UNIX for Bioinformatics

Instructor:
Nicholas E. Navin, Ph.D.
Department of Genetics
Department of Bioinformatics

Teaching Assistant:
Aislyn Schalck
Visweswaran Ravikumar

UNIX Background and History

- Unix was developed in the 1960's by MIT and AT&T Bell Laboratory
- Main developers were Dennis Ritchie and Ken Thompson
- Unics = Uniplexed Information and Computing System
- The first computer systems used command line interfaces
- UNIX provided one of the first powerful computing system for clients to log into servers and run applications remotely
- Unix was widely adopted by the academic community in the 1970s
- Unix is the operating system of choice for servers since it is highly stable and very secure for file permissions
- Unix has been developed into several different versions: Linux, Solaris, Mac OSX and many others

Dennis Ritchie & Ken Thompson 1972

High-Performance Computing Clusters (HPCC)

- Most academic universities have High-Performance Computing Clusters (HPCCs) that allow users to upload data and run programs, using thousands of processors

- Parallel Computing = computationally intensive programs that normally take weeks to run can be broken down into thousands of small jobs and run in parallel in hours

MD Anderson HPCC

Nautilus (HP Cluster)
- 336 Nodes
- 8064 processors
- 17 TB or RAM

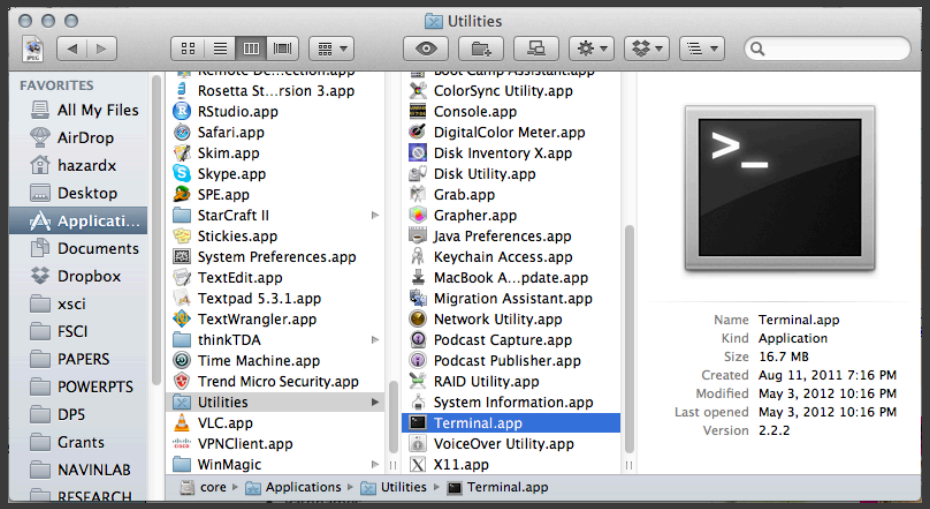
Shark (IBM Cluster)
- 240 nodes
- 5760 processors
- 92 TB RAM

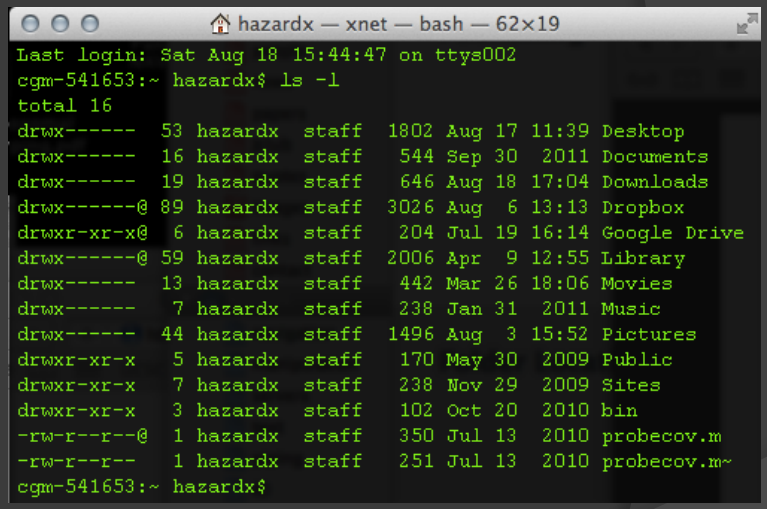
Storage
- 6.6 Petabytes of Storage

Network
- Infiniband switch 51.8 TB/sec

Apple OS (OSX) is based on the Unix Operating System

- BASH – is a unix shell command language that allows users to enter commands that instruct the UNIX kernel to run programs
- To make a shortcut in OSX, search for ‘Terminal.app’ on your Apple computer
- Drag the app onto your launch bar
- Click to launch the application

