spaces. Underscores are often used instead if you want to separate words.


/
home
bin
lib
...
genomics
master
ubuntu
~
Data





```
genomics@harvard_ami:~$ mkdir Data
genomics@harvard_ami:~$ cd Data
genomics@harvard_ami:~/Data$ pwd
/home/genomics/Data
genomics@harvard_ami:~/Data$
```

Now let's create some directories and files

Make an empty file

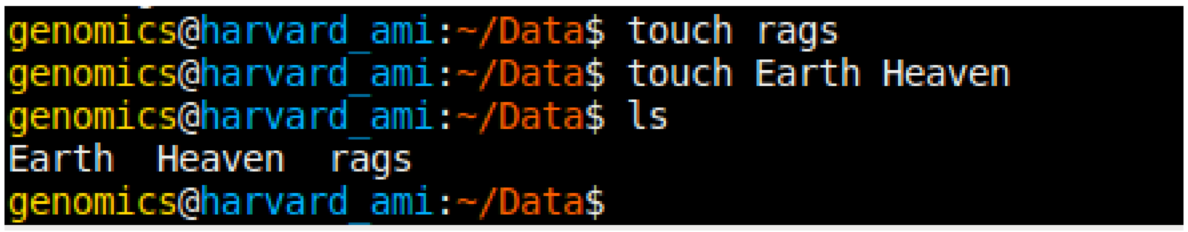
```
$ touch rags
```

And another two

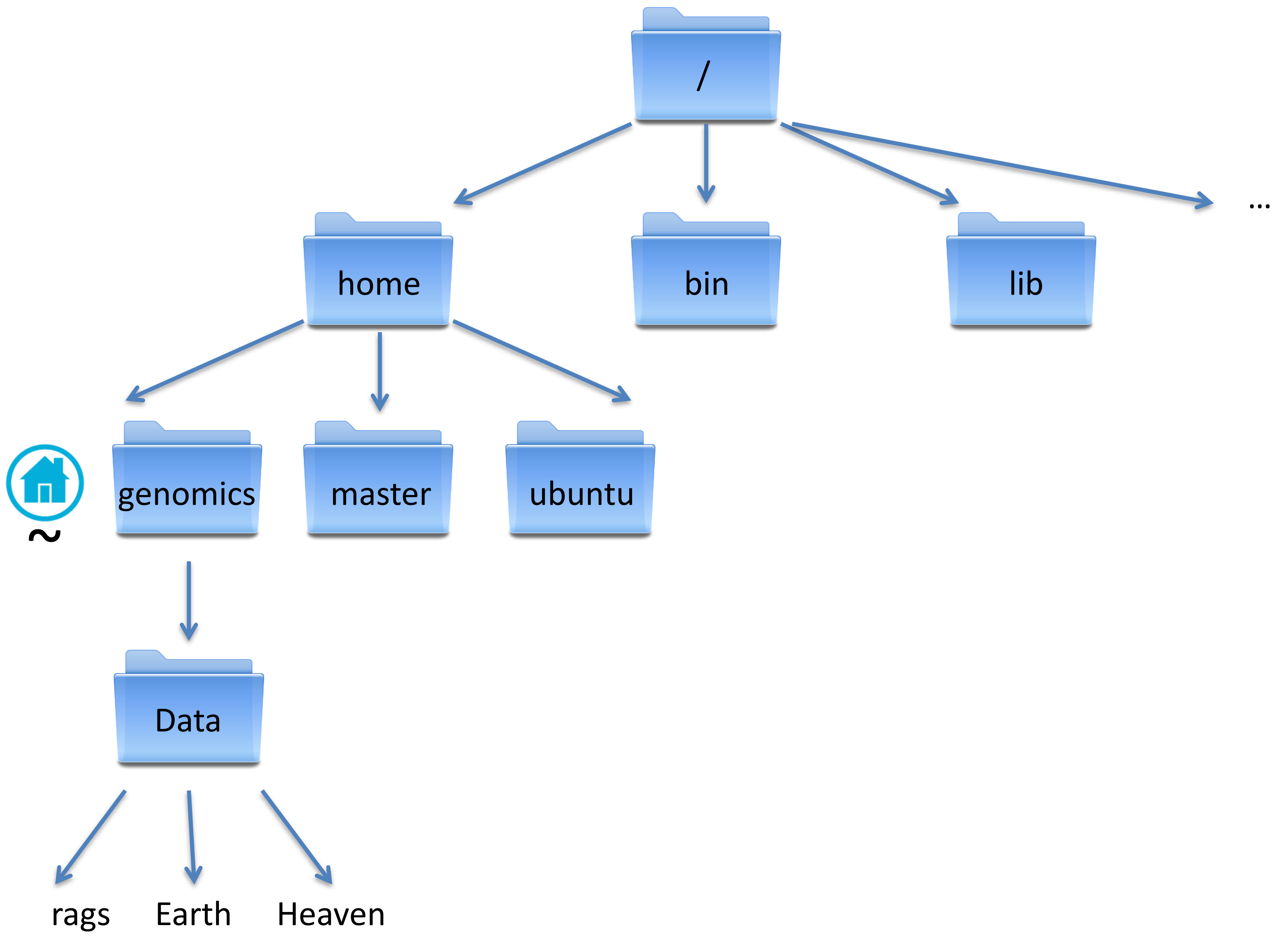
```
$ touch Earth Heaven
```

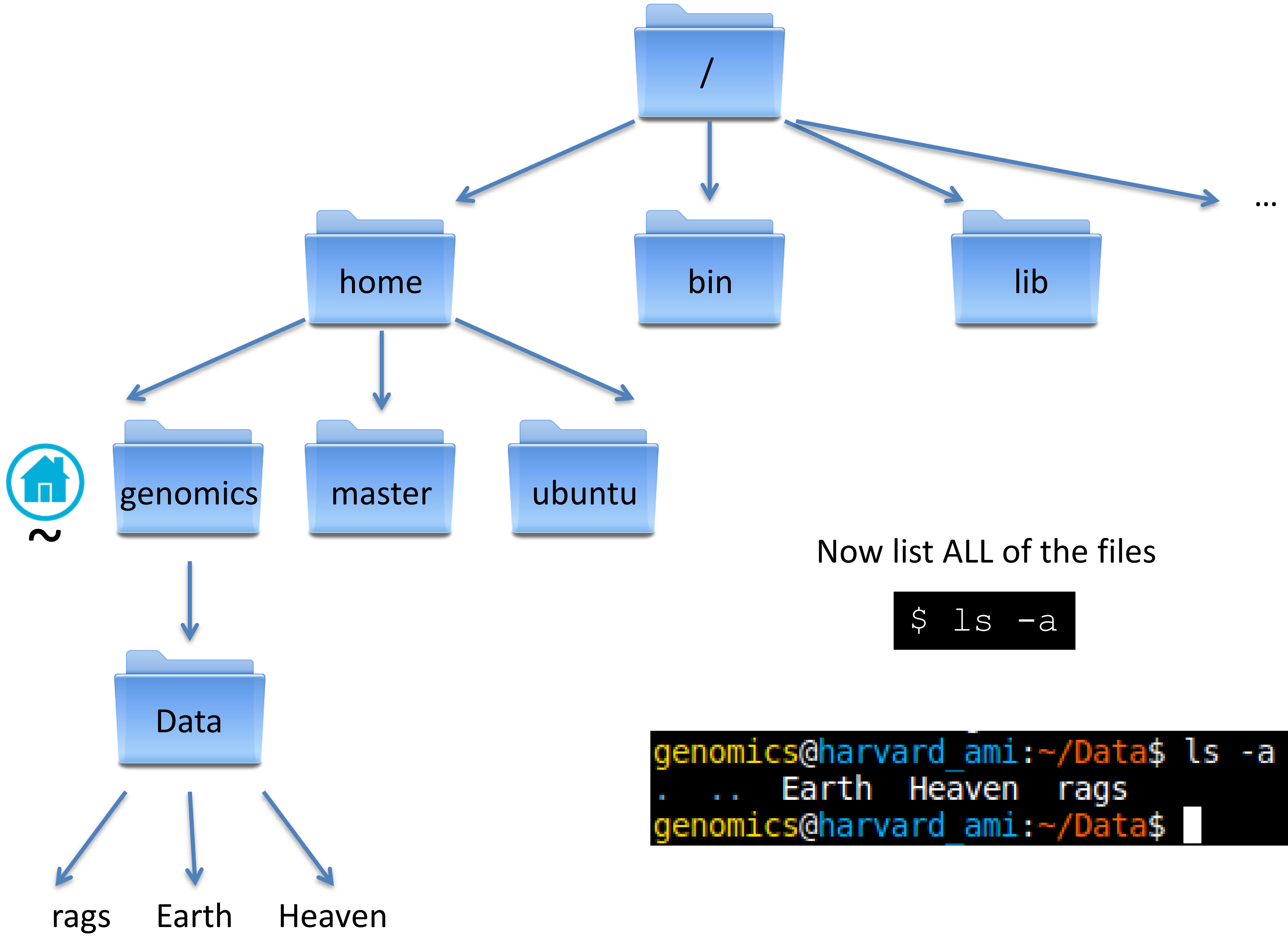
Now let's list the contents of the current directory (Data)

```
$ ls
```



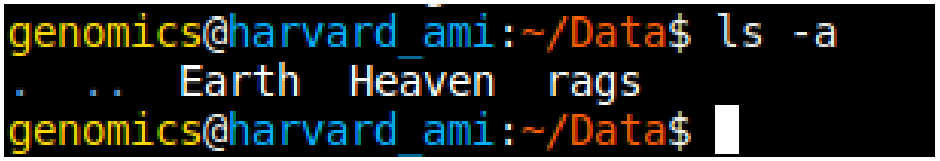
```
genomics@harvard_ami:~/Data$ touch rags
genomics@harvard_ami:~/Data$ touch Earth Heaven
genomics@harvard_ami:~/Data$ ls
Earth  Heaven  rags
genomics@harvard_ami:~/Data$
```


/
home
bin
lib
...
genomics
master
ubuntu
~
Data
rags
Earth
Heaven

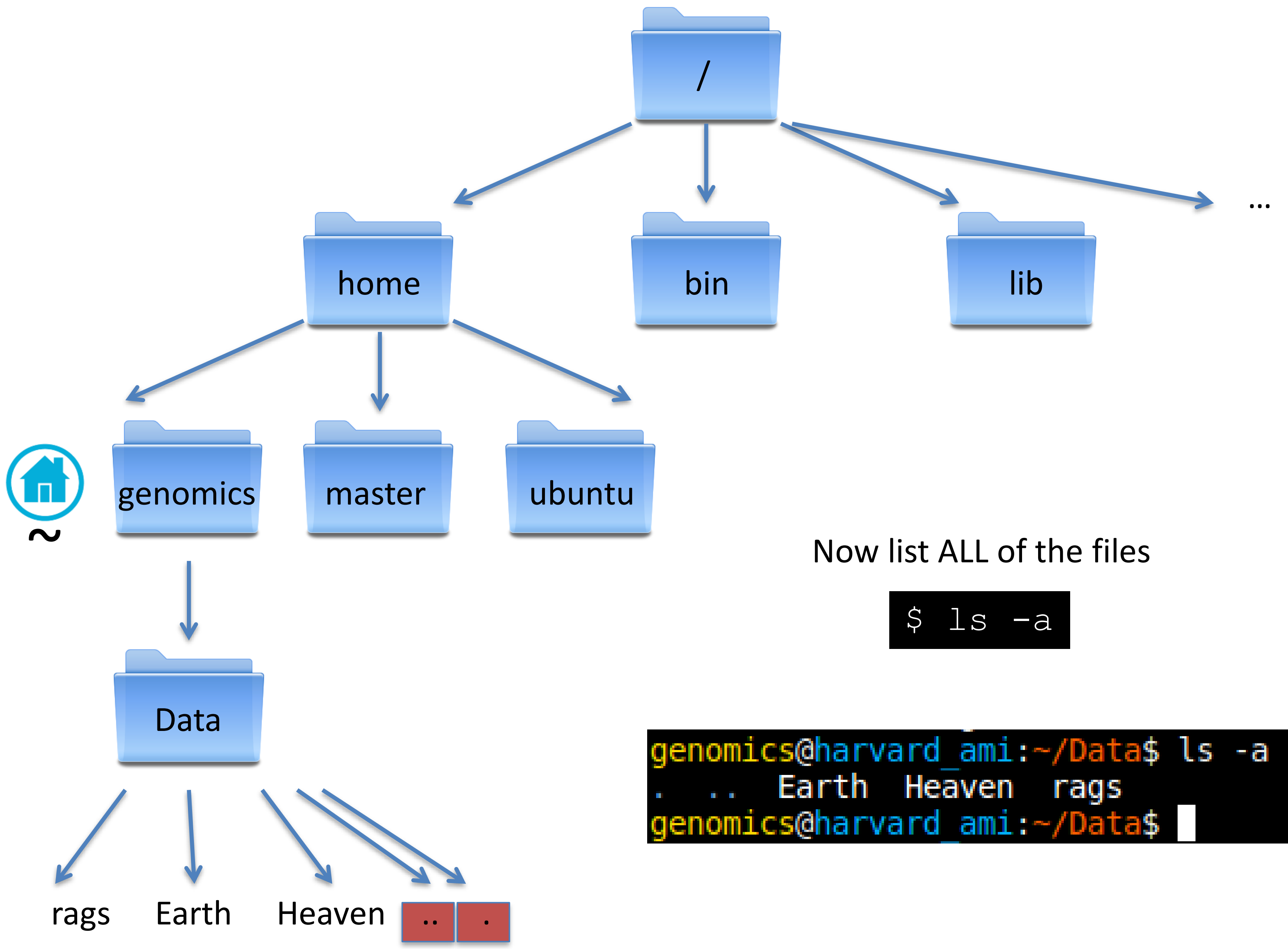


Now list ALL of the files

```
$ ls -a
```

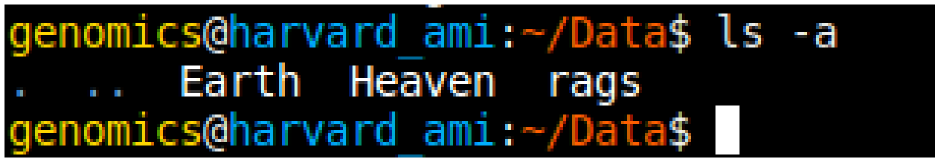


```
genomics@harvard_ami:~/Data$ ls -a
.   ..   Earth   Heaven   rags
genomics@harvard_ami:~/Data$ 
```

