“Prof” Joe's Tutorial on

Linux (well Unix) basics

Joseph Phillips
2016 September 6

What is Unix?

- Invented by ATT, has a few common flavors
  - System V (ATT)
  - BSD (University of California Berkeley)
  - Linux (Linus Torvalds/GNU Project)
  - Solaris (by Sun, now Oracle)
- Popularized at Universities in 1970s and 1980s
- Learned from OS's that came before it
  - Purposefully “stripped down” from complicated Multics
  - “*Each command should do one thing, and well*”
- Influential
  - Running Apple's OS X? *It's Unix!*

Why use Unix?

- Relatively robust
- Relatively flexible
- Relatively general
- Has open source implementations:
  - Linux (of course)
  - OpenBSD
- Lets you see what is going on “under the hood”

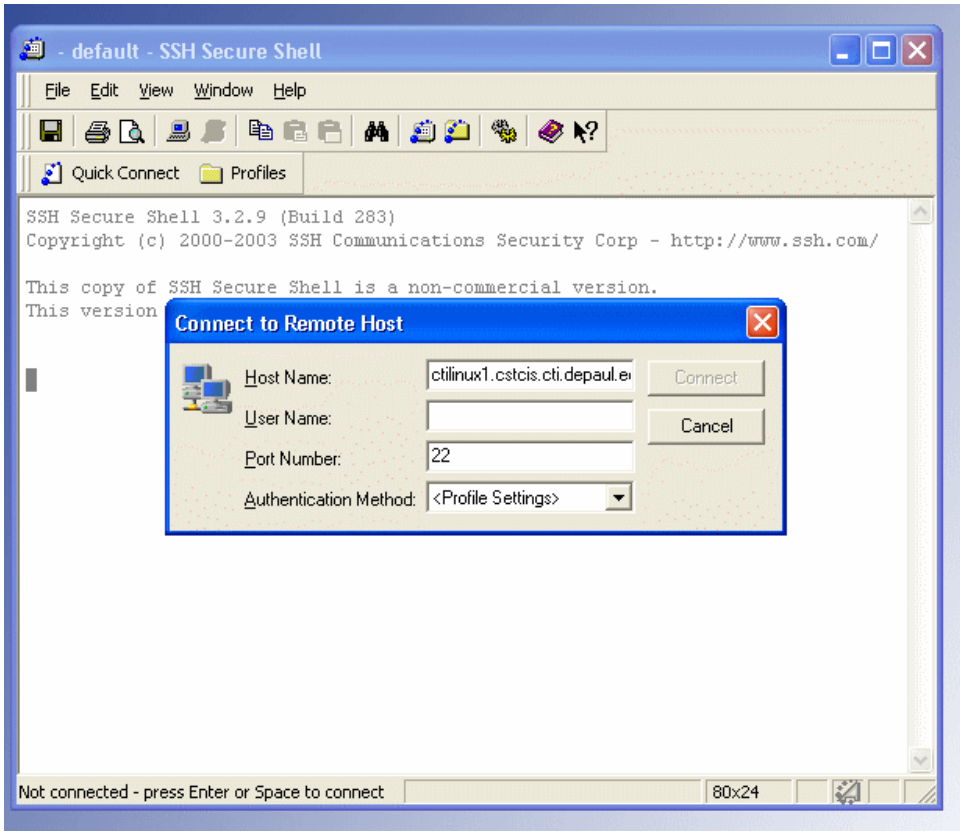
Logging in and exiting

- To start: use an ssh (*S*ecure *SH*ell) program like **putty**
  - free at http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
- Use **CampusConnect** user-name/passwd to login to one of:

140.192.36.184
140.192.36.185
140.192.36.186
140.192.36.187

- To stop: type ***exit***

```
$ exit
```

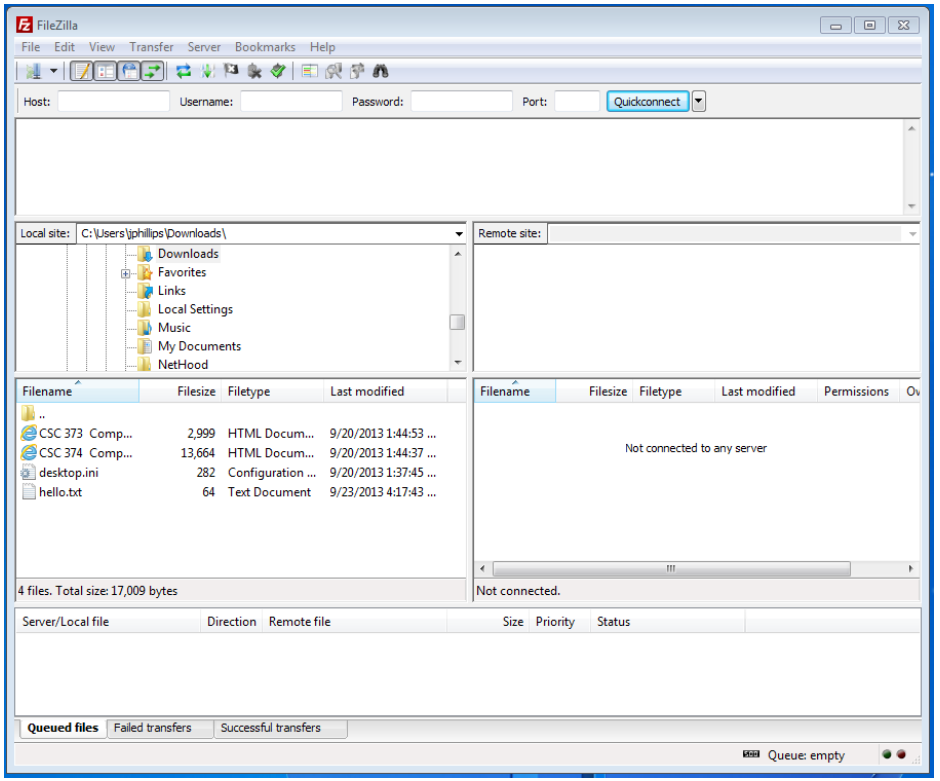


Transferring files

- Use an sftp (*S*ecure *F*ile *T*ransfer *P*rogram/*P*rotocol) like ***filezilla***
  - free at https://filezilla-project.org/
- ***Hostname***: one of:

  140.192.36.184
  140.192.36.185
  140.192.36.186
  140.192.36.187

- ***Username***: your Campus Connect
- ***Password***: Don't tell me!
- ***Port***: ***22***



Getting around

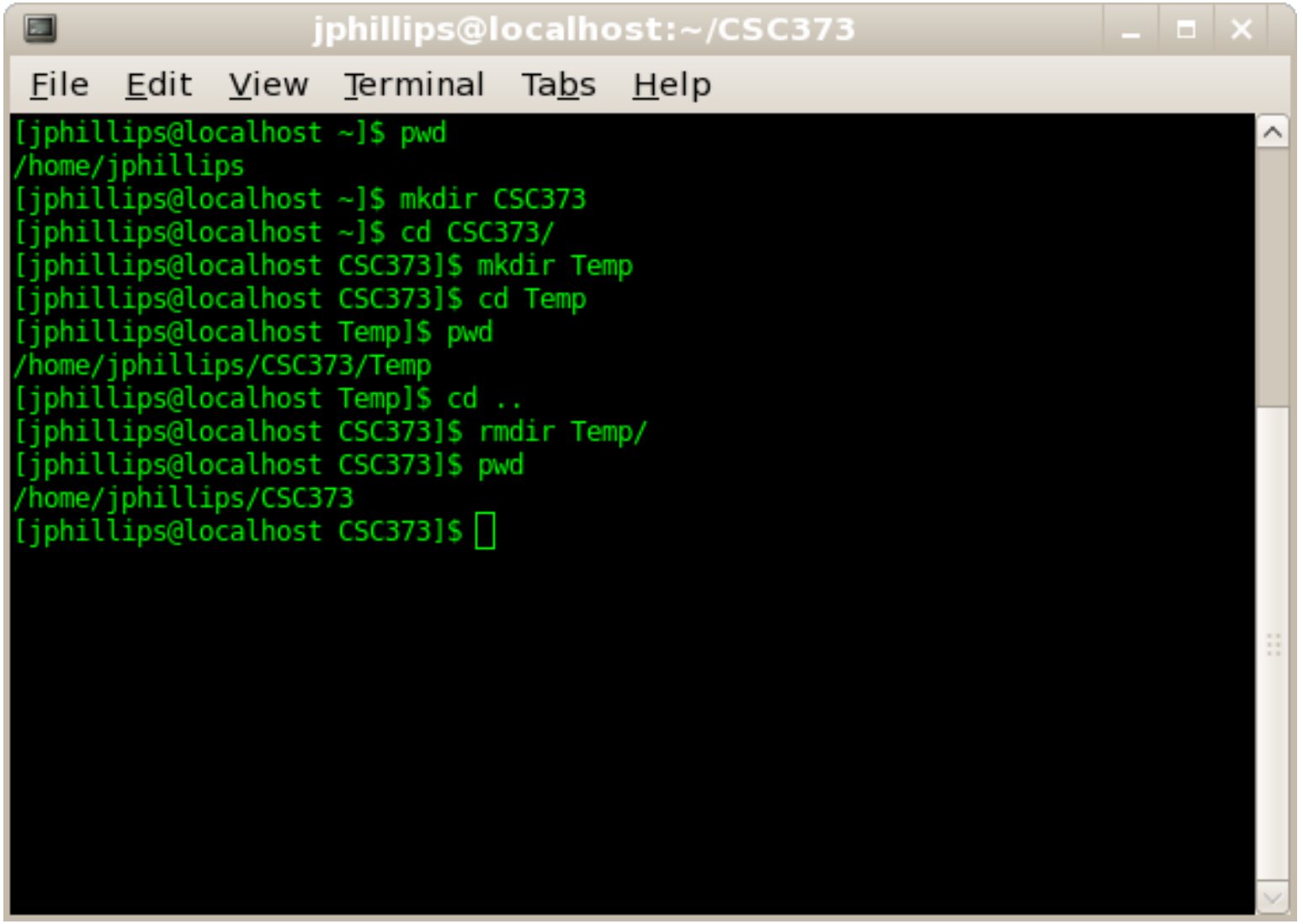
- Folders are called "*directories*"
  - They look like folders on GUIs like KDE or Gnome
  - *Use them*: They will organize your work for this class!
- Special directory names:

..	(two periods)	The parent of current directory
.	(one period)	The current directory
~	(tilde)	User's home directory
/	(forward slash)	Root directory
	Also used separator for subdirectories	

- Directory commands:

`mkdir dirName`	Make directory **`dirName`**
`cd dirName`	Change to dir **`dirName`**
`rmdir dirName`	Remove (delete) **`dirName`**
`pwd`	*P*rint *W*orking *D*irectory

Example


jphillips@localhost:~/CSC373
File Edit View Terminal Tabs Help
[jphillips@localhost ~]$ pwd
/home/jphillips
[jphillips@localhost ~]$ mkdir CSC373
[jphillips@localhost ~]$ cd CSC373/
[jphillips@localhost CSC373]$ mkdir Temp
[jphillips@localhost CSC373]$ cd Temp
[jphillips@localhost Temp]$ pwd
/home/jphillips/CSC373/Temp
[jphillips@localhost Temp]$ cd ..
[jphillips@localhost CSC373]$ rmdir Temp/
[jphillips@localhost CSC373]$ pwd
/home/jphillips/CSC373
[jphillips@localhost CSC373]$

Managing files

- Commands:

`ls`	*L*i*S*t files in current directory
`ls dirName`	*L*i*S*t files in dirName
`rm fileName`	*R*e*M*ove (delete) file dirName

- Wildcard chars for **`ls`** and **`rm`**:

`*`	Matches anything
`?`	Matches just one letter

Managing Files: the **`cat`** cmd

The (con)**`cat`**(enate) Unix command:

- Types **`file1`** to screen:

```
cat file1
```

- Types **`file1 file2 . . . fileN`** to screen:

```
cat file1 file2 . . . fileN
```

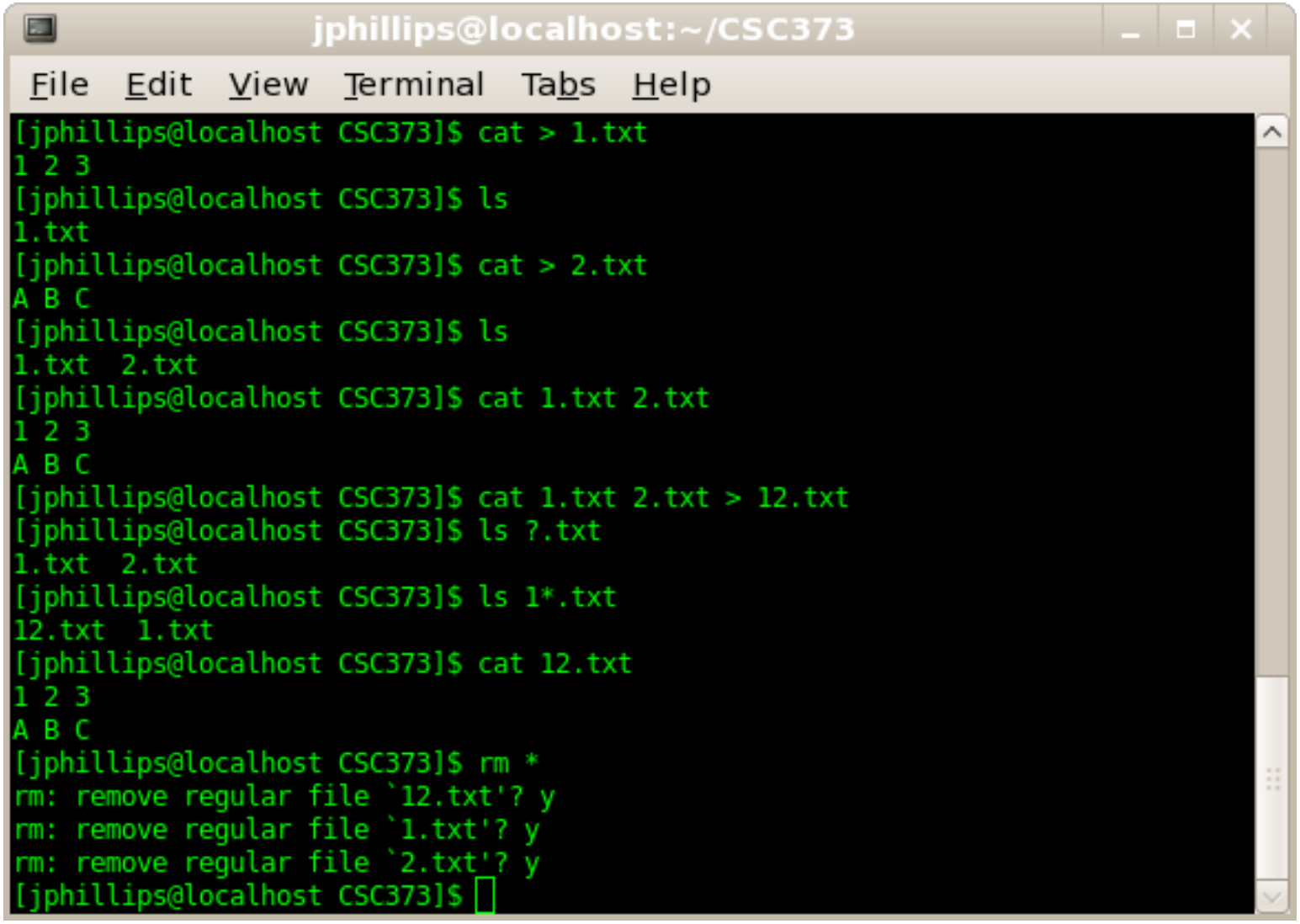
- Makes **`outFile`** the concatenation of **`file1 file2 . . . fileN`**:

```
cat file1 file2 . . . fileN > outFile
```

- Whatever you type on the keyboard goes into **`outFile`**. Stop with **`Ctrl-D`**. (An alternative to filezilla)

```
cat > outFile
```

Example


jphillips@localhost:~/CSC373
File Edit View Terminal Tabs Help
[jphillips@localhost CSC373]$ cat > 1.txt
1 2 3
[jphillips@localhost CSC373]$ ls
1.txt
[jphillips@localhost CSC373]$ cat > 2.txt
A B C
[jphillips@localhost CSC373]$ ls
1.txt 2.txt
[jphillips@localhost CSC373]$ cat 1.txt 2.txt
1 2 3
A B C
[jphillips@localhost CSC373]$ cat 1.txt 2.txt > 12.txt
[jphillips@localhost CSC373]$ ls ?.txt
1.txt 2.txt
[jphillips@localhost CSC373]$ ls 1*.txt
12.txt 1.txt
[jphillips@localhost CSC373]$ cat 12.txt
1 2 3
A B C
[jphillips@localhost CSC373]$ rm *
rm: remove regular file `12.txt'? y
rm: remove regular file `1.txt'? y
rm: remove regular file `2.txt'? y
[jphillips@localhost CSC373]$