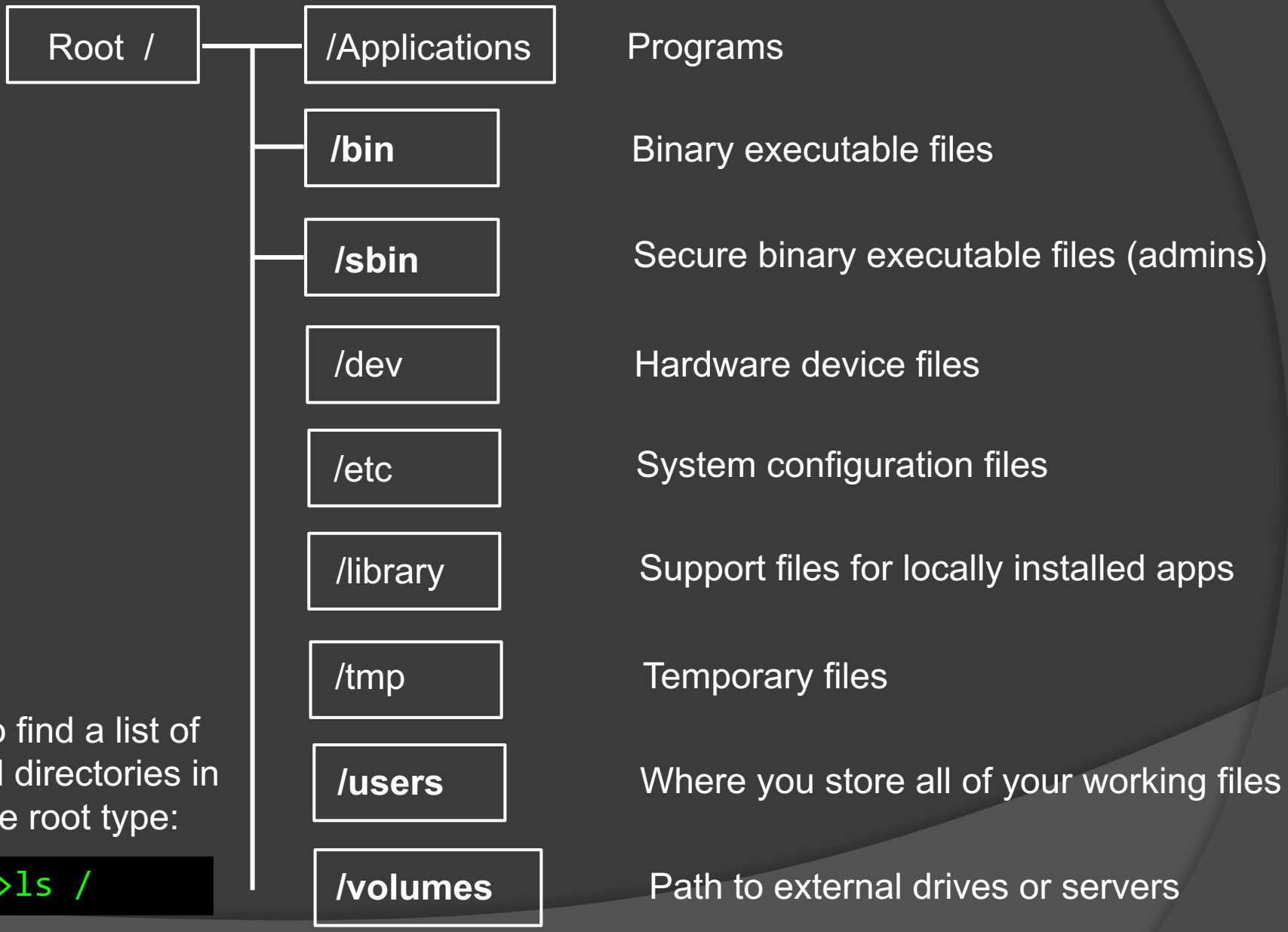




- For PC users download and install the ‘Putty’ SSH client http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

