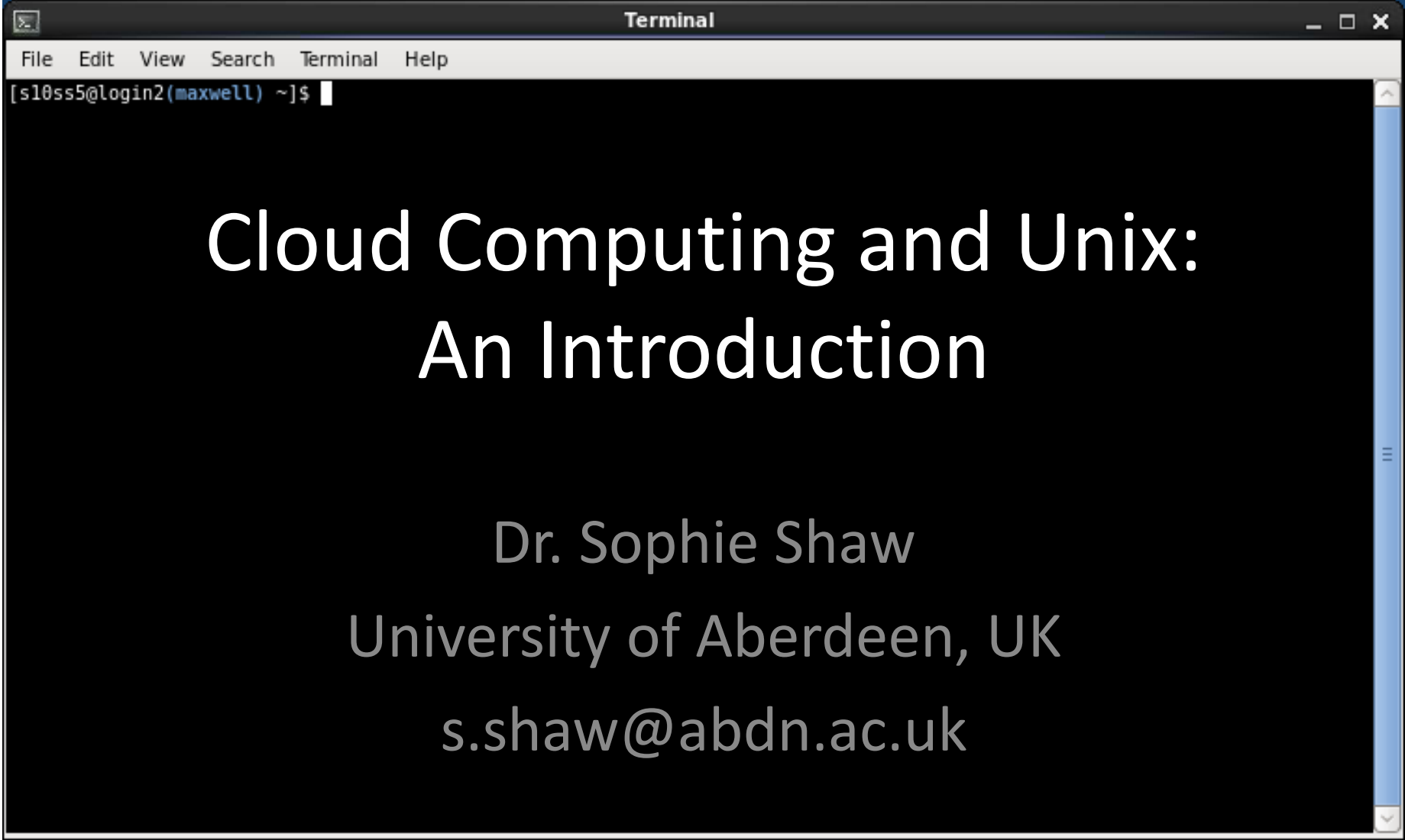


A terminal window titled "Terminal" with a menu bar containing "File", "Edit", "View", "Search", "Terminal", and "Help". The prompt is "[s10ss5@login2(maxwell) ~]\$". The main content area has a black background with white text.

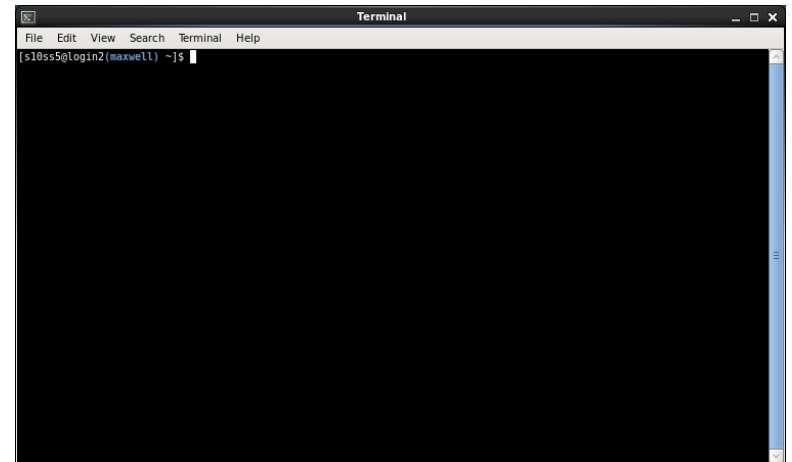
Cloud Computing and Unix: An Introduction

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



What We're Going To Do

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



Etiquette

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

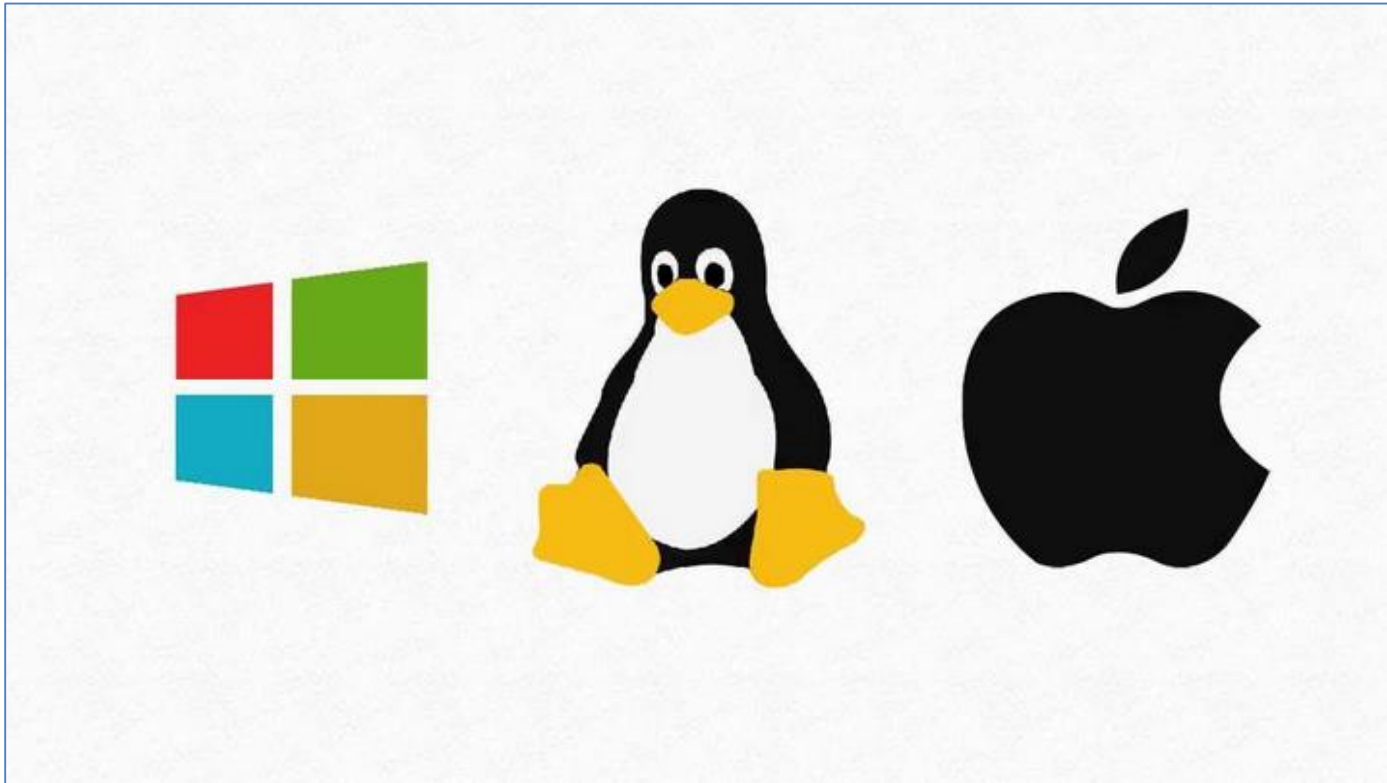


File Commands	System Info
ls - directory listing ls -al - formatted listing with hidden files cd <i>dir</i> - change directory to <i>dir</i> cd - change to home pwd - show current directory mkdir <i>dir</i> - create a directory <i>dir</i> rm <i>file</i> - delete <i>file</i> rm -r <i>dir</i> - delete directory <i>dir</i> rm -f <i>file</i> - force remove <i>file</i> rm -rf <i>dir</i> - force remove directory <i>dir</i> * cp <i>file1 file2</i> - copy <i>file1</i> to <i>file2</i> cp -r <i>dir1 dir2</i> - copy <i>dir1</i> to <i>dir2</i> ; create <i>dir2</i> if it doesn't exist 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> ln -s <i>file link</i> - create symbolic link <i>link</i> to <i>file</i> touch <i>file</i> - create or update <i>file</i> cat > <i>file</i> - places standard input into <i>file</i> more <i>file</i> - output the contents of <i>file</i> head <i>file</i> - output the first 10 lines of <i>file</i> tail <i>file</i> - output the last 10 lines of <i>file</i> tail -f <i>file</i> - output the contents of <i>file</i> as it grows, starting with the last 10 lines	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 <i>user</i> - display information about <i>user</i> uname -a - show kernel information cat /proc/cpuinfo - cpu information cat /proc/meminfo - memory information man <i>command</i> - show the manual for <i>command</i> df - show disk usage du - show directory space usage free - show memory and swap usage whereis <i>app</i> - show possible locations of <i>app</i> which <i>app</i> - show which <i>app</i> will be run by default
Process Management	Compression
ps - display your currently active processes top - display all running processes kill <i>pid</i> - kill process id <i>pid</i> killall <i>proc</i> - kill all processes named <i>proc</i> * bg - lists stopped or background jobs; resume a stopped job in the background fg - brings the most recent job to foreground fg <i>n</i> - brings job <i>n</i> to the foreground	tar cf <i>file.tar files</i> - create a tar named <i>file.tar</i> containing <i>files</i> tar xf <i>file.tar</i> - extract the files from <i>file.tar</i> 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 tar xjf <i>file.tar.bz2</i> - extract a tar using Bzip2 gzip <i>file</i> - compresses <i>file</i> and renames it to <i>file.gz</i> gzip -d <i>file.gz</i> - decompresses <i>file.gz</i> back to <i>file</i>
File Permissions	Network
chmod <i>octal file</i> - change the permissions of <i>file</i> to <i>octal</i> , 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 .	ping <i>host</i> - ping <i>host</i> and output results whois <i>domain</i> - get whois information for <i>domain</i> dig <i>domain</i> - get DNS information for <i>domain</i> dig -x <i>host</i> - reverse lookup <i>host</i> wget <i>file</i> - download <i>file</i> wget -c <i>file</i> - continue a stopped download
SSH	Installation
ssh <i>user@host</i> - connect to <i>host</i> as <i>user</i> ssh -p <i>port user@host</i> - connect to <i>host</i> on port <i>port</i> as <i>user</i> ssh-copy-id <i>user@host</i> - add your key to <i>host</i> for <i>user</i> to enable a keyed or passwordless login	Install from source: ./configure make make install dpkg -i <i>pkg.deb</i> - install a package (Debian) rpm -Uvh <i>pkg.rpm</i> - install a package (RPM)
Searching	Shortcuts
grep <i>pattern files</i> - search for <i>pattern</i> in <i>files</i> grep -r <i>pattern dir</i> - search recursively for <i>pattern</i> in <i>dir</i> <i>command</i> grep <i>pattern</i> - search for <i>pattern</i> in the output of <i>command</i> locate <i>file</i> - find all instances of <i>file</i>	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.	



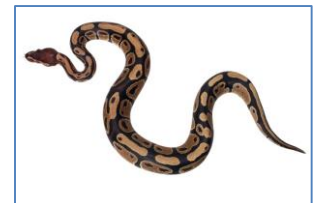
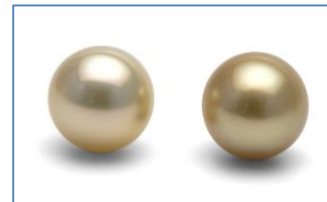
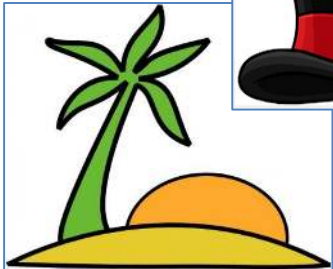
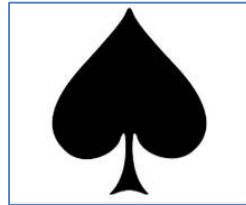
What is Unix?

- 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?

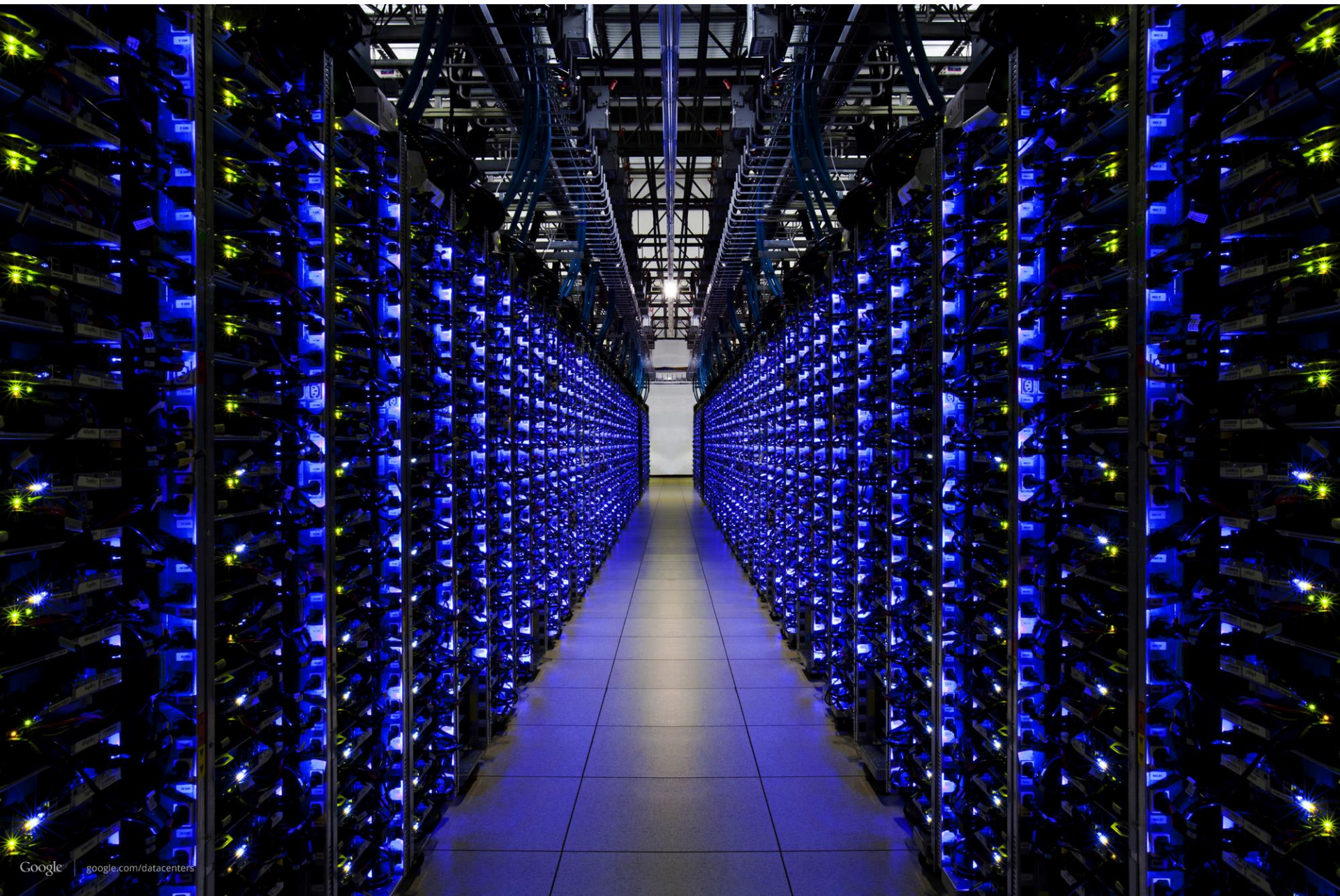
- Linux computers or servers.
- Compute clusters.
- The cloud.
 - What we're going to use this week

Download more graphics at www.psdgraphics.com



So What is Cloud Computing?



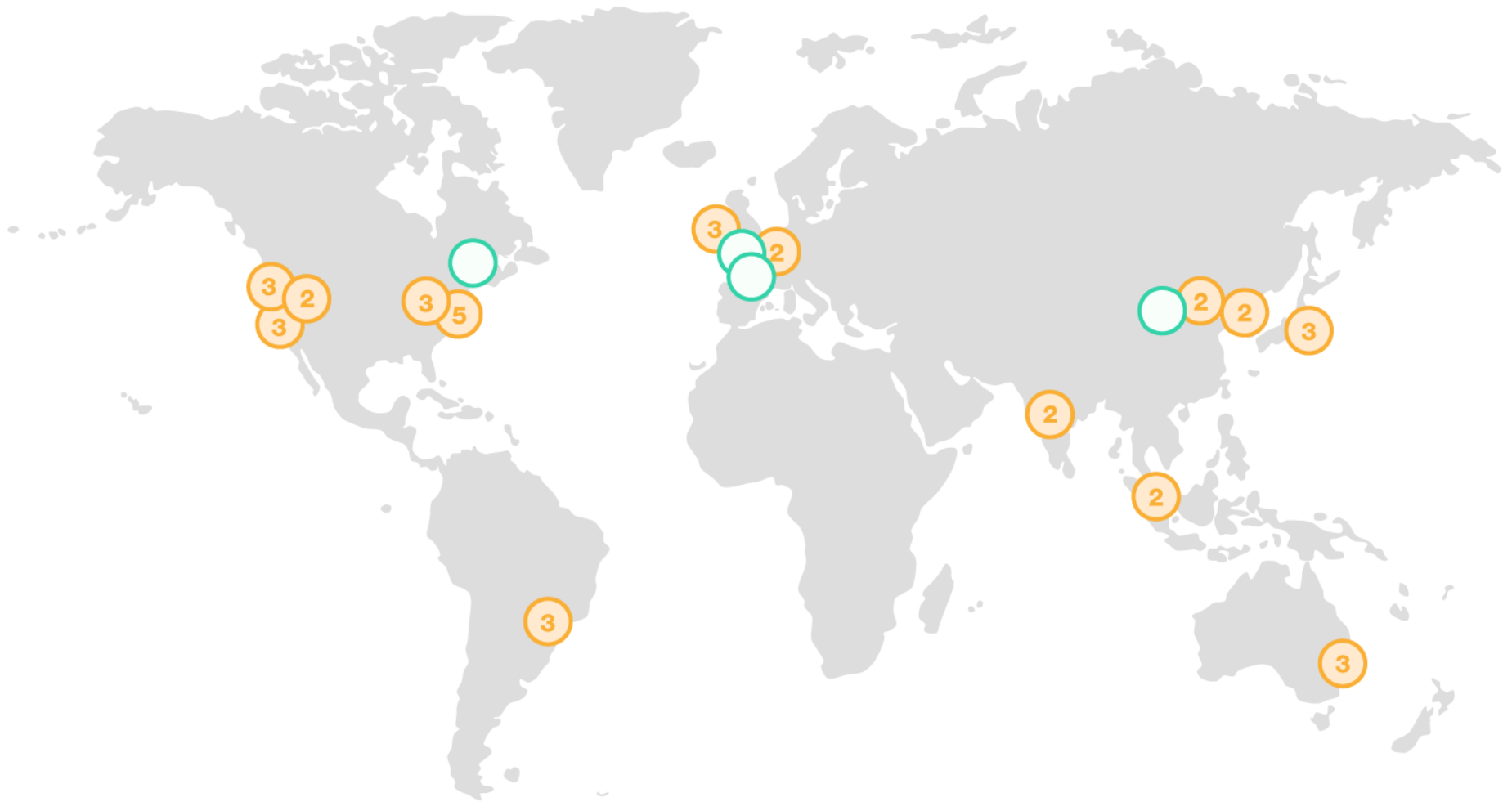


Cloud Computing Solutions



Google Compute Engine

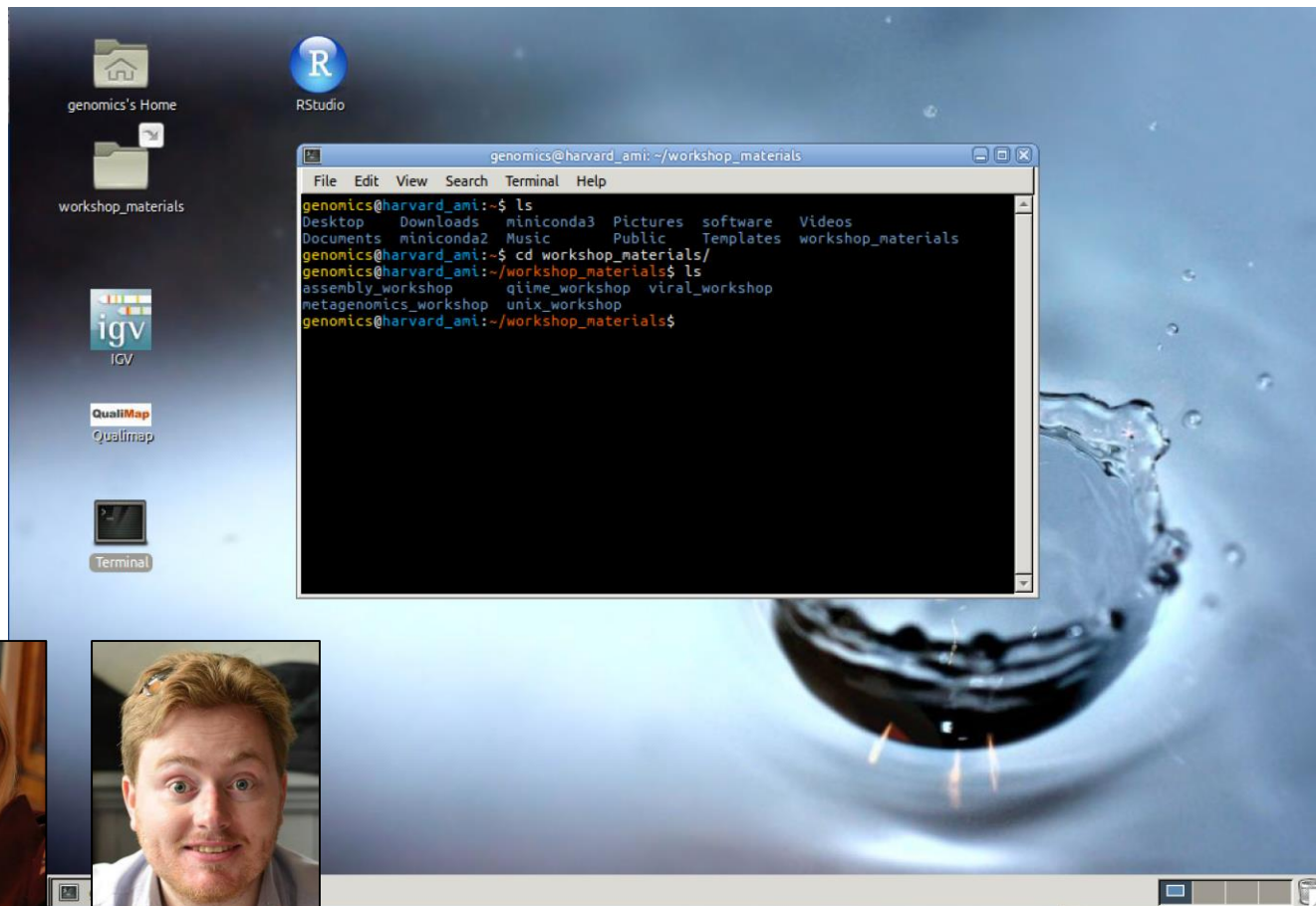




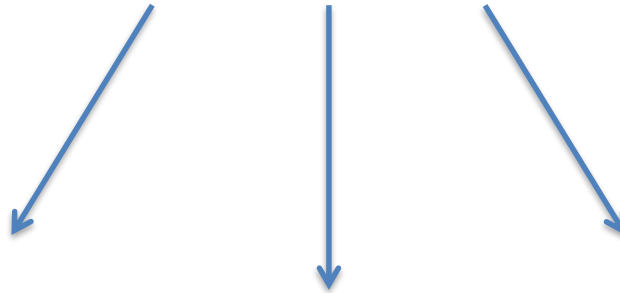
AWS “Availability Zones” and Data Centres

How it Works

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



How it Works



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.*

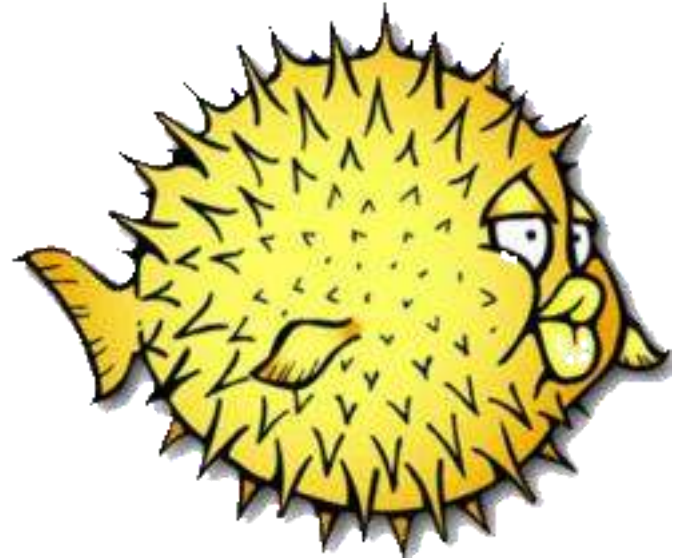
The Rules

- 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

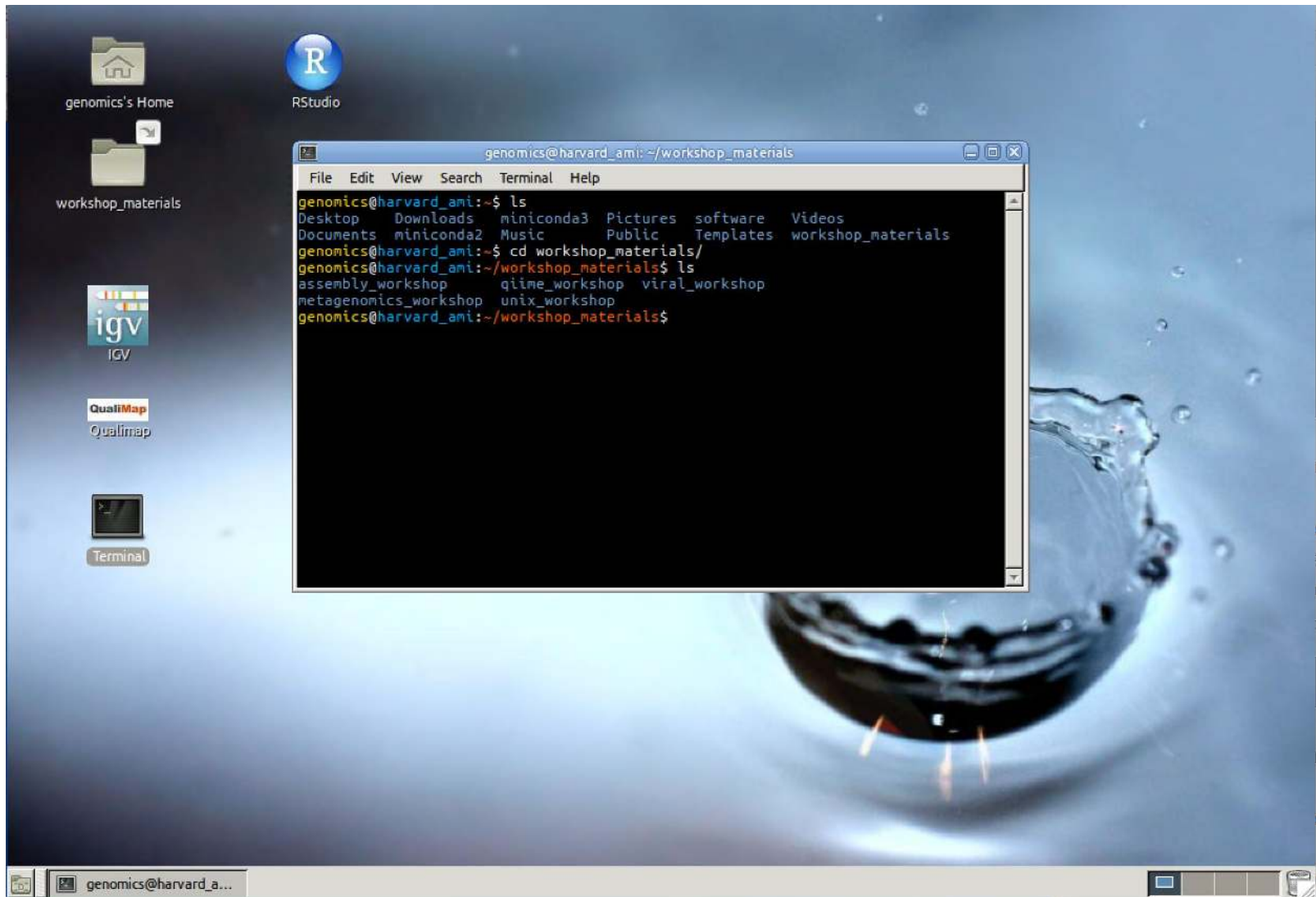


Secure Shell –
“SSH”
e.g. SSH or PuTTY

Now What?!

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

INSERT LIVE DEMO

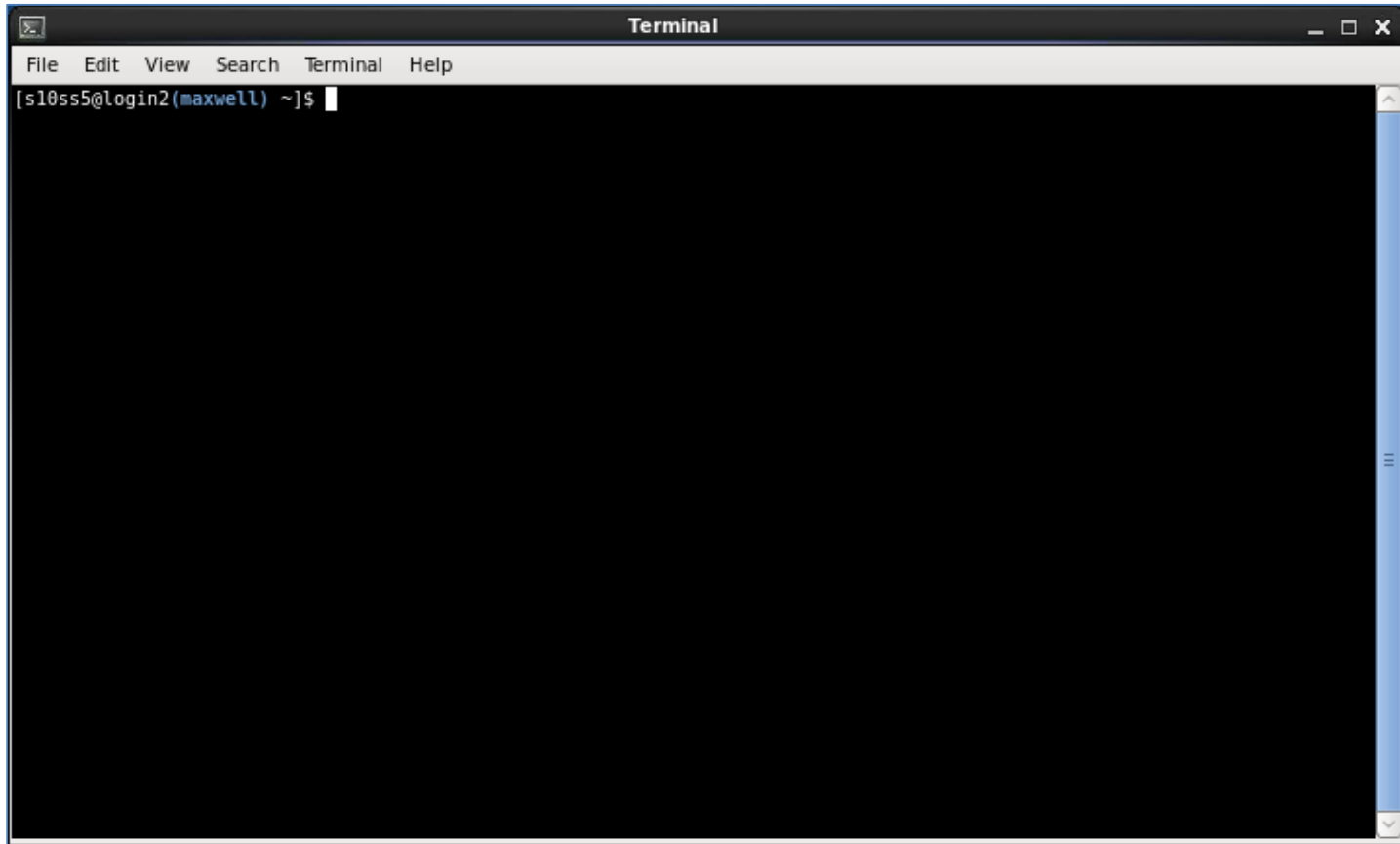


You're now connected to your instance and you're ready to learn some Unix!

Any Questions So Far?



The Terminal Window



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

Catriona Wo... Catriona Scr... Eva Working Eva Script Exeter [Tutorial] Jonathon Wo... Jonathon Scr... Terminal Neil Working Neil Script

Location is Important

First Task – Where am I?

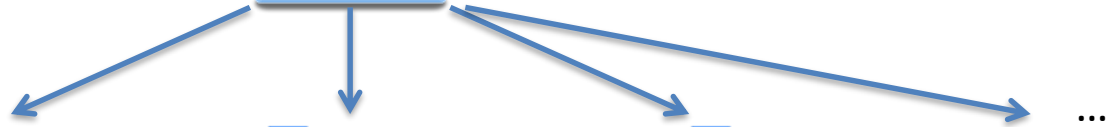
```
$ pwd
```

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

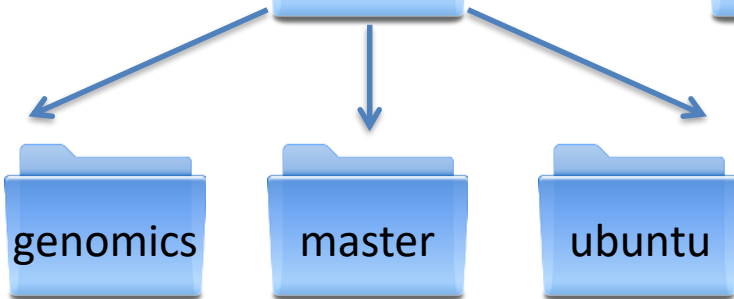
This is your “present working directory”