Editing files

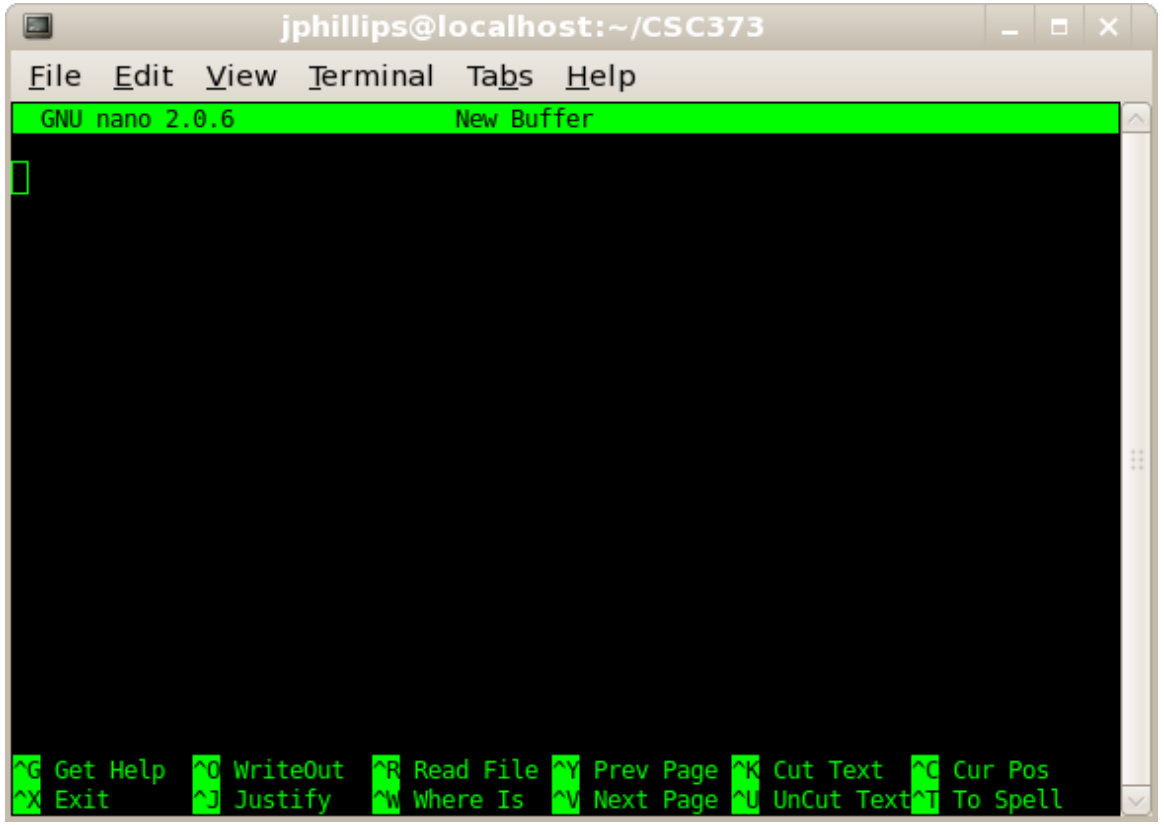
- Most popular Unix editors:

  `emacs`
  - Very powerful
  - For big, multi-file projects

  `vi`
  - Very flexible
  - For big files
  - See Joe's vi tutorial
- Serious hackers should learn both
- For the ***lazy***, I recommend **`nano`**

  `nano filename`

nano



Commands at bottom:

- Most important:
- Ctrl-X (exit)
- Ctrl-O (Write file)
- Ctrl-R (Read file)
- Ctrl-K (Del line)
- Ctrl-U (Paste line)

Compiling Files, 1

- Let's compile and run a file:
  - Type this file (either with **`cat`**, **`nano`** or **`vi`**)

```
#include          <stdio.h>
#include          <stdlib.h>

int       main      ()
{
  printf("Hello world!\n");
  return(EXIT_SUCCESS);
}
```

Compiling Files, 2

- Compiling:
  `gcc source.c -g -o executable`
  Run *G*nu *C* *C*ompiler on **`source.c`** with debugging info (**`-g`**) output (**`-o`**) to file **`executable`**.
- Running:
  `./executable`