Command line basics:

2016 HPC boot camp

Max Joseph

July 18, 2016

Preliminaries

My assumption

You have completed the pre-workshop prep material
`https://www.rc.colorado.edu/training/workshop/2016basicsbootcamp`
if not, do so now

Our etherpad

`https://etherpad.net/p/hpcboot-bash`

Help

- chat
- stickies
- helpers
- neighbors

Figure 1:

Get data

(link on etherpad)
`https://www.dropbox.com/s/bhbg6qsjalpm56q/data-shell.zip?dl=0`

1. Download .zip file
2. Extract onto Desktop
3. Verify that you have a folder called `data-shell`
4. Green sticky when done

Let's go back

Figure 2:

Open a terminal emulator

Macintosh

Cmd+space: terminal

Windows

Windows key, then type “Git Bash” and press enter

Navigation

Key commands

List directory contents

`ls`

Change directory

`cd`

Print working directory

`pwd`

Challenge

What is your working directory?
What is in your working directory?

Solution

What is your working directory?

`pwd`

What is in your working directory?

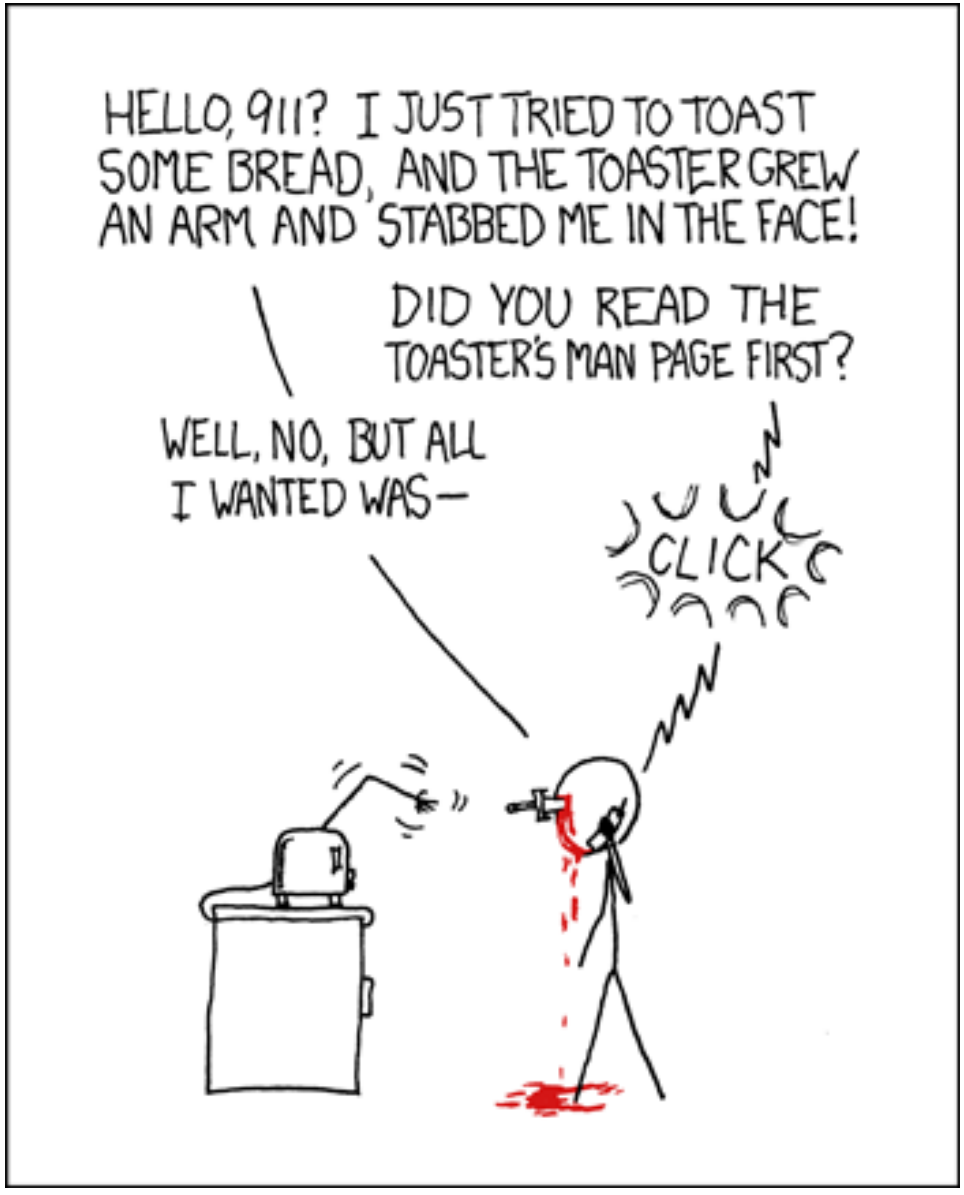
`ls`

Flags

```
ls -a   # show all

ls -l   # long form

ls -h   # human readable

ls -F   # append indicators

ls -R   # recursively list files

ls -R -a -h -l -F

ls -RahlF
```

Man pages (q to exit)

```
man ls
```



Challenge

1. Navigate to the folder `data-shell/`
2. List all of the files in that folder (including hidden files)

Solution

```
cd Desktop/data-shell

ls -a
```

Listing contents of directory

```
ls creatures
```

Challenge

1. List the contents of `data-shell/data/pdb/` **without changing your current working directory**

Solution

```
ls data/pdb
```

Where is home?

Use `cd ~` to navigate home.

1. What is the path to your home directory?
2. Is your home path the same as your neighbor's?

What's in your root directory?

Challenge:
List the contents of your root directory *without navigating to it*
i.e., do not use `cd`