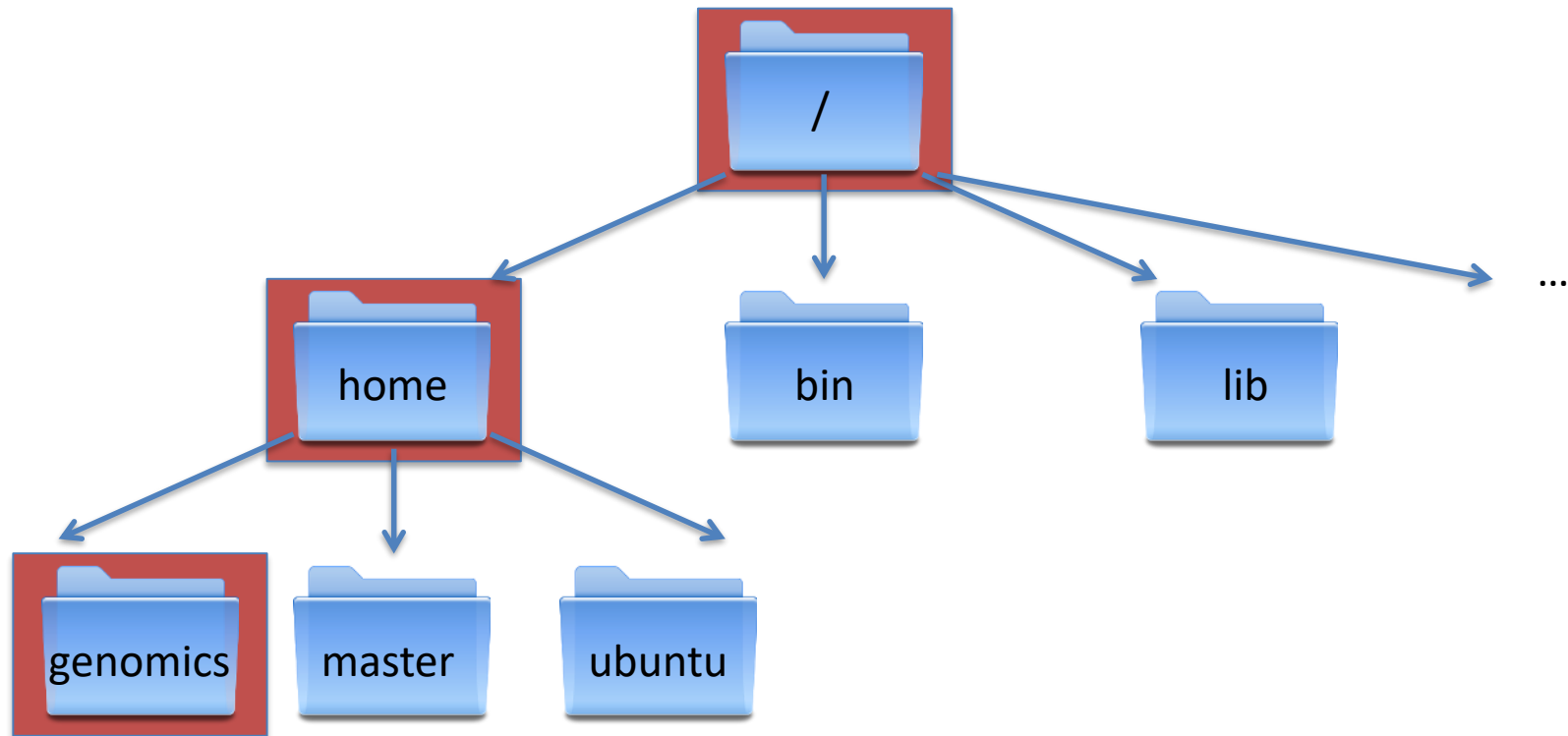


= ROOT

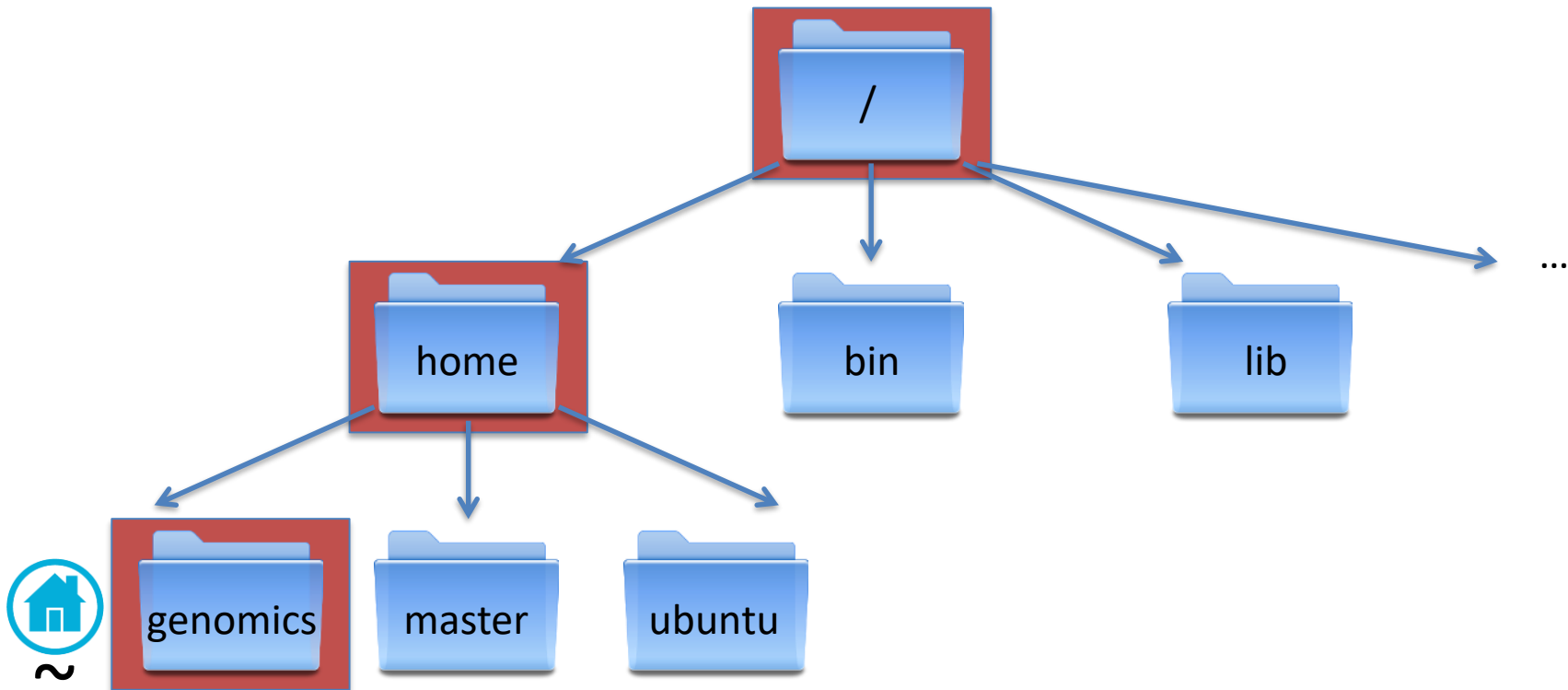


...





```
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

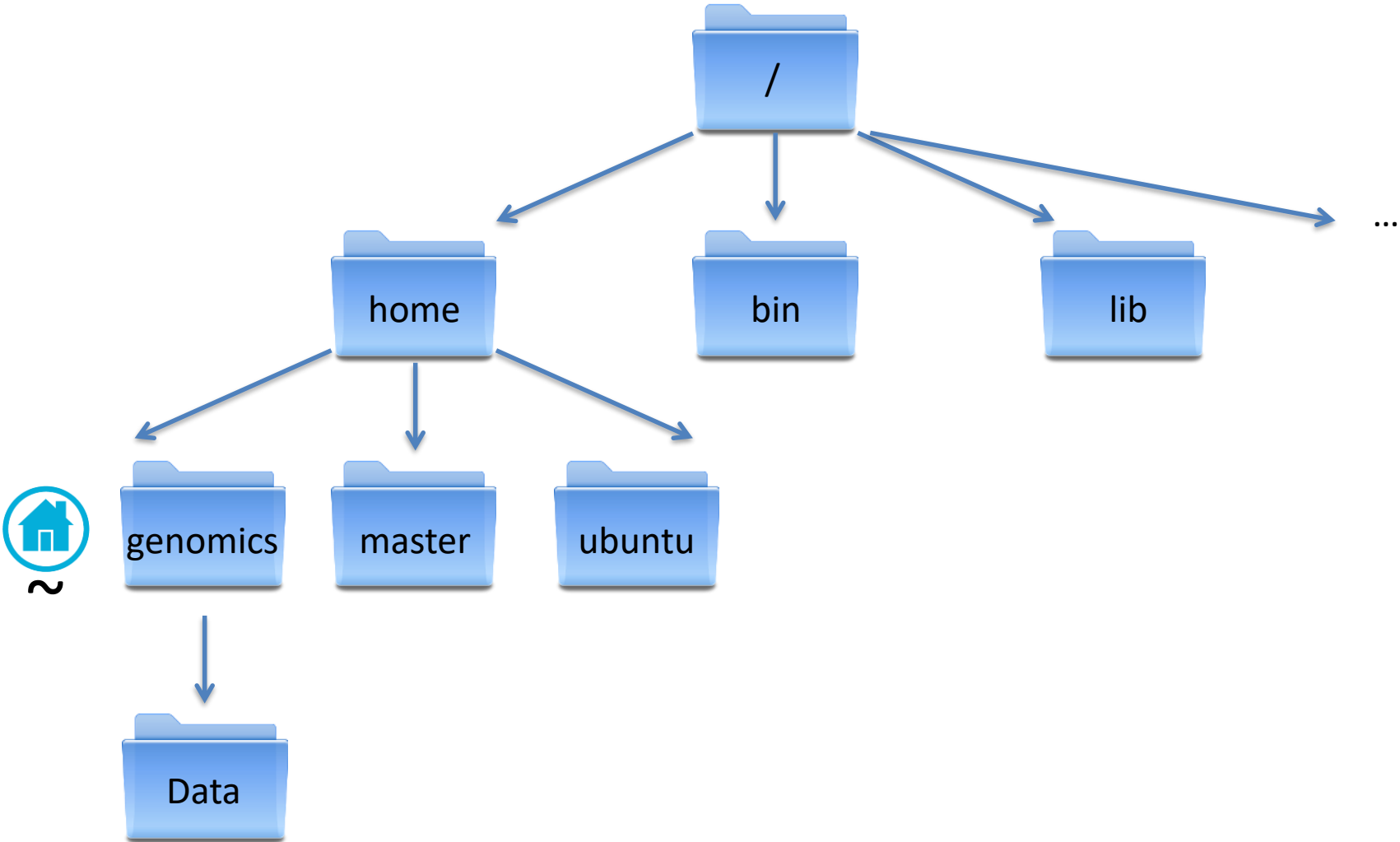
```
$ mkdir Data
```

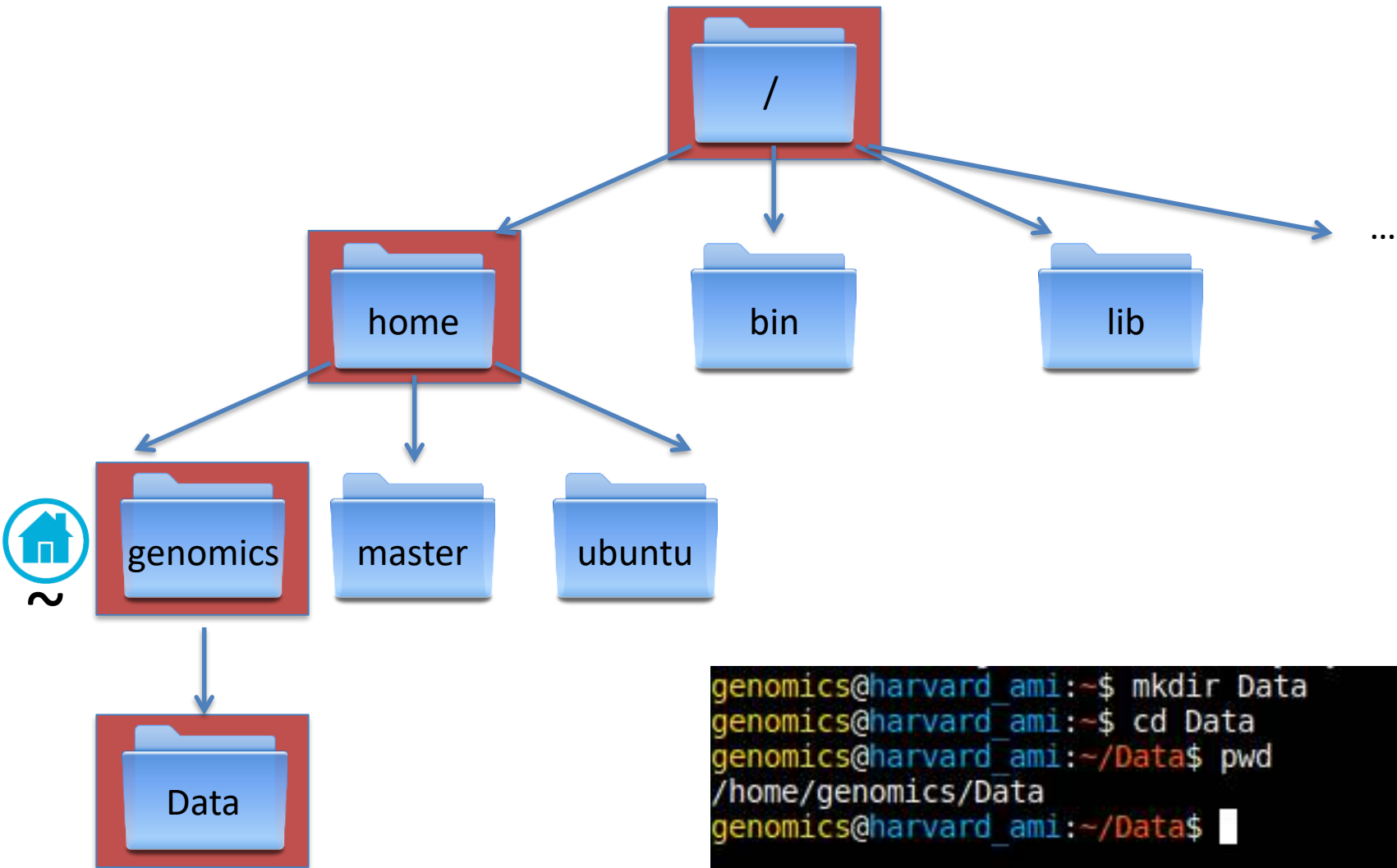
Change into this directory

```
$ 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.





```
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
```

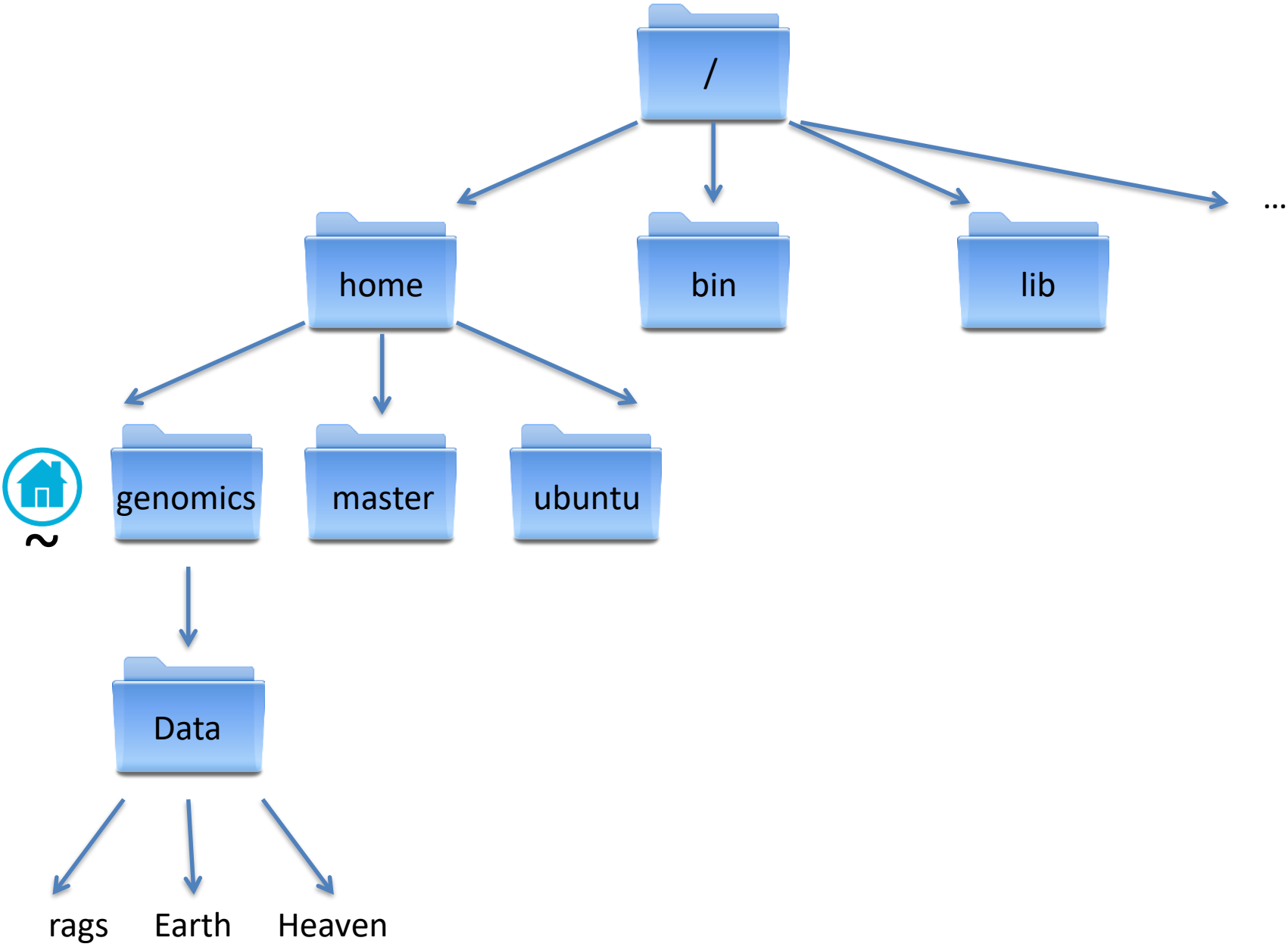
And another two

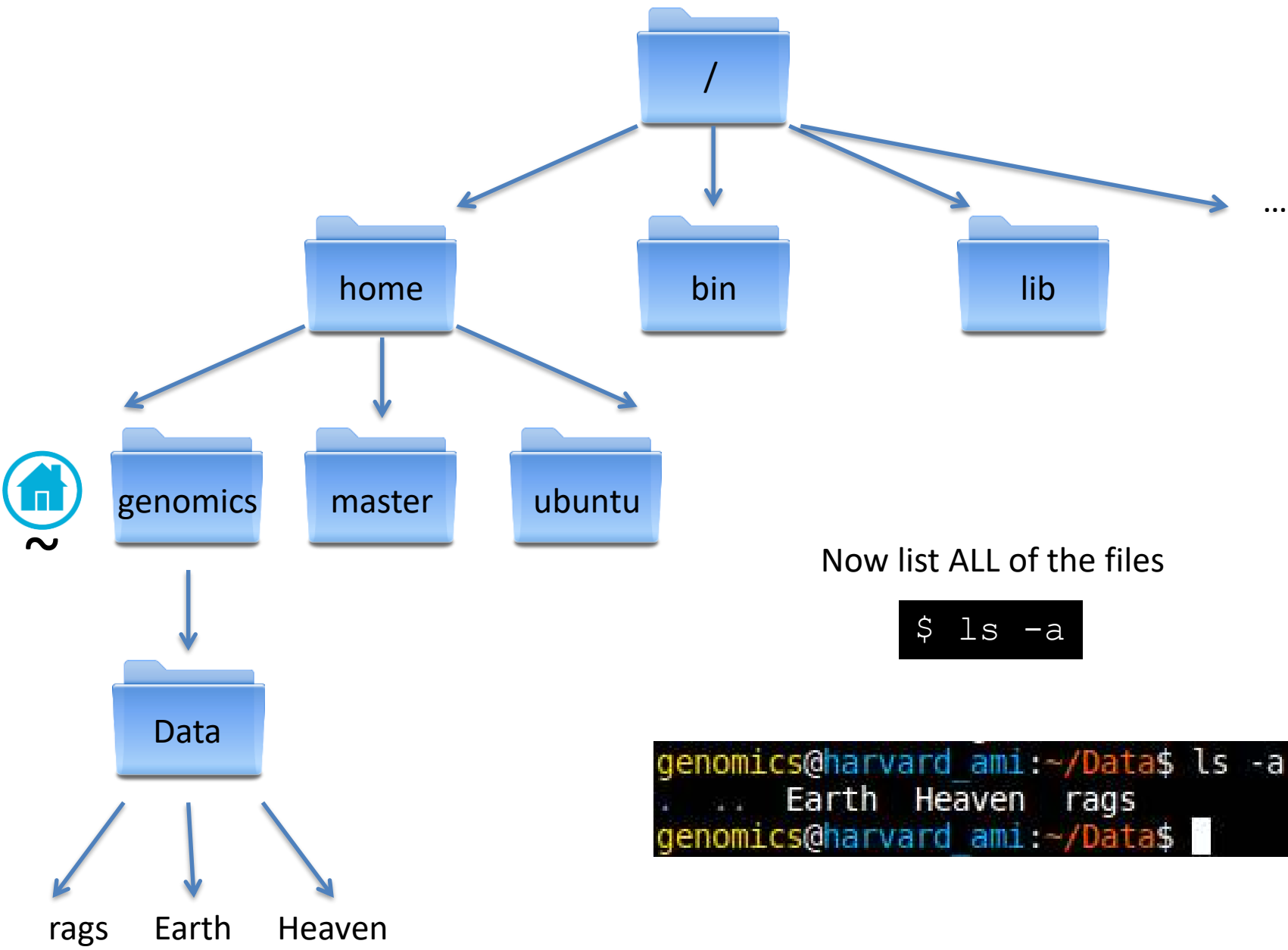
```
$ touch Earth Heaven
```

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$
```

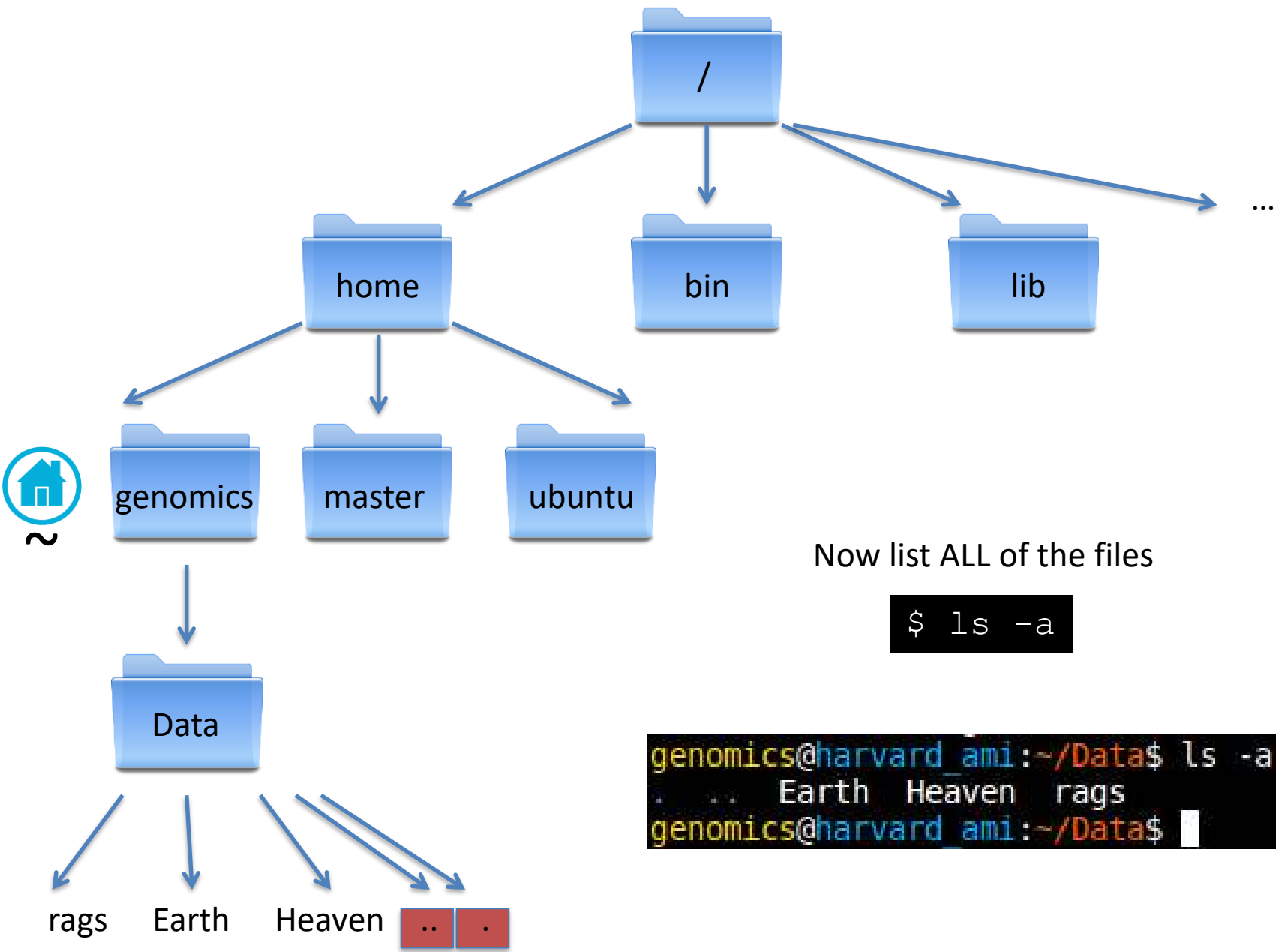




Now list ALL of the files

```
$ ls -a
```

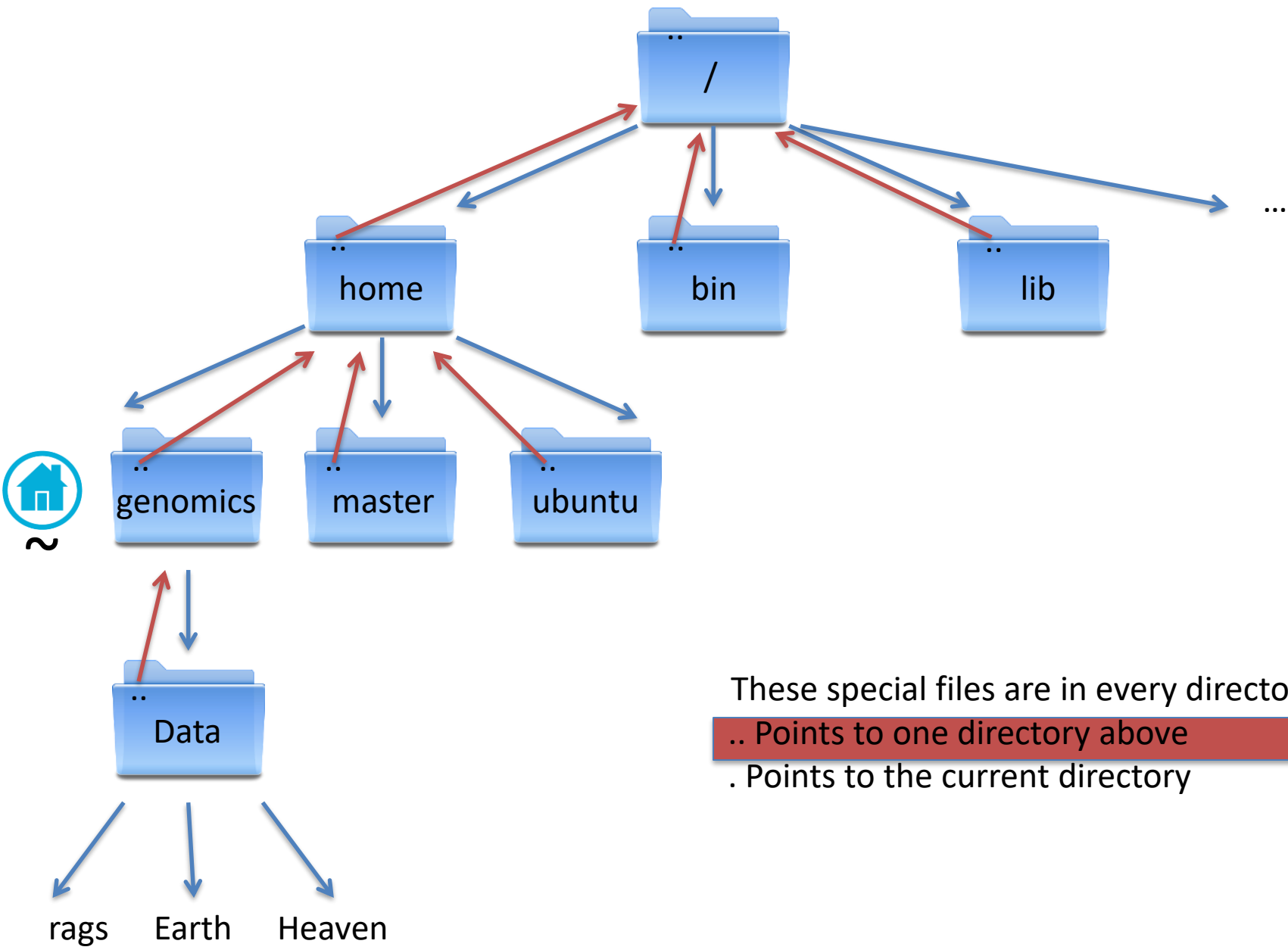
```
genomics@harvard_ami:~/Data$ ls -a
.      Earth Heaven rags
genomics@harvard_ami:~/Data$
```

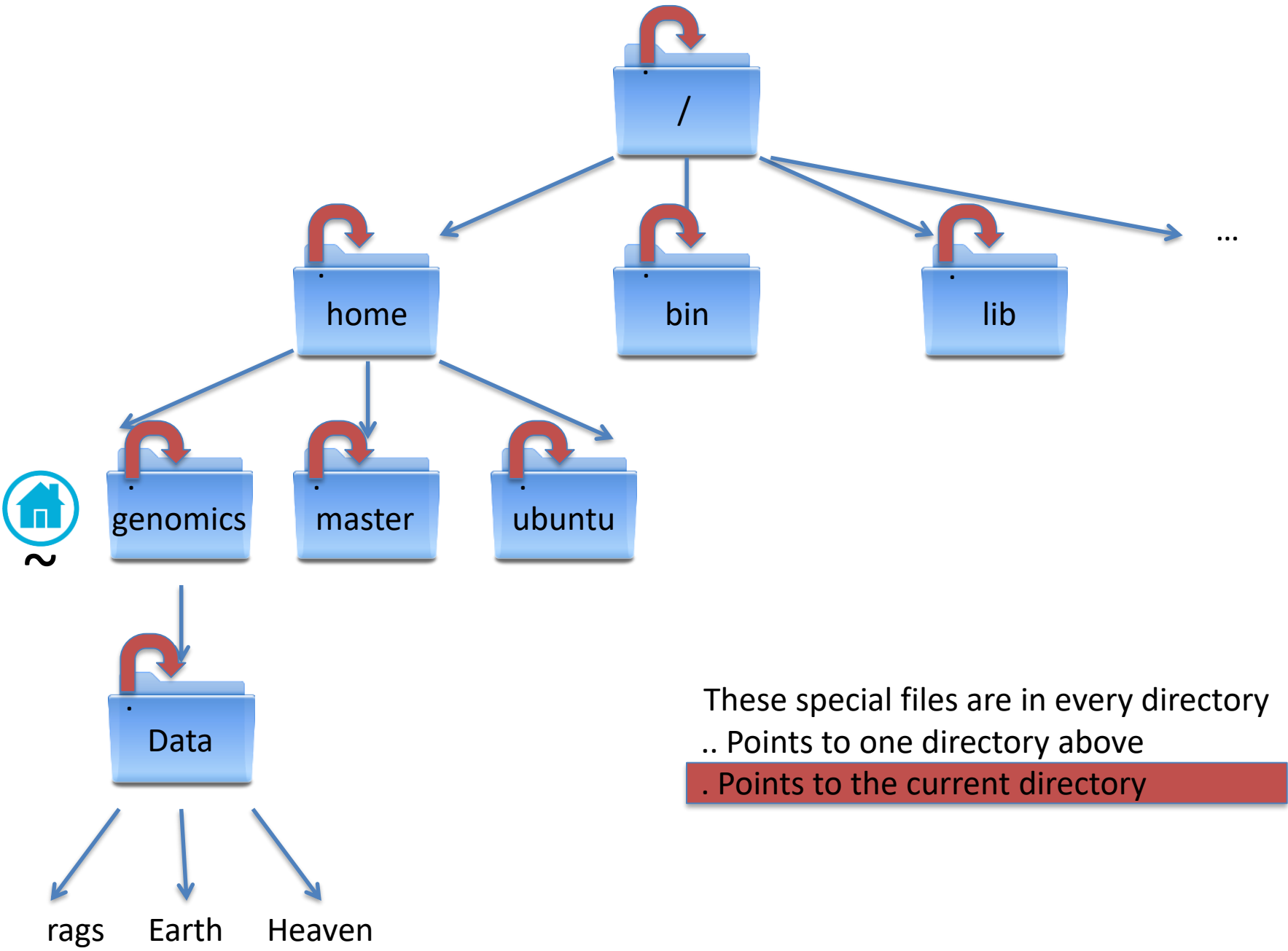


Now list ALL of the files

```
$ ls -a
```

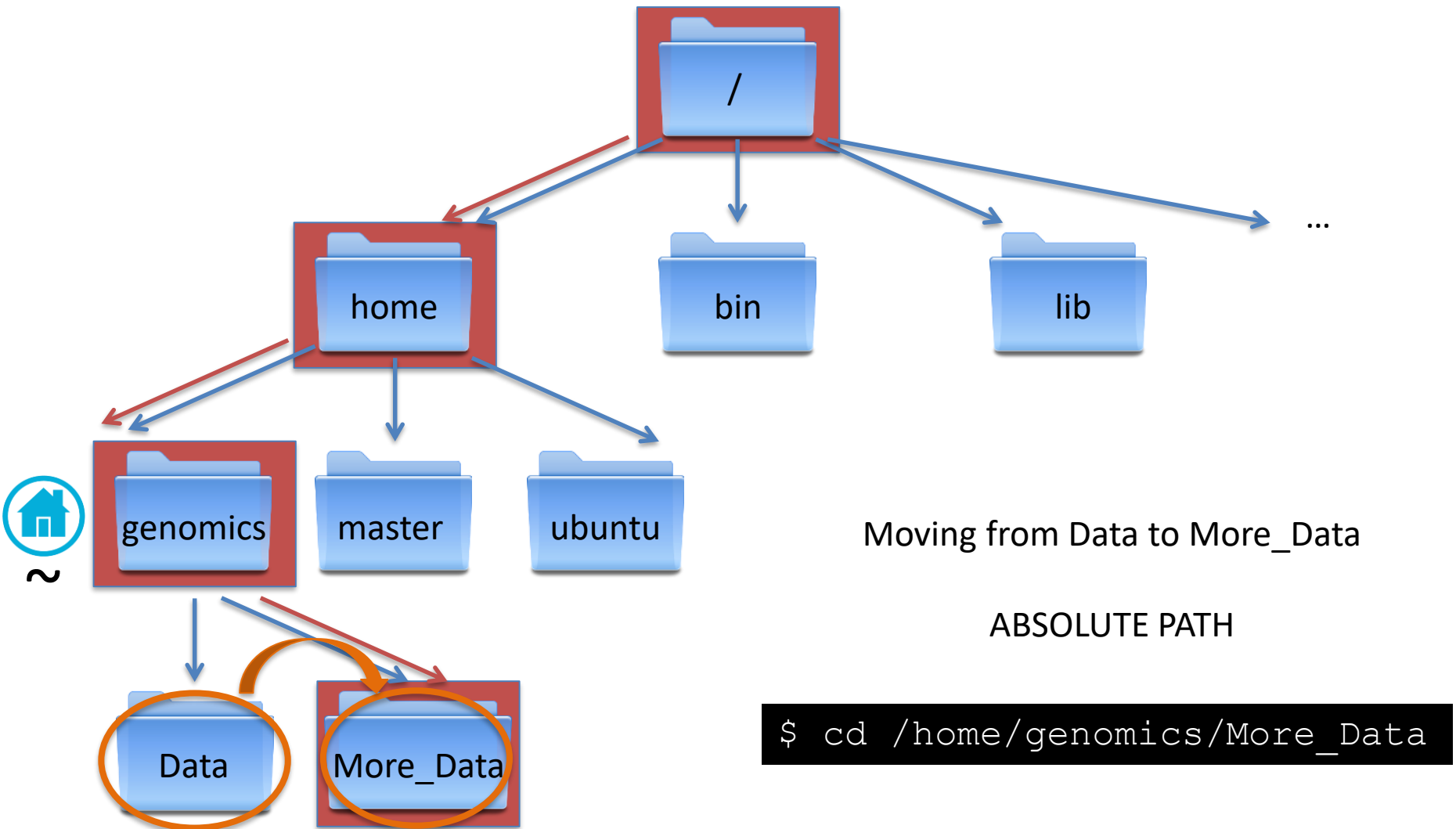
```
genomics@harvard_ami:~/Data$ ls -a
.  ..  Earth  Heaven  rags
genomics@harvard_ami:~/Data$
```



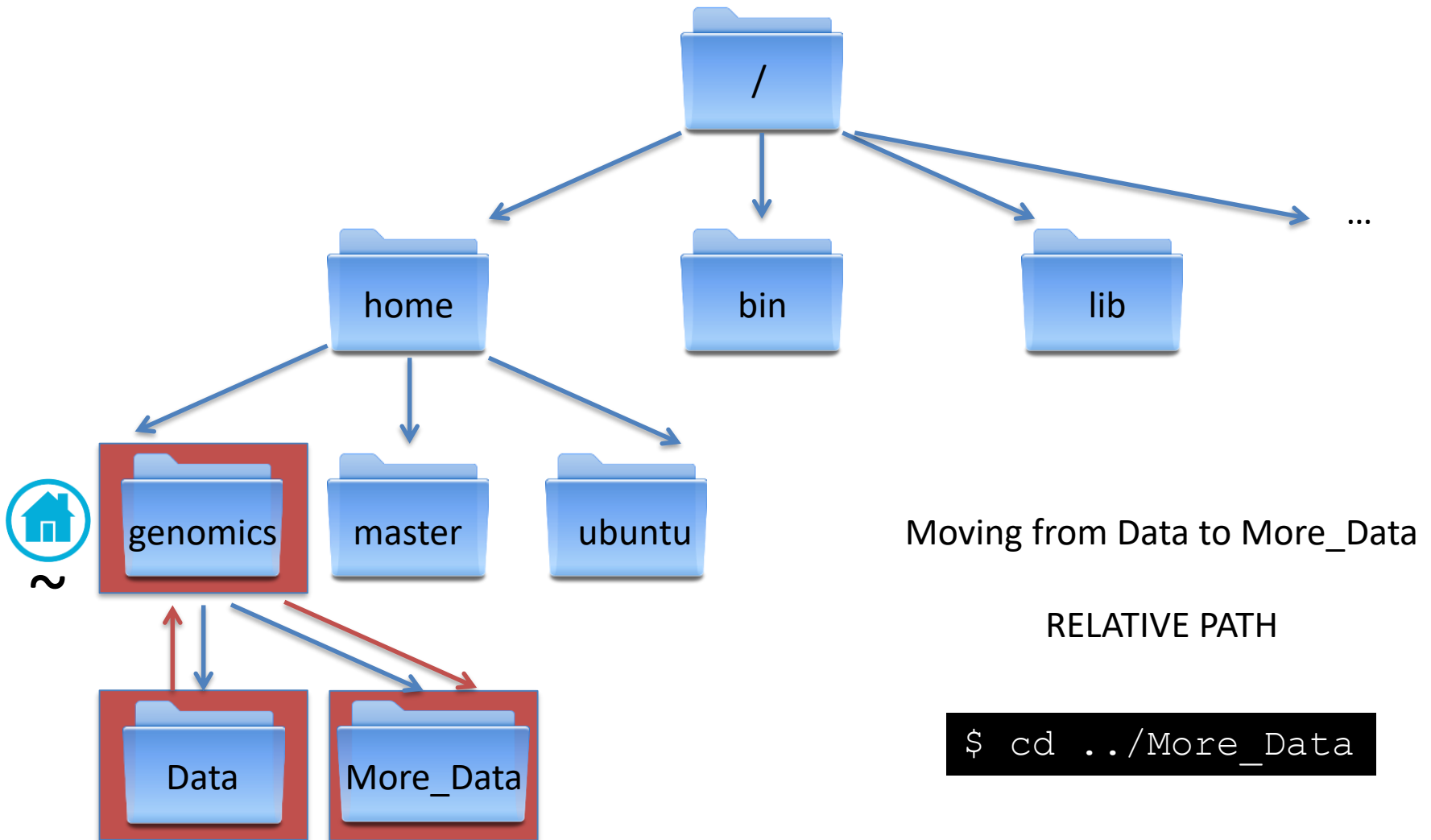


ABSOLUTE AND RELATIVE PATHS: GETTING FROM ONE PLACE TO ANOTHER

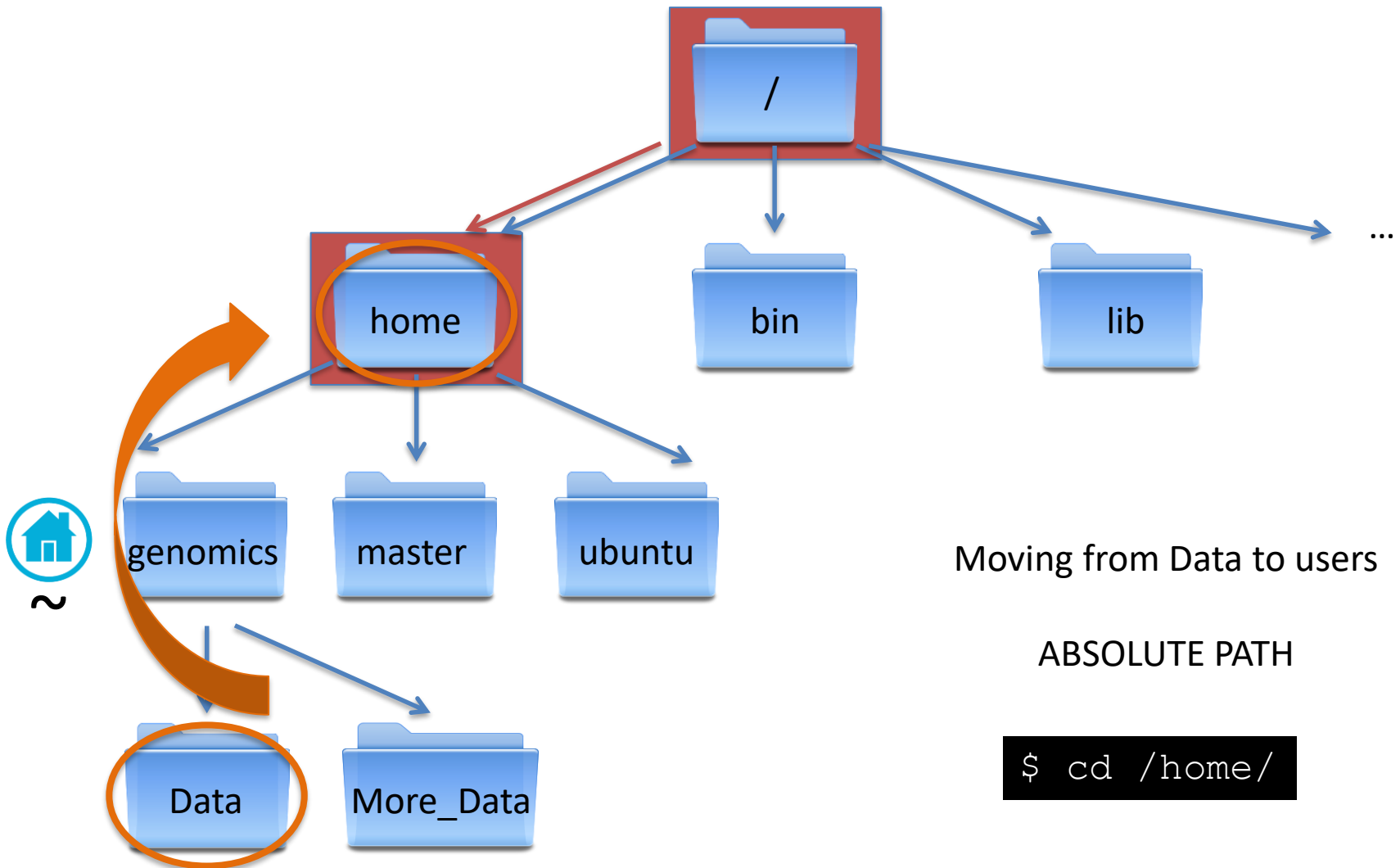
Absolute and Relative Paths



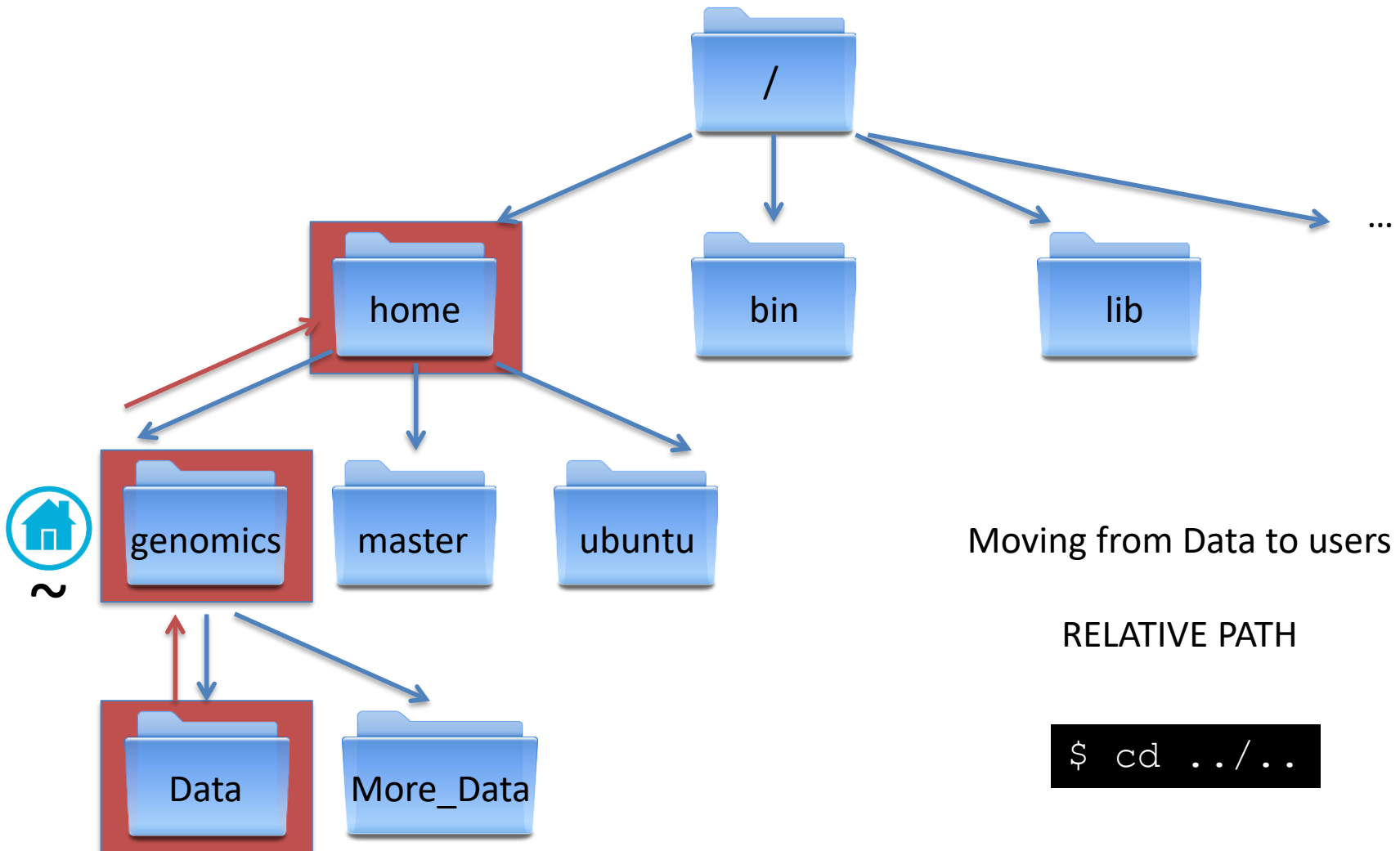
Absolute and Relative Paths



Absolute and Relative Paths



Absolute and Relative Paths



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

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          kill           netcat         rm            touch
btrfs-debug-tree dash          kmod           netstat        rmdir         true
btrfs-find-root date           less           networkctl     rnano         udevadm
btrfs-image     dd            lessecho       nisdomainname  run-parts     ulockmgr_server
btrfs-map-logical df             lessfile       ntfs-3g        sed           umount
btrfs-select-super dir           lesskey        ntfs-3g.probe  setfacl       uname
btrfs-show-super dmesg         lesspipe       ntfs-3g.secaudit setfont        uncompress
btrfstune       dnsdomainname lesspipe        ntfs-3g.usermap setupcon       unicode_start
btrfs-zero-log  domainname    ln             ntfsctl        sh            vdir
bunzip2         dumpkeys      loadkeys       ntfscluster    sh.distrib    vmmouse_detect
busybox         echo          login          ntfsncmp       sleep          wdctl
bzipcat         ed            loginctl       ntfsfallocate  ss            which
bzipcmp         egrep         lowntfs-3g     ntfsfix        static-sh     whiptail
bzipdiff        false         ls             ntfsinfo       stty          ypsdomainname
bzegrep         fgconsole     lsblk          ntfsls         su            zcat
bzexe           fgrep         lsmode         ntfsmove       sync          zcmp
bzfgrep         findmnt       mkdir          ntfsmove       systemctl     zdiff
bzgrep          fsck.btrfs   mkfs.btrfs     ntfstruncate   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

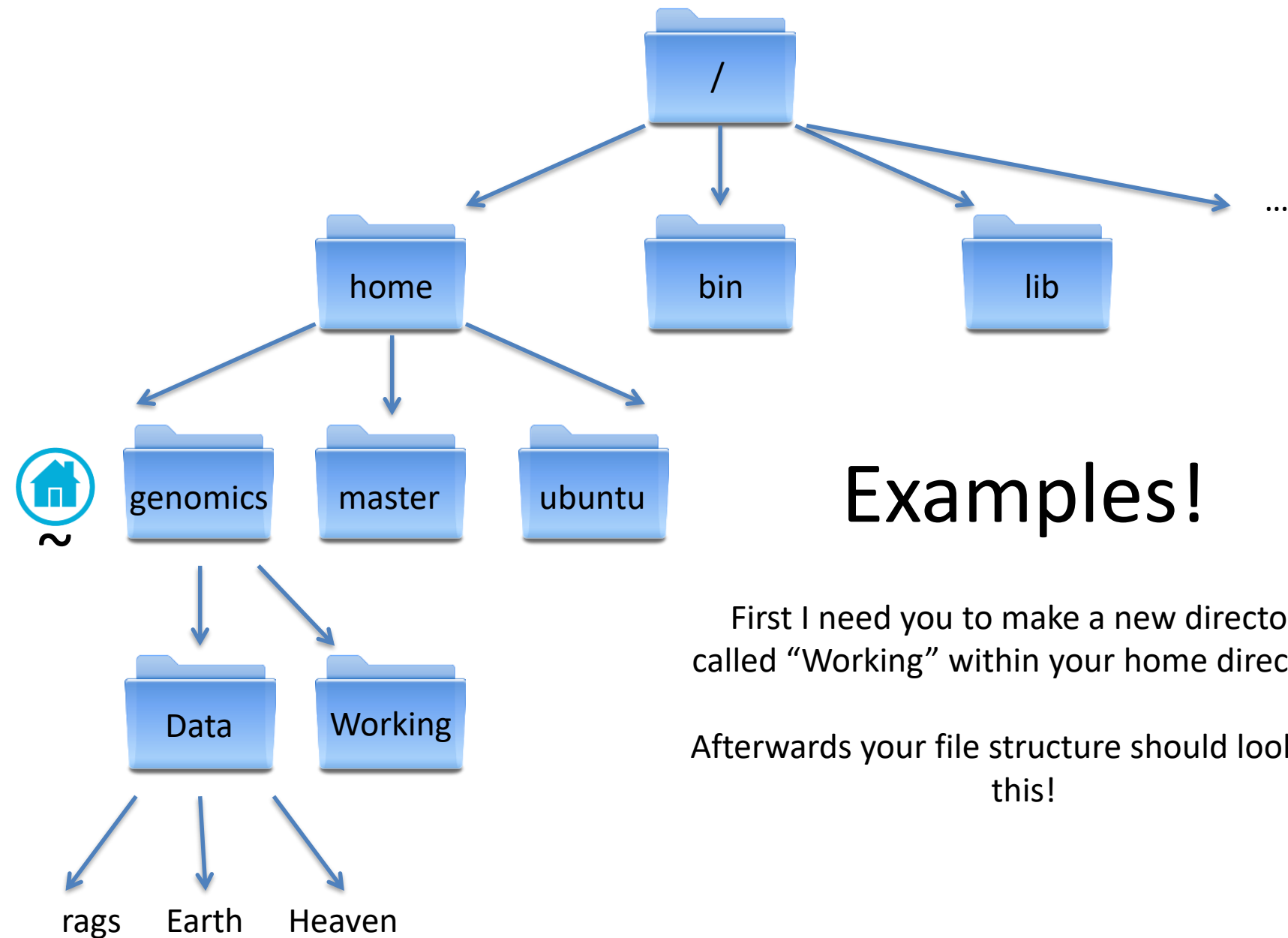
```
$ man ls
```

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
```

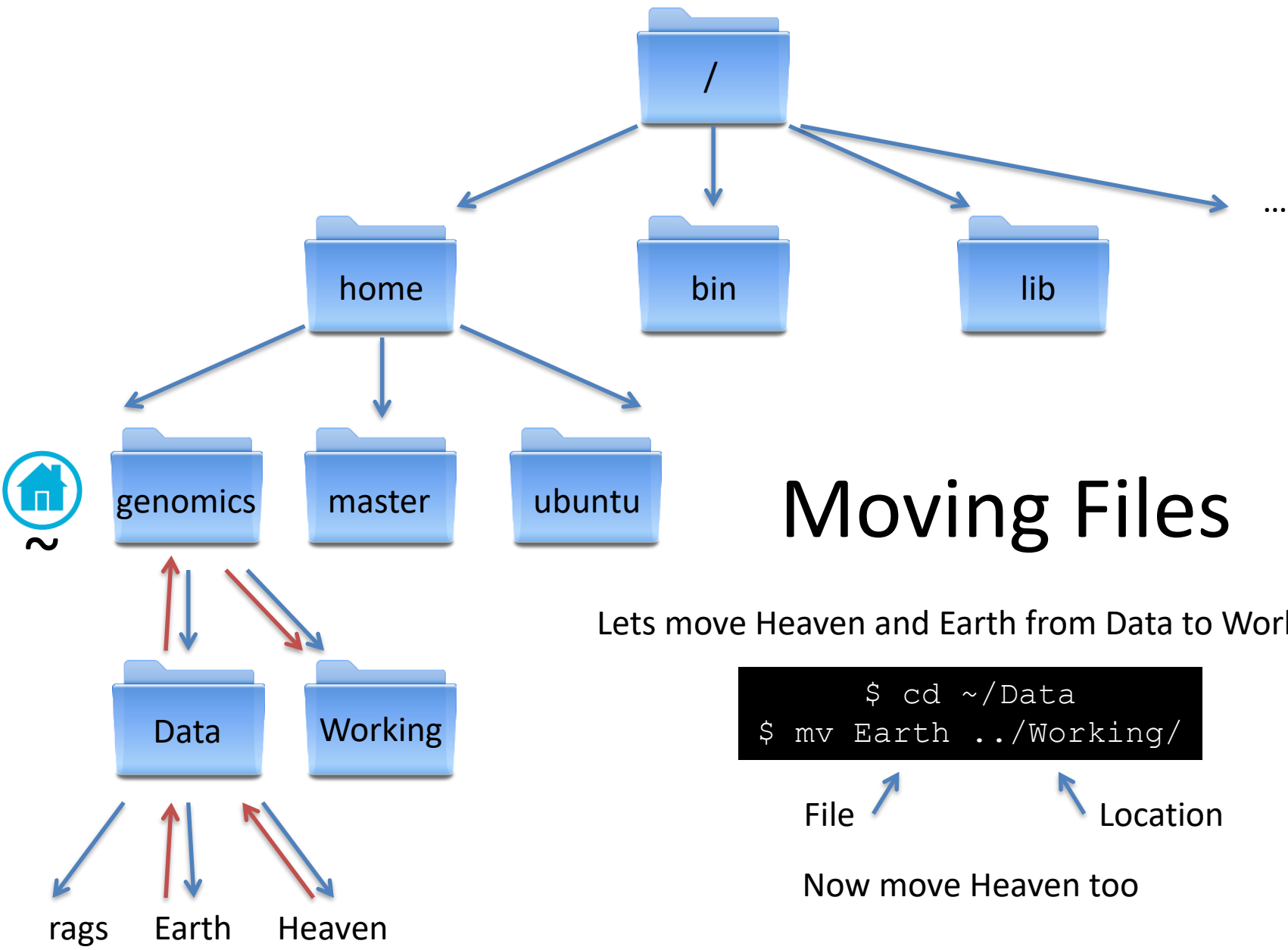




Examples!