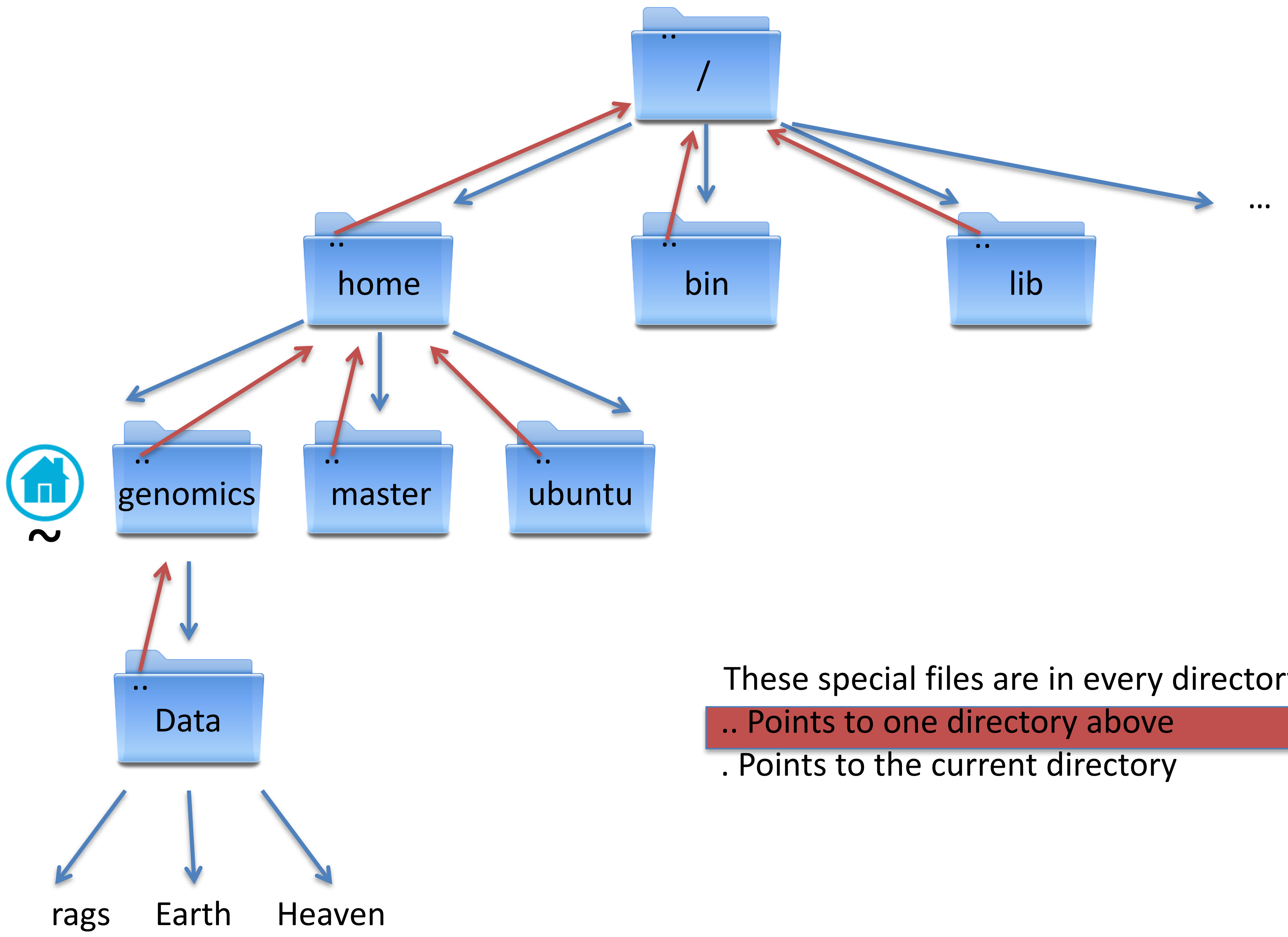


Now list ALL of the files

```
$ ls -a
```



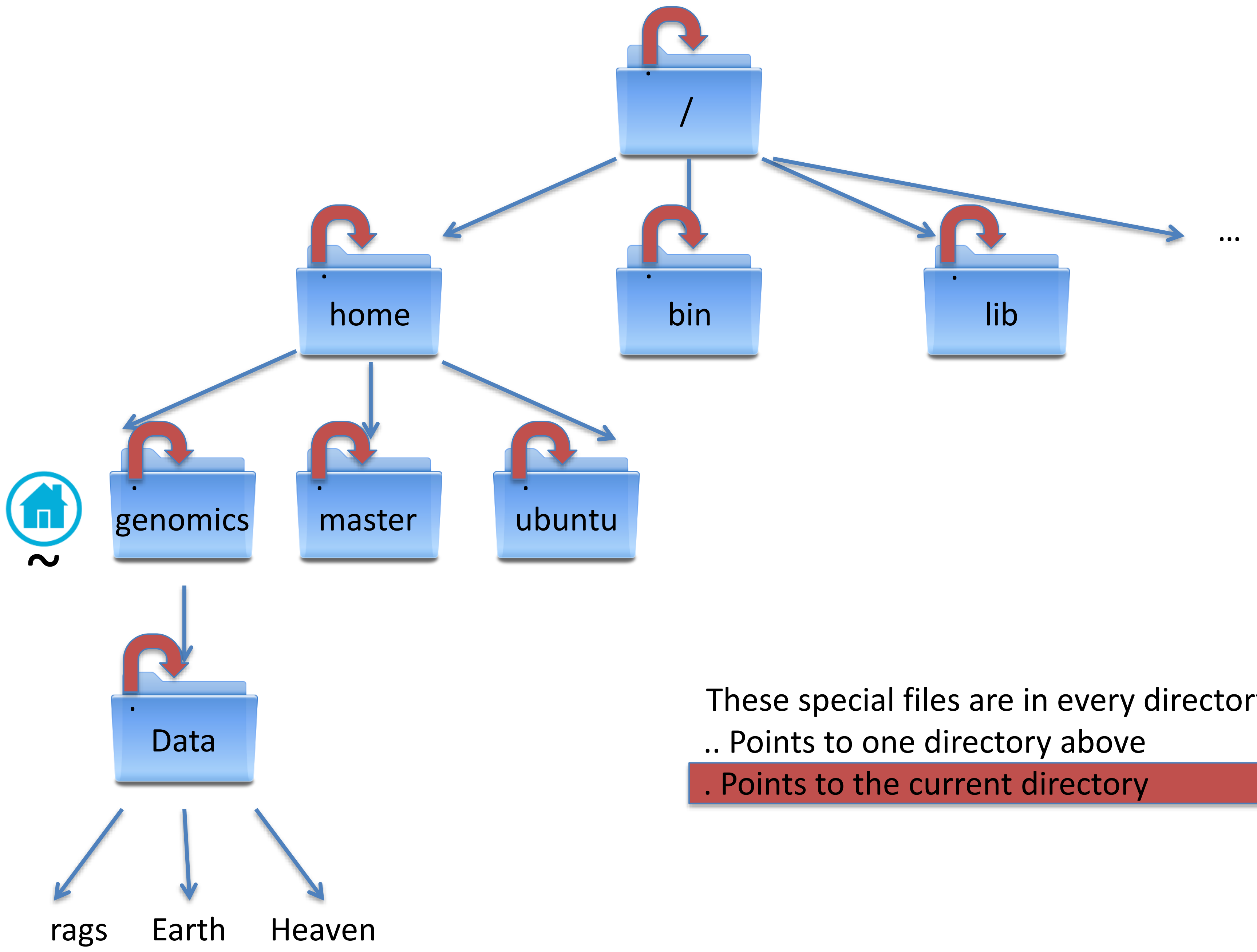
```
genomics@harvard_ami:~/Data$ ls -a
.   ..   Earth  Heaven  rags
genomics@harvard_ami:~/Data$
```



These special files are in every directory

.. Points to one directory above

. Points to the current directory



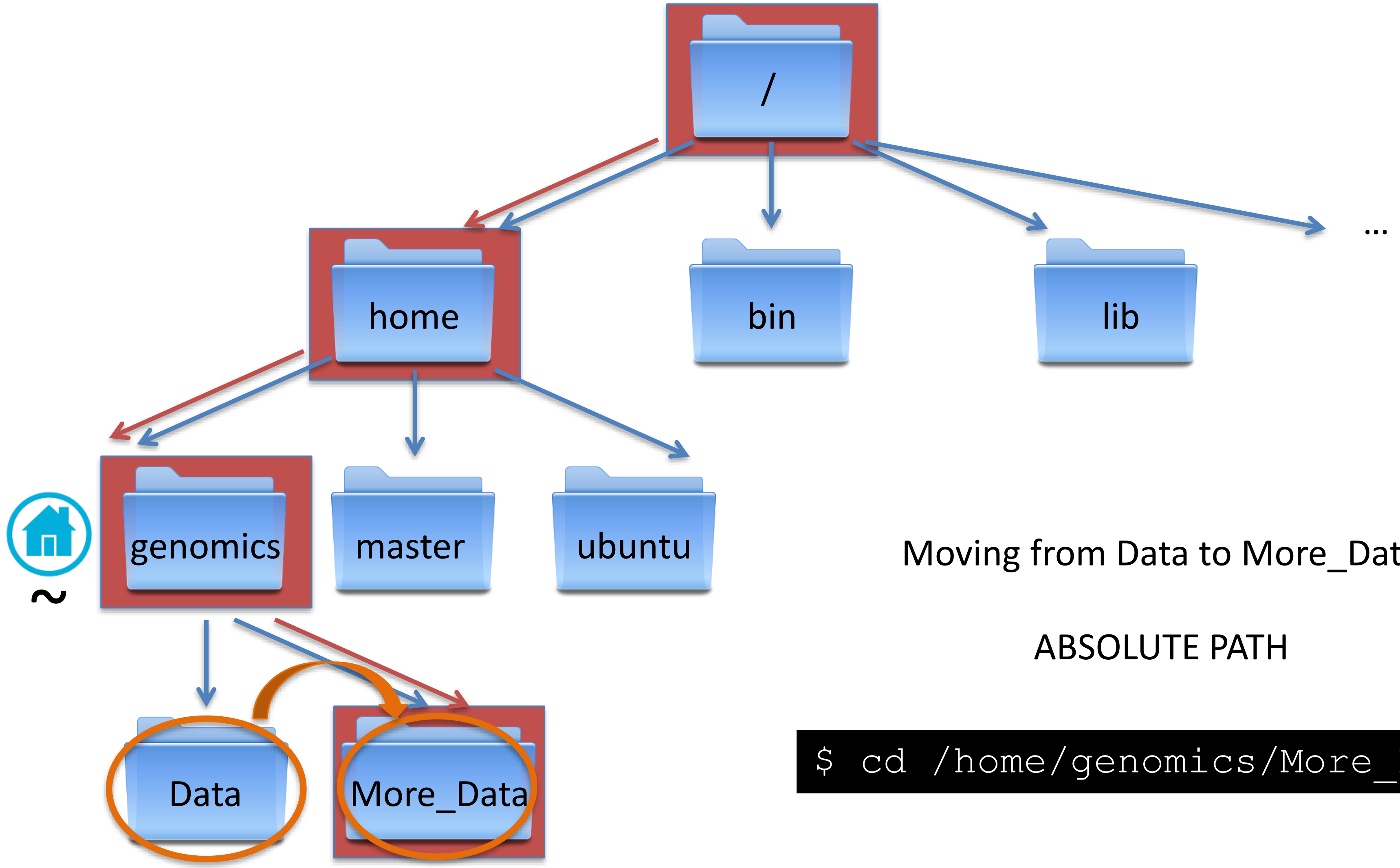
These special files are in every directory