Solution

```
ls /
```

Relative vs. absolute paths

Relative:

```
cd Desktop
```

```
cd ../Documents
```

Absolute:

```
cd /Users/max/Desktop
```

Challenge: using absolute vs. relative paths

Challenge: predicting output

`pwd` displays `/Users/thing/`
What will `ls ../backup` display?

Going up in the file tree

`..` always* refers to one level up in the directory tree

Files and directories

Key commands

```
nano
rm
mkdir
mv
cp
```

File operations

1. Use `nano` to edit data-shell/writing/haiku.txt
2. Put `haiku.txt` in `data-shell/writing/old/`

Bonus: Open a file with nano in read-only mode

Wildcards

In `data-shell/molecules/`

```
ls *.pdb
```

```
ls p*
```

```
ls *than*
```

Single character equivalent is `?`

```
ls ?thane.pdb
```

Using wildcards

When run in the molecules directory, which ls command will produce this output?
ethane.pdb methane.pdb

1. `ls *t*ane.pdb`
2. `ls *t?ne.*`
3. `ls *t??ne.pdb`
4. `ls ethane.*`

Bonus
Put all files with "thane" in the filename in a new directory called `thanes/`

Counting words, lines, and characters

In `data-shell/molecules/`
Count the number of lines in each .pdb file

```
wc *.pdb
```

Redirecting output

```
wc *.pdb > counts.txt
```

Piping

```
wc *.pdb | sort
```

Output from command 1 $\rightarrow$ input for command 2

Challenge

Create a file `data-shell/data/pdb/line-counts.txt` that:

- contains line counts for each .pdb file in `data-shell/data/pdb` with a molecule name ending in "ol"
- is sorted from smallest to largest line count

Hint: use wildcards, piping, and output redirection

Solution

Create a file `data-shell/data/pdb/line-counts.txt` that:

- contains line counts for each .pdb file in `data-shell/data/pdb` with a molecule name ending in "ol"
- is sorted from smallest to largest line count

```
cd data/pdb
```

```
wc -l *ol.pdb | sort > line-counts.txt
```