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

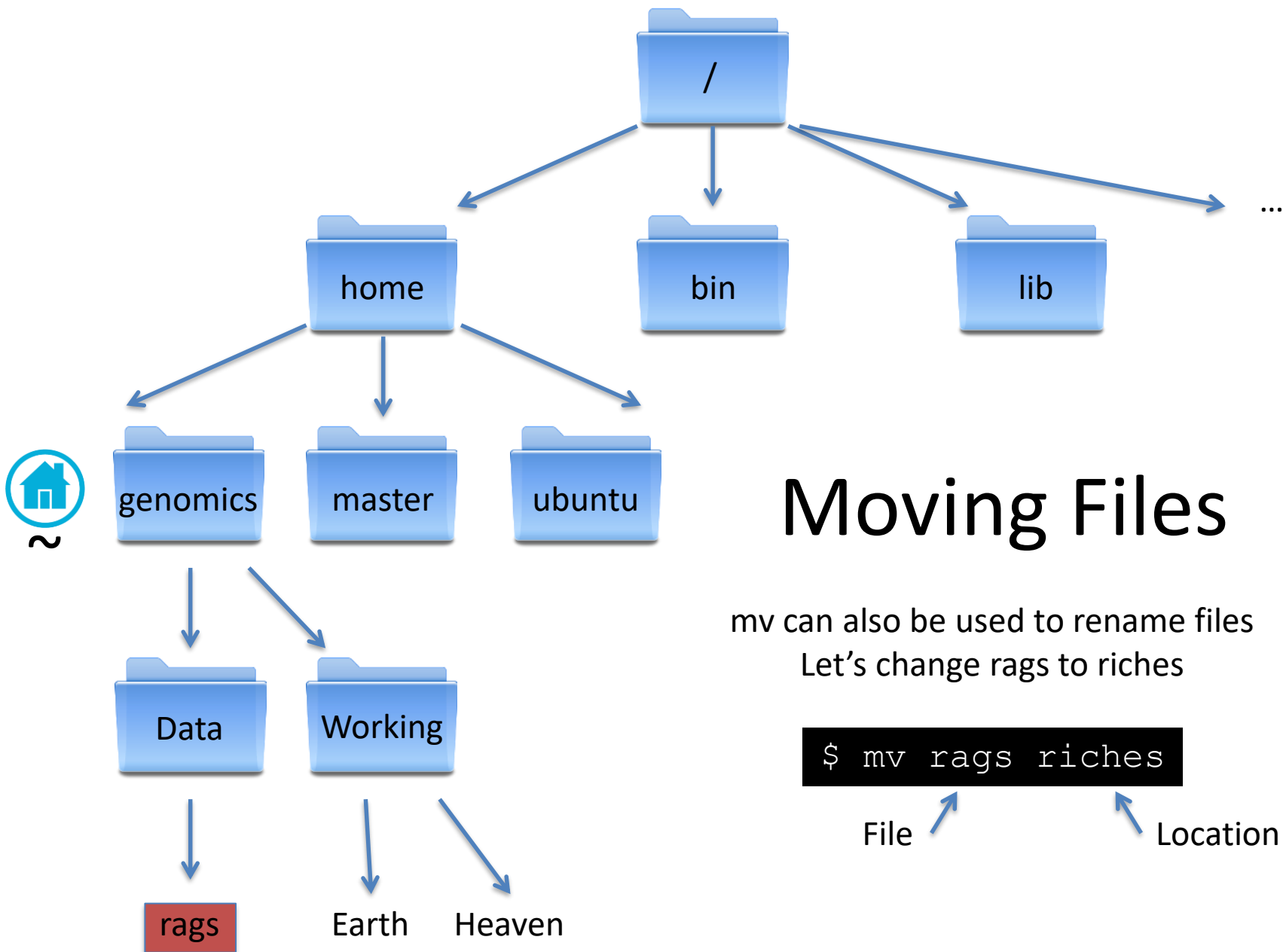
Lets move Heaven and Earth from Data to Working

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

File Location

Now move Heaven too

REMEMBER TAB COMPLETE!



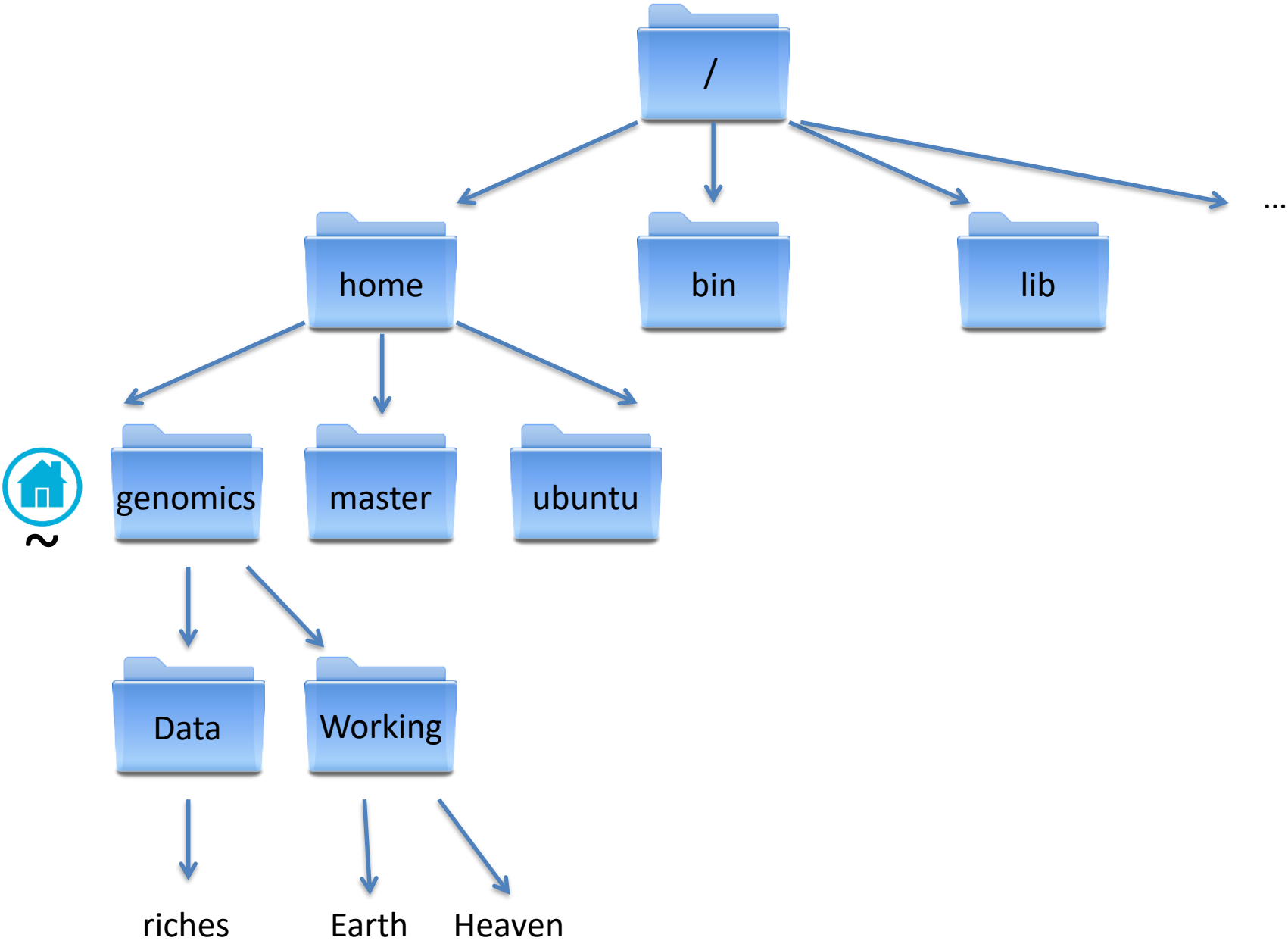
Moving Files

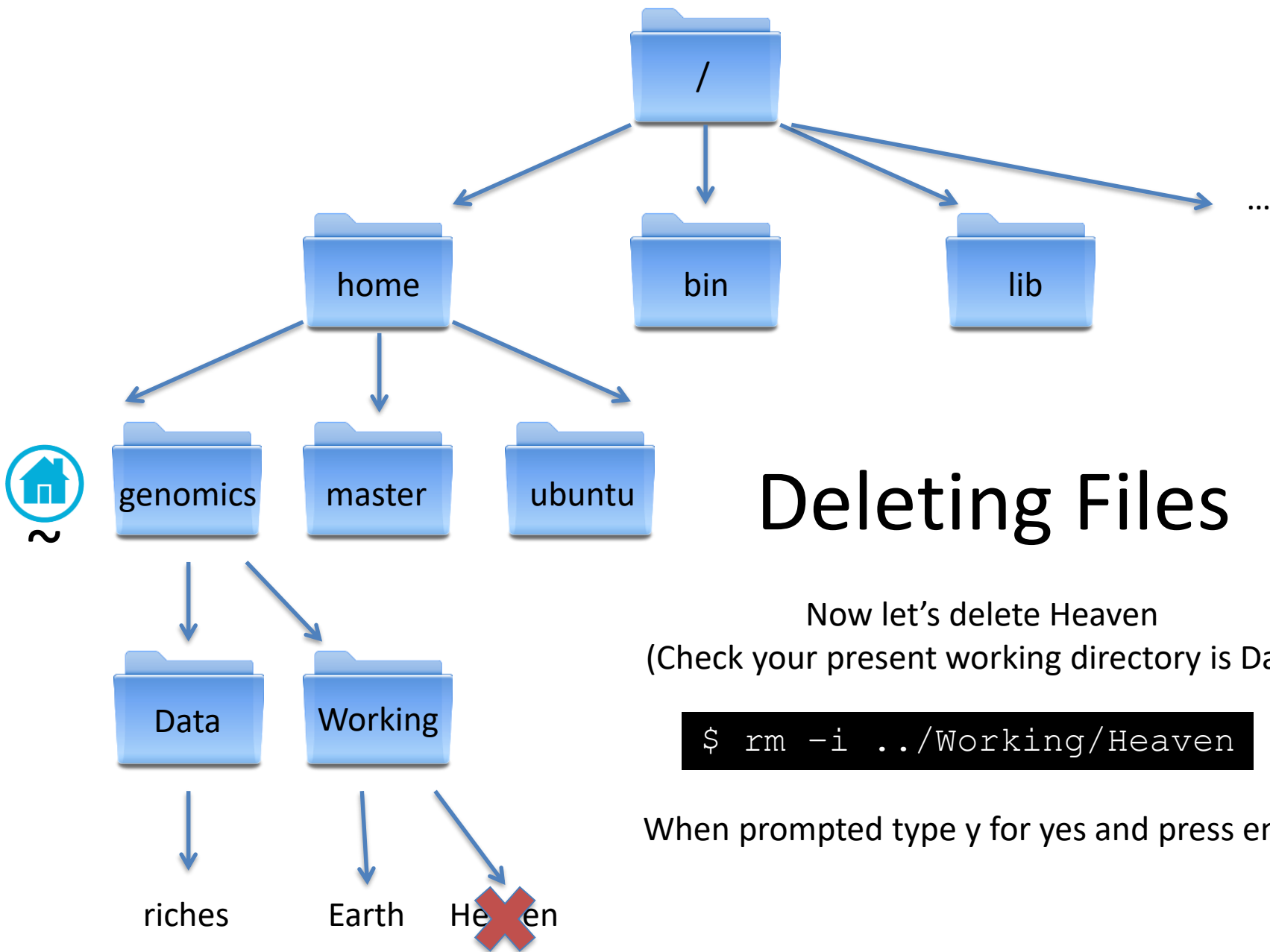
`mv` can also be used to rename files
Let's change rags to riches

```
$ mv rags riches
```

File

Location



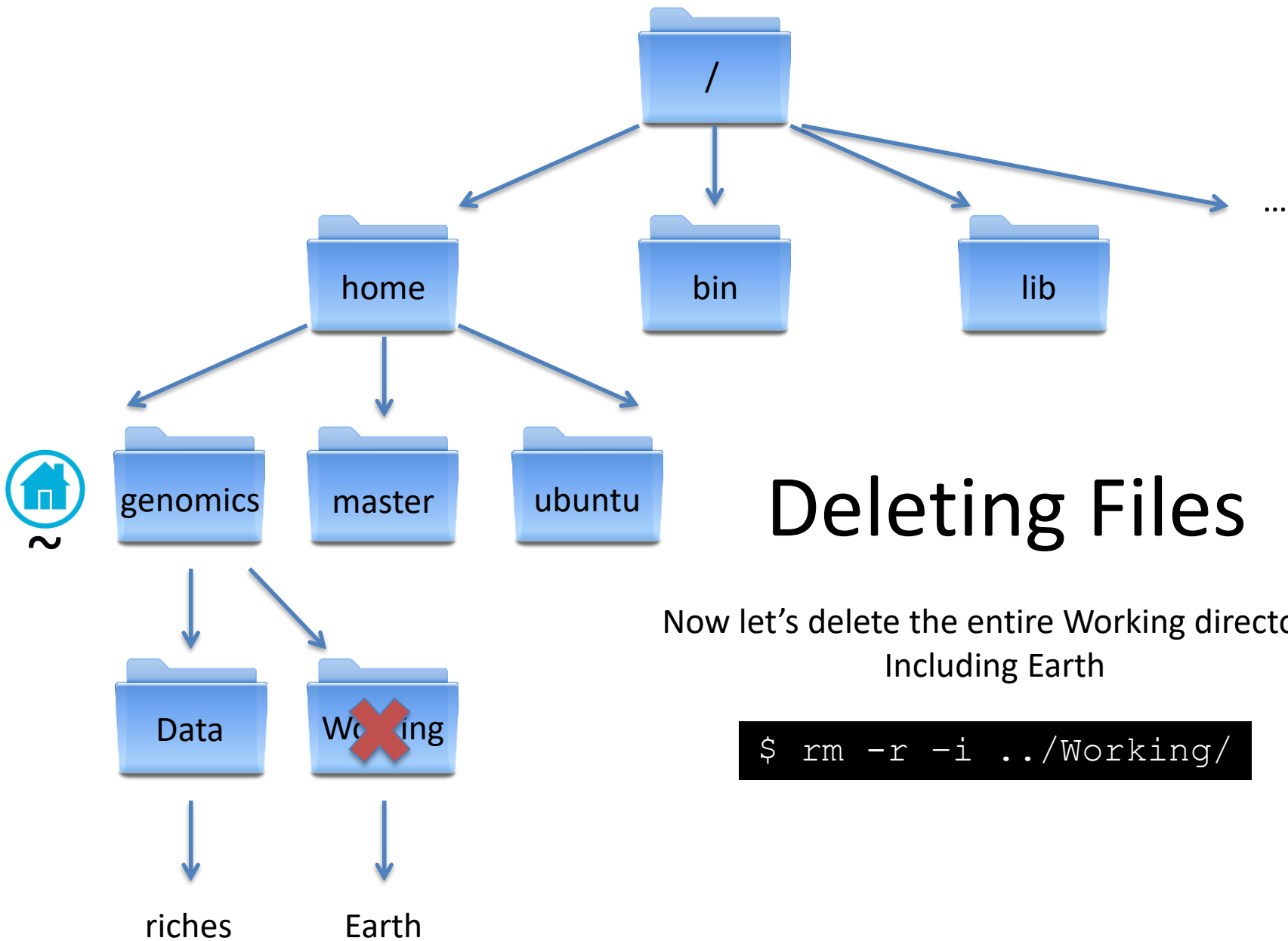


Deleting Files

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

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

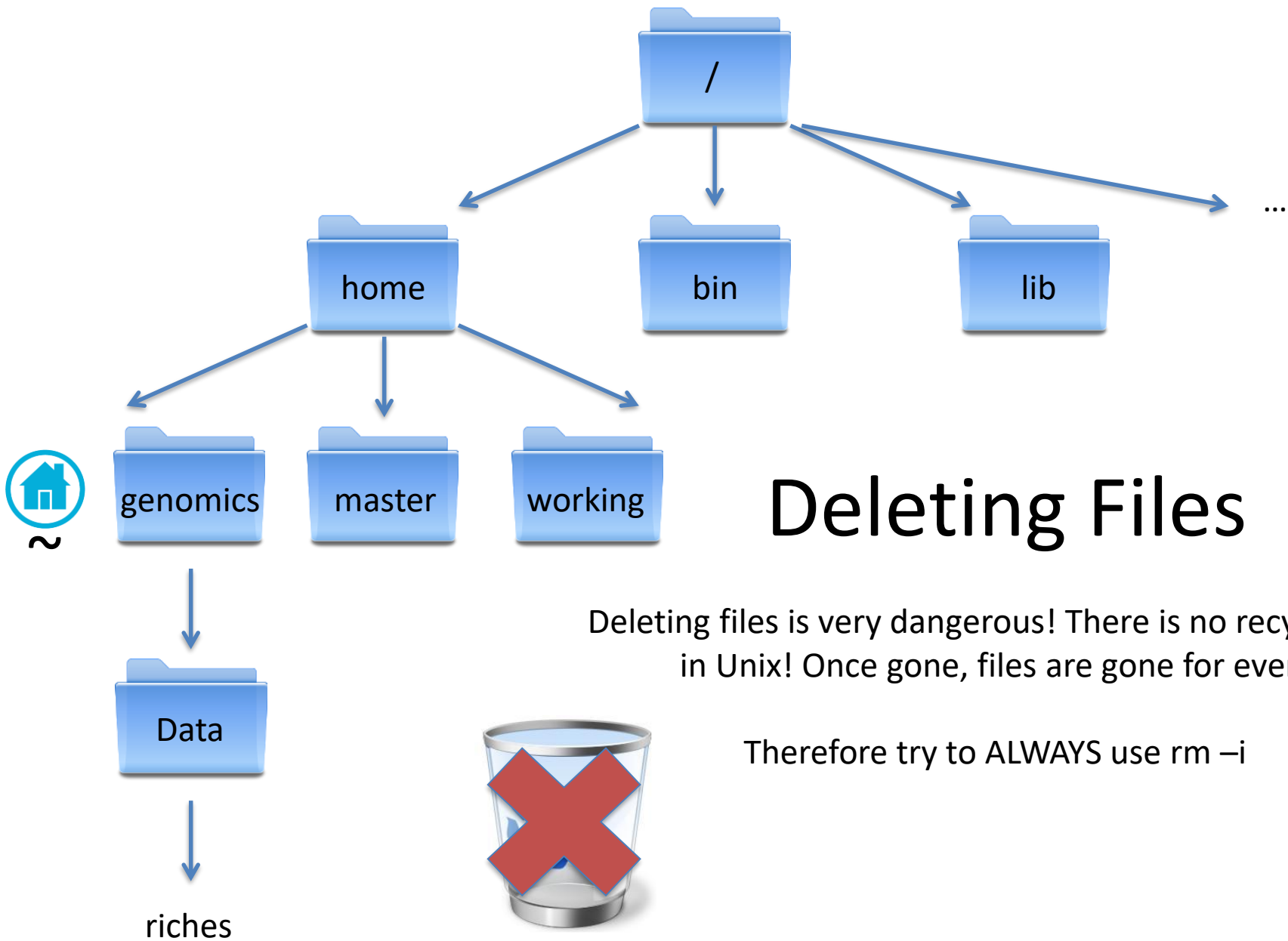
When prompted type y for yes and press enter



Deleting Files

Now let's delete the entire Working directory
Including Earth

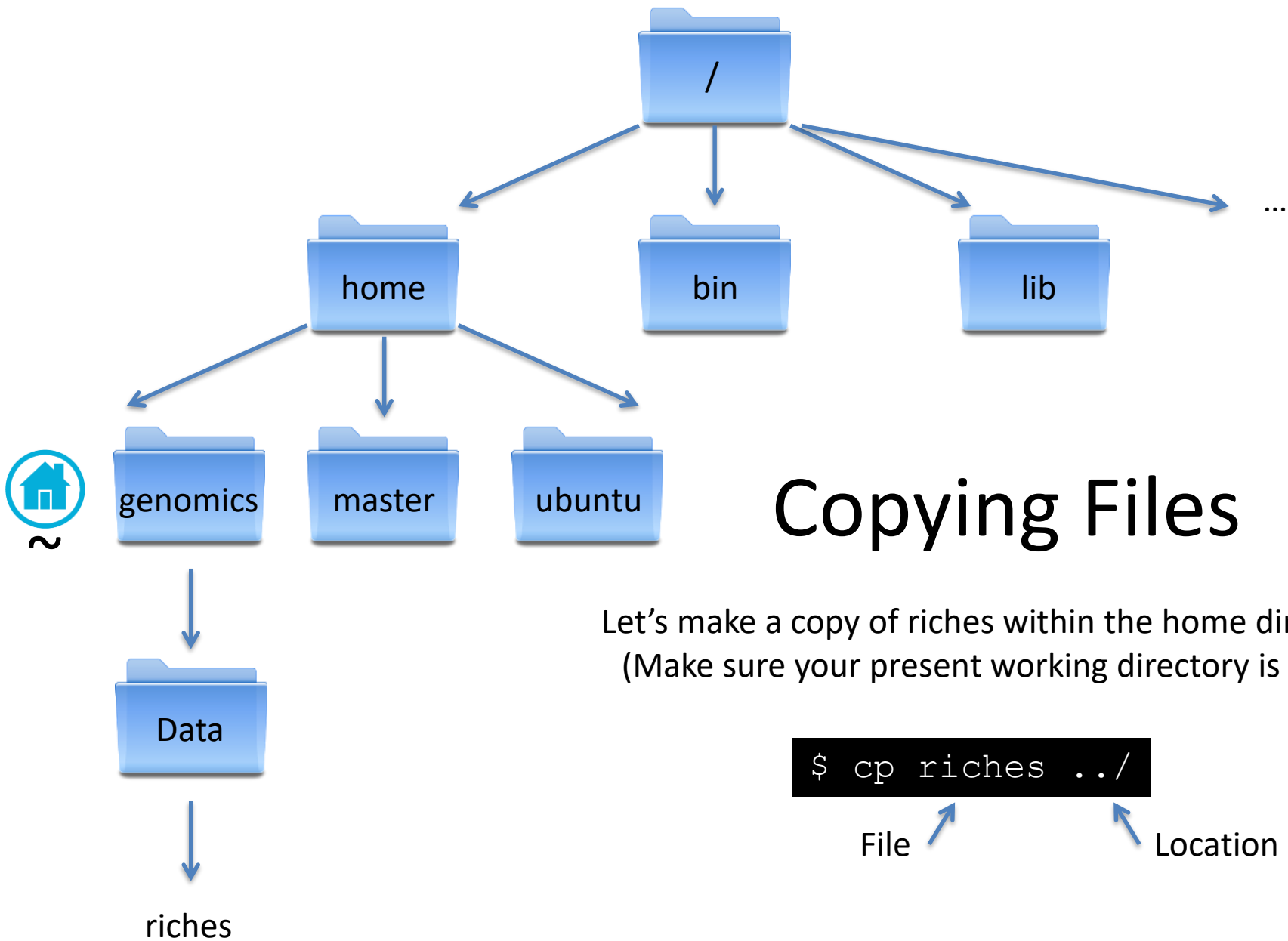
```
$ rm -r -i ../Working/
```

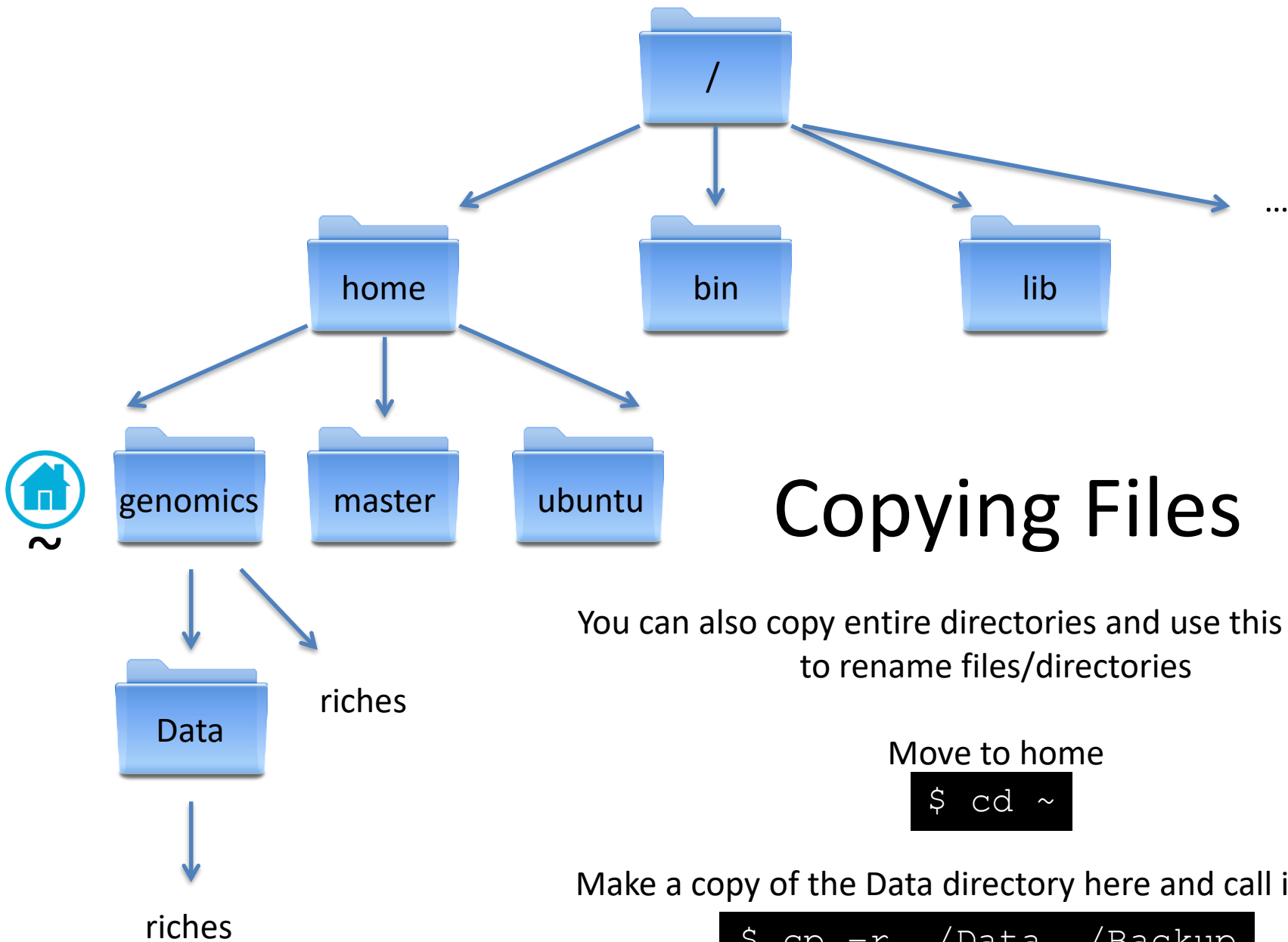


Deleting Files

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

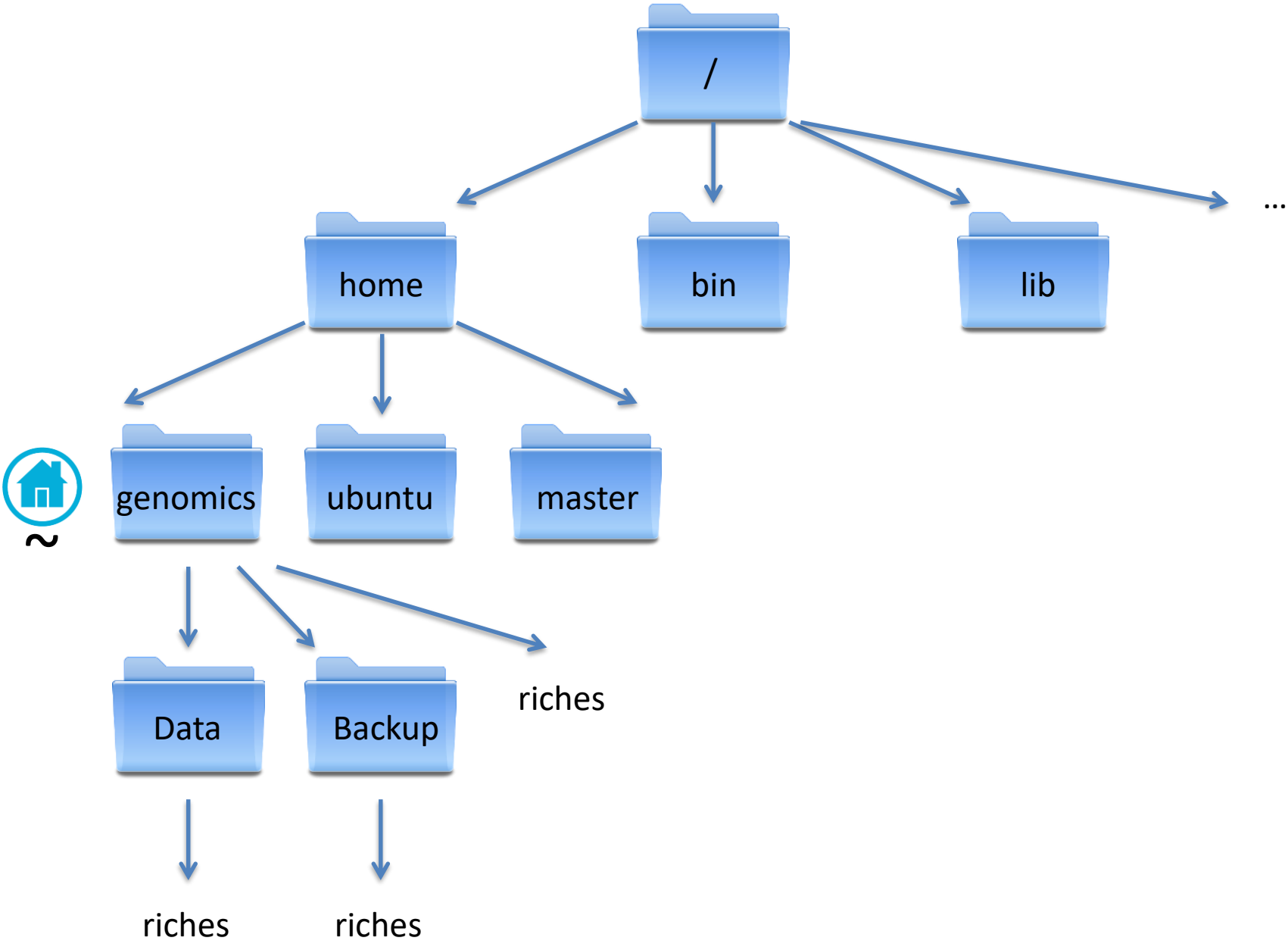
You can also copy entire directories and use this function to rename files/directories

Move to home

```
$ cd ~
```

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

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



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
```

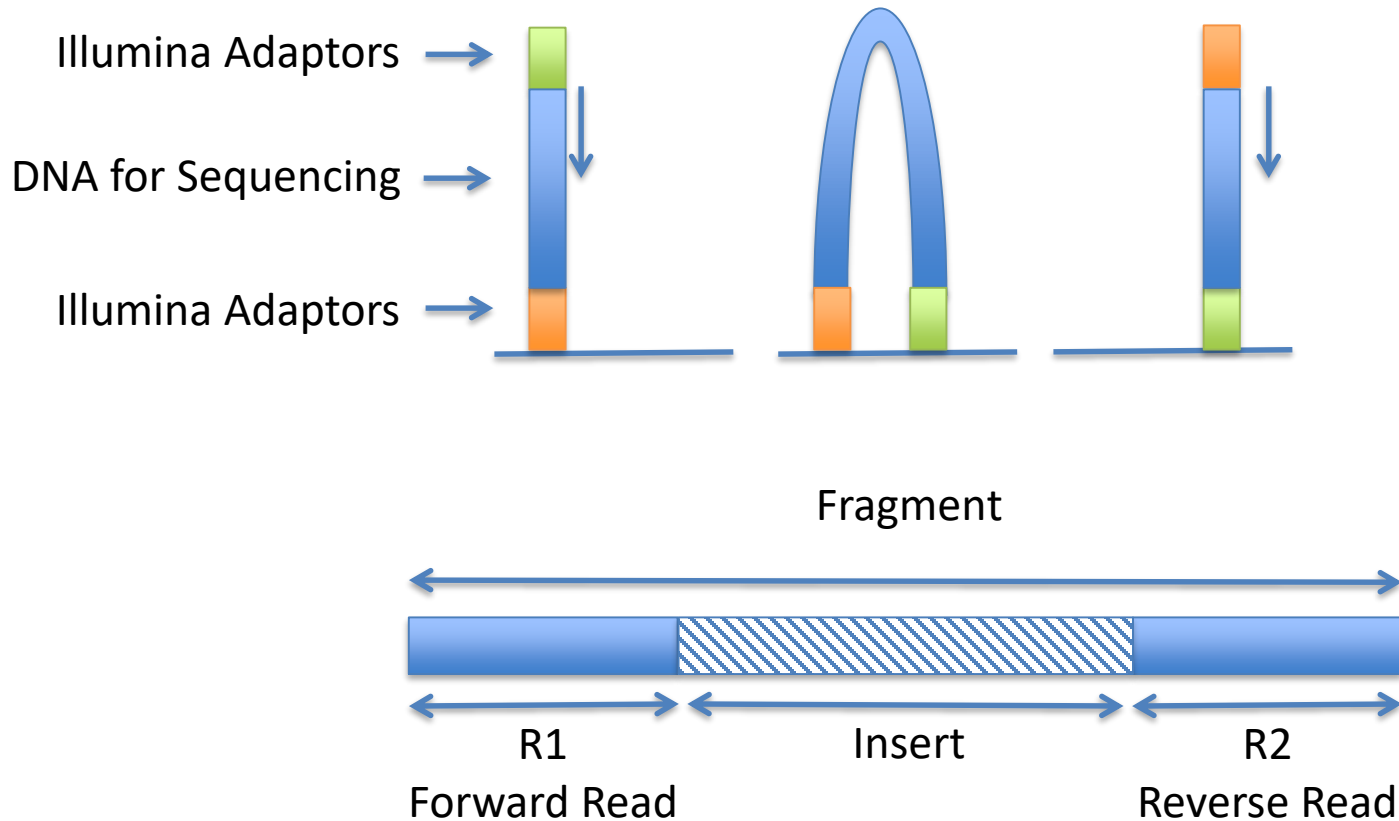
You should find a compressed directory:

```
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	

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
GCTGGTCAGCCAGGATAAAACCACCACTGACCCGATGGCGGTGTTGACTGGATCAACATGTTTGCCTGGCAGTGAACGAAGAGAACGCTGCTGG
CGGTCGCGTGGTGACTGCGCCGACTAACGGTGCGTGCGGGATTATCCCGGCAGTTCTGGCGTACTACGACAAGTTTATCCGCGAAGTGAACGCTAA
CTCACTGGCTCGTTACCTGCTGGTAGCCAGCGCCATTGGTACTCTTTATAAGATGAAC
+
????,BBBBDD<BD?<FGFFGFCFFIIHHIHHIGIIDHGIIDIIIIIGIFHHHIIHHHHIHHIII-HHDIIHHIHHIIFIHHHFIGIHHHH:HGI=G
HHHHFGIBEGHHHHBFBH=HHHFEEFGGEGHGDEBFG?FIFEG:8EBEEDG:GEEBGGGGGGGG(GEGGGGE?FECGFFCGFDGFGGEFEE??GG?
GCGE*FG?:6E/FGCEEHC: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

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

742640 lines THEREFORE 185660 reads

This is the
letter l

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.



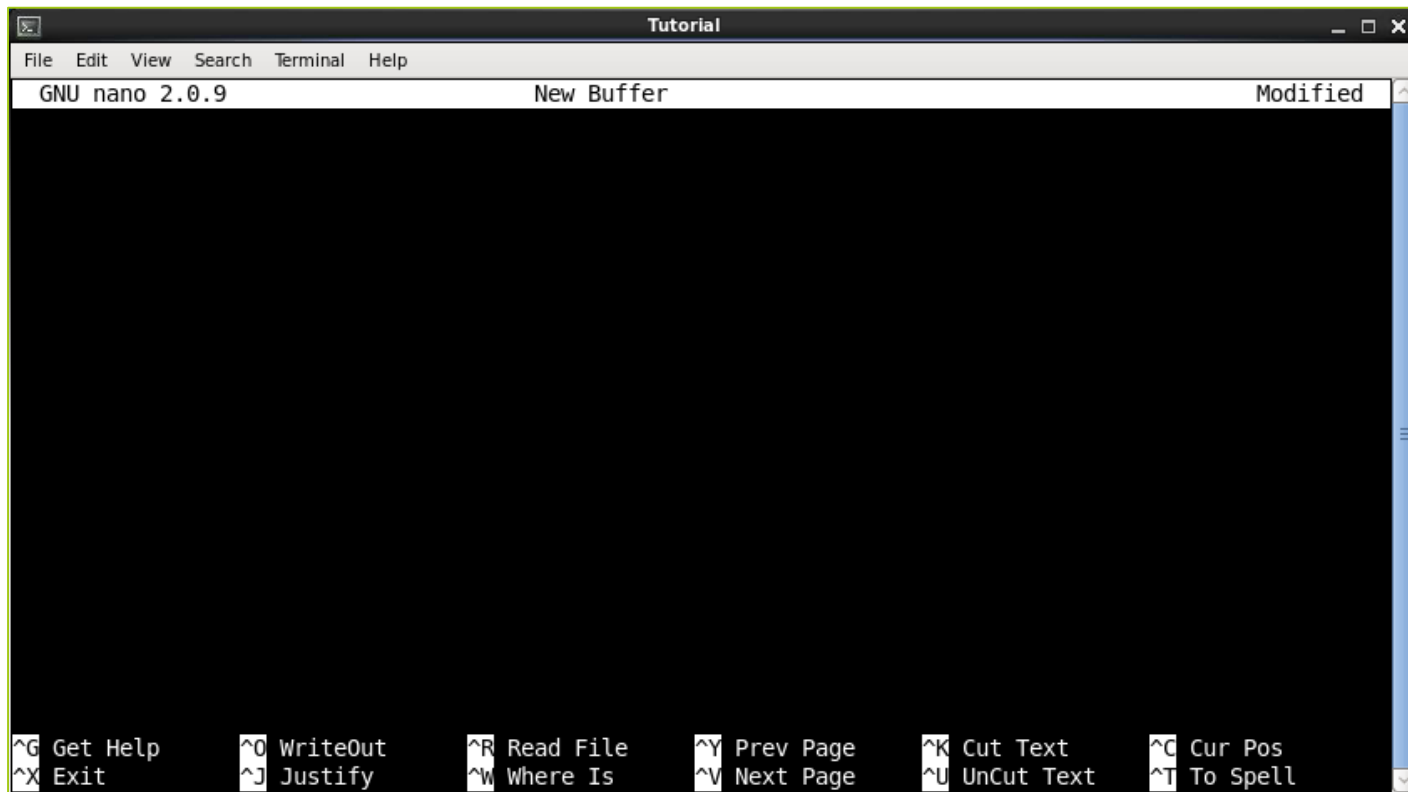
```

1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
194
```