.. Points to one directory above

. Points to the current directory

ABSOLUTE AND RELATIVE PATHS: GETTING FROM ONE PLACE TO ANOTHER

Absolute and Relative Paths

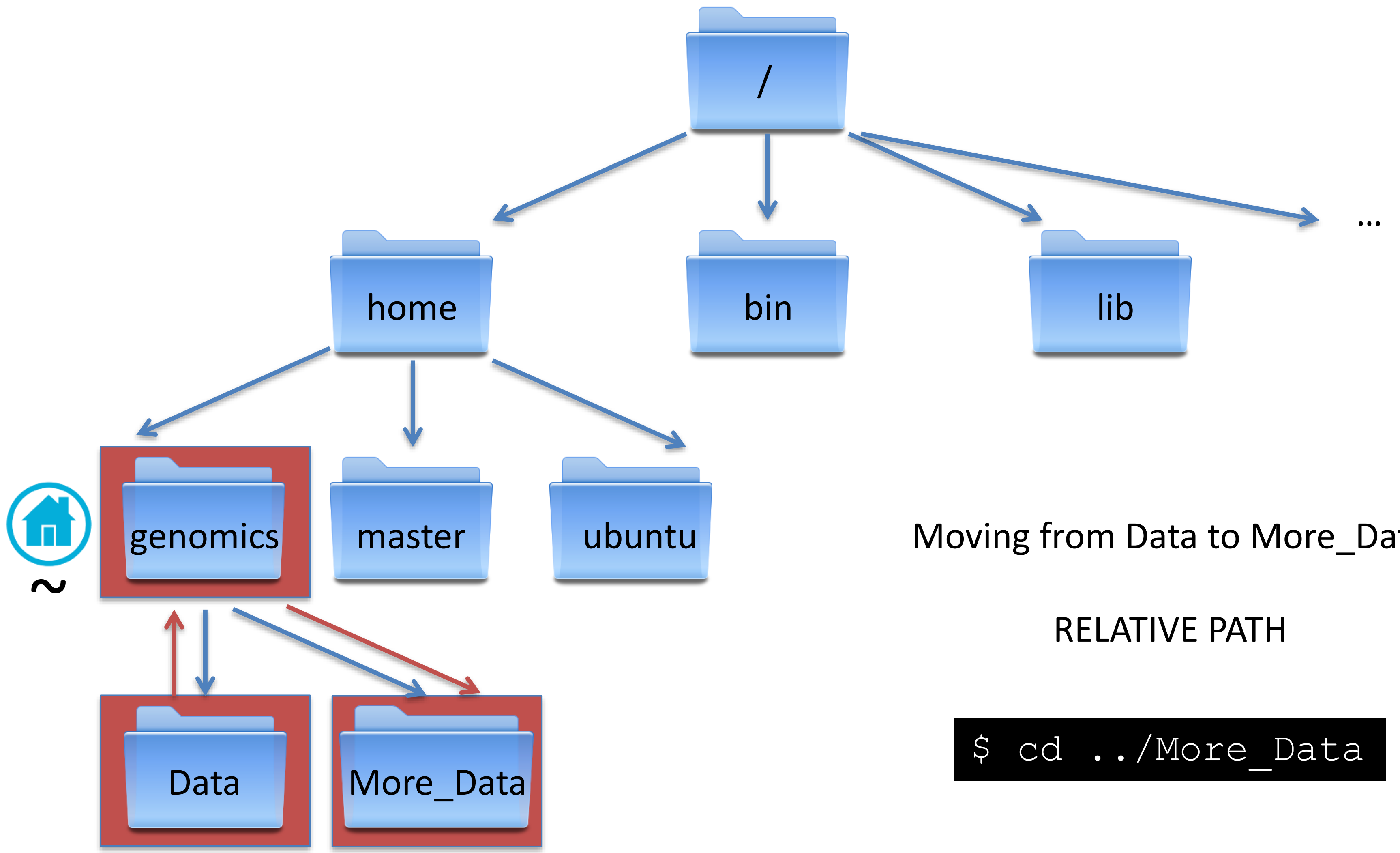


Moving from Data to More_Data

ABSOLUTE PATH

```
$ cd /home/genomics/More_Data
```

Absolute and Relative Paths

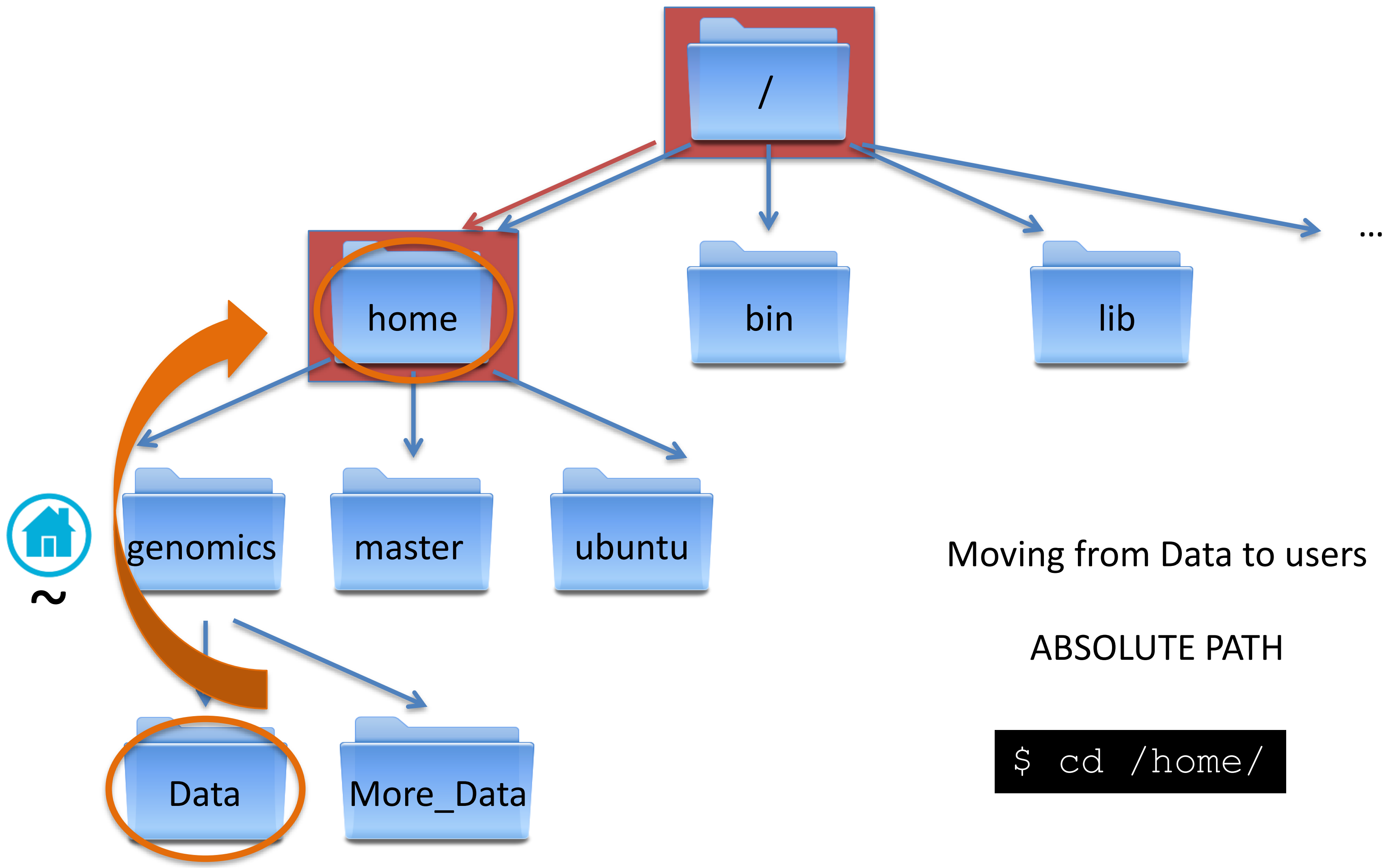


Moving from Data to More_Data

RELATIVE PATH

```
$ cd ../More_Data
```

Absolute and Relative Paths

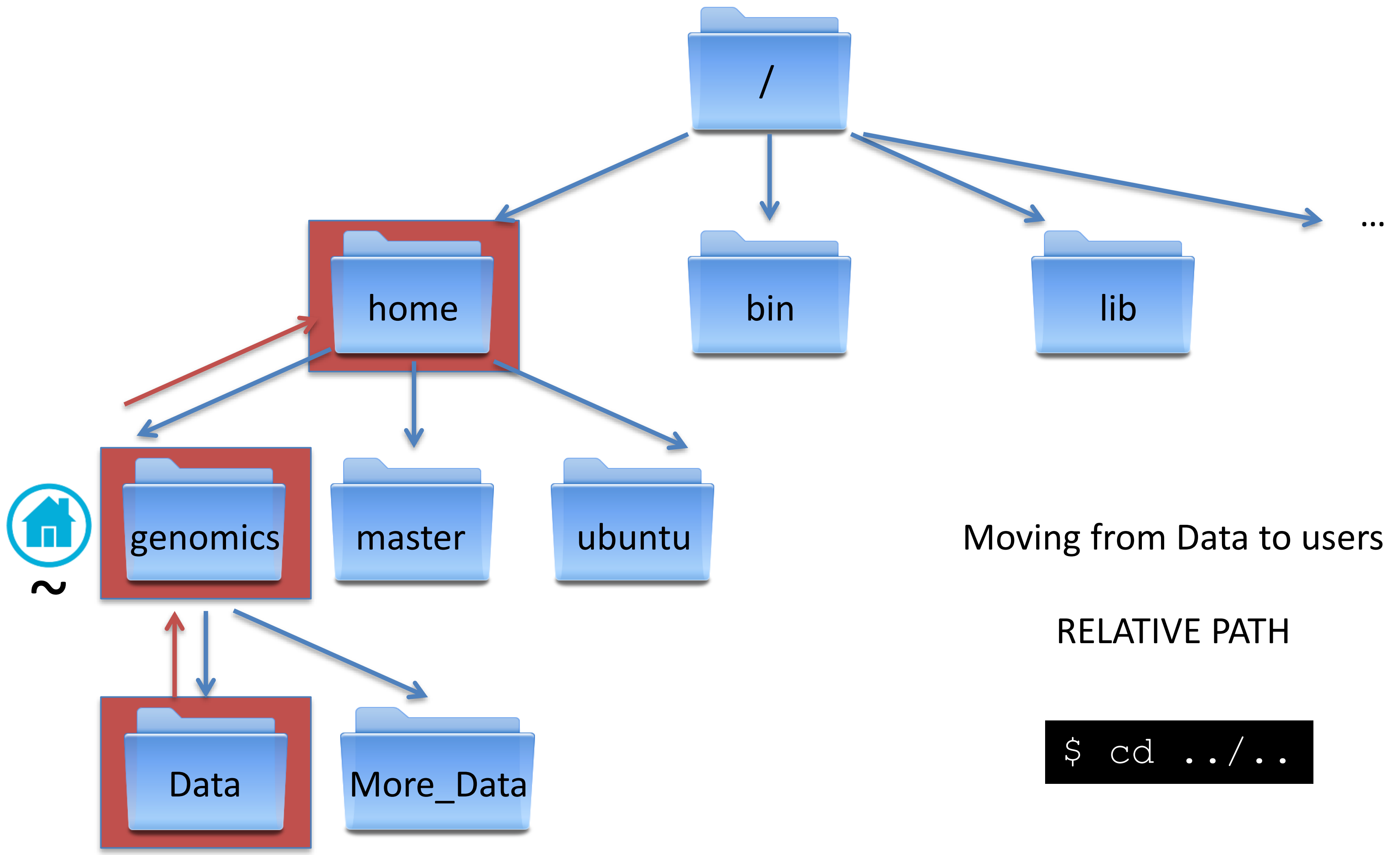


Moving from Data to users

ABSOLUTE PATH

```
$ cd /home/
```

Absolute and Relative Paths



Moving from Data to users

RELATIVE PATH

```
$ cd ../..
```

Let’s put this to practice

Where am I right now? (Should be the Data directory)

```
$ pwd
```

Change to the directory above

```
$ cd ..
```

Let’s list the contents of the Data directory

```
$ ls ./Data
```

Let’s put this to practice

Where am I right now? (Should be the Data directory)

```
$ pwd
```

Change to the directory above

```
$ cd ..
```

Let’s list the contents of the Data directory

```
$ ls ./Data
```

CHALLENGE 1!

1. Move into the Data directory and list the contents of your home directory
2. In Data, make a new directory and move into this location
3. From this new directory, move into your home directory IN ONE COMMAND and check your location

Challenge 1!

1. Move into the Data directory and list the contents of your home directory

```
$ cd Data
$ ls ..
```

OR

```
$ ls /home/genomics
```

OR

```
$ ls ~
```

2. In Data, make a new directory and move into this location

```
$ mkdir new
$ cd new
```

3. From this new directory, move into your home directory IN ONE COMMAND and check your location

```
$ cd ../..
```

OR

```
$ cd /home/genomics
```

OR

```
$ cd ~
```

OR

```
$ cd
```

```
$ pwd
```

If You’re Typing, You’re Doing Something Wrong!

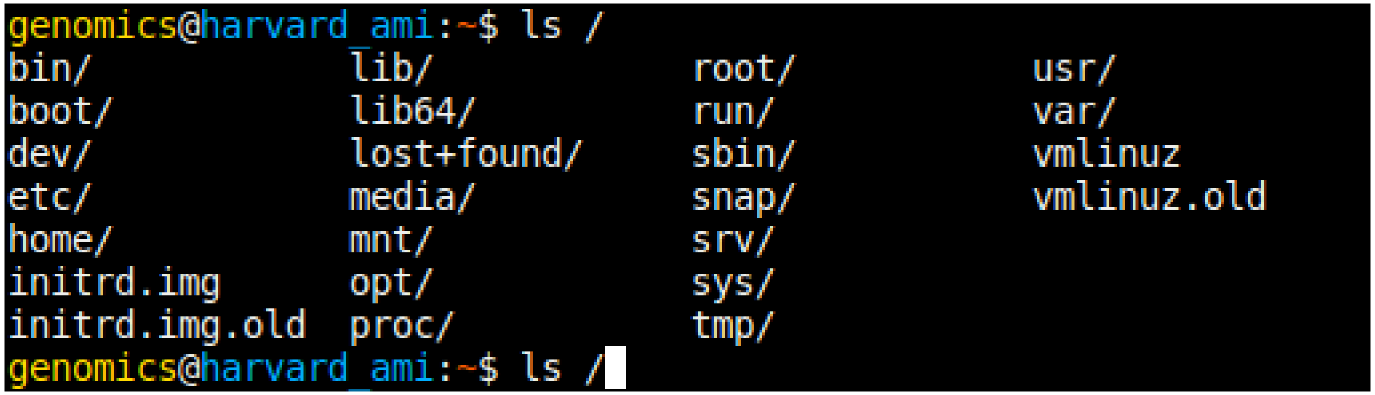
Tab complete is a nice trick to save you typing paths

For this examples we are going to list everything in directory /var/run

Start by typing:

```
$ ls /
```

Followed by tab twice quickly



```
genomics@harvard_ami:~$ ls /
bin/            lib/            root/           usr/
boot/           lib64/          run/            var/
dev/            lost+found/     sbin/           vmlinuz
etc/            media/          snap/           vmlinuz.old
home/           mnt/            srv/
initrd.img      opt/            sys/
initrd.img.old  proc/           tmp/
genomics@harvard_ami:~$ ls /
```

This shows the contents of the root directory

If You're Typing, You're Doing Something Wrong!

Now type:

```
$ ls /v
```

Followed by tab once. The path to the /var/ directory has filled in.

```
$ ls /var/
```

Now type:

```
$ ls /var/r
```

Followed by tab once. The path to the /var/run/ directory has filled in.

```
$ ls /var/run
```

Tab complete will fill in paths, save you time in typing and prevent typos!

If You're Typing, You're Doing Something Wrong!

Two more tricks for less typing!

* Represents a special character
For example:

```
$ ls /home/genomics/*.txt
```

Will list everything in my home directory ending .txt

The up arrow can be used to re-run commands

Press your up arrow and see

If you want all of these commands listed, simply type

```
$ history
```

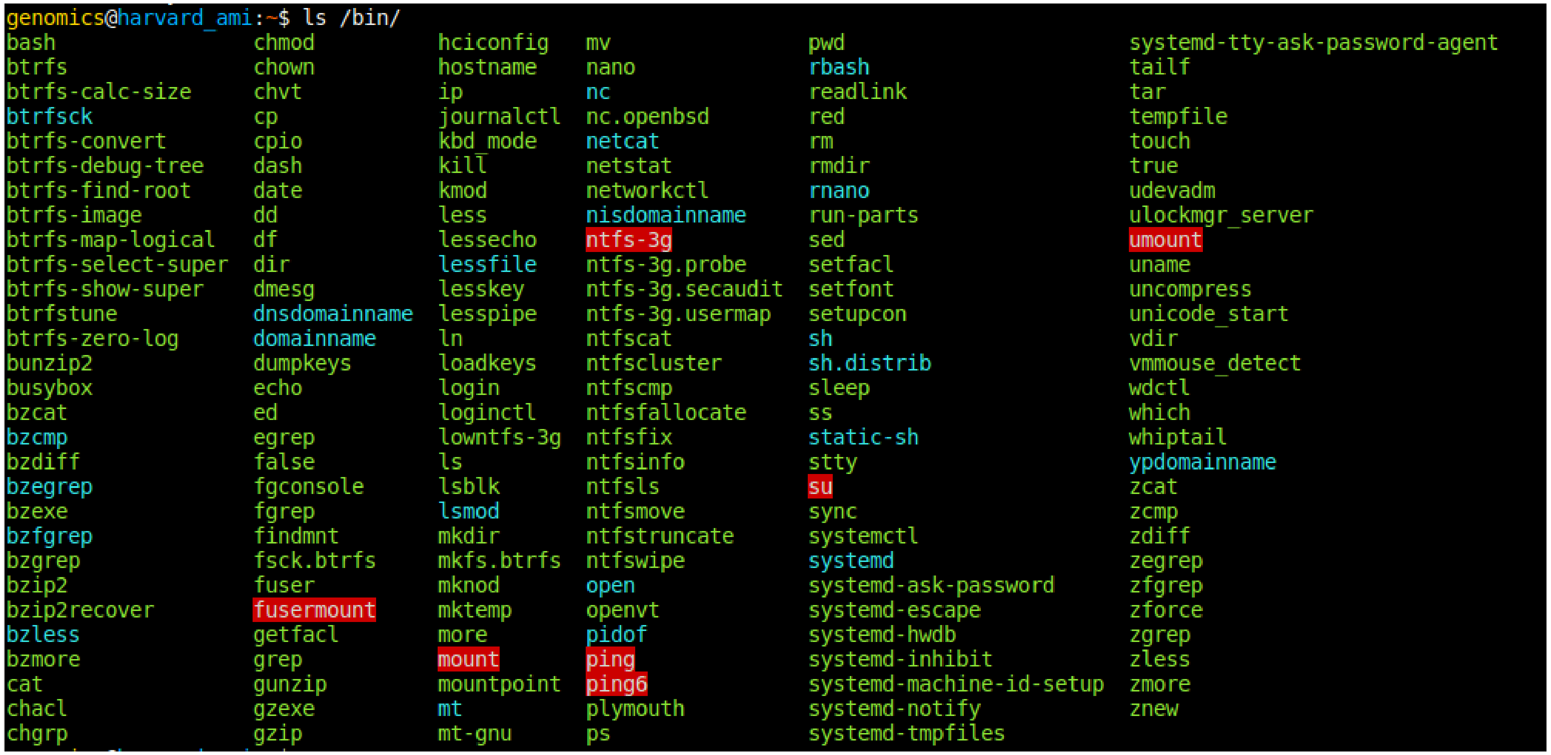
Any Questions So Far?

Binary programs

These are all programs installed on the Unix machine.

They can be found in /bin

```
$ ls /bin
```



```
genomics@harvard_ami:~$ ls /bin/
bash                chmod          hciconfig   mv                pwd                       systemd-tty-ask-password-agent
btrfs               chown          hostname    nano              rbash                     tailf
btrfs-calc-size     chvt           ip          nc                readlink                  tar
btrfsck             cp             journalctl  nc.openbsd        red                       tempfile
btrfs-convert       cpio           kbd_mode    netcat            rm                        touch
btrfs-debug-tree    dash           kill        netstat           rmdir                     true
btrfs-find-root     date           kmod        networkctl        rnano                     udevadm
btrfs-image         dd             less        nisdomainname     run-parts                 ulockmgr_server
btrfs-map-logical   df             lessecho    ntfs-3g           sed                       umount
btrfs-select-super  dir            lessfile    ntfs-3g.probe     setfacl                   uname
btrfs-show-super    dmesg          lesskey     ntfs-3g.secaudit  setfont                   uncompress
btrfstune           dnsdomainname  lesspipe    ntfs-3g.usermap   setupcon                  unicode_start
btrfs-zero-log      domainname     ln          ntfscat           sh                        vdir
bunzip2             dumpkeys       loadkeys    ntfscluster       sh.distrib                vmmouse_detect
busybox             echo           login       ntfscmp           sleep                     wdctl
bzcat               ed             loginctl    ntfsfallocate     ss                        which
bzcmp               egrep          lowntfs-3g  ntfsfix           static-sh                 whiptail
bzdiff              false          ls          ntfsinfo          stty                      ypdomainname
bzegrep             fgconsole      lsblk       ntfsls            su                        zcat
bzexe               fgrep          lsmod       ntfsmove          sync                      zcmp
bzfgrep             findmnt        mkdir       ntfstruncate      systemctl                 zdiff
bzgrep              fsck.btrfs     mkfs.btrfs  ntfswipe          systemd                   zegrep
bzip2               fuser          mknod       open              systemd-ask-password      zfgrep
bzip2recover        fusermount     mktemp      openvt            systemd-escape            zforce
bzless              getfacl        more        pidof             systemd-hwdb              zgrep
bzmore              grep           mount       ping              systemd-inhibit           zless
cat                 gunzip         mountpoint  ping6             systemd-machine-id-setup  zmore
chacl               gzexe          mt          plymouth          systemd-notify            znew
chgrp               gzip           mt-gnu      ps                systemd-tmpfiles
```

These include pwd, mkdir, ls …

Every binary program has a manual

To view the manual page, type man followed by the name of the program

```
$ man <PROGRAMME>
```

Open the manual page for ls

Scroll through (enter) and find the options for: long listing format, human-readable sizes and sort by modification time

Exit the manual page (type q) and give these ls options a go in your Data directory

Every binary program has a manual

To view the manual page, type man followed by the name of the program

```
$ man <PROGRAMME>
```

Open the manual page for ls

```
$ man ls
```

Scroll through (enter) and find the options for: long listing format (-l), human-readable Sizes (-h) and sort by modification time (-t)

Exit the manual page (type q) and give these ls options a go in your Data directory