- You can use this SSH tool to connect to a UNIX server and run commands on the server (PCs do not have UNIX installed locally)

UNIX Directory Structure



To find a list of all directories in the root type:

```
>ls /
```

Navigating the UNIX File System

Navigation	
ls	List directory files
cd	Change directory
cd ~	Go to home directory in /user
cd ..	Move down one directory
cd ./	Move up one directory
pwd	What is my current directory?
df -h	Disk usage/storage
history	List my previous commands
Up arrow	Cycle through previous commands
ps	List current processes
kill	End current process
mdfind	Find a file on the computer
clear	Clear the screen

Executing Programs

To execute a program simply type the program name at the command line (if it has been loaded into the bin or sbin folder)

```
top
```

Otherwise execute the program using the full directory path

```
/bin/top
```

By default the program will run in the foreground and you will be unable to run other programs at the same time

To run a program in the background use the ‘&’ symbol after

```
/bin/top &
```

Type ‘ps’ to view all the processes running in the background

```
ps
```

To stop running a program type ‘kill’ and the process ID

```
kill 6869
```

Manipulating Files and Directories

Navigation	
rm file1	Remove a file
cp file1 file2	Copy a file to a location
mv file1 file2	Move file or rename file
mkdir directory	Make a new directory
rmdir directory	Delete a directory
touch file1	Create a new file
more file1	Begin listing the contents of file1
less file 1	Another program to list contents of file1
head file1	List first x number of lines from file1
tail file1	List last x number of lines from file 1

Advanced File Manipulation

File Manipulation	
wc -l file1	Count number of lines in file 1
wc –c file1	Count number of characters in file 1
cat file1 file2	Display or concatenate files
sort	Sort the contents of a file
uniq	Will remove duplicate lines from a column
cut	Extract columns from a file
grep ‘pattern’ file	Search for pattern inside a file
which	Shows path to an executable
whereis	Where is a file located
tr char1 char2	Translate characters in text file
sed pattern1/pattern2	Replace a pattern of characters inside a file
diff file1 file2	Difference between two files

Piping Commands and Input/Output

A powerful aspect of UNIX is the ability to redirect output and combine multiple commands together using the pipe character |

IO Commands	
Command > file	Output text to file 1
Command >> file	Append output to file 1
Command < file	Input file 1 into the command
Command 1 | command 2	Pipe output of command 1 into command 2
Cat file1 file2 file3 > bigfile	Concatenate the output of files 1-3 into bigfile

Simple Example: combine history and grep to find commands that contain 'ls'

```
history | grep ls
```

A More Sophisticated Example of Piping Commands Together:

```
cat file1 | sort | grep seq | cut –f 2 | sed s/at/cg > file0
```

Connecting to Servers

SSH

To log into a server use SSH (secure shell) and enter the login name and IP address

```
ssh navin@139.52.107.59
```

FTP

To log into an server and download/upload files use FTP (file transfer protocol)

```
ftp navin@139.52.107.59
```

SCP

To copy a file from a server to your local computer use SCP (secure copy) Enter the login name, IP address, colon, file location on server and file location for local transfer

```
scp navin@10.132.83.154:/tmp/file.txt  /users/navin
```

Server Commands

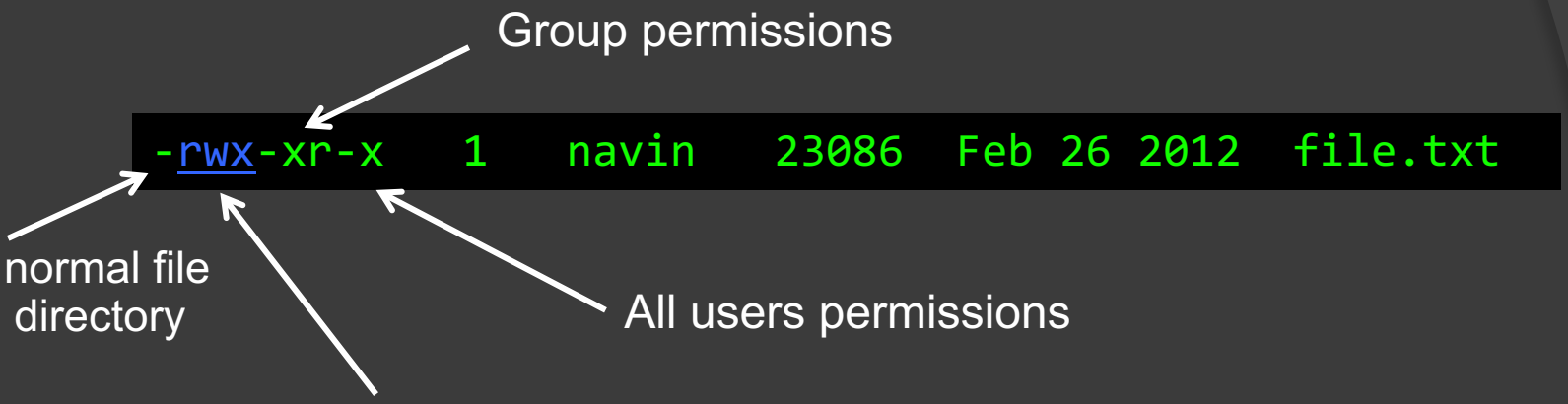
While logged into a server with SSH there are specific commands for running programs and communicating with other users

Server Commands	
who	Who else is on the server?
logout	Logout of the server
finger	Find out more about the user
ps	List processes on server
top	Real time updates of server processes
&	Always run programs in the background on servers
nice	Give program a lower priority for running
write	Write a message to another user
talk	Two-way chat program
ls -l	List files with permissions
chmod	Change file permissions
passwd	Change user password

File Permissions

Unix is a highly secure file system in which each file has specific permissions

To find out the permissions for files in a directory type : “ls – l”



The first three characters describe YOUR permission, where
- = separate permission groups
r = read permission
w = write permission
x = execute permission

The next three characters describe the GROUP permissions

And the last three characters describe the permission by ALL USERS

The tool ‘chmod’ allows admins to change the permissions of a file or folder

File Permissions

To change file permissions use ‘chmod’

To add read write and execute permissions for the current user type:

```
chmod +rwx file.txt
```

To remove read and write permissions type:

```
chmod u-rx file.txt
```

To grant all permissions to a file use ‘777’

```
chmod 777 file.txt
```

SUDO

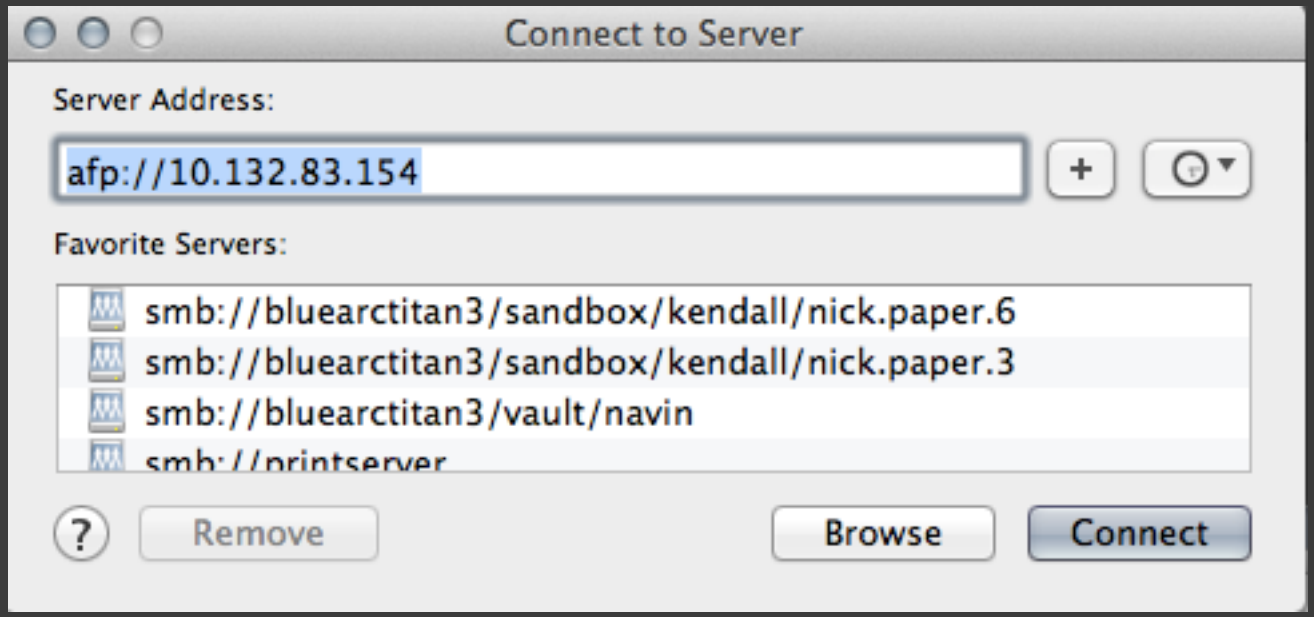
In some cases you will need administrator privileges to make changes to files or directories, in which case you can use the SUDO command to temporarily execute a command as the administator

```
sudo cp file1 /sbin
```