Your First Script

Let's start by opening nano

```
$ nano
```



Key Nano Commands

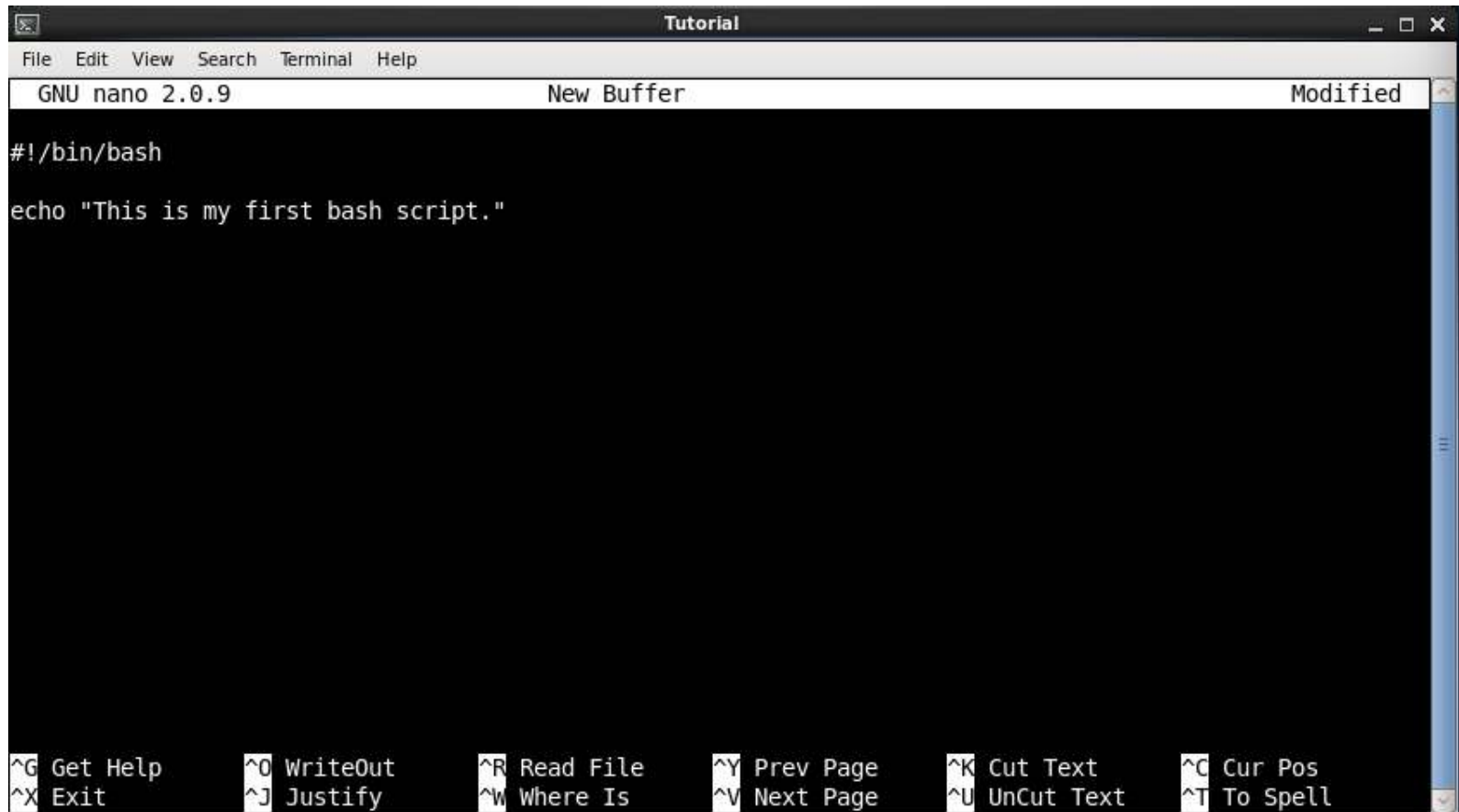
Ctrl + O – This saves the file. You will be asked for a file name.
Type the name and press enter.



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



The image shows a screenshot of the GNU nano 2.0.9 text editor. The window title is "Tutorial". The menu bar includes File, Edit, View, Search, Terminal, and Help. The status bar at the top shows "GNU nano 2.0.9", "New Buffer", and "Modified". The main editing area contains the following text:

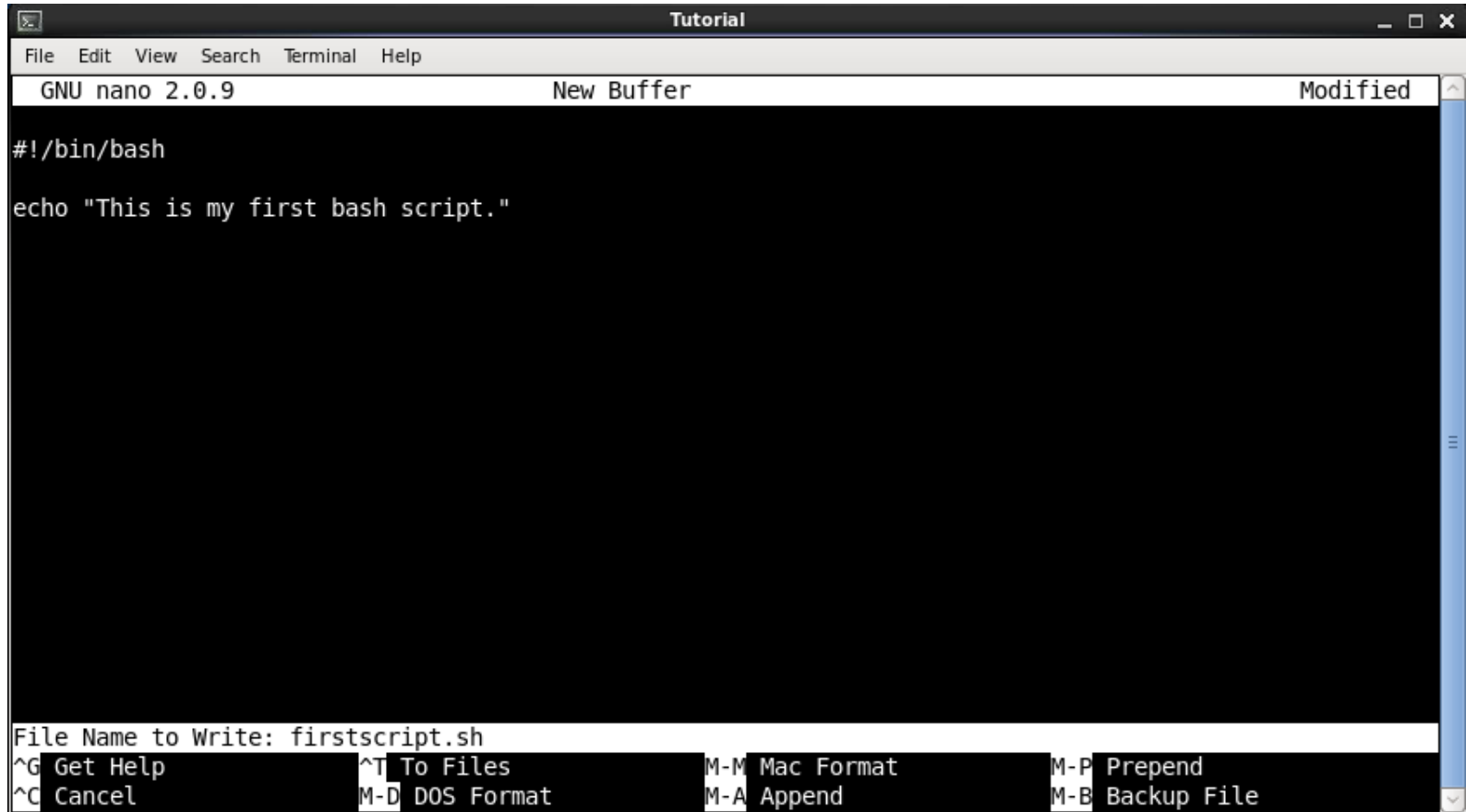
```
#!/bin/bash  
echo "This is my first bash script."
```

The bottom status bar displays various keyboard shortcuts for navigation and editing:

^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

A screenshot of the nano text editor window titled "Tutorial". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu bar, the status bar shows "GNU nano 2.0.9", "New Buffer", and "Modified". The main editing area is black with white text. It contains two lines: "#!/bin/bash" and "echo \"This is my first bash script.\"". At the bottom, a white bar shows "File Name to Write: firstscript.sh". Below this bar is a black bar with white text showing keyboard shortcuts: "^G Get Help", "^C Cancel", "^T To Files", "M-D DOS Format", "M-M Mac Format", "M-A Append", "M-P Prepend", and "M-B Backup File".

```
#!/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
```

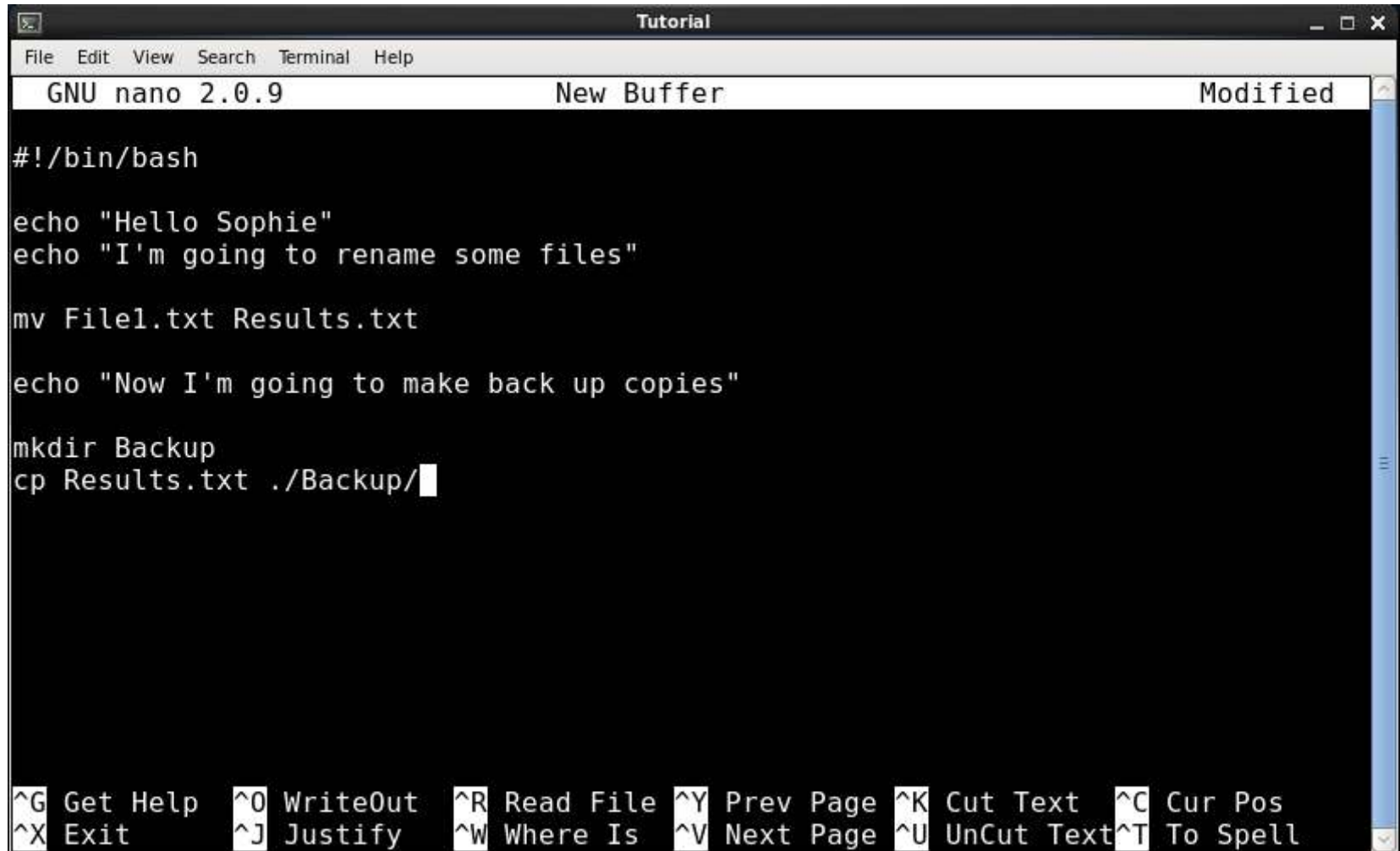
Reopen the same script:

```
$ 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...



The screenshot shows a terminal window titled "Tutorial" running the GNU nano 2.0.9 text editor. The editor's status bar at the top indicates "New Buffer" and "Modified". The script content is as follows:

```
#!/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/
```

At the bottom of the window, a help menu lists various keyboard shortcuts:

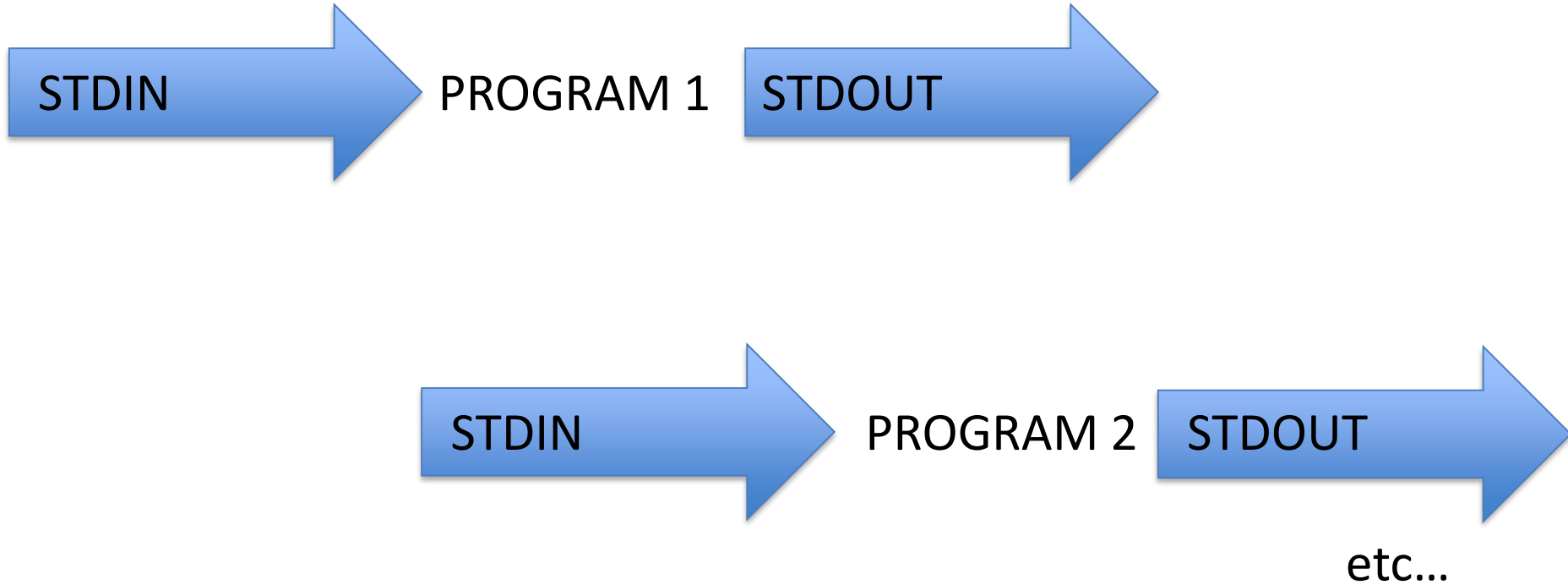
^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell

PIPELINES (TIME DEPENDING)

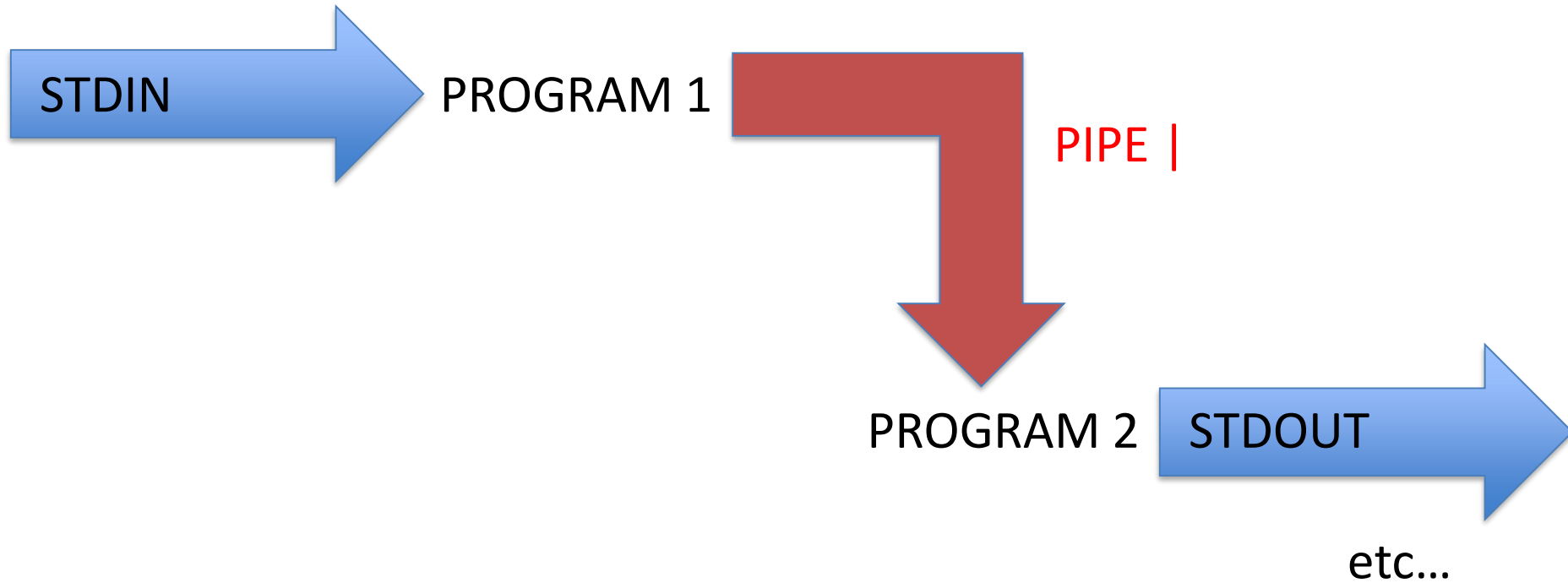
Pipelines



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:

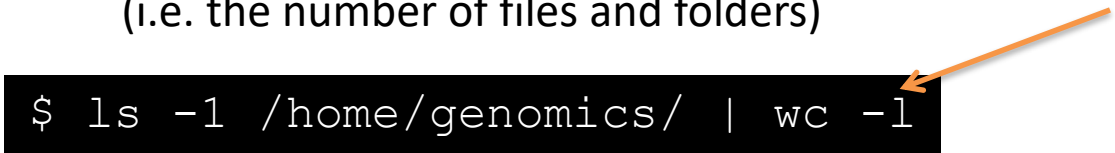
Number 1



```
$ ls -l /home/genomics/
```

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

Letter l



```
$ ls -l /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
#genome-build-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 frameX3B unlikely to encode a functional proteinX2C based on available experimen
l and comparative sequence data [Source:SGD;SBAcc: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 frameX3B unlikely to encode a functional proteinX2C based on available experimen
tal and comparative sequence dataX3B identified by gene-trappingX2C microarray-based expression analysisX2C and genome-wide homology searching [Source:SGD;SBAcc: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 functionX3B member of the seripauperin multigene family encoded mainly in
subtelomeric regions [Source:SGD;SBAcc: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 1891291
##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
#genome-build-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 experimen
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--(8%)
```

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

Any Questions So Far?