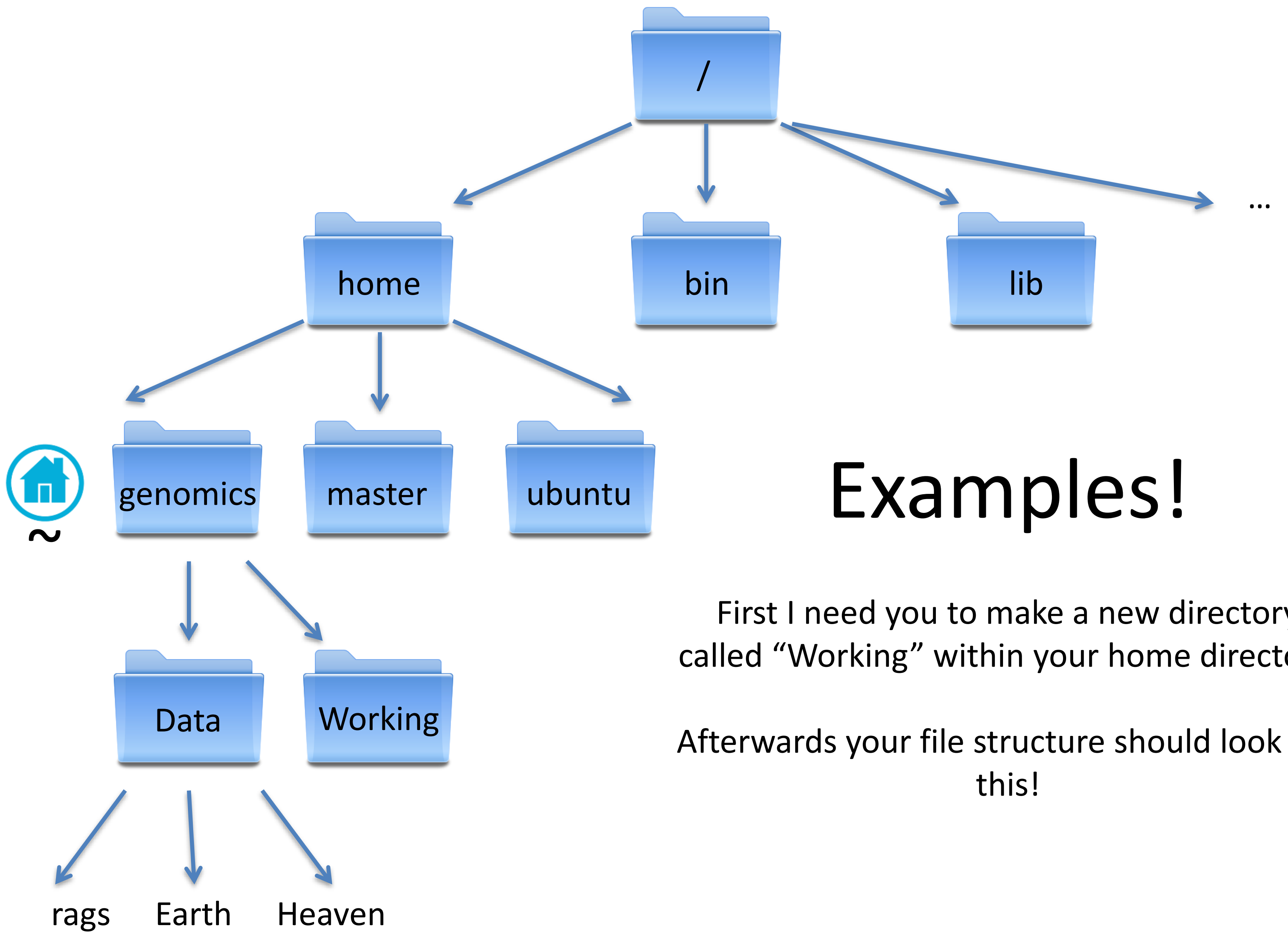
```
$ ls -l
```

OR

```
$ ls -h
```

OR

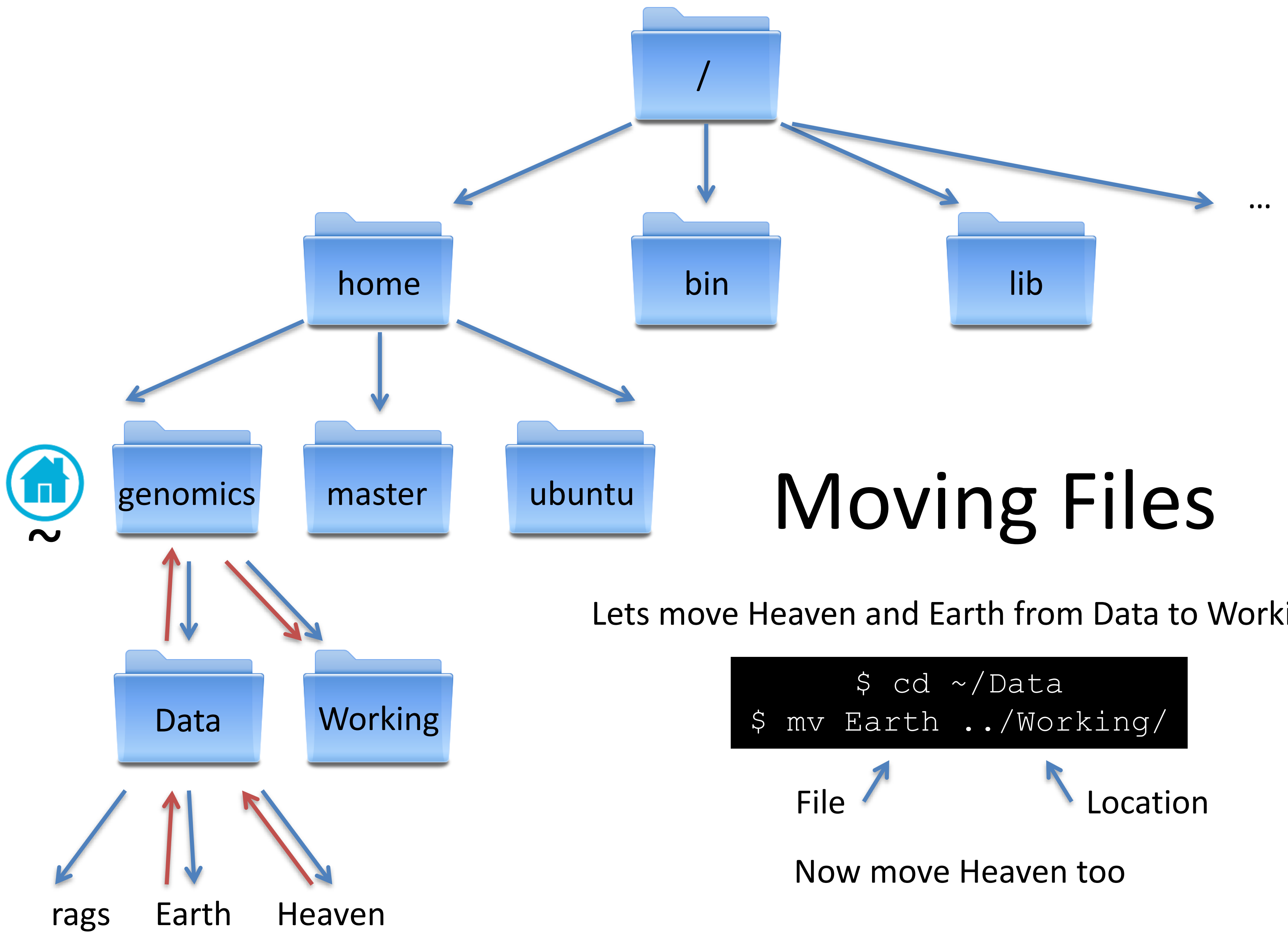
```
$ ls -t
```



Examples!

First I need you to make a new directory called “Working” within your home directory.

Afterwards your file structure should look like this!



Moving Files

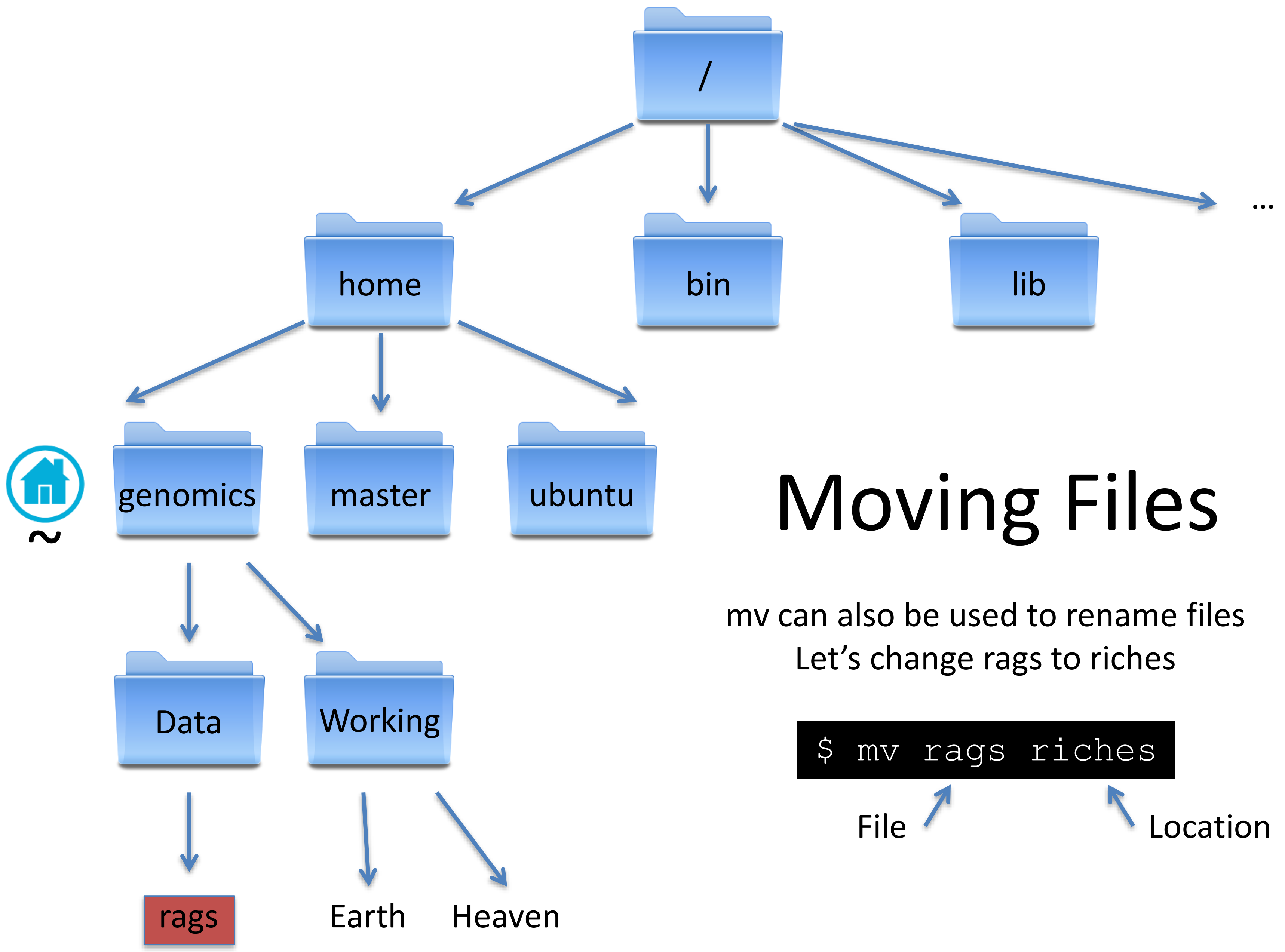
Lets move Heaven and Earth from Data to Working

```
$ cd ~/Data
$ mv Earth ../Working/
```

File ↗ ↖ Location

Now move Heaven too

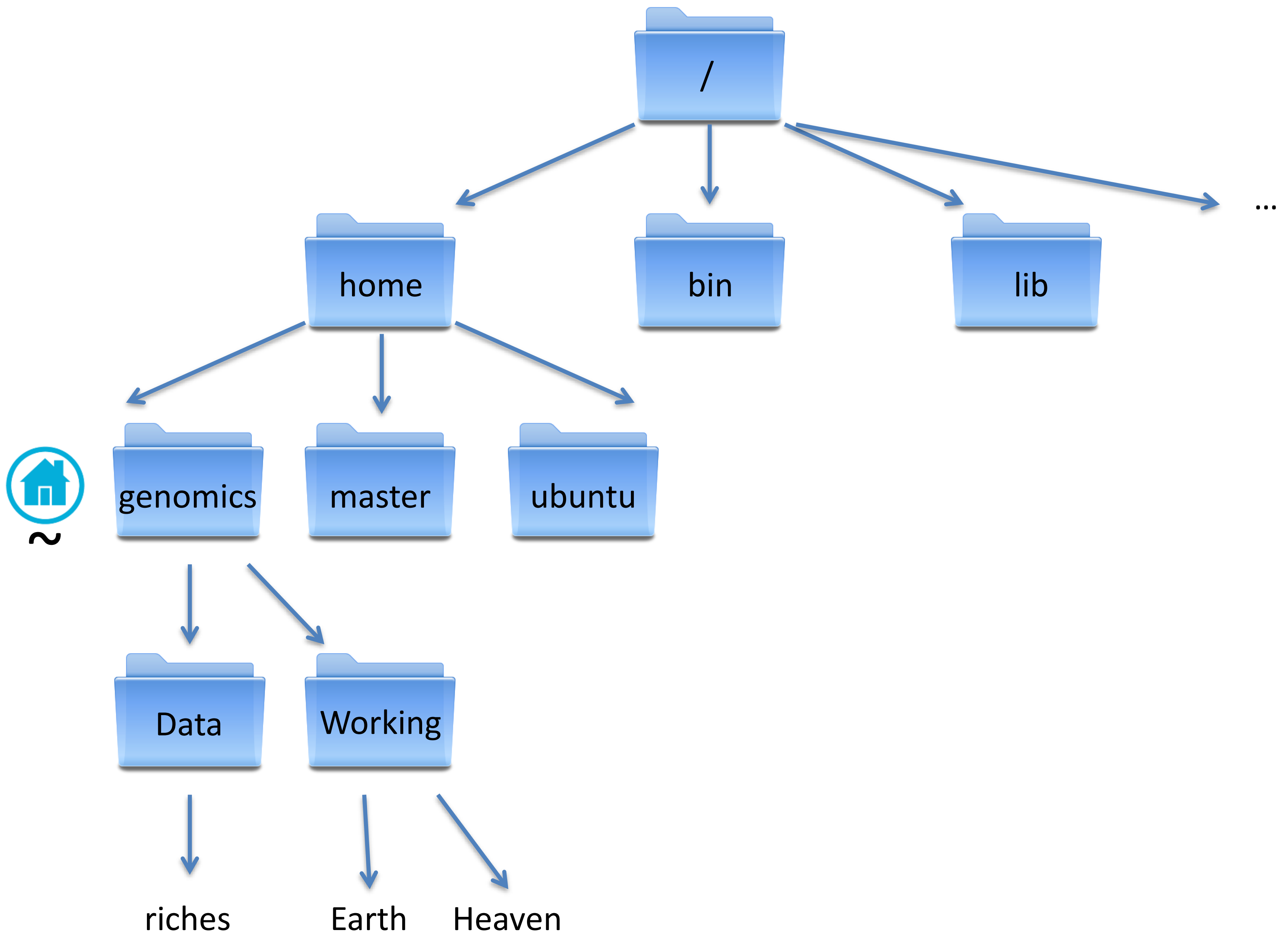
REMEMBER TAB COMPLETE!

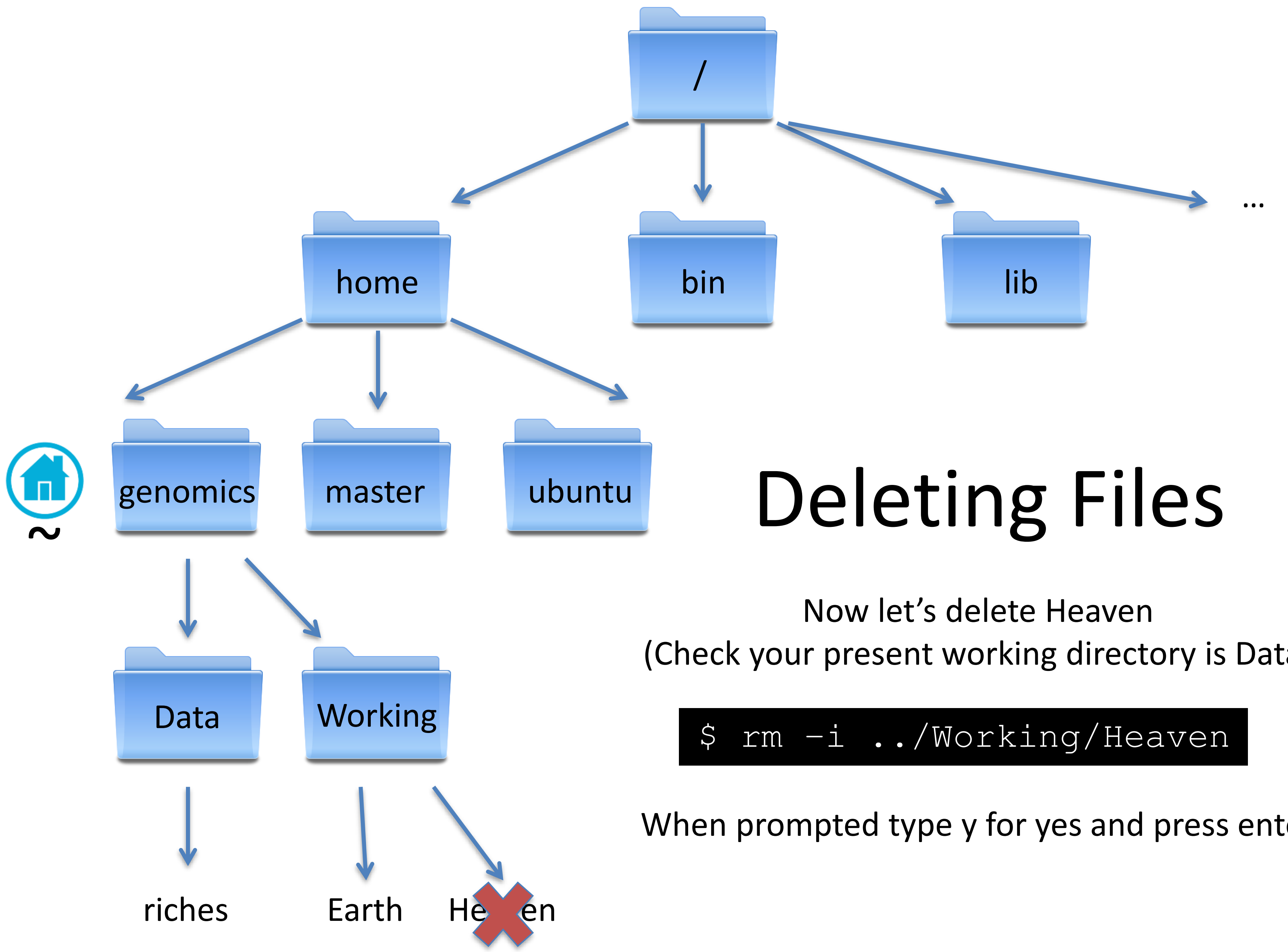


Moving Files

mv can also be used to rename files
Let's change rags to riches

```
$ mv rags riches
```


/
home
bin
lib
...
genomics
master
ubuntu
~
Data
Working
riches
Earth
Heaven

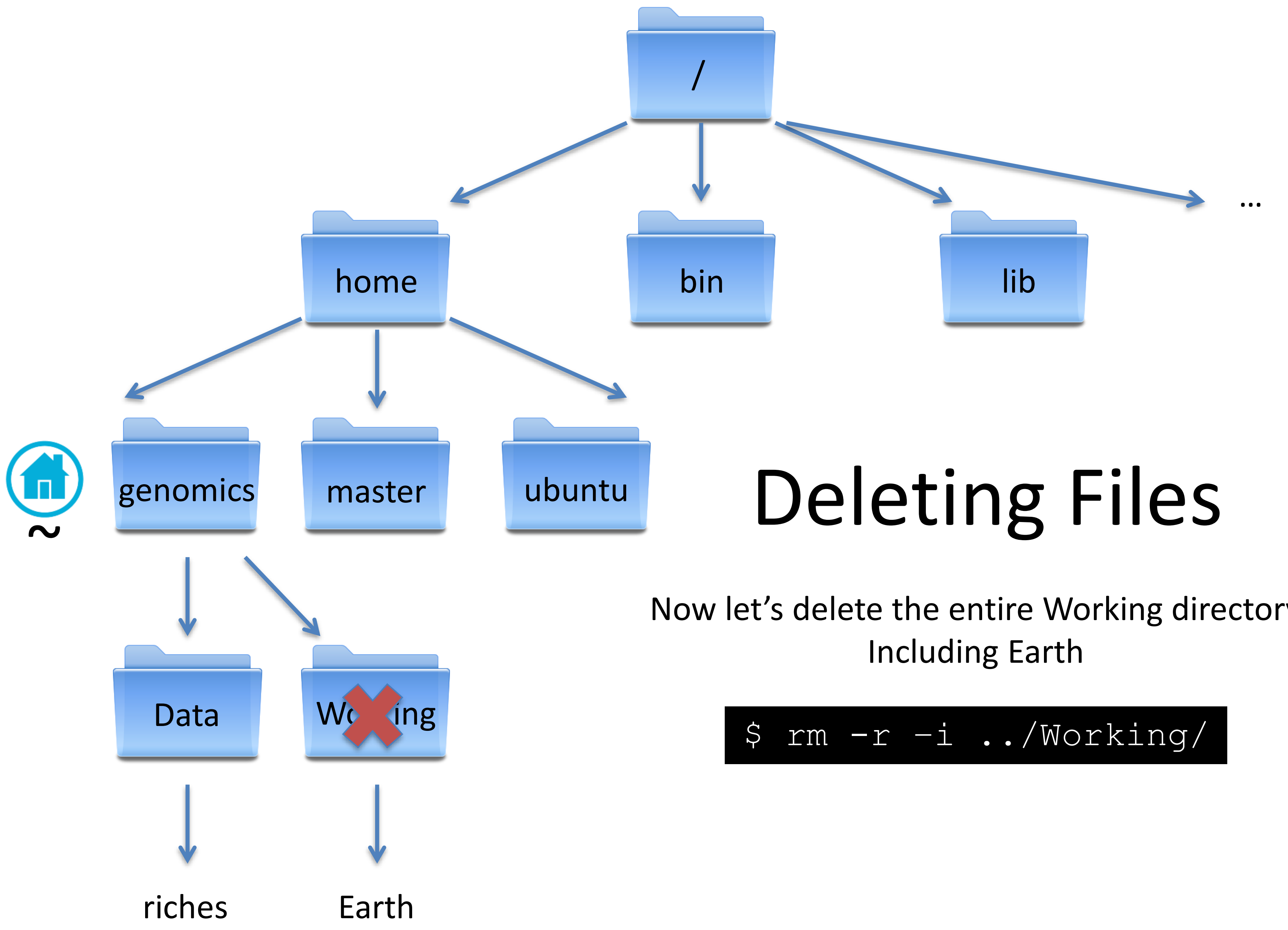


Deleting Files

Now let’s delete Heaven
(Check your present working directory is Data)

```
$ rm -i ../Working/Heaven
```

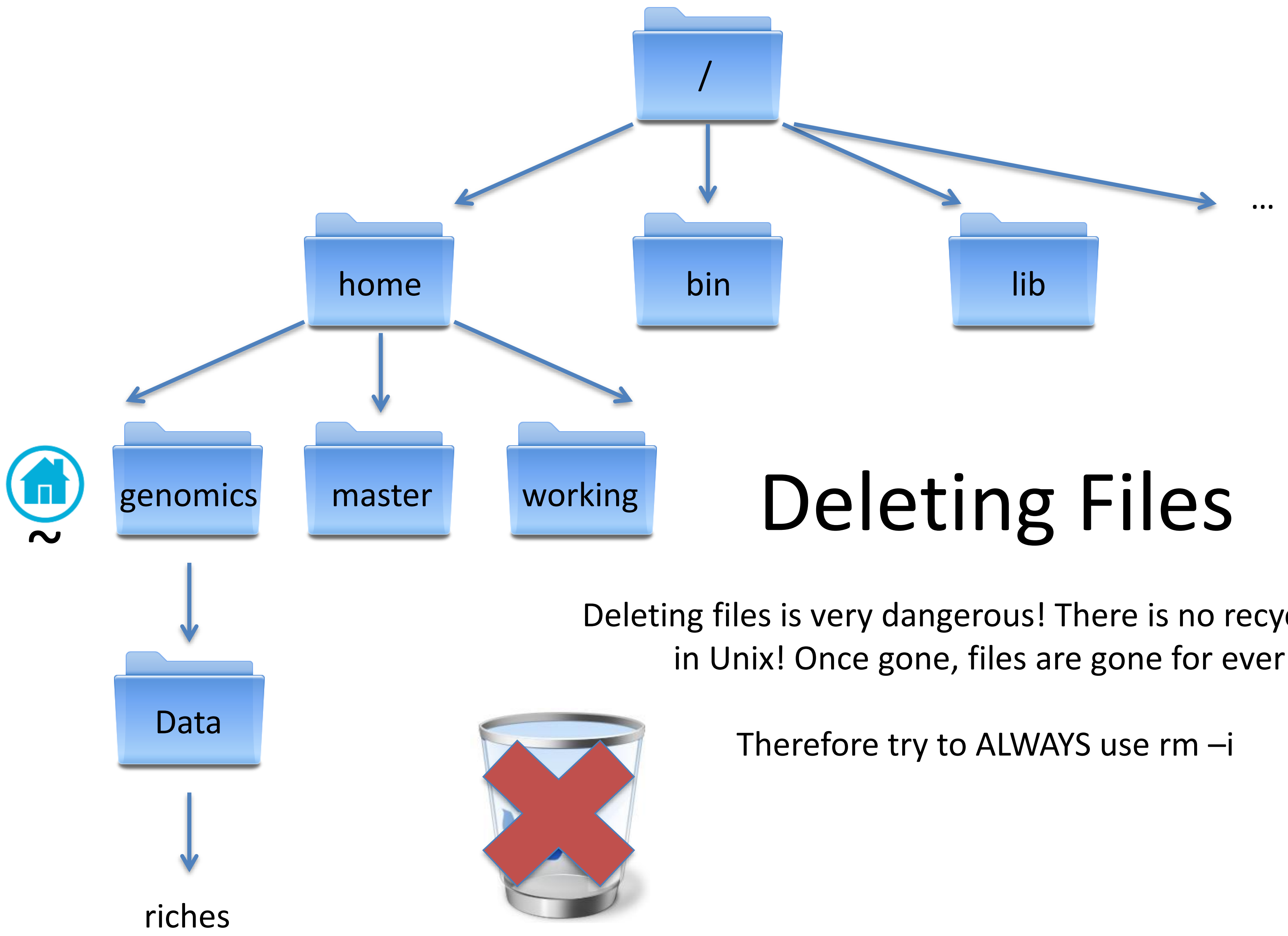
When prompted type y for yes and press enter



Deleting Files

Now let’s delete the entire Working directory
Including Earth

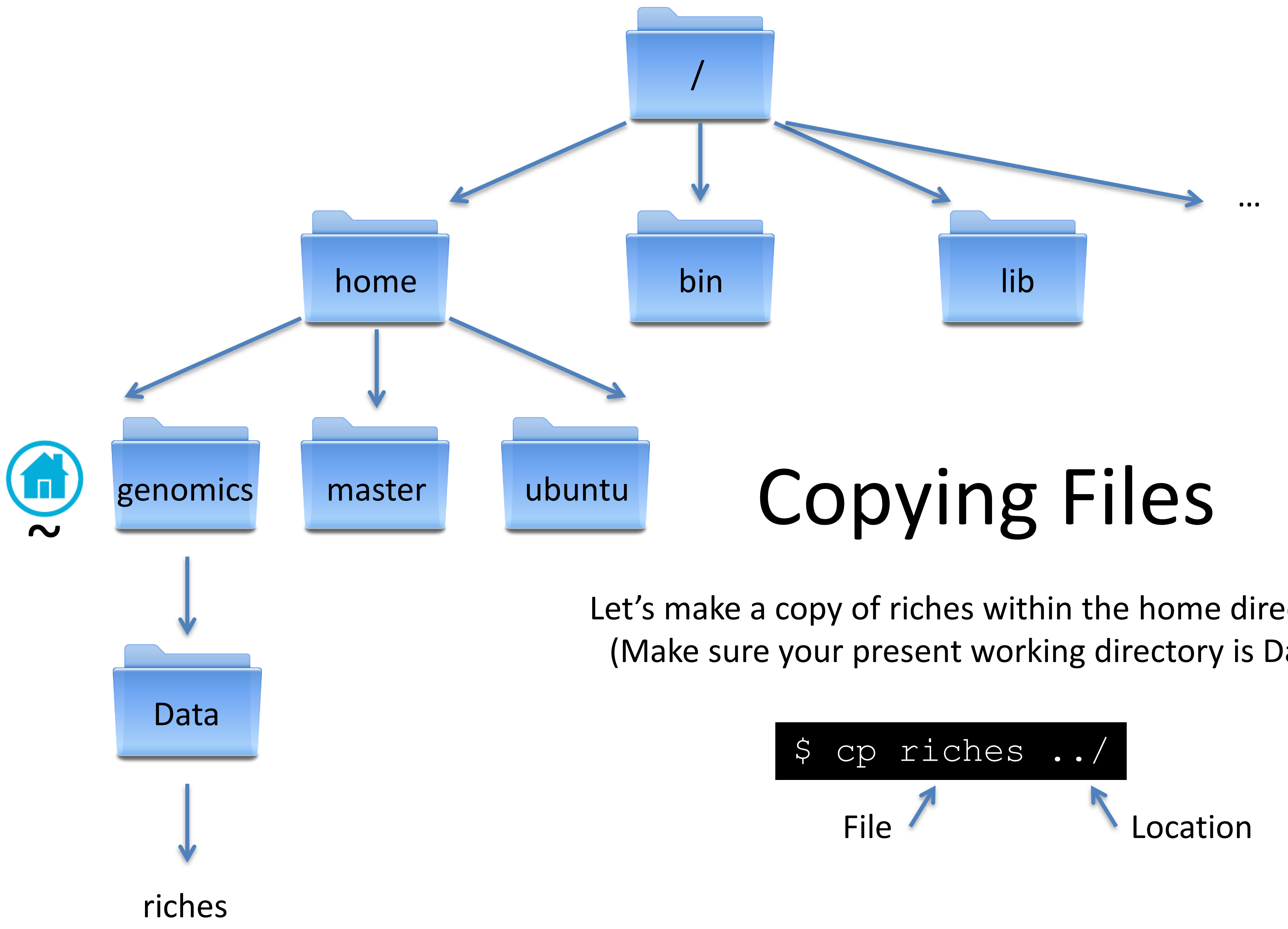
```
$ rm -r -i ../Working/
```



Deleting Files

Deleting files is very dangerous! There is no recycle bin in Unix! Once gone, files are gone for ever!

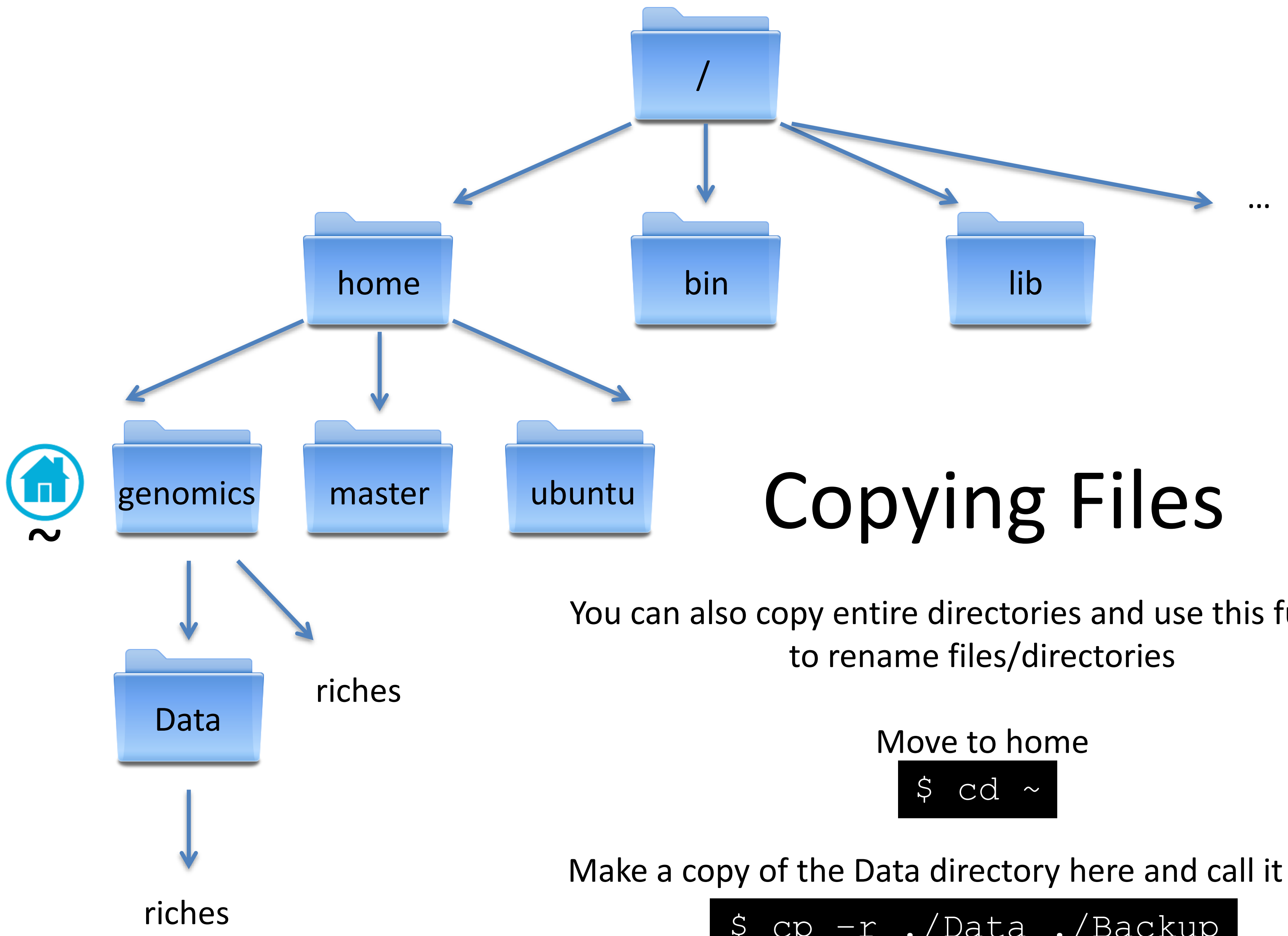
Therefore try to ALWAYS use rm –i



Copying Files

Let’s make a copy of riches within the home directory
(Make sure your present working directory is Data)

```
$ cp riches ../
```



Copying Files

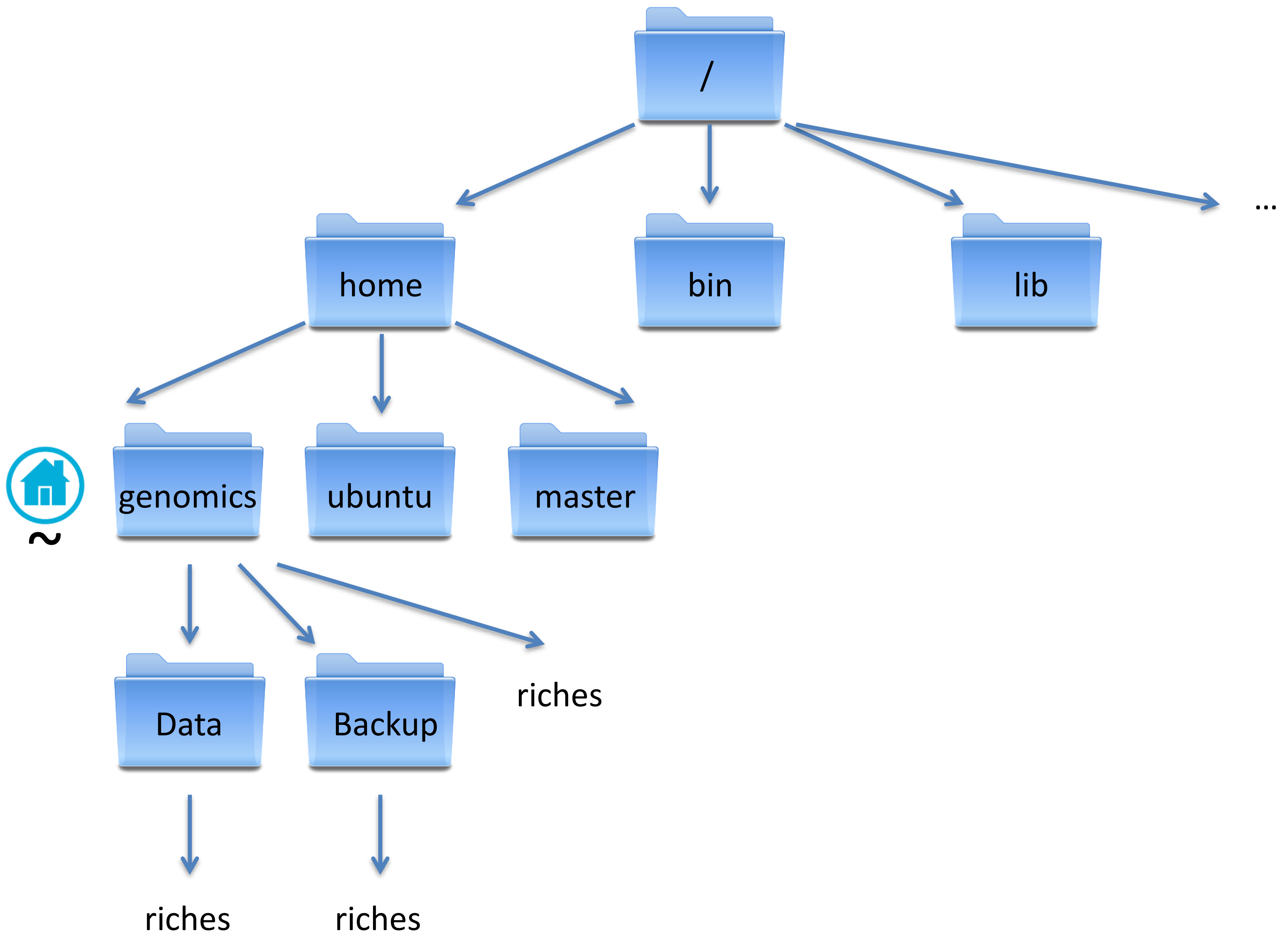
You can also copy entire directories and use this function to rename files/directories

Move to home

```
$ cd ~
```

Make a copy of the Data directory here and call it Backup

```
$ cp -r ./Data ./Backup
```


/
home
bin
lib
...
genomics
ubuntu
master
~
Data
Backup
riches
riches
riches

Typical File Sizes

One Sequencing Sample on the Illumina NextSeq
3,000,000 reads = 1 Gb

But typically you will sequence more than one sample
You may have different patients, different locations, replicates etc…

The size of the sequencing data file can easily become 100s of Gb

(or even bigger depending on the sequencer used)

Archived/Compressed Files

Commonly, people will compress large files so that they are easier to store or share
Here's an example:
sequences.tar.gz

.tar – means that it is a tape archive
.gz – means that it is gzipped

These can be used alone or in combination

To uncompress
A Tar Archive

```
$ tar -xvf <filename>
```

(x = extract, v = verbose, f = all files)

A Gzipped file

```
$ gunzip <filename>
```

A Gzipped Tar archive

```
$ tar -xzvf <filename>
```

Any Questions So Far?

Challenge 2!

1. Change to the unix_workshop directory at the following path:

```
$ cd ~/workshop_materials/unix_workshop
```

You should find a compressed directory:

```
Sequences.tar
```

2. Make a copy of this file in a Backup directory

3. Un archive the directory

4. Unzip the read files

4. Rename the unarchived files – sequence_1.fq and sequence_2.fq

5. Delete the original .tar file

tar	gunzip
cp	mv
rm –i	mkdir
cd	

Challenge 2!

1. Change to the unix_workshop directory at the following path:

```
$ cd ~/workshop_materials/unix_workshop
```

2. Make a copy of the Sequences.tar file in a Backup directory

```
$ mkdir Backup
$ cp Sequences.tar ./Backup
```

3. Un archive the directory

```
$ tar -xvf Sequences.tar
```

4. Unzip the read files

```
$ gunzip Sequences/E_coli_Sequence1.fq.gz
$ gunzip Sequences/E_coli_Sequence2.fq.gz
```

OR

```
$ gunzip Sequences/E_coli_Sequence*
```

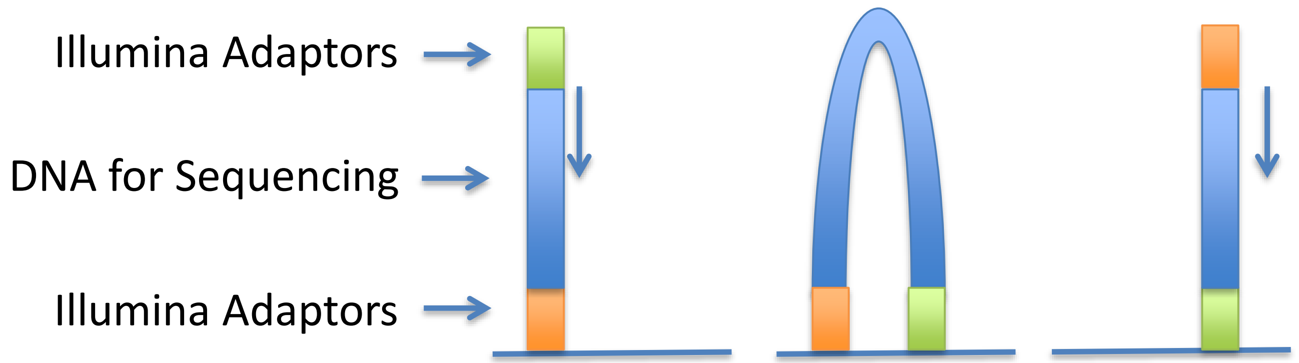
4. Rename the unarchived files – sequence_1.fq and sequence_2.fq

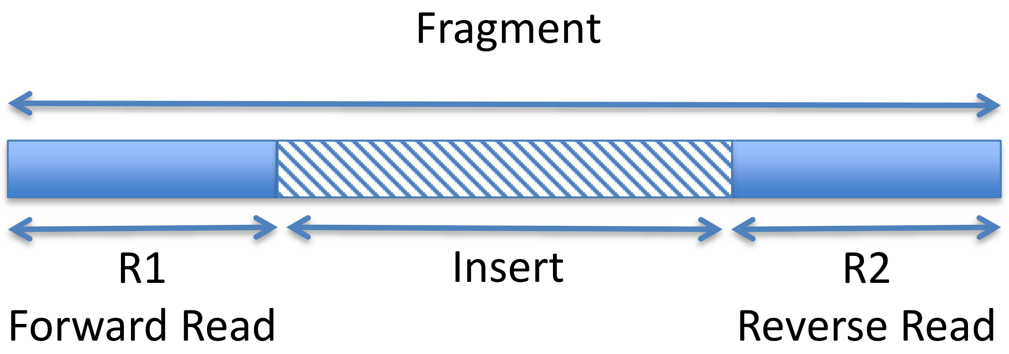
```
$ mv Sequences/E_coli_Sequence1.fq Sequences/sequence_1.fq
$ mv Sequences/E_coli_Sequence2.fq Sequences/sequence_2.fq
```

5. Delete the original .tar file

```
$ rm Sequences.tar
```

Paired Reads





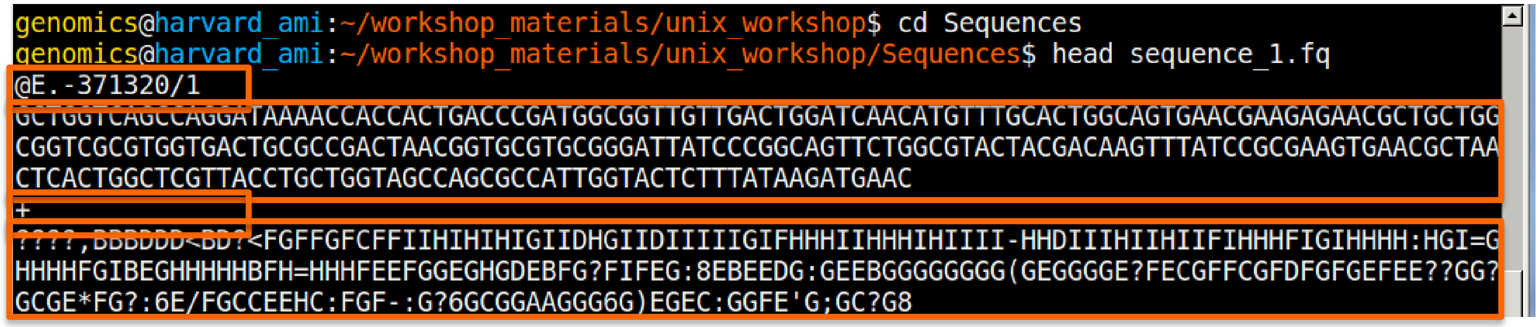
An example: 300 bp paired end reads with a 700 bp fragment size
R1 = 300 bp, R2 = 300 bp, Insert = 100bp

Looking at File Contents

head	tail	more	less	cat
Shows the top lines of a file	Shows the bottom lines of a file	Shows the file one full screen at a time	Shows the file one full screen at a time	Shows an entire file all at once
-n specifies the number of lines (default 10)	-n specifies the number of lines (default 10)	Enter to scroll one line Space to scroll a page q to quit	Enter to scroll one line Space to scroll a page q to quit / to search	Ctrl + C to stop

Use these command line programs to look at the sequence files

Let’s put this to use



```
genomics@harvard_ami:~/workshop_materials/unix_workshop$ cd Sequences
genomics@harvard_ami:~/workshop_materials/unix_workshop/Sequences$ head sequence_1.fq
@E.-371320/1
GCTGGTCAGCCAGGATAAAACCACCACTGACCCGATGGCGGTTGTTGACTGGATCAACATGTTTGCACTGGCAGTGAACGAAGAGAACGCTGCTGG
CGGTCGCGTGGTGACTGCGCCGACTAACGGTGCGTGCGGGATTATCCCGGCAGTTCTGGCGTACTACGACAAGTTTATCCGCGAAGTGAACGCTAA
CTCACTGGCTCGTTACCTGCTGGTAGCCAGCGCCATTGGTACTCTTTATAAGATGAAC
+
????,BBBDDD<BD?<FGFFGFCFFIIHIHIHIGIIDHGIIDIIIIIGIFHHHIIHHHIHIIII-HHDIIIHIIHIIFIHHHFIGIHHHH:HGI=G
HHHHFGIBEGHHHHHBFH=HHHFEEFGGEGHGDEBFG?FIFEG:8EBEEDG:GEEBGGGGGGGG(GEGGGGE?FECGFFCGFDFGFGEFEE??GG?
GCGE*FG?:6E/FGCCEEHC:FGF-:G?6GCGGAAGGG6G)EGEC:GGFE'G;GC?G8
```

Fastq File Format:

Header

Sequence

Second Header (often +)

Phred Quality Score

Lot’s of analysis software like paired reads to be in the same order

Use head to check that the top three headers are in the same order in sequence_1.fq and sequence_2.fq

Sequencing Stats

How many reads?

Count the number of lines

This is the letter l

```
$ wc -l sequence_1.fq
```

742640 lines THEREFORE 185660 reads

Are there the same number of reverse reads?

How about just counting the header lines?

Each header line starts @E

```
$ grep -c "@E" sequence_1.fq
```

219153

BUT the numbers from the two programs don't match?!

```
$ grep "@E" sequence_1.fq
```

How about with this

```
$ grep -c "^@E" sequence_1.fq
```

185660

^ matches this pattern at the start of the line – this is an example of a regular expression

Any Questions So Far?

AND NOW FOR A BRIEF SEGWAY INTO SCRIPTS…

Shell Scripts

Imagine you have a complicated command to run. Take this as an example:

```
ref_map.pl -o ./stacks_gsnap/ -T 4 -O ./popmap –B middleton2_
radtags -b 1 –s ./aligned_gsnap/s13_an_01.bam -s ./aligned_
gsnap/s13_an_02.bam -s ./aligned_gsnap/s13_an_03.bam –s
./aligned_gsnap/s13_an_04.bam –s ./aligned_gsnap/s13_an_05.bam
-s ./aligned_gsnap/s13_an_06.bam -s ./aligned_gsnap/s13_an_07
.bam -s ./aligned_gsnap/s13_an_08.bam -s ./aligned_gsnap/s13_
fw_01.bam -s ./aligned_gsnap/s13_fw_02.bam -s ./aligned_gsnap/
s13_fw_03.bam -s ./aligned_gsnap/s13_fw_04.bam -s ./aligned_
gsnap/s13_fw_05.bam -s ./aligned_gsnap/s13_fw_06.bam -s
./aligned_gsnap/s13_fw_07.bam -s ./aligned_gsnap/s13_fw_08.bam
```

But what if you make a mistake?
Or want to run this command 10 times?
You have to type it out every time ☹

Shell Scripts

Instead we can put this command inside a script.

Then it can easily be edited and ran multiple times

To understand shell scripts, we're going to look at a few topics:

- Shell scripting languages
- Text editors
- How to write a script
- How to run a script

Scripts make a great record of what you've done, when and with what.
You should also aim to keep a computational biology lab book.

What is a Shell Script?

A computer program designed to be run by the Unix shell, the command line interpreter.

There are various types of shell scripts. These are scripting languages.

Today we are going to look at bash

First, let’s run a simple bash command:

```
$ echo Hello World
```

```
$ echo Hello World
Hello World
```

Try using echo with a different phrase

Text Editors

These are pieces of software which can be used to write your script.
Think of them as Unix versions of Notepad.

Some have an interactive user interface – E.G. gedit

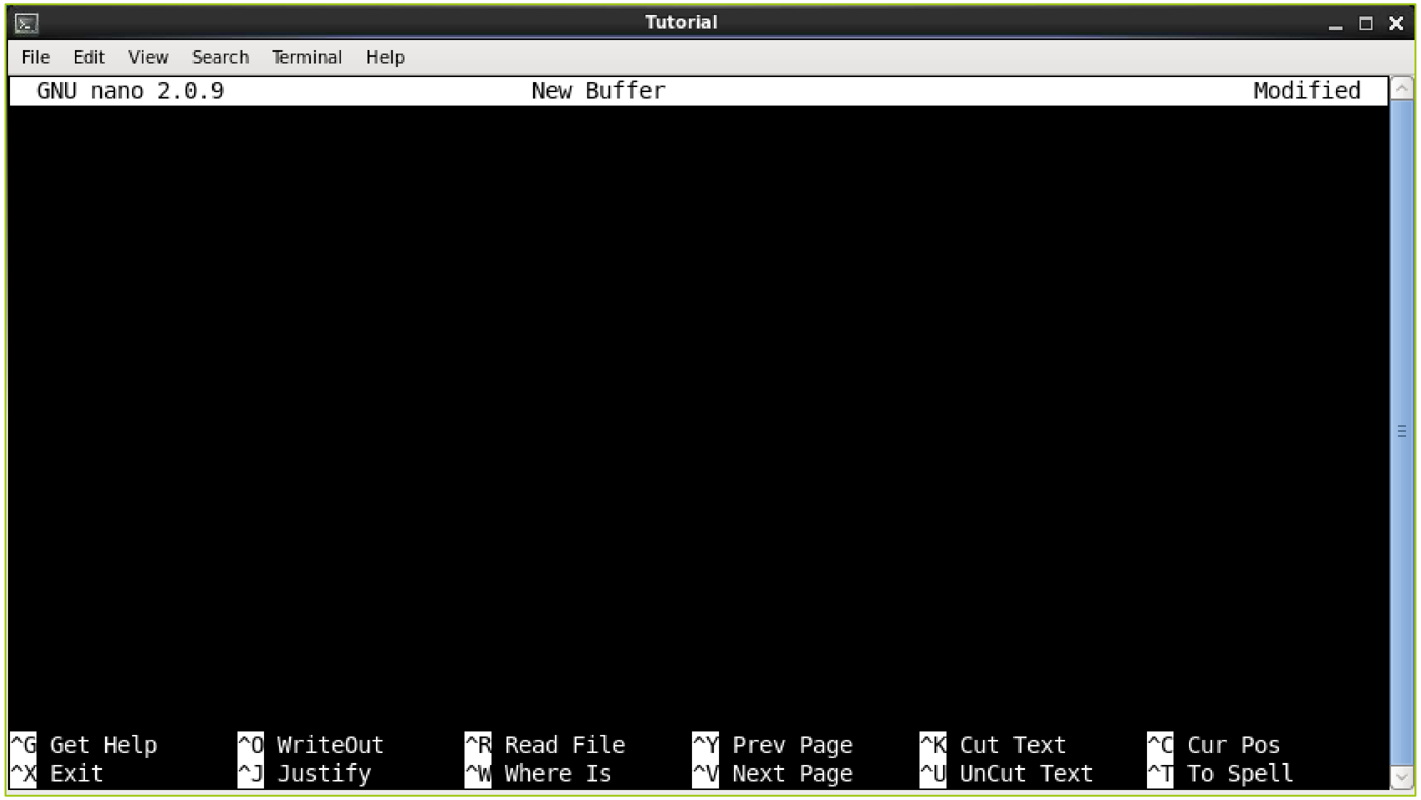
Some work from within the command line – E.G. nano, vim, emacs

Today we are going to work with nano but have a play around with the others when you have a chance. Emacs and vim are notoriously difficult to use for the first time, so look up a cheat sheet.

Your First Script

Let's start by opening nano

```
$ nano
```



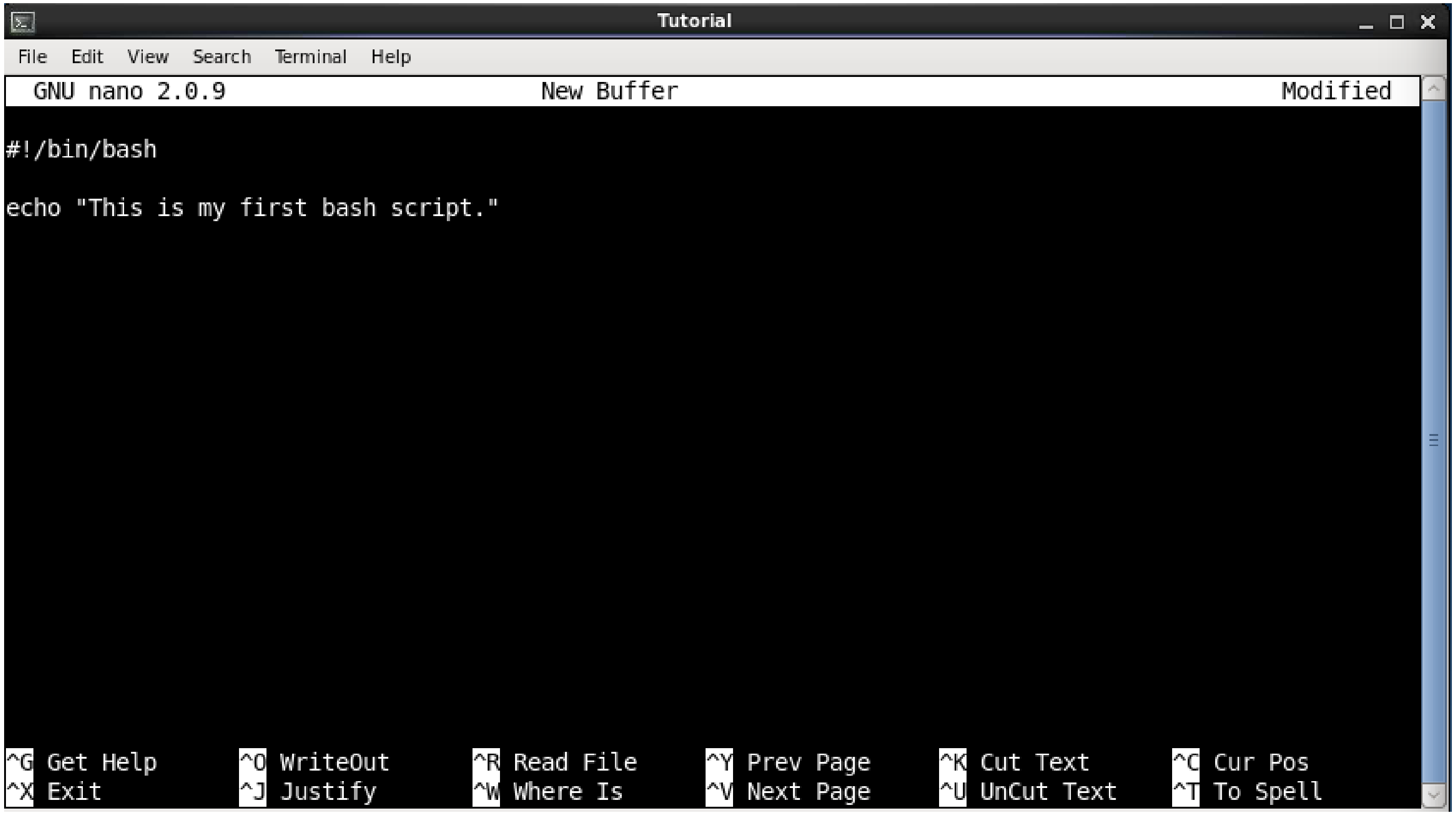
```
  GNU nano 2.0.9                 New Buffer                          Modified



^G Get Help     ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit         ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

Key Nano Commands

Ctrl + O – This saves the file. You will be asked for a file name. Type the name and press enter.

Ctrl + X – This exits nano. If the file is unsaved, you will be asked at this point if you'd like to save it.

Your First Script

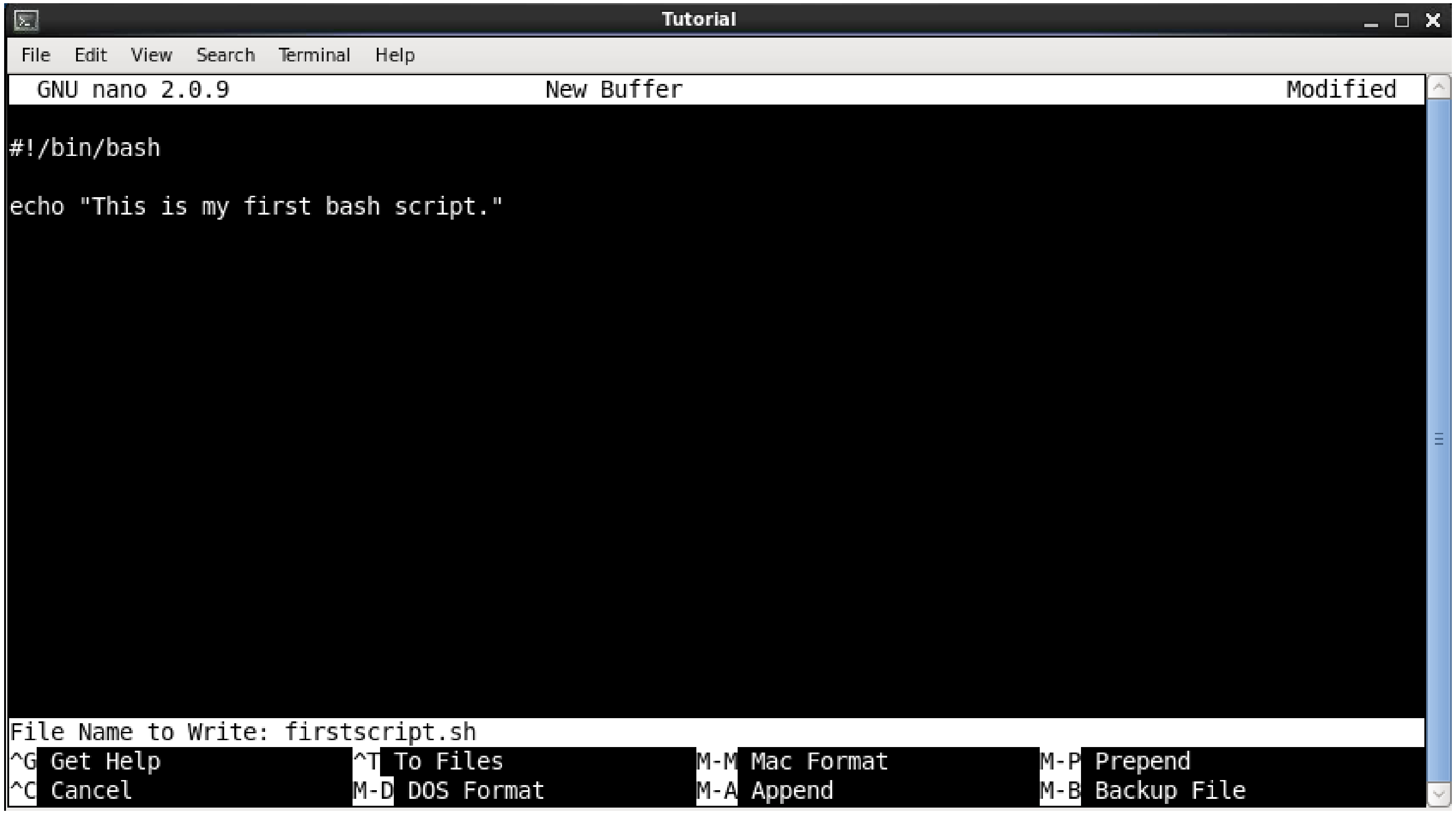


```
  GNU nano 2.0.9                    New Buffer                           Modified

#!/bin/bash

echo "This is my first bash script."
```

```
^G Get Help     ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit         ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

#!/bin/bash tells the computer that this script is in the language bash. It always needs to go at the top of any bash script.

Your First Script



```
  GNU nano 2.0.9                        New Buffer                           Modified

#!/bin/bash

echo "This is my first bash script."
```

```
File Name to Write: firstscript.sh
^G Get Help          ^T To Files          M-M Mac Format       M-P Prepend
^C Cancel            M-D DOS Format       M-A Append           M-B Backup File
```

Then use Ctrl + O to save and give the file the name firstscript.sh.

Then use Ctrl + X to exit.

Now Run Your Script

Simply Type:

```
$ bash firstscript.sh
```

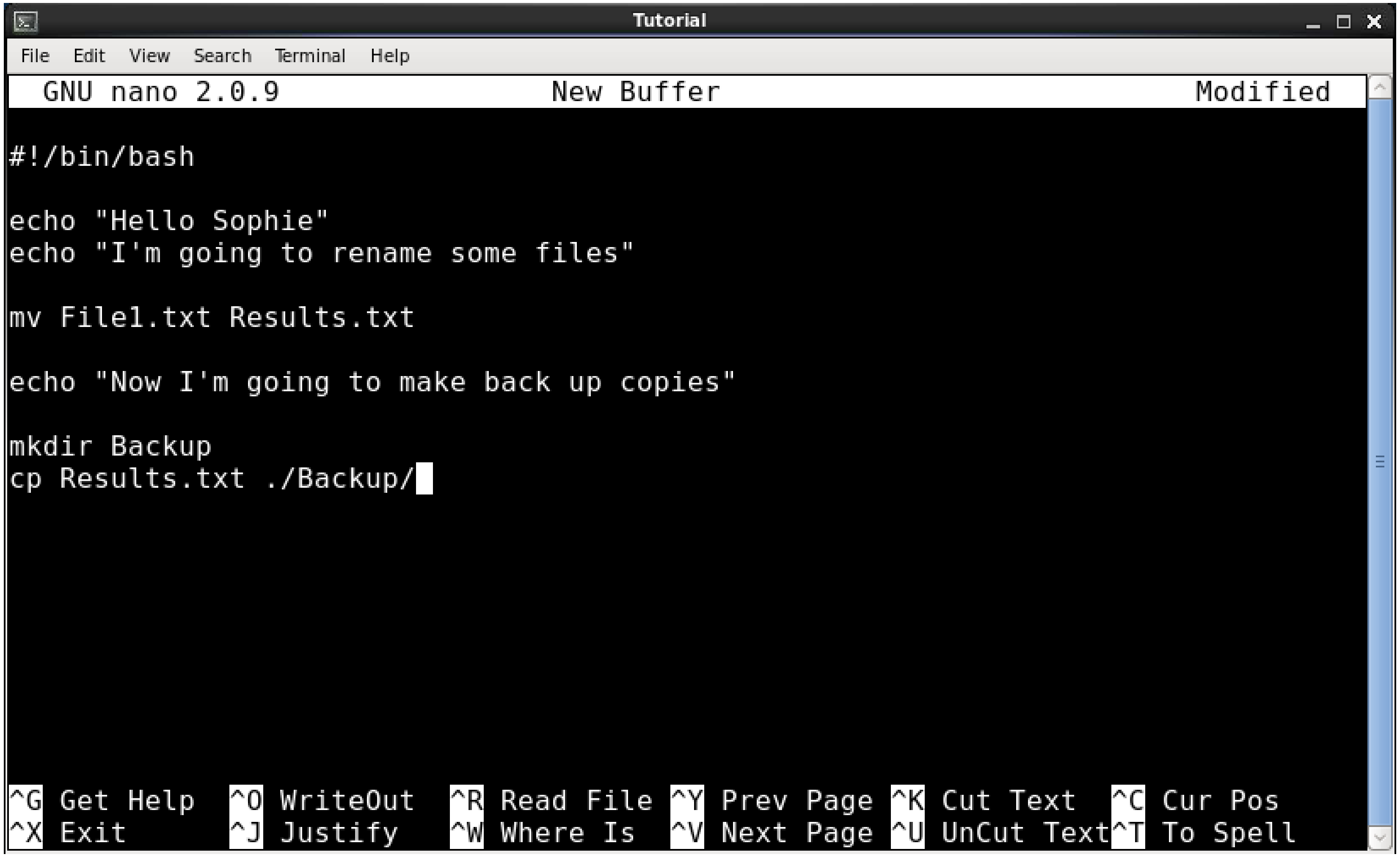
Reopen the same script:

```
$ nano firstscript.sh
```

Change the phrase, save it and run the script again

Bash Scripts

Bash scripts can be used to run binary programs like cd, mv, cp etc…

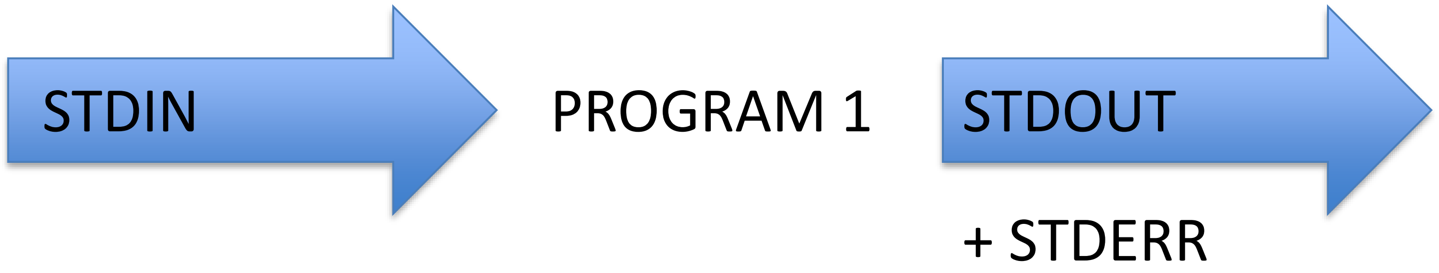


```
  GNU nano 2.0.9                 New Buffer                        Modified

#!/bin/bash

echo "Hello Sophie"
echo "I'm going to rename some files"

mv File1.txt Results.txt

echo "Now I'm going to make back up copies"

mkdir Backup
cp Results.txt ./Backup/
```

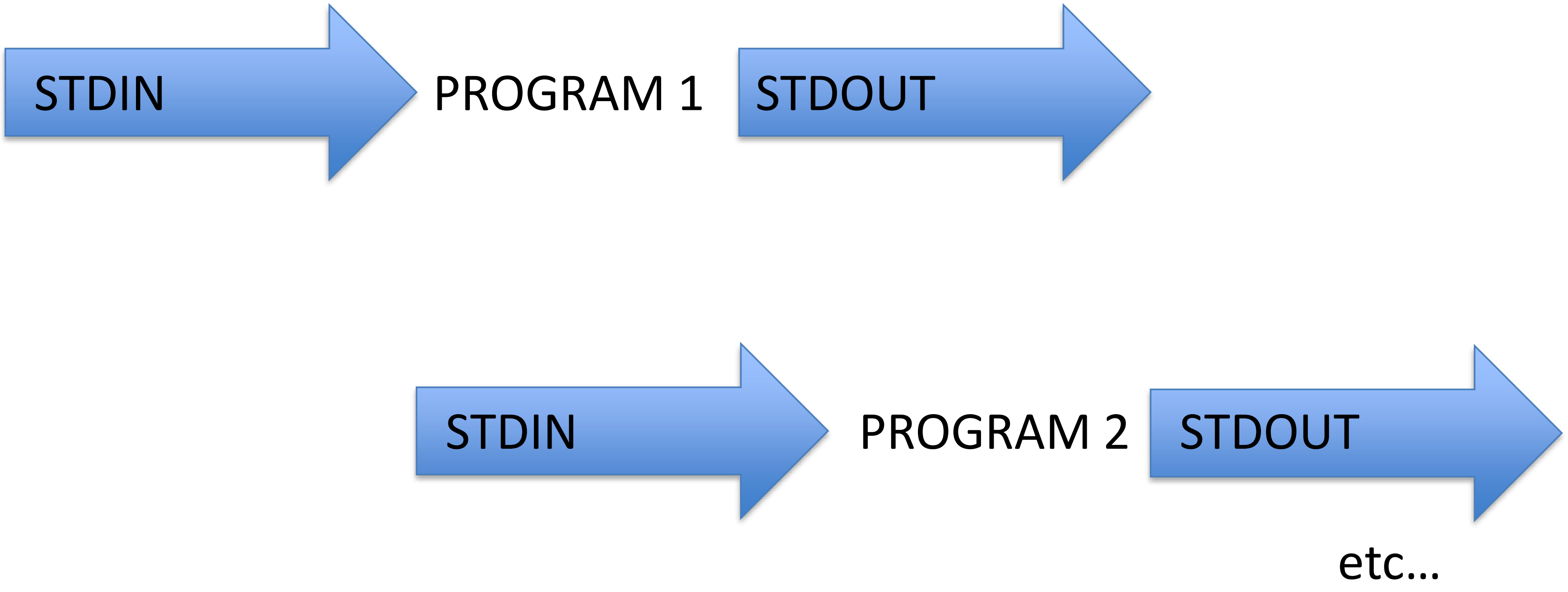
```
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

PIPELINES
(TIME DEPENDING)

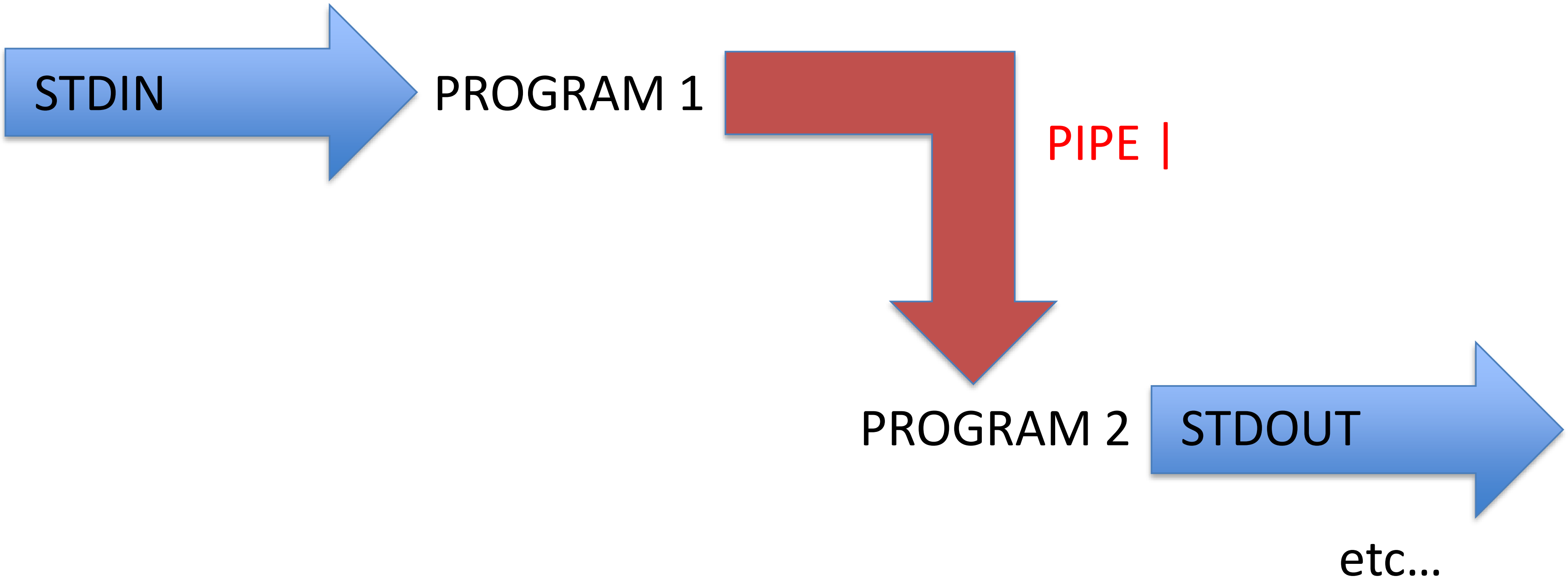
Pipelines



Pipelines



Pipelines



Let’s put this to practice: Building Pipelines

Count the number of files and folders in your home directory

Let’s build the first part of the pipeline, listing the files:

Number 1

```
$ ls -1 /home/genomics/
```

PIPE this into wc –l to count the number of lines:
(i.e. the number of files and folders)

Letter l

```
$ ls -1 /home/genomics/ | wc -l
```

Let’s put this to practice: Building Pipelines

How many base pairs in first sequence?

Firstly let’s get the top two lines of the sequence file:

```
$ head -n 2 sequence_1.fq
```

Now let’s PIPE this into tail to get just the sequence line

```
$ head -n 2 sequence_1.fq | tail -n 1
```

Finally PIPE this into word count of characters to count the base pairs

```
$ head -n 2 sequence_1.fq | tail -n 1 | wc -c
```

Is the first reverse read the same length?

Some More Examples

Within the Unix Workshop directory you should find a file called scientists.txt

```
$ cd ~/workshop_materials/unix_workshop/
```

Take a look at the contents

```
$ more scientists.txt
```

First	Last	DOB
Charles	Darwin	12 February 1809
Marie	Curie	07 November 1867
Stephen	Hawking	08 January 1942
Rosalind	Franklin	25 July 1920
Isaac	Newton	04 January 1643
Richard	Dawkins	26 March 1941

Some More Examples

```
$ cat scientists.txt | cut -f 1
```

First
Charles
Marie
Stephen
Rosalind
Isaac
Richard

Now take a look at the original file

```
$ more scientists.txt
```

Some More Examples

```
$ cat scientists.txt | cut -f 1,3
```

First	DOB
Charles	12 February 1809
Marie	07 November 1867
Stephen	08 January 1942
Rosalind	25 July 1920
Issac	04 January 1643
Richard	26 March 1941

Some More Examples

```
$ cat scientists.txt | cut -f 1 | sort
```

Charles
First
Isaac
Marie
Richard
Rosalind
Stephen

What if you wanted to keep the sorted list?

```
$ cat scientists.txt | cut -f 1 | sort > newfile.txt
```

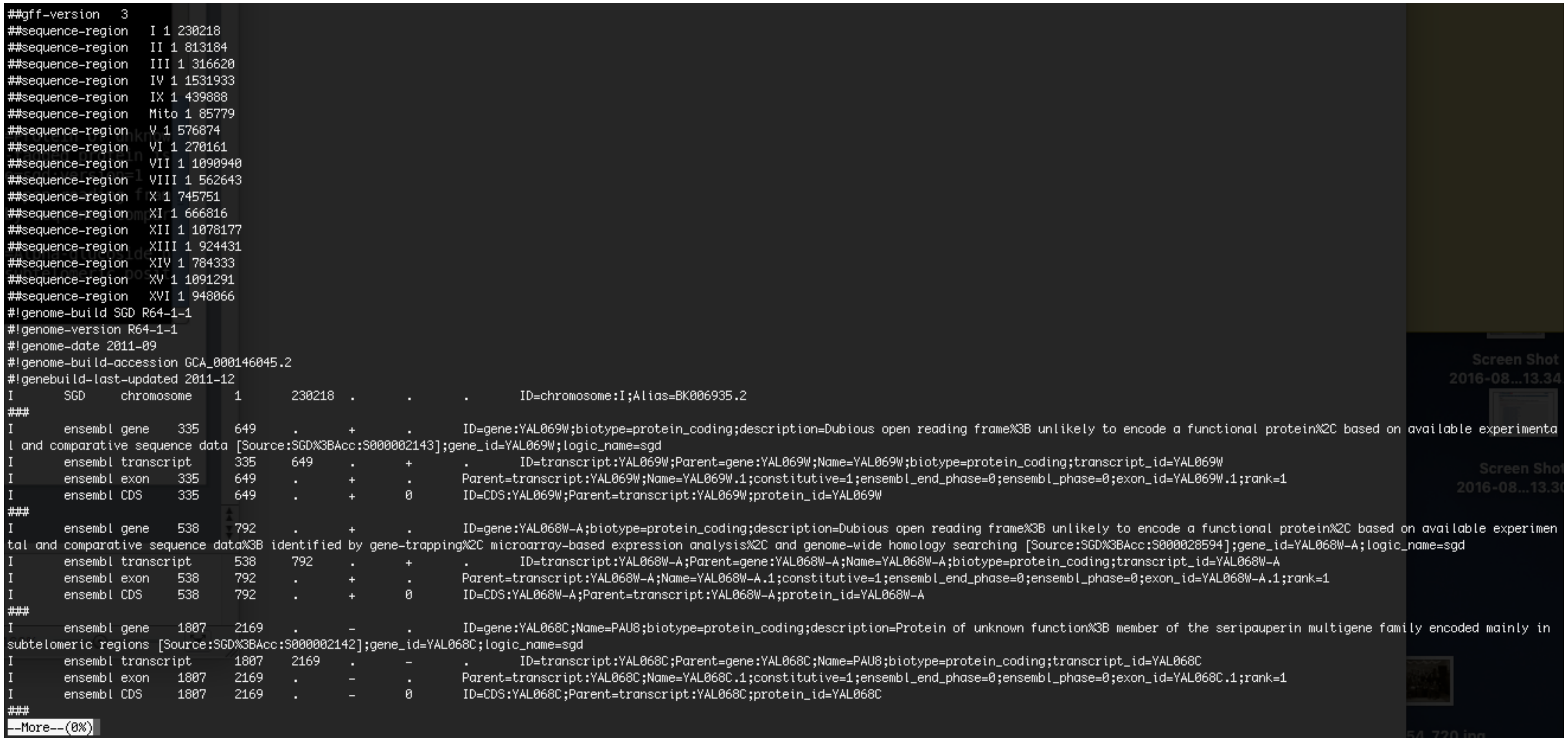
Redirects

CHALLENGE 3!

Looking at the *Saccharomyces cerevisiae* gff3 file

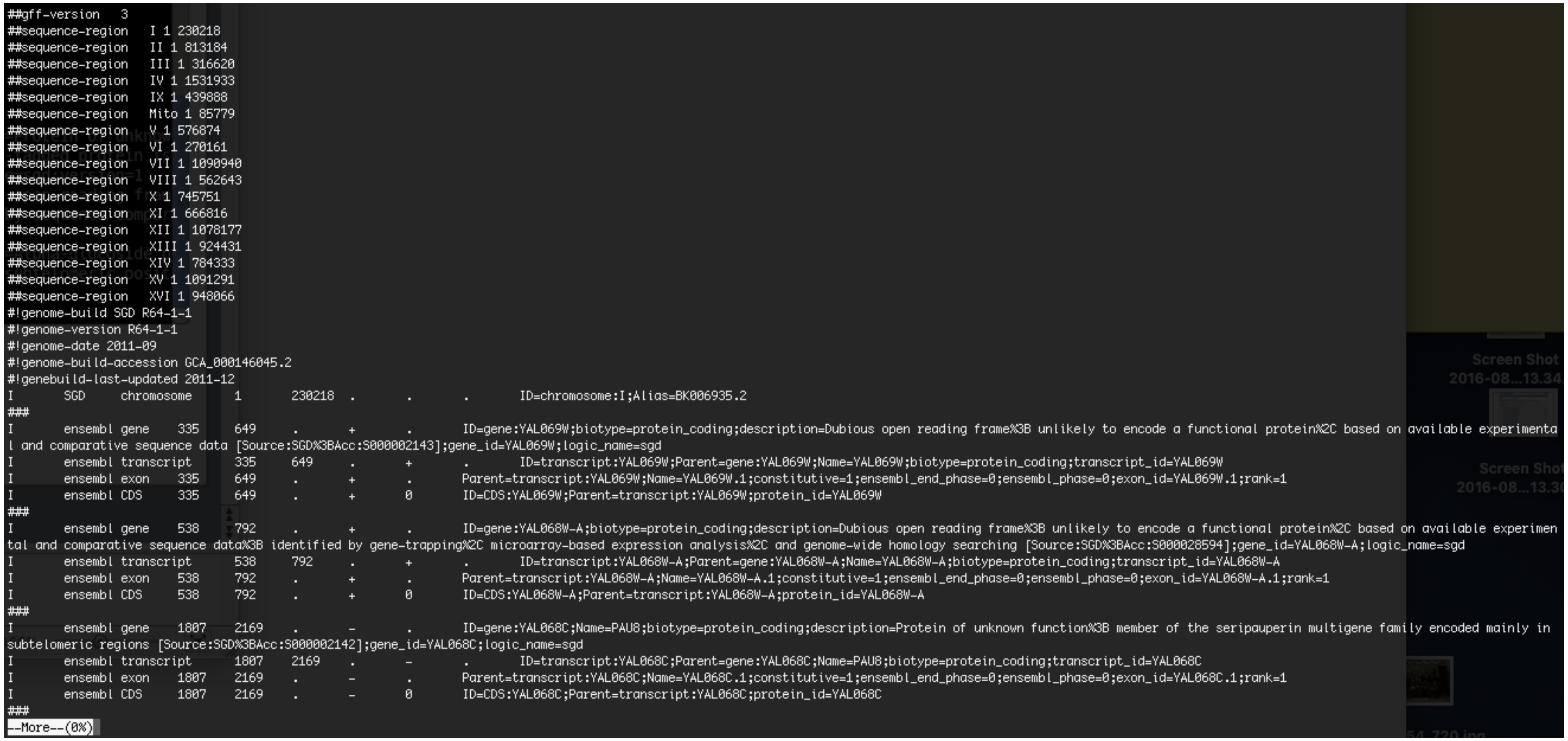
GFF = general feature format

This is a file which lists all of the genome features, their coordinates, and info about them (genes, tRNAs, exons etc…)



```
##gff-version   3
##sequence-region   I 1 230218
##sequence-region   II 1 813184
##sequence-region   III 1 316620
##sequence-region   IV 1 1531933
##sequence-region   IX 1 439888
##sequence-region   Mito 1 85779
##sequence-region   V 1 576874
##sequence-region   VI 1 270161
##sequence-region   VII 1 1090940
##sequence-region   VIII 1 562643
##sequence-region   X 1 745751
##sequence-region   XI 1 666816
##sequence-region   XII 1 1078177
##sequence-region   XIII 1 924431
##sequence-region   XIV 1 784333
##sequence-region   XV 1 1091291
##sequence-region   XVI 1 948066
#!genome-build SGD R64-1-1
#!genome-version R64-1-1
#!genome-date 2011-09
#!genome-build-accession GCA_000146045.2
#!genebuild-last-updated 2011-12
I       SGD     chromosome      1       230218  .       .       .       ID=chromosome:I;Alias=BK006935.2
###
I       ensembl gene    335     649     .       +       .       ID=gene:YAL069W;biotype=protein_coding;description=Dubious open reading frame%3B unlikely to encode a functional protein%2C based on available experimenta
l and comparative sequence data [Source:SGD%3BAcc:S000002143];gene_id=YAL069W;logic_name=sgd
I       ensembl transcript      335     649     .       +       .       ID=transcript:YAL069W;Parent=gene:YAL069W;Name=YAL069W;biotype=protein_coding;transcript_id=YAL069W
I       ensembl exon    335     649     .       +       .       Parent=transcript:YAL069W;Name=YAL069W.1;constitutive=1;ensembl_end_phase=0;ensembl_phase=0;exon_id=YAL069W.1;rank=1
I       ensembl CDS     335     649     .       +       0       ID=CDS:YAL069W;Parent=transcript:YAL069W;protein_id=YAL069W
###
I       ensembl gene    538     792     .       +       .       ID=gene:YAL068W-A;biotype=protein_coding;description=Dubious open reading frame%3B unlikely to encode a functional protein%2C based on available experimen
tal and comparative sequence data%3B identified by gene-trapping%2C microarray-based expression analysis%2C and genome-wide homology searching [Source:SGD%3BAcc:S000028594];gene_id=YAL068W-A;logic_name=sgd
I       ensembl transcript      538     792     .       +       .       ID=transcript:YAL068W-A;Parent=gene:YAL068W-A;Name=YAL068W-A;biotype=protein_coding;transcript_id=YAL068W-A
I       ensembl exon    538     792     .       +       .       Parent=transcript:YAL068W-A;Name=YAL068W-A.1;constitutive=1;ensembl_end_phase=0;ensembl_phase=0;exon_id=YAL068W-A.1;rank=1
I       ensembl CDS     538     792     .       +       0       ID=CDS:YAL068W-A;Parent=transcript:YAL068W-A;protein_id=YAL068W-A
###
I       ensembl gene    1807    2169    .       -       .       ID=gene:YAL068C;Name=PAU8;biotype=protein_coding;description=Protein of unknown function%3B member of the seripauperin multigene family encoded mainly in
subtelomeric regions [Source:SGD%3BAcc:S000002142];gene_id=YAL068C;logic_name=sgd
I       ensembl transcript      1807    2169    .       -       .       ID=transcript:YAL068C;Parent=gene:YAL068C;Name=PAU8;biotype=protein_coding;transcript_id=YAL068C
I       ensembl exon    1807    2169    .       -       .       Parent=transcript:YAL068C;Name=YAL068C.1;constitutive=1;ensembl_end_phase=0;ensembl_phase=0;exon_id=YAL068C.1;rank=1
I       ensembl CDS     1807    2169    .       -       0       ID=CDS:YAL068C;Parent=transcript:YAL068C;protein_id=YAL068C
###
--More--(0%)
```

CHALLENGE 3!



```
##gff-version   3
##sequence-region   I 1 230218
##sequence-region   II 1 813184
##sequence-region   III 1 316620
##sequence-region   IV 1 1531933
##sequence-region   IX 1 439888
##sequence-region   Mito 1 85779
##sequence-region   V 1 576874
##sequence-region   VI 1 270161
##sequence-region   VII 1 1090940
##sequence-region   VIII 1 562643
##sequence-region   X 1 745751
##sequence-region   XI 1 666816
##sequence-region   XII 1 1078177
##sequence-region   XIII 1 924431
##sequence-region   XIV 1 784333
##sequence-region   XV 1 1091291
##sequence-region   XVI 1 948066
#!genome-build SGD R64-1-1
#!genome-version R64-1-1
#!genome-date 2011-09
#!genome-build-accession GCA_000146045.2
#!genebuild-last-updated 2011-12
I       SGD     chromosome      1       230218  .       .       .       ID=chromosome:I;Alias=BK006935.2
###
I       ensembl gene    335     649     .       +       .       ID=gene:YAL069W;biotype=protein_coding;description=Dubious open reading frame%3B unlikely to encode a functional protein%2C based on available experimenta
l and comparative sequence data [Source:SGD%3BAcc:S000002143];gene_id=YAL069W;logic_name=sgd
I       ensembl transcript      335     649     .       +       .       ID=transcript:YAL069W;Parent=gene:YAL069W;Name=YAL069W;biotype=protein_coding;transcript_id=YAL069W
I       ensembl exon    335     649     .       +       .       Parent=transcript:YAL069W;Name=YAL069W.1;constitutive=1;ensembl_end_phase=0;ensembl_phase=0;exon_id=YAL069W.1;rank=1
I       ensembl CDS     335     649     .       +       0       ID=CDS:YAL069W;Parent=transcript:YAL069W;protein_id=YAL069W
###
I       ensembl gene    538     792     .       +       .       ID=gene:YAL068W-A;biotype=protein_coding;description=Dubious open reading frame%3B unlikely to encode a functional protein%2C based on available experimen
tal and comparative sequence data%3B identified by gene-trapping%2C microarray-based expression analysis%2C and genome-wide homology searching [Source:SGD%3BAcc:S000028594];gene_id=YAL068W-A;logic_name=sgd
I       ensembl transcript      538     792     .       +       .       ID=transcript:YAL068W-A;Parent=gene:YAL068W-A;Name=YAL068W-A;biotype=protein_coding;transcript_id=YAL068W-A
I       ensembl exon    538     792     .       +       .       Parent=transcript:YAL068W-A;Name=YAL068W-A.1;constitutive=1;ensembl_end_phase=0;ensembl_phase=0;exon_id=YAL068W-A.1;rank=1
I       ensembl CDS     538     792     .       +       0       ID=CDS:YAL068W-A;Parent=transcript:YAL068W-A;protein_id=YAL068W-A
###
I       ensembl gene    1807    2169    .       -       .       ID=gene:YAL068C;Name=PAU8;biotype=protein_coding;description=Protein of unknown function%3B member of the seripauperin multigene family encoded mainly in
subtelomeric regions [Source:SGD%3BAcc:S000002142];gene_id=YAL068C;logic_name=sgd
I       ensembl transcript      1807    2169    .       -       .       ID=transcript:YAL068C;Parent=gene:YAL068C;Name=PAU8;biotype=protein_coding;transcript_id=YAL068C
I       ensembl exon    1807    2169    .       -       .       Parent=transcript:YAL068C;Name=YAL068C.1;constitutive=1;ensembl_end_phase=0;ensembl_phase=0;exon_id=YAL068C.1;rank=1
I       ensembl CDS     1807    2169    .       -       0       ID=CDS:YAL068C;Parent=transcript:YAL068C;protein_id=YAL068C
###
--More--(0%)
```

Lines that start # are comments – just run information

Column 1 = Chromosome
Column 3 = Feature/Type e.g. gene, chromosome, exon
Column 4 = Start Location
Column 5 = Stop Location

CHALLENGE 3!

1. In the Unix workshop directory you should find a gff3 file.

```
$ cd ~/Unix_Workshop/Challenge5
```

```
Saccharomyces_cerevisiae.R64-1-1.85.gff3.gz
```

2. Unzip the file.

3. How many feature entries are there?

4. List and count all the different types of features

5. Which chromosome is the longest?

Hints!

- Use head to work with 10 lines whilst testing what your pipe does!
- This is a tab delimited file with a column layout.
- Google “gff3 format” to find out what each of the columns are.
- Remember that cat opens an entire file at once.
- There are a number of info lines at the start which begin with a hash. Look into grep with invert matches to skip these.
- Cut can be used to isolate certain columns. You’ll want the field option.
- The programs sort and uniq may be helpful.
- Sort must be used before uniq.
- Uniq has a counting option.
- Sort uses the key option to sort by a column.

more	gunzip	head	uniq
cp	mv	grep	wc
rm –i	mkdir	cut	|
cd	cat	sort	man

CHALLENGE 3!

1. Find the gff3 file.

2. Unzip the file.

```
$ gunzip Saccharomyces_cerevisiae.R64-1-1.85.gff3.gz
```

3. How many feature entries are there?

```
$ cat Saccharomyces_cerevisiae.R64-1-1.85.gff3 | grep -v "^#" | wc -l
```

28926

4. List and count all the different types of features (type on one line)

```
$ cat Saccharomyces_cerevisiae.R64-1-1.85.gff3 | grep -v "^#" | cut -f 3
| sort | uniq -c | sort -n
```

5. Which chromosome is the longest? (type on one line)

```
$ cat Saccharomyces_cerevisiae.R64-1-1.85.gff3 | grep -v "^#"
| cut -f 1,3,4,5 | grep chromosome | sort -n -k 4
```

Chromosome IV 1531933 bp

Any Questions So Far?