Using the OSX GUI to copy files from a server

- Connect to the course server with AFP on a MAC
  AFP = apple file protocol

Finder > Go > Connect to Server



Enter afp:// and the server IP address

Enter your login and password

Additional Tools and Commands

MAN

The **man** command will look up the documentation and flags for a UNIX application

```
man cp
```

Wildcards

In UNIX there are two types of wildcards that can be used in filenames:
The * wildcard will search for any number of characters before/after the star
The **?** wildcard will only search for a single character

```
>ls file*
>ls file.f?
```

File naming conventions
NEVER use spaces in file names or directories

```
>touch TRR_restriction_enzyme.txt
>touch TRR restriction enzyme.txt
```

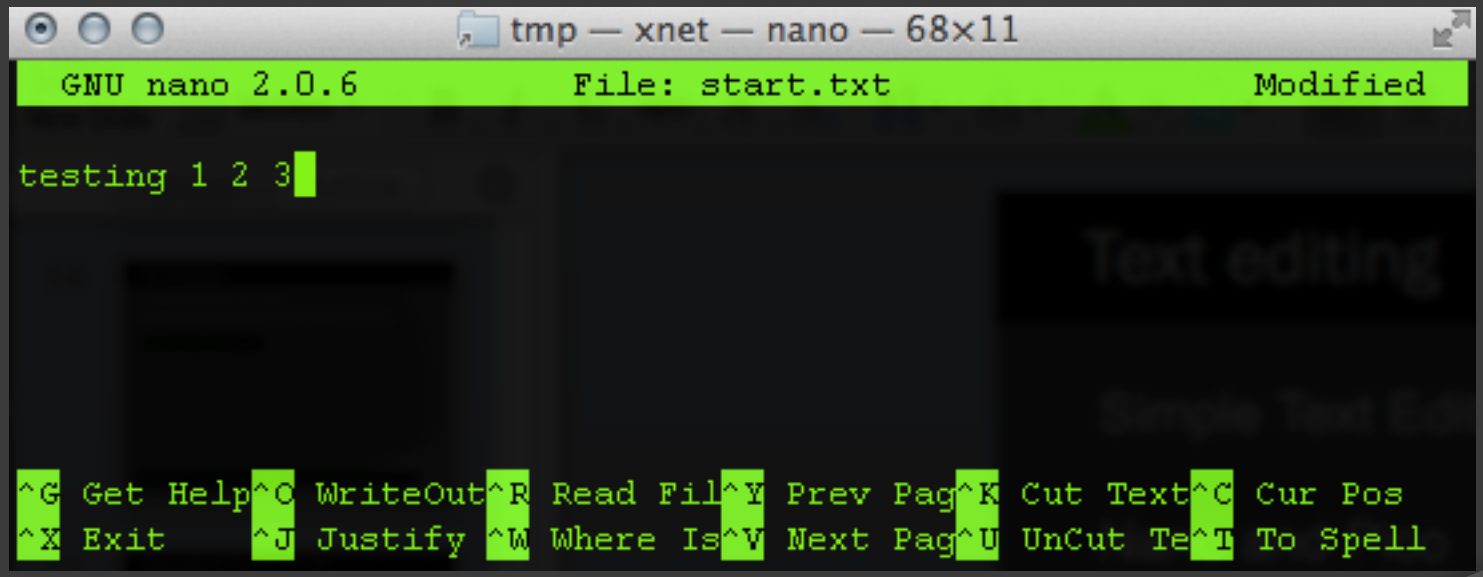
Bad!

Text editing

Simple Text Editors:

Nano

```
nano filename
```

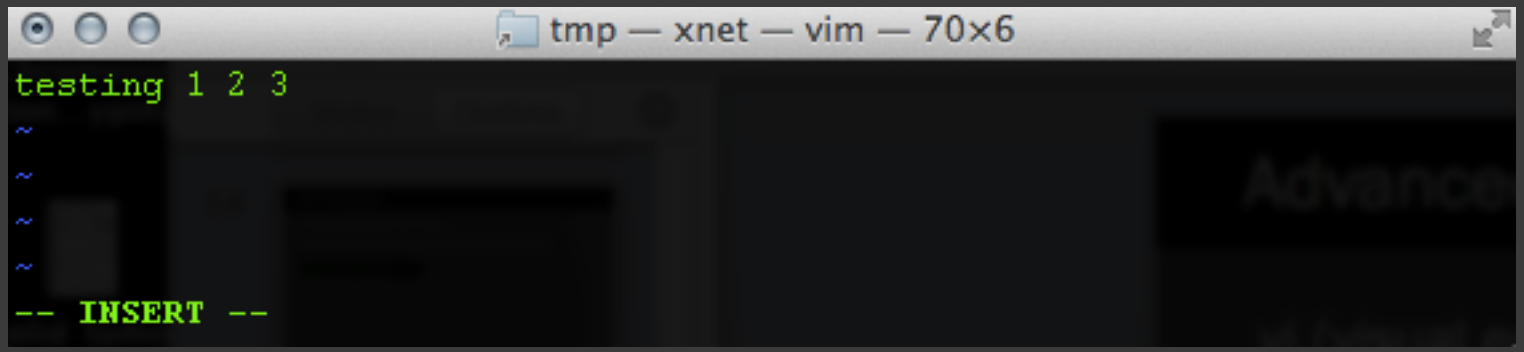


Type text and type control-O to write the file out
Then type control-X to exit the application

Advanced Text Editors

VI or VIM (visual editor)

```
>vi filename
```



VI has 2 different editing modes:

- insert mode (press i to insert text)
- command mode (press escape to return to command mode)

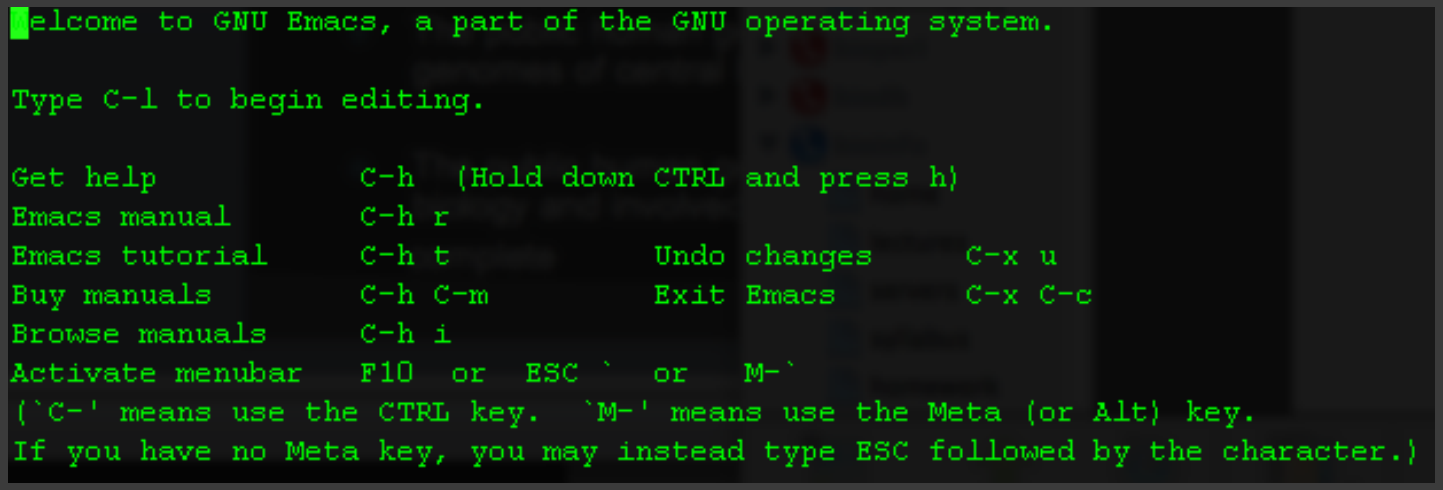
When in command mode press shift-colon
To enter a command

To exit and write out the file press shift-colon
Then enter qw and press return

Advanced Text Editors

emacs

```
>emacs filename
```



Emacs commands:

control-”letter key”

Features:

- regular expression pattern matching
- Code highlighting

Advanced Text Editors - EMACS

C (control key) M (escape key)

Commands	
save and quit	C-x, C-c
save filename	C-x, C-s
cut text	C-w
copy text	M-w
paste text	C-y
select text	C-space, C-space
switch buffers	C-b
undo	C-_ (control-underscore)
redo	C-g, C-_
search-forwards	C-s
search-backward	C-h
delete line	C-k

FASTA Files

FASTA files are used to store DNA, RNA or Protein sequences

First line starts with '>' and a header line with the gene ID or name

Subsequent lines store characters of DNA, RNA or Protein sequence

>KRAS gene |NM_004985.3| Homo sapiens v-Ki-ras2
GGCCGCGGCGGCGGAGGCAGCAGCGGCGGCGGCAGTG
GCGGCGGCGAAGGTGGCGGCGGCTCGGCCAGTACTCCC
GGCCCCCGCCATTTCGGACTGGGAGCGAGCGCGGCGCA
GGCACTGAAGGCGGCGGCGGGGCCAGAGGCTCAGCGG
CTCCCAGGTGCGGGAGAGAGGCCTGCTGAAAATGACTGA
ATATAAACTTGTGGTAGTTGGAGCTGGTGGCGTAGGCAAG
AGTGCCTTGACGATACAGCTAATTCAGAATCATTTTGTGGA
CGAATATGATCCAACAATAGAGGATTCCTACAGGAAGCAAG

BED Files

The BED file (Browser Extensible Data) is a popular format for storing genomic data

BED files have 3 required columns separated by tabs and a forth column with an identifier

```
Chr1      112233      123414        MDM4
Chr2     2398712     2509102         ERSF
Chr5      219120      301291       TRRFS3
```

Column1 is the chromosome
Column2 is the start coordinate (in basepairs)
Column3 is the end coordinate (in basepairs)
Column4 is the descriptor column (such as gene name)

Additional columns can be added with data values

BED files can be uploaded directly to the UCSC Browser as an annotation track for viewing

File Compression

UNIX files are often compressed as ‘tarballs’
Or ‘gunzip’ files to save bandwidth when
Transferring files over the network

Tarballs have the *.tar extension and can be decompressed with the ‘tar’ command

```
tar -xzf file1.tar
```

Texas tarball

Gunzip giles can be decompressed with ‘gzip’

```
gzip –d file1.gz
```

The ZIP format is also used occasionally, and can be compressed/decompressed with ‘zip’ or ‘unzip’ commands

```
zip file1
unzip file1.zip
```

Installing and Compiling Software with Make

A common task is to install a new tool in UNIX that was published in an academic paper

In OSX you need to have XCODE installed in the operating system in order to compile programs, which is included on the OSX install disc or can be downloaded for free from Apple

For example, if we wanted to install BEDtools, very useful software package for working with BED files, we would go to the URL

http://code.google.com/p/bedtools/downloads/list

And download the compressed tarball file:
BEDTools.v2.16.2.tar.gz

First we would unzip the file using GZIP

```
tar -xzf BEDTools.v2.16.2.tar.gz
```

Then go into the main directory

```
cd BEDTools.v2.16
```

Using Make to Compile Files

To build the executable files from the code you use MAKE

```
make all
```

Go into the binary directory

```
cd bin
```

Run the program

```
./intersectBED
```

To make the program run from anywhere copy it into your /bin folder

```
cp * /bin
```

UNIX Bioinformatics Workshop

The workshop for today can be found here:

http://www.navinlab.com/biounix

Follow the instructions on the website to complete the workshops and don't be afraid to ask our TA (Erin Williams) for help

TWO OPTIONS:

#1 WORK ON THE SERVER
(RECOMMENDED FOR REGISTERED STUDENT AND AUDITORS)

#2 WORK ON YOUR LOCAL COMPUTER
(RECOMMENDED FOR AUDITORS WHO ARE NOT REGISTERED)