

# A Gentle Introduction to Bash

# So what are we using?

The **shell** is a program that presents a *command line interface* which allows you to control your computer using commands entered with a keyboard.

**Bash** is probably the only *shell* you need to know.

A **terminal** is a program you run that gives you access to the shell.

# What does a shell command look like?

Commands often look like this:

```
<command name> <list of arguments>
```

```
e.g., cd ~/Documents/thesis/
```

Some commands don't require arguments, but this is rare.

# Navigating your computer

**pwd: print working directory**

*Find out where you are in your computer*

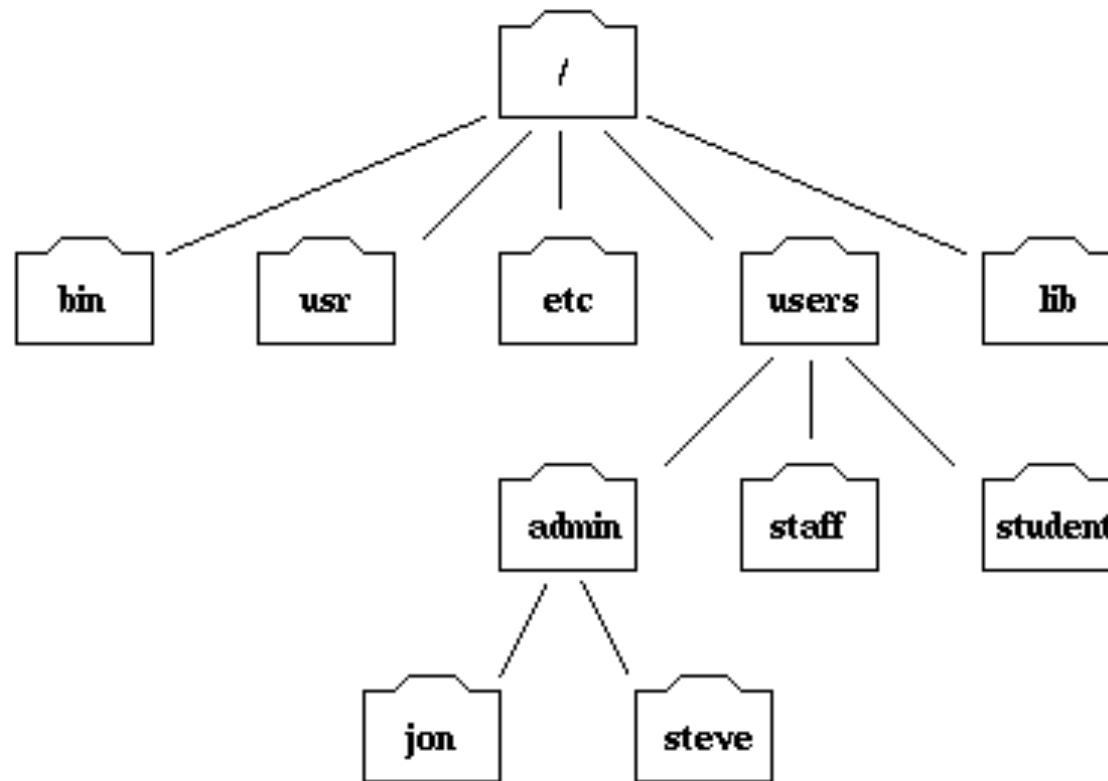
**cd <path>: change directory**

*Move to a different directory “path”*

**ls: list**

*List contents of current directory*

# File paths



**Part of the filesystem tree**

# Navigation, continued

If you are in `/users/` (see previous slide):

```
cd ../
```

Go up one level to /

```
cd ../lib/
```

Go up one level, and  
down into `/lib/`

```
cd admin/
```

Go down one level, into  
admin

```
cd admin/jon/
```

Go down two levels into  
admin/jon/

# Getting more out of `ls`

```
ls *.pdf
```

Show all files ending  
with `.pdf`

```
ls test*
```

Show all files beginning  
with `test`

```
ls ../dir2/*.txt
```

Show `.txt` files in another  
directory

```
ls -lh
```

Show more information  
in a readable format

```
ls -a
```

Show all hidden files

```
ls -G
```

Color-code output

# Changing Files and Directories

`mkdir <path>`: **make directory**

*Make a new directory "path"*

`cp <A> <B>`: **copy**

*Copy a file or folder from folder A to folder B*

`mv <A> <B>`: **move**

*Copy A to B, and then remove A*

`rm <A>`: **remove**

*Remove A - **PERMANENT!** There is no recycle bin!*

*Get in the habit of using `rm -i <A>` - you will be asked to confirm that you do indeed want to delete <A>*



# Individual Files

**touch** <A>: make new file A

**less** <A>: see first few lines of A

*To see more of A, keep scrolling down*

*To quit the view, press “:q”*

**head** <A>: see first 10 lines of A in shell

*head -n <NUM> <A>: see first NUM lines of A*

**file** <A>: get file type of A

*Not available in GitBash*

# Shortcuts

`cd ~`: go to home directory

`cd :` go to home directory

`cd -`: go to the last folder you were in before your current folder

`↑`: go back in shell history

`↓`: go forward in shell history

`ctrl+R`: search in shell history

# More shortcuts

`ctrl+C`: cancel command, get new prompt

`tab`: match to entered text

`ctrl+A`: go to beginning of entered line

`ctrl+E`: go to end of entered line

`ctrl+L`: clear terminal screen

# Iterating

What if you want to do something to a bunch of files?

```
for <variable> in <set>; do  
<commands>; done
```

Ex: copy all pdf files in current folder to a new folder ../newfolder:

```
for f in *.pdf; do cp $f ../  
newfolder/$f; done
```

# Further References

- <http://explainshell.com/>
- <http://stackoverflow.com/>
- <http://nicercode.github.io/2014-02-18-UTS/lessons/60-shell/>
- <http://www.skorks.com/2009/09/bash-shortcuts-for-maximum-productivity/>
- <https://openhatch.org/missions/windows-setup/open-git-bash-prompt> (GitBash users)
- `man <command>` (most users)
- `<command> --help` (GitBash users)