# Systems Programming

## Lecture 2: Introduction to UNIX

Stuart James

stuart.a.james@durham.ac.uk

## Module Organisation and Content

- Term 1: Systems Programming -> Stuart
- Term 2: Functional Programming (Haskell) -> Max
- Term 2: Object-Oriented Programming -> Nelly

## Covered Last Lecture

- Intro into C & How to Compile
- Pre-processor & macros

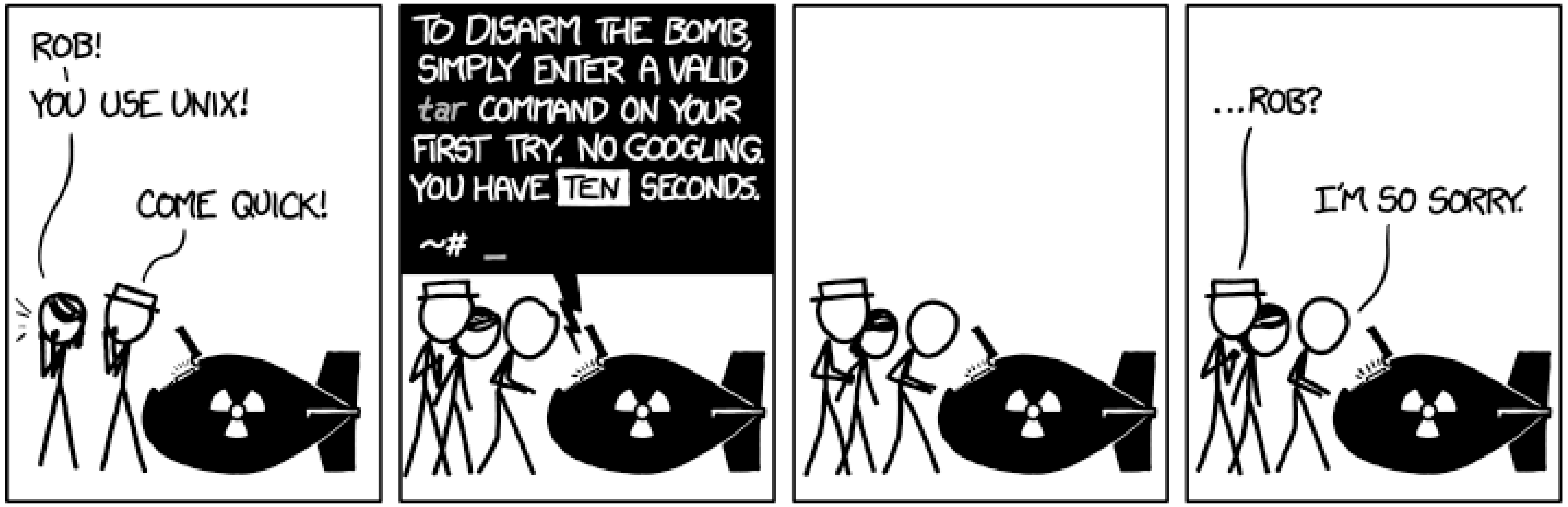
## Points raised

- The coursework will require using Makefiles and .c/cpp files NOT the Jupyter notebook
- YouTube videos can help get Jupyter running on your own machine

## Recap

https://PollEv.com/stuartjames

## UNIX



ROB!
YOU USE UNIX!
COME QUICK!
TO DISARM THE BOMB, SIMPLY ENTER A VALID tar COMMAND ON YOUR FIRST TRY. NO GOOGLING. YOU HAVE TEN SECONDS.
~# _
...ROB?
I'M SO SORRY.

## A (very) short history of UNIX

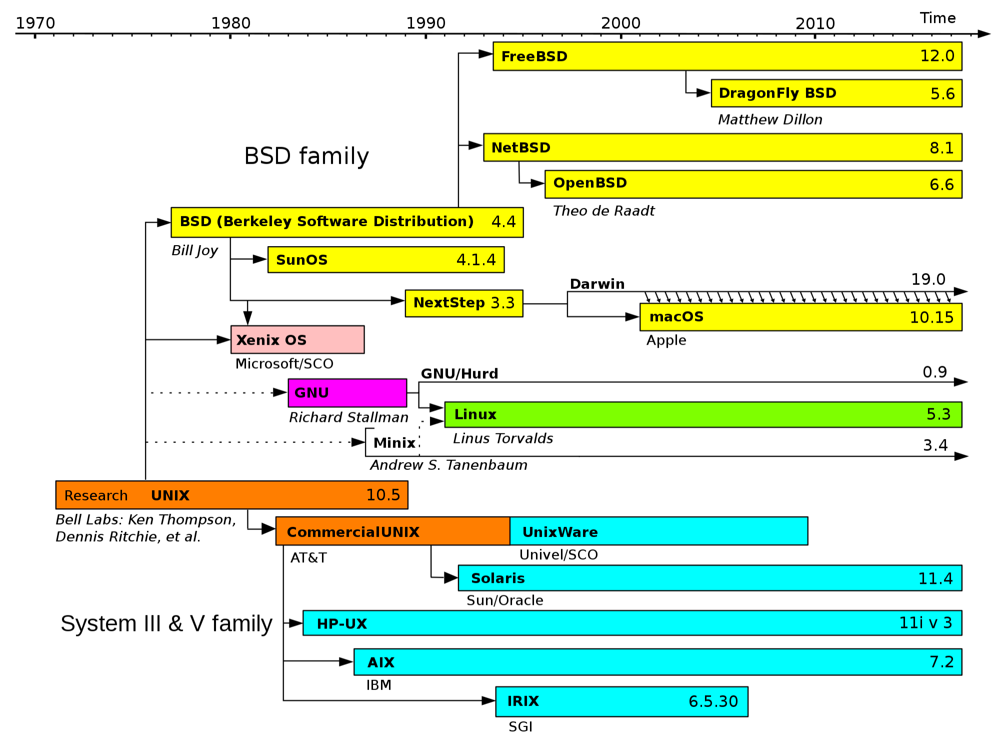
- 1963-1969 MULTICS (Multiplexed Information and Computing Service)
    - a high-availability, modular, multi-component system;
    - continued until 1985;
    - last system decommissioned in 2000

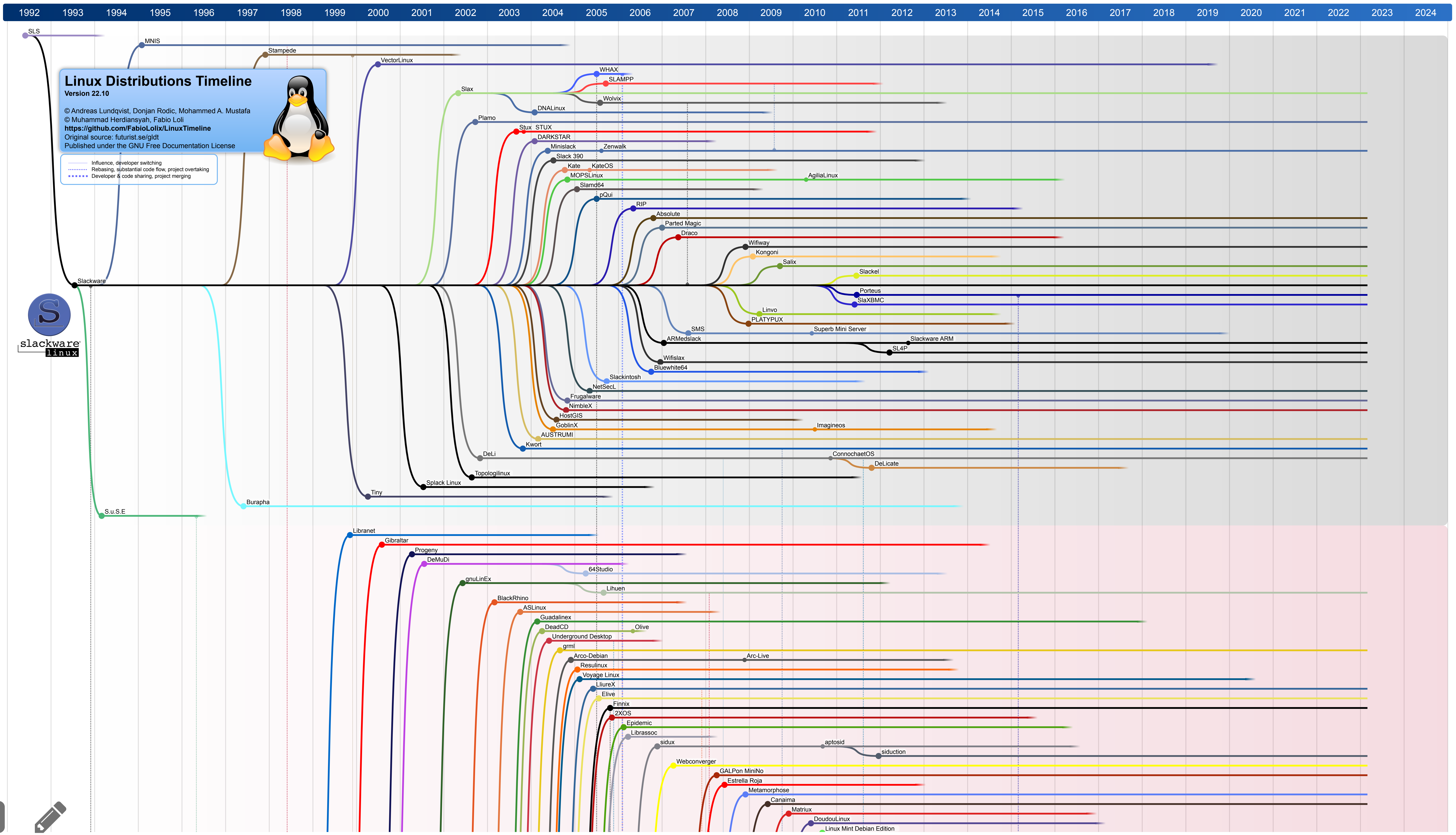
## A (very) short history of UNIX

- UNIX: the opposite of MULTICS
  - Simpler and faster approach than MULTICS
  - initial assembler implementation by Ken Thompson and Dennis Ritchie for PDP-7 and PDP-11 (1960's, Bell Labs)
  - rewritten in C in 1973: the first operating system written in a high-level portable language
  - continuous evolution of various dialects of UNIX and its routines for over 50 years

## A (very) short history of UNIX

- Focus on:
  - Multiuser Operating System
  - High-end users (skilled)


1970
1980
1990
2000
2010
Time
FreeBSD 12.0
DragonFly BSD 5.6
Matthew Dillon
BSD family
NetBSD 8.1
OpenBSD 6.6
Theo de Raadt
BSD (Berkeley Software Distribution) 4.4
Bill Joy
SunOS 4.1.4
Darwin 19.0
NextStep 3.3
macOS 10.15
Apple
Xenix OS
Microsoft/SCO
GNU/Hurd 0.9
GNU
Richard Stallman
Linux 5.3
Linus Torvalds
Minix 3.4
Andrew S. Tanenbaum
Research UNIX 10.5
Bell Labs: