



Linux

Introduction

Introduction: What is Unix?

- An operating system
- Developed at AT&T Bell Labs in the 1960's
- Command Line Interpreter
- GUIs (Window systems) are now available

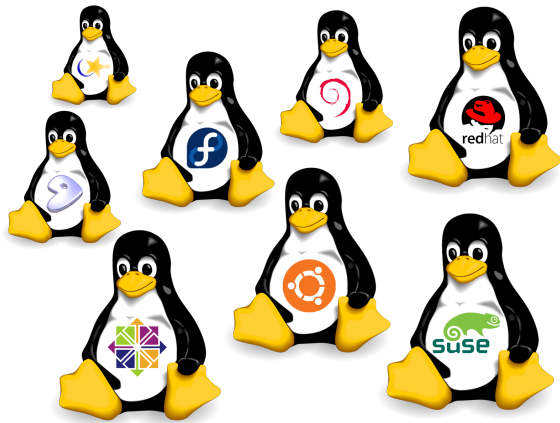
Introduction: Unix vs. Linux

- Unix was the predecessor of Linux
- Linux is a variant of Unix
 - So is Mac OS X, so much of this tutorial applies to Macs as well
- Linux is open source
- Most of the machines you'll use in the Bioinformatics program are running the Linux OS

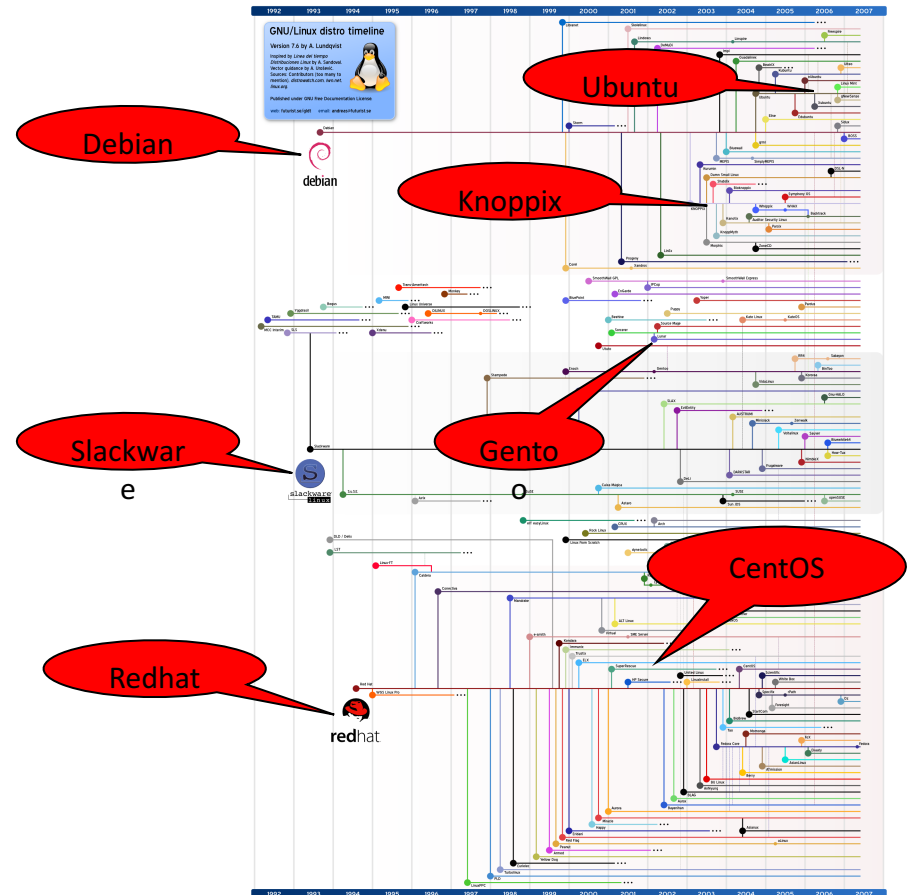
Linux Distribution



Linux



Ubuntu
Centos OS
Redhat
SUSE
Debian
....

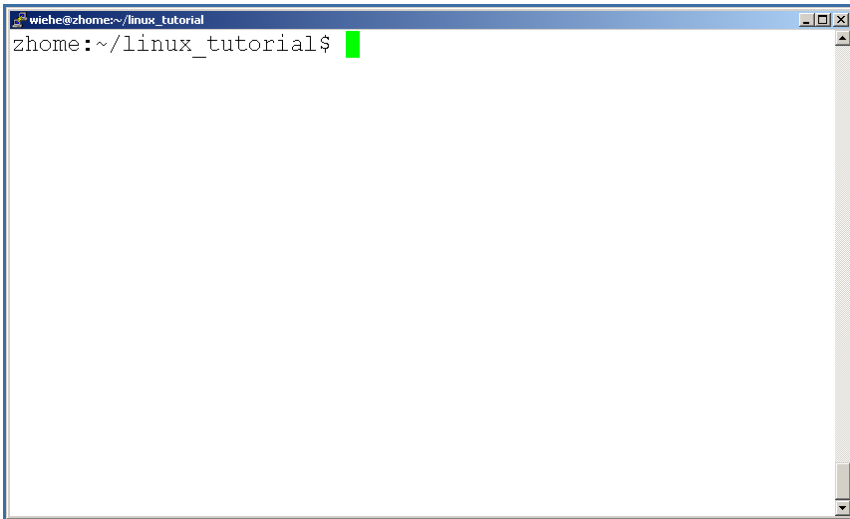


Introduction: Why Unix/Linux?

- Linux is **free**
- It's fully **customizable**
- It's **stable** (i.e. it almost never crashes)
- These characteristics make it an ideal OS for programmers and scientists

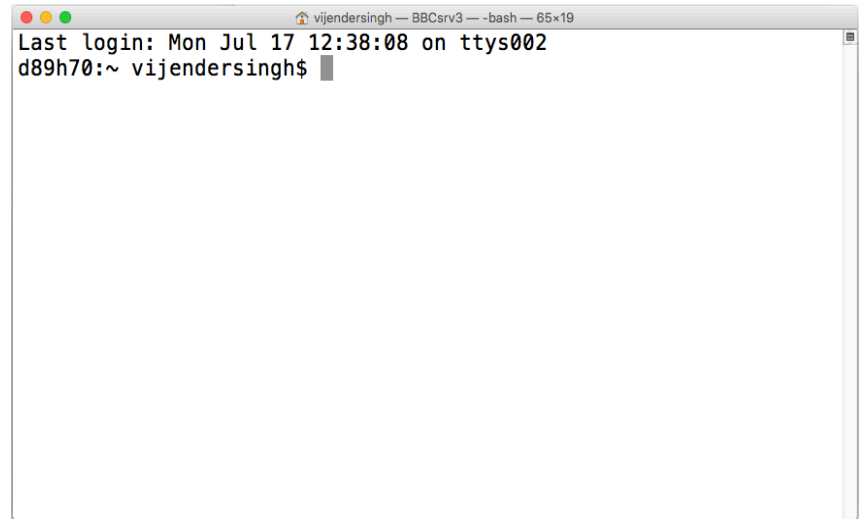
Connecting to a Unix/Linux system

- Open up a terminal:

A terminal window from the Putty application. The title bar shows 'wiehe@zhome:~/linux_tutorial'. The terminal content shows the prompt 'zhome:~/linux_tutorial\$' followed by a green cursor.

```
wiehe@zhome:~/linux_tutorial
zhome:~/linux_tutorial$
```

Putty

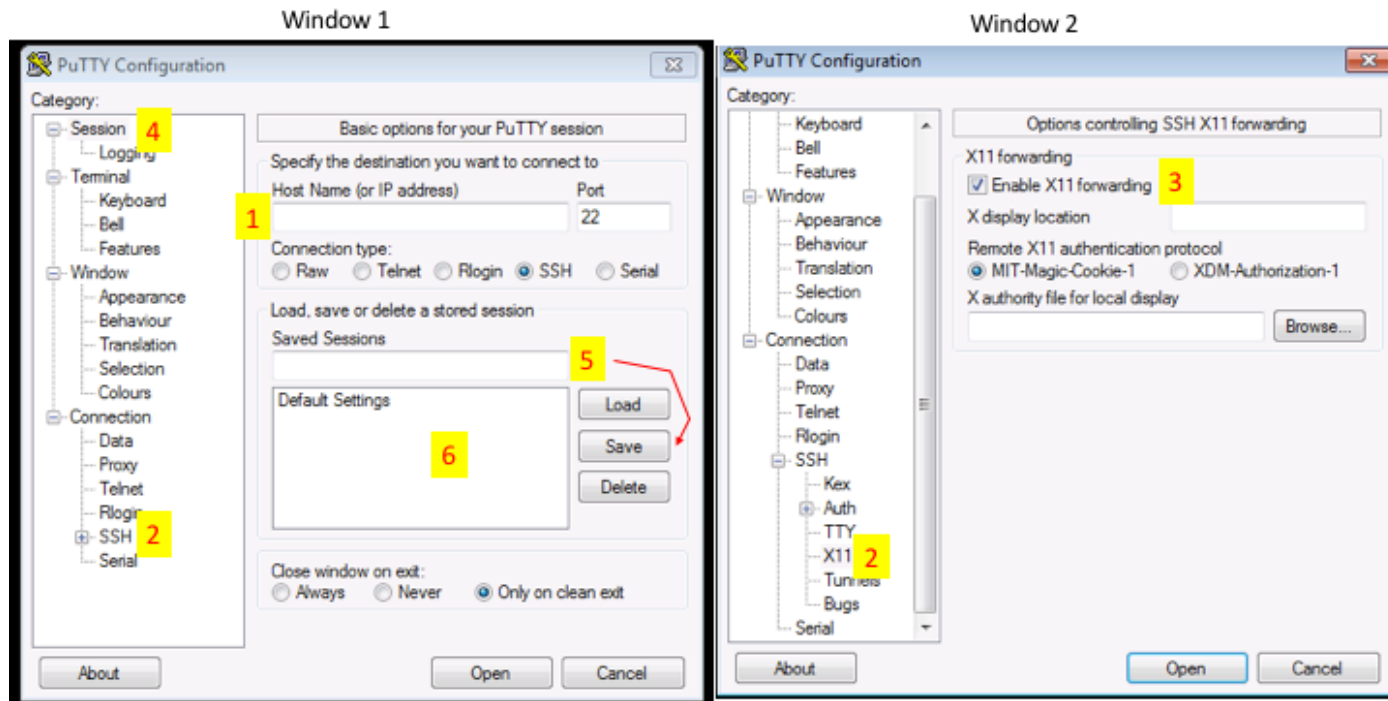
A terminal window from a Mac. The title bar shows 'vijendersingh — BBCsrv3 — -bash — 65x19'. The terminal content shows the last login message: 'Last login: Mon Jul 17 12:38:08 on ttys002' followed by the prompt 'd89h70:~ vijendersingh\$' and a grey cursor.

```
vijendersingh — BBCsrv3 — -bash — 65x19
Last login: Mon Jul 17 12:38:08 on ttys002
d89h70:~ vijendersingh$
```

Mac

Mac : Terminal : ssh [your_username@bbcsrv3.biotech.uconn.edu](ssh://your_username@bbcsrv3.biotech.uconn.edu)

Windows : Putty

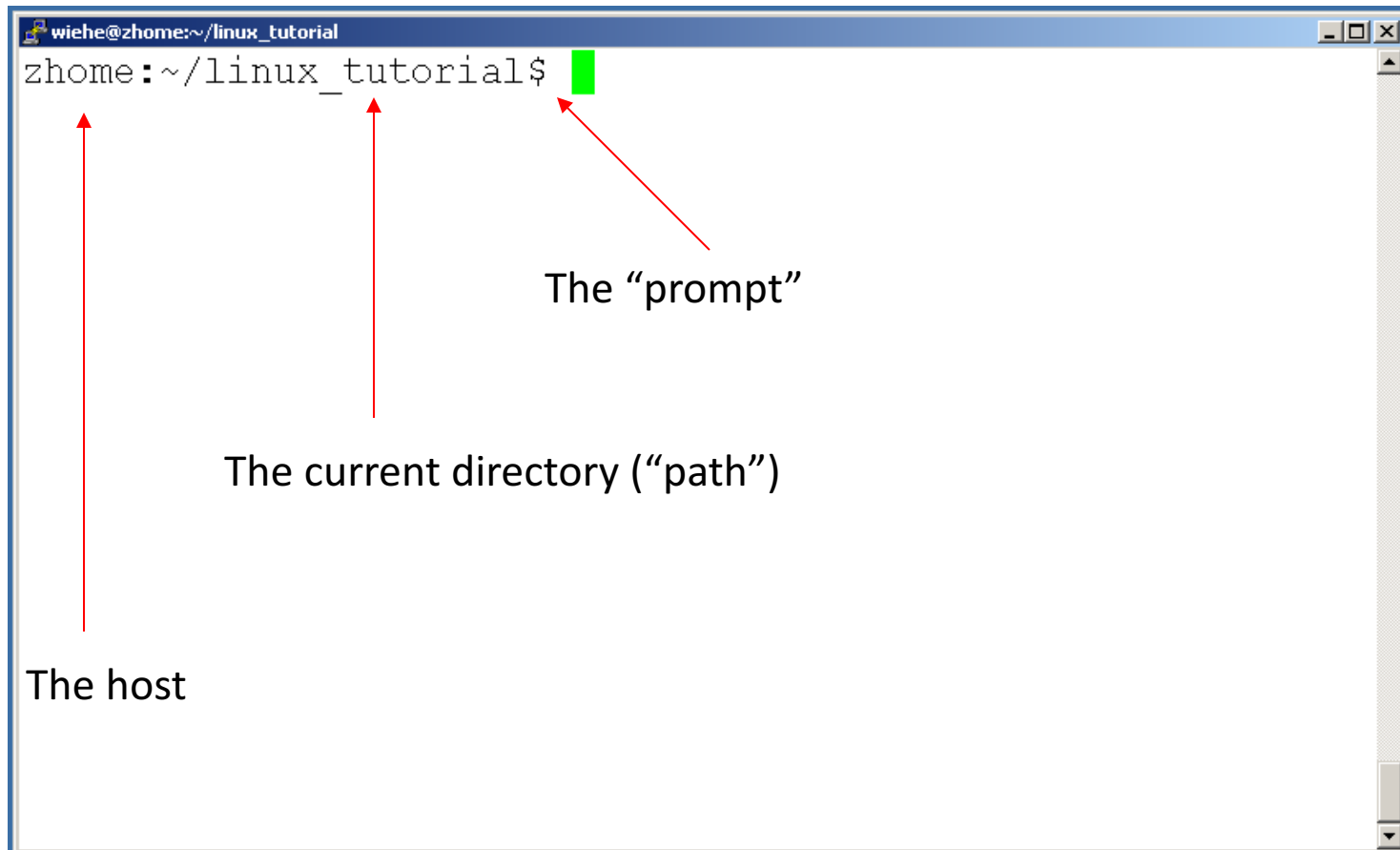


Open Putty it will open window1.

1. Provide host name e.g. ssh [your_username@bbcsrv3.biotech.uconn.edu](ssh://your_username@bbcsrv3.biotech.uconn.edu) Expand SSH tab and select X11 (shown in window2)
2. Enable X11 forwarding by selecting it. (window2)
3. Scroll up the left panel and select Session.(window1)
4. Name your session e.g. BBC_cluster and click save tab to save.
5. Your session name should appear in saved sessions.
Double click on your session name to connect to server with SSH session.

Connecting to a Unix/Linux system

- Open up a terminal:

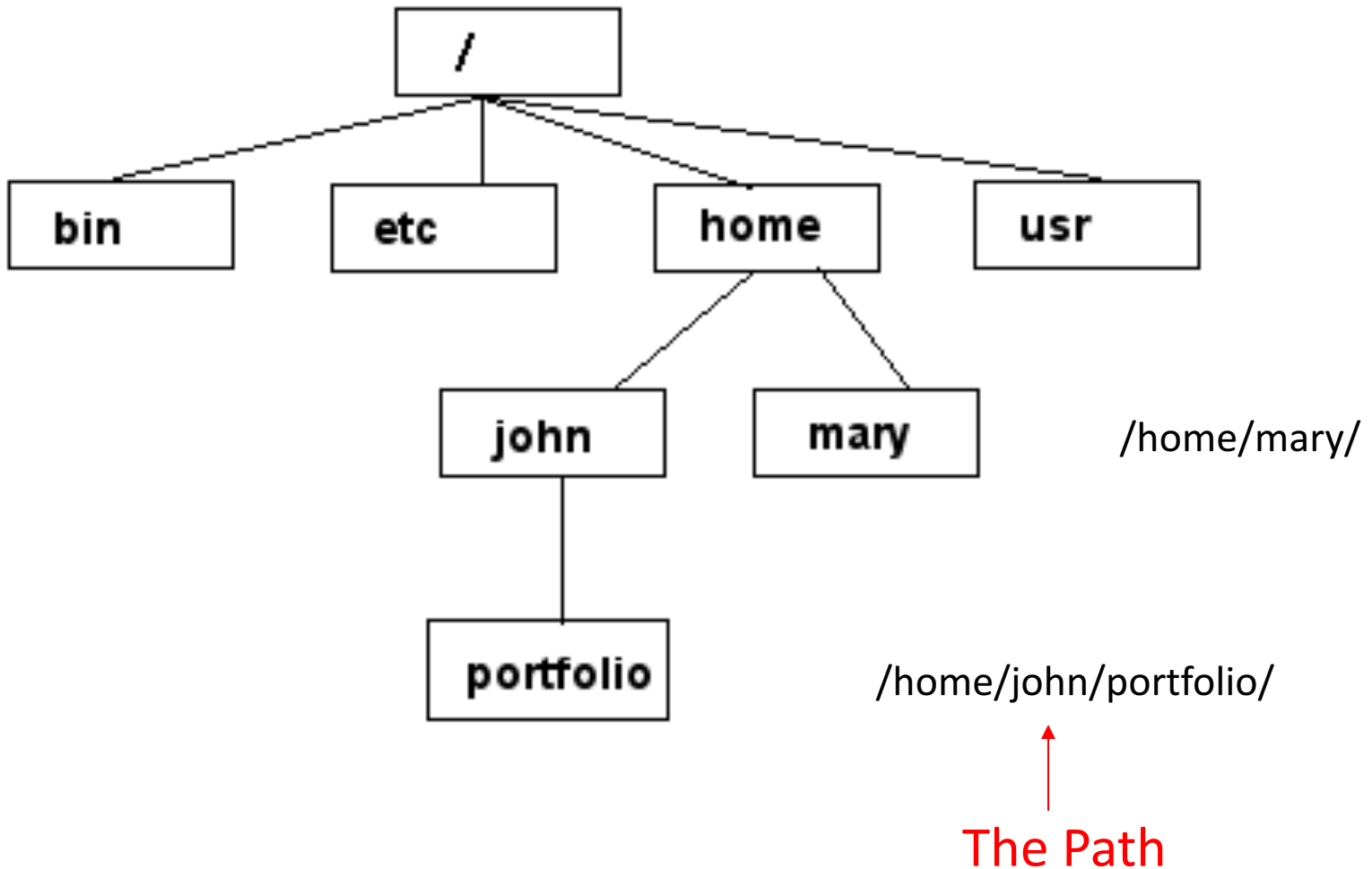


What exactly is a “shell”?

- After logging in, Linux/Unix starts another program called the **shell**
- The shell interprets commands the user types and manages their execution
 - The shell communicates with the internal part of the operating system called the **kernel**
 - The most popular shells are: tcsh, csh, korn, and bash
 - The differences are most times subtle
 - For this tutorial, we are using bash
- Shell commands are **CASE SENSITIVE!**

Unix/Linux File System

NOTE: Unix file names are **CASE SENSITIVE!**



Creating files in Unix/Linux

- Requires the use of an Editor
- Various Editors:
 - 1) nano
 - 2) vim
 - 3) emacs



Login into remote server

```
ssh -X <your_username>@<host_name>
```

```
ssh vsingh@bbcsrv3.biotech.uconn.edu
```

Basic Linux commands

```
pwd          # Get full path of the present working directory  
             (same as "echo $HOME")  
ls           # Content of pwd  
ls -l       # Similar as ls, but provides additional info on  
             files and directories  
ls -a       # Includes all files including hidden files  
             (.name) as well  
ls -R       # Lists subdirectories recursively  
ls -t       # Lists files in chronological order
```

Basic Linux commands cont.....

`cd <dir_name>`

`cd` # Brings you to the highest level of your home directory.

`cd ..` # Moves one directory up

`cd ../../` # Moves two directories up (and so on)

`cd -` # Go back to you were previously (before the last directory change)

`echo ~` # View the full (complete) path of your home

`find ~` # List all your files (including everything in sub-directories)

`ls ~` # List the top level files of your home directory

Basic Linux commands cont.....

```
mkdir <dir_name> # Creates specified directory
rmdir <dir_name> # Removes empty directory
rm <file_name> # Removes file name
rm -r <dir_name> # Removes directory including its
                  content, but asks for confirmation, 'f'
                  argument turns confirmation off
cp <name> <path> # Copy file/directory as specified in
                  path (-r or -a to include content in
                  directories)
mv <name1> <name2> # Renames directories or files
mv <name> <path> # Moves file/directory as specified in path
```

Basic Linux commands cont.....

`man <something>` # general help (press the 'q' key to exit)

`man wc` # manual on program 'word count' wc

`wc --help` # short help on wc

`cat <file1>` # Print contents of file1

`cat <file1> <file2> > <cat.out>` # concatenate files in output file
'cat.out'

`paste <file1> <file2> > <paste.out>` # merges lines of files and
separates them by tabs
(useful for tables)

`cmp <file1> <file2>` # tells you whether two files are identical

`diff <fileA> <fileB>` # finds differences between two files

`head -<number> <file>` # prints first lines of a file

`tail -<number> <file>` # prints last lines of a file

Basic Linux commands cont.....

Files and permission

"r" means "read only" permission

"w" means "write" permission

"x" means "execute" permission

In case of directory, "x" grants permission to list directory contents

```
xanadu-submit-ext ~ $ ls -l
```

```
total 892
```

```
-rw-r--r-- 1 vsingh domain users 257 Jun 2 09:18 .  
drwxr-xr-x 7 vsingh domain users 1601 May 2 21:26 aening_masurca  
-rw-r--r-- 1 vsingh domain users 499 Jun 2 00:39 arrayjob.sh  
drwxr-xr-x 2 vsingh domain users 1431 Jun 2 00:39 arrayout  
-rwxr-xr-x 1 vsingh domain users 19384 Jun 16 12:07 assemble_xandu2.sh  
-rw-r--r-- 1 vsingh domain users 649 Jun 14 13:30 basemount_cp.sh
```

```
Other/world (safe: r-x) : o
```

```
Group (safe: r-x) : g
```

```
User : u
```

```
File(-) or directory (d)
```


Basic Linux commands cont.....

Change permission

If you own the file, you can change it's permissions with "chmod"

Syntax: `chmod [user/group/others/all]+[permission] [file(s)]`

Below we grant execute permission to all:

```
xanadu-submit-ext ~ $ clear
xanadu-submit-ext ~ $ ls -l test.sh
-r-x----- 1 vsingh domain users 484 Jun 16 12:06 test.sh
xanadu-submit-ext ~ $ chmod u+w test.sh
xanadu-submit-ext ~ $ ls -l test.sh
-rwx----- 1 vsingh domain users 484 Jun 16 12:06 test.sh
xanadu-submit-ext ~ $ chmod g+rwx test.sh
xanadu-submit-ext ~ $ ls -l test.sh
-rwxrwx--- 1 vsingh domain users 484 Jun 16 12:06 test.sh
xanadu-submit-ext ~ $ chmod a+rwx test.sh
xanadu-submit-ext ~ $ ls -l test.sh
-rwxrwxrwx 1 vsingh domain users 484 Jun 16 12:06 test.sh
xanadu-submit-ext ~ $ chmod go+rx test.sh
xanadu-submit-ext ~ $ ls -l test.sh
-rwxrwxrwx 1 vsingh domain users 484 Jun 16 12:06 test.sh
xanadu-submit-ext ~ $ chmod go-w test.sh
xanadu-submit-ext ~ $ ls -l test.sh
-rwxr-xr-x 1 vsingh domain users 484 Jun 16 12:06 test.sh
xanadu-submit-ext ~ $
```

Basic Linux commands cont.....

Running a program (a.k.a a job)

Make sure the program has executable permissions

```
chmod u+x program
```

Use "./" to run the program

```
./ program
```

INPUT/OUTPUT: STDIN STDOUT and redirecting (Pipes)

```
program_a > file.txt
```

program_a's output is written to a file called "file.txt"

```
program_a < input.txt
```

program_a gets its input from a file called "input.txt"

Programs can output to other programs

Called "piping"

```
program_a | program_b
```

program_a's output becomes program_b's input

Exercise 1:

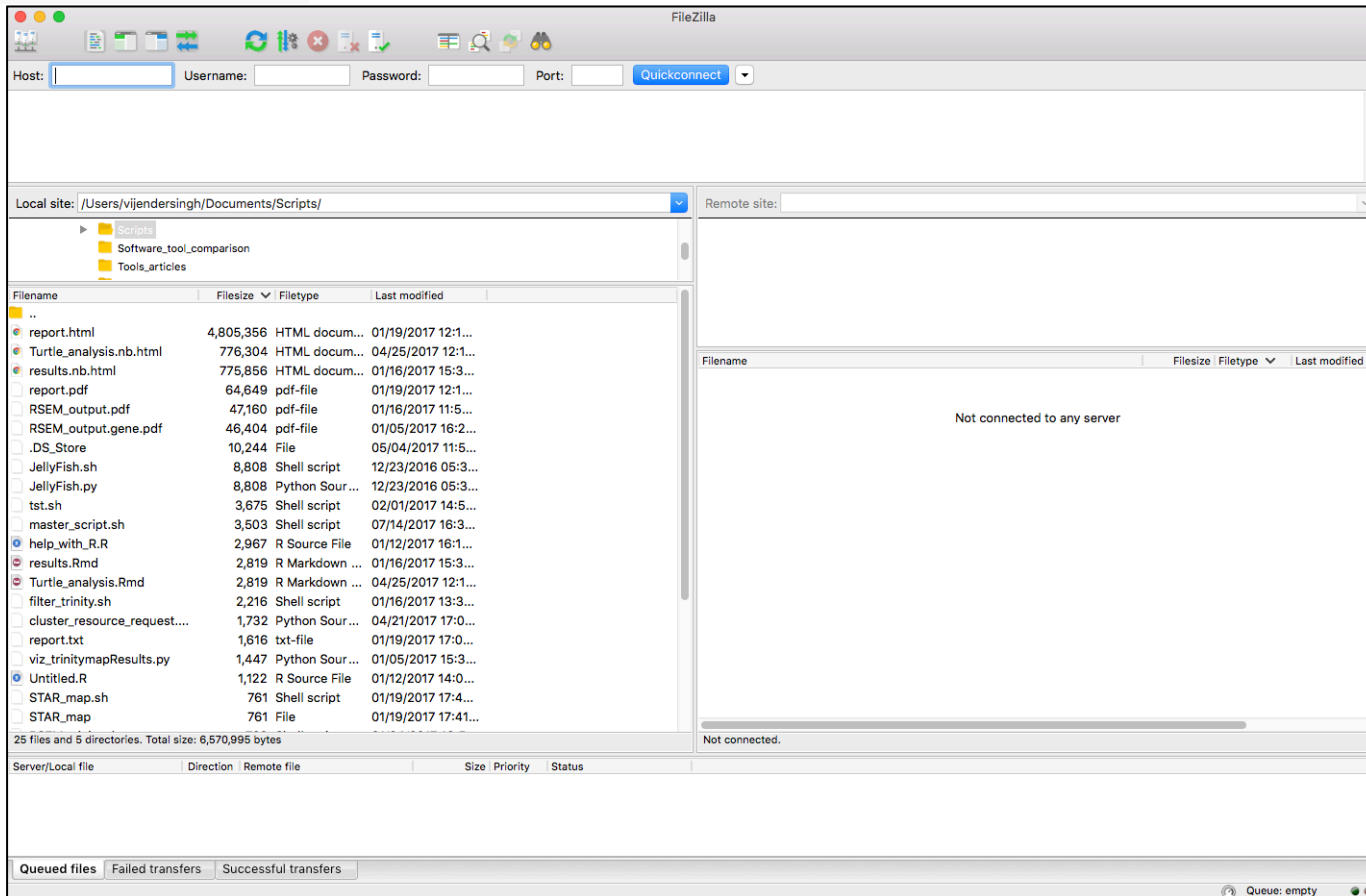
Create a file myscript.sh	: touch myscript.sh
Check for file	: ls
Open file to edit	: vi myscript.sh : i
Write commands: interpreter	: #!/bin/bash : echo "Hello world" : echo "This is my first script in Linux."
save and exit	: esc-:-w-q-!
Check permissions of file	: ls -l myscript.sh
Change permission if required	: chmod u+x myscript.sh
Run script/Program	: ./myscript.sh : sh myscript.sh
Store output to a file	: sh myscript.sh > myscript_output.txt
Check the output (limited)	: less myscript_output.txt
print the contents	: cat myscript_output.txt
Count the number of lines	: cat myscript_output.txt wc -l

Transferring files between systems



Filezilla: <https://filezilla-project.org/>

**Download
FileZilla Client**
All platforms



1: Host: sftp://bbcsrv3.biotech.uconn.edu

2: Username:

3. Password

4.Quickconnect

The screenshot shows the FileZilla SFTP client interface. At the top, the connection details are: Host: sftp://bbcsrv3.biotech.uconn.edu, Username: vsingh, Password: [masked], and Port: [blank]. The status bar indicates the connection is successful. The local site is /Users/vijendersingh/Documents/Scripts/ and the remote site is /home/vsingh. The local site contains 25 files and 5 directories, while the remote site contains 38 files and 41 directories. The interface is divided into two main panes: Local site and Remote site, each with a file list table.

Local site: /Users/vijendersingh/Documents/Scripts/

Filename	Filesize	Filetype	Last modified
..			
report.html	4,805,356	HTML docum...	01/19/2017 12:1...
Turtle_analysis.nb.html	776,304	HTML docum...	04/25/2017 12:1...
results.nb.html	775,856	HTML docum...	01/16/2017 15:3...
report.pdf	64,649	pdf-file	01/19/2017 12:1...
RSEM_output.pdf	47,160	pdf-file	01/16/2017 11:5...
RSEM_output.gene.pdf	46,404	pdf-file	01/05/2017 16:2...
.DS_Store	10,244	File	05/04/2017 11:5...
JellyFish.sh	8,808	Shell script	12/23/2016 05:3...
JellyFish.py	8,808	Python Sour...	12/23/2016 05:3...
tst.sh	3,675	Shell script	02/01/2017 14:5...
master_script.sh	3,503	Shell script	07/14/2017 16:3...
help_with_R.R	2,967	R Source File	01/12/2017 16:1...
results.Rmd	2,819	R Markdown ...	01/16/2017 15:3...
Turtle_analysis.Rmd	2,819	R Markdown ...	04/25/2017 12:1...
filter_trinity.sh	2,216	Shell script	01/16/2017 13:3...
cluster_resource_request....	1,732	Python Sour...	04/21/2017 17:0...
report.txt	1,616	txt-file	01/19/2017 17:0...
viz_trinitymapResults.py	1,447	Python Sour...	01/05/2017 15:3...
Untitled.R	1,122	R Source File	01/12/2017 14:0...
STAR_map.sh	761	Shell script	01/19/2017 17:4...
STAR_map	761	File	01/19/2017 17:41...

Remote site: /home/vsingh

Filename	Filesize	Filetype	Last modified
..			
.recently-used.xbel	774	xbel-file	04/25/2017
sickle_latmier_report.txt	4,498	txt-file	12/08/2016
prot.txt	3,323	txt-file	06/12/2017
merged_fasta_file_stats.txt	825	txt-file	01/17/2017
all.txt	2,316	txt-file	04/18/2017
Trinity_filtered_fasta.txt	821	txt-file	01/16/2017
Trinity_filteded_fastpwa.txt	0	txt-file	01/16/2017
README_test_239741.txt	225	txt-file	02/22/2017
Alignment_to_Green_sea_Turtle_result.txt	7,466	txt-file	02/16/2017
array_job_script.sh	1,078	Shell script	04/12/2017
test.r.Rout	865	Rout-file	07/13/2017
test.r	80	R Source ...	07/13/2017
graph.py	93	Python S...	03/10/2017
RSEM_output.pdf	46,404	pdf-file	01/05/2017
BBC_test_272507.out	12	out-file	07/10/2017
BBC_test_272506.out	12	out-file	07/10/2017
BBC_test_272505.out	12	out-file	07/10/2017

STDIN, STDOUT, STDERR, Redirections

```
<beginning-of-filename>*      # * is wildcard to specify many files
ls > file                      # prints ls output into specified file
command < my_file              # uses file after '<' as STDIN
command >> my_file             # appends output of one command to file
command | tee my_file         # writes STDOUT to file and prints it to
                              # screen
command > my_file; cat my_file # writes STDOUT to file and prints it to
                              # screen
command > /dev/null           # turns off progress info of applications by
                              # redirecting their output to /dev/null
grep my_pattern my_file | wc   # Pipes (|) output of 'grep' into 'wc'
grep my_pattern my_non_existing_file 2 > my_stderr # prints STDERR to file
```

```
> file redirects stdout to file
1> file redirects stdout to file
2> file redirects stderr to file
&> file redirects stdout and stderr to file
```

`/dev/null` is the null device it takes any input you want and throws it away. It can be used to suppress any output.

