Linux

Introduction

Introduction: What is Unix?

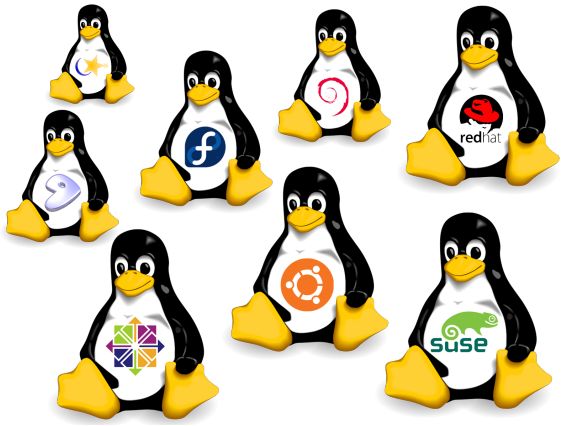
- An operating system
- Developed at AT&T Bell Labs in the 1960's
- Command Line Interpreter
- GUIs (Window systems) are now available

Introduction: Unix vs. Linux

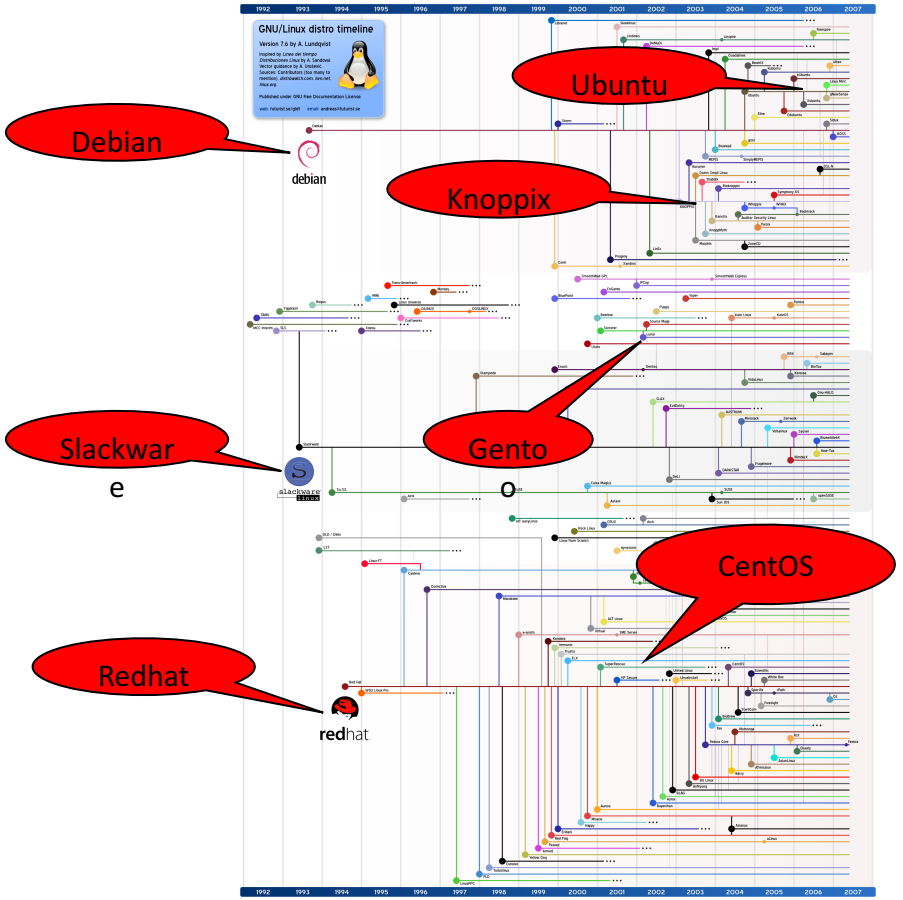
- Unix was the predecessor of Linux
- Linux is a variant of Unix
  - So is Mac OS X, so much of this tutorial applies to Macs as well
- Linux is open source
- Most of the machines you'll use in the Bioinformatics program are running the Linux OS

Linux Distribution

Linux



Ubuntu
Centos OS
Redhat
SUSE
Debian
....



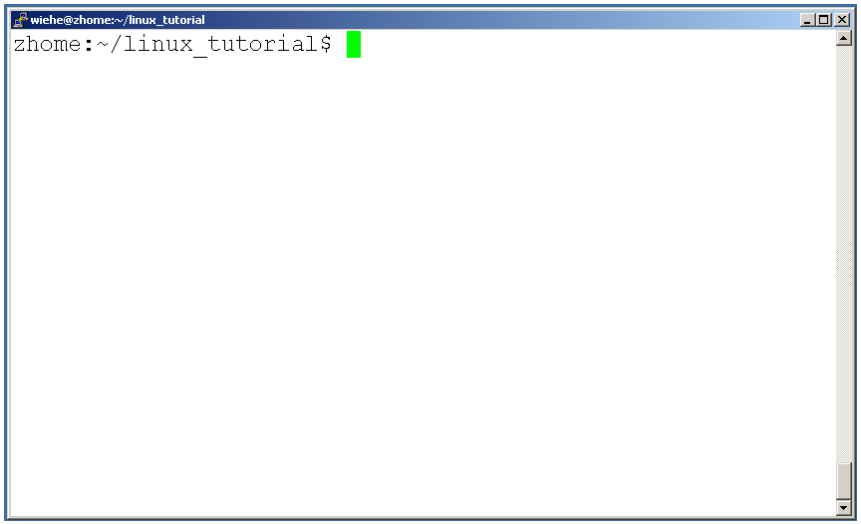
Introduction: Why Unix/Linux?

- Linux is free
- It's fully customizable
- It's stable (i.e. it almost never crashes)

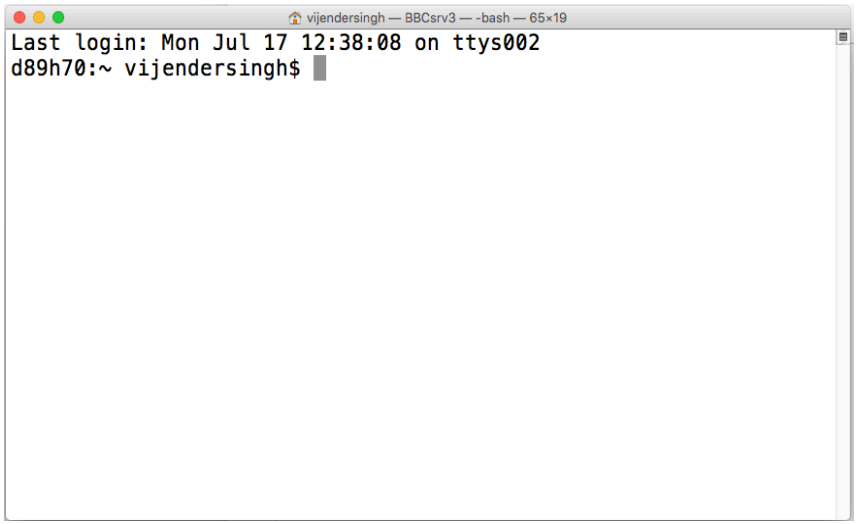
- These characteristics make it an ideal OS for programmers and scientists

Connecting to a Unix/Linux system

- Open up a terminal:



Putty

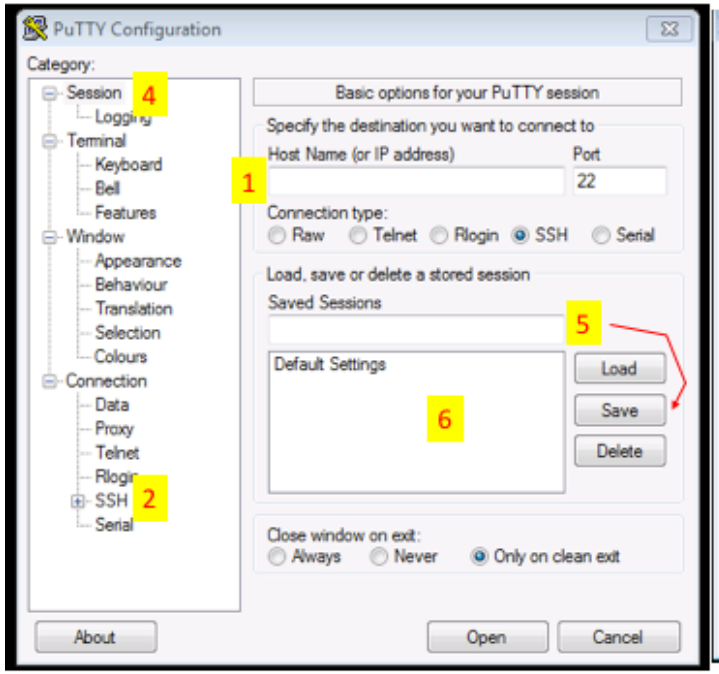


Mac

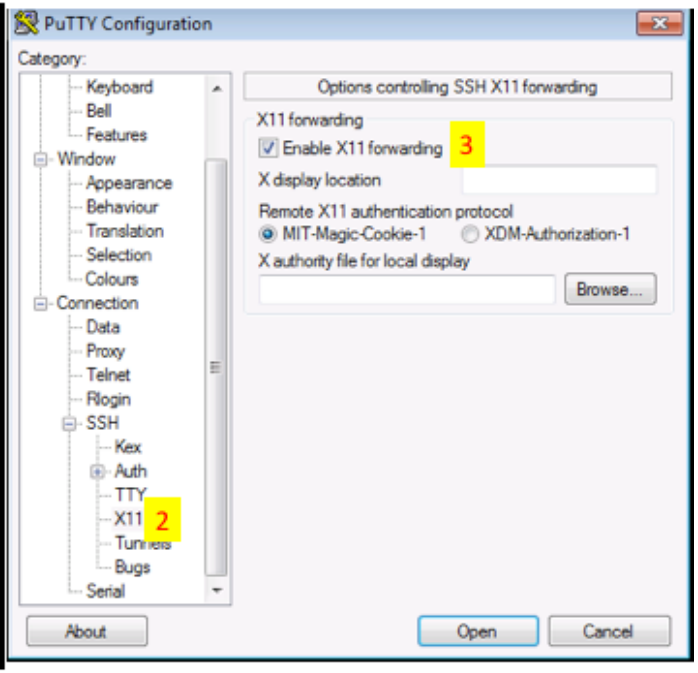
Mac : Terminal : ssh your_username@bbcsrv3.biotech.uconn.edu

Windows : Putty

Window 1



Window 2



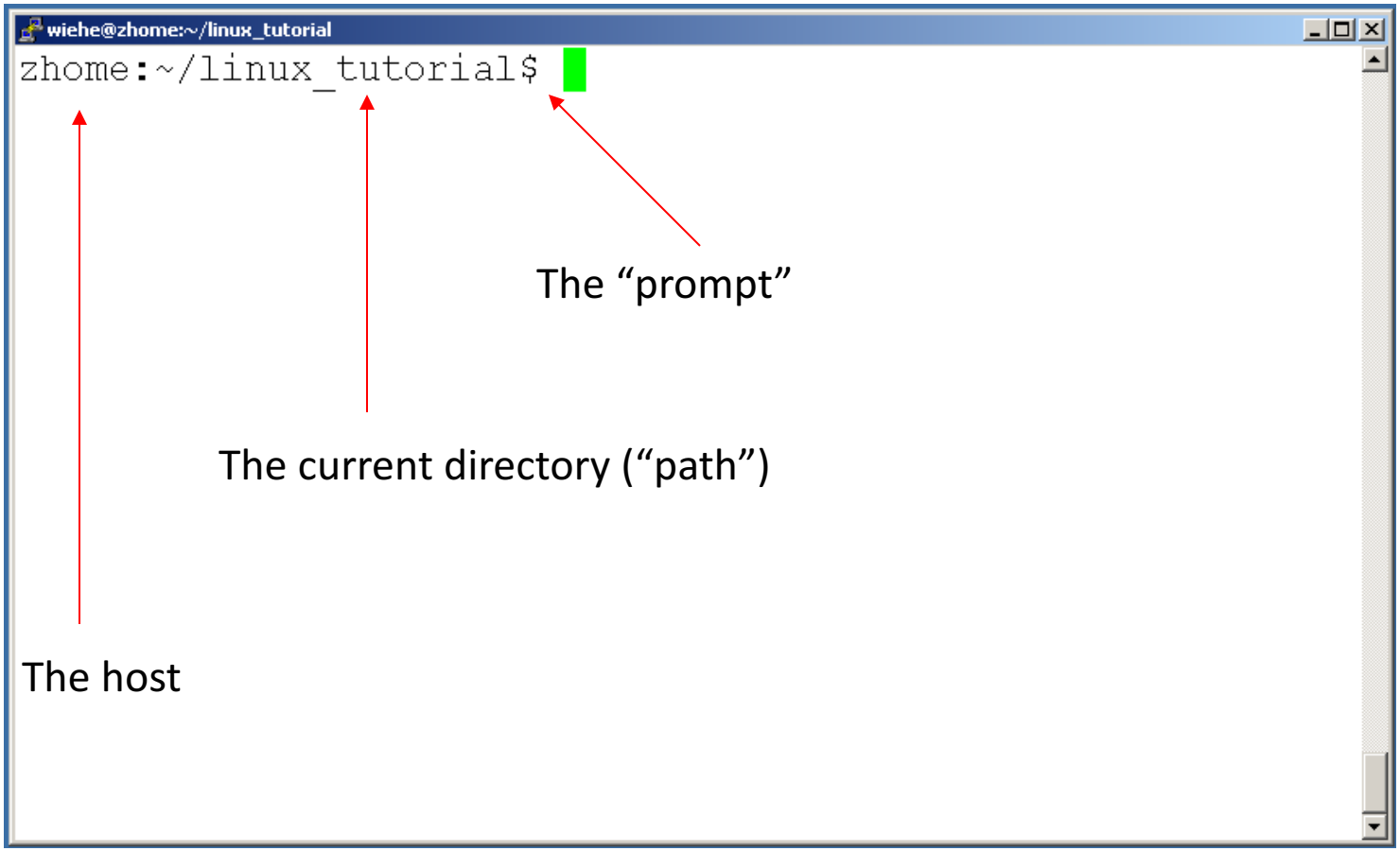
Open Putty it will open window1.

1. Provide host name e.g. ssh your_username@bbcsrv3.biotech.uconn.eduExpand 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:

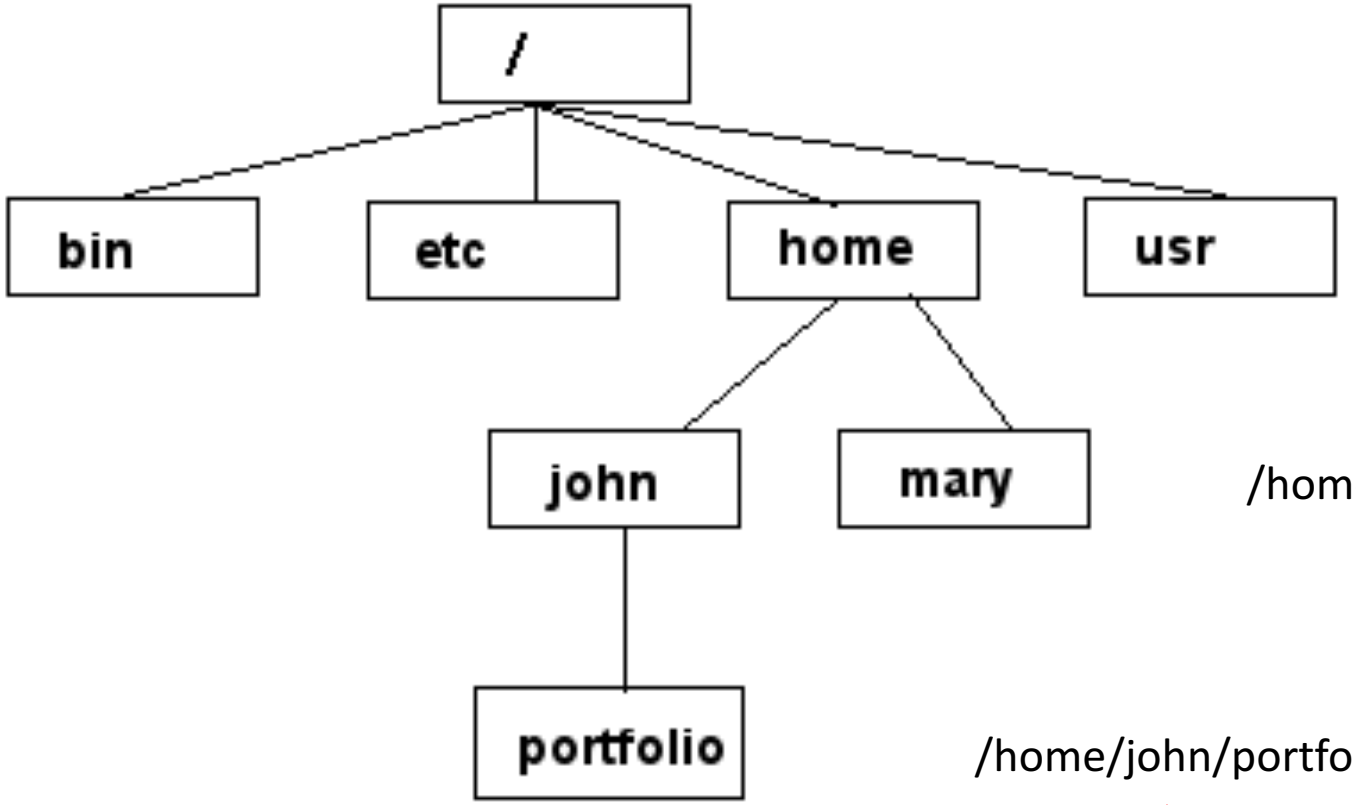


What exactly is a “shell”?

- 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!**



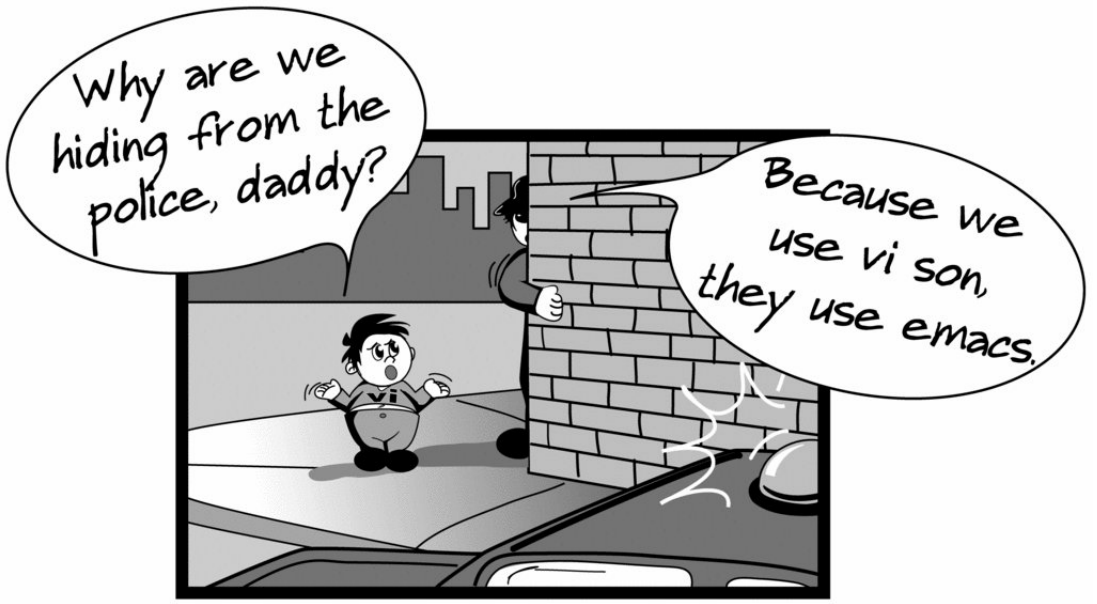
/home/mary/

/home/john/portfolio/

The Path

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.....

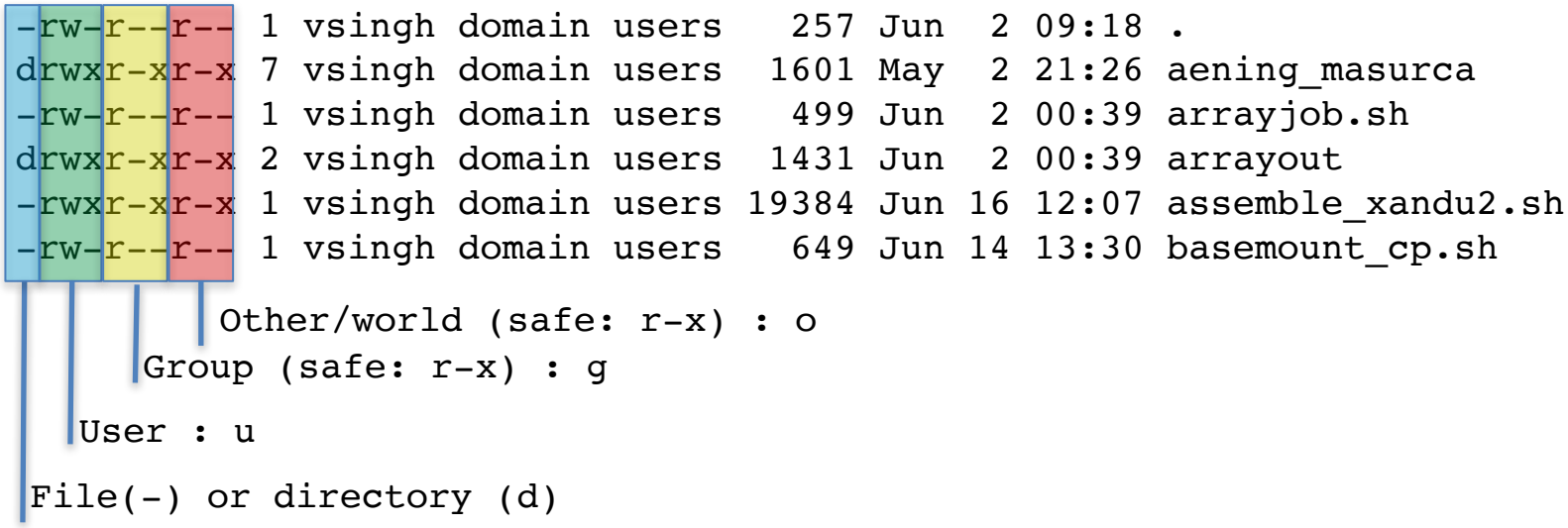
```
man <something> # general help (press the 'q' key to exit)
man wc          # manual on program 'word count' wc
wc --help       # short help on wc

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
```



Basic Linux commands cont…..

Change permission

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

Syntax: chmod [**u**ser/**g**roup/**o**thers/**a**ll]+[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
```

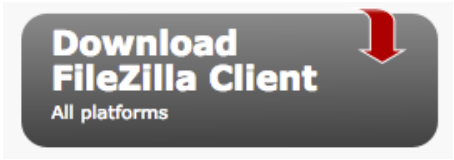
program_a's output becomes program_b's input

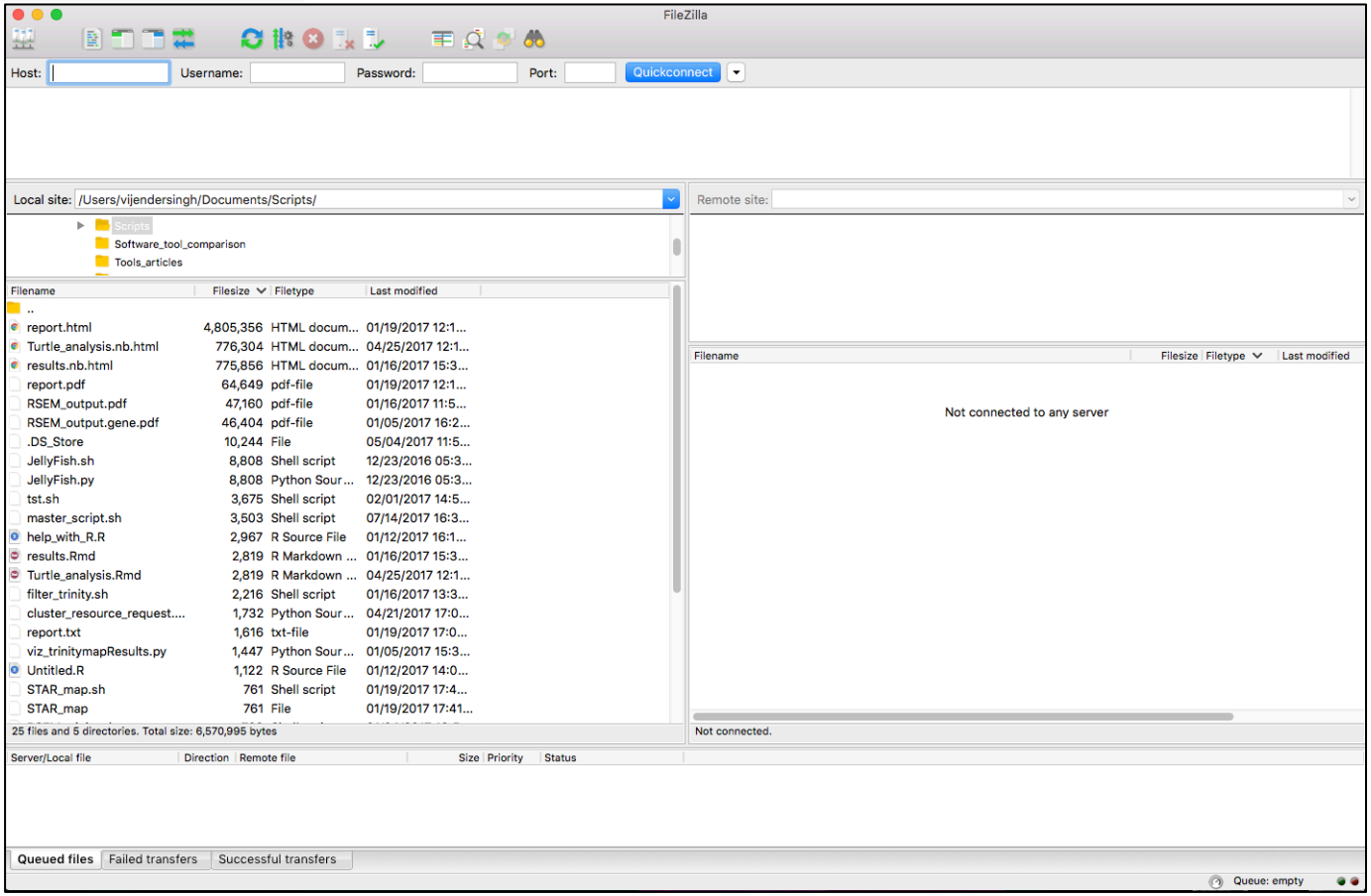
Exercise 1:

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

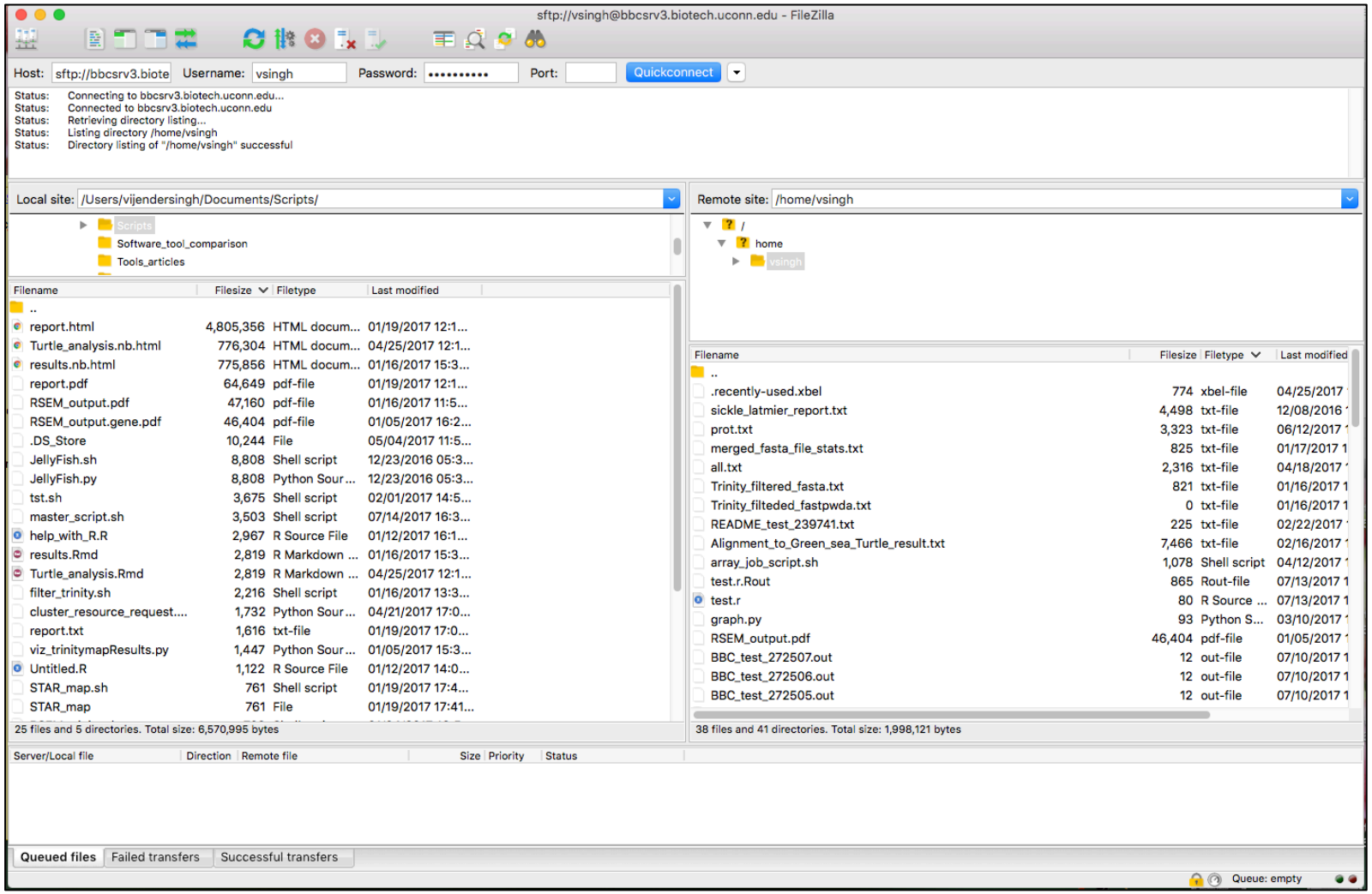
Transferring files between systems

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





1: Host: sftp://bbcsrv3.biotech.uconn.edu
2: Username:
3. Password
4.Quickconnect



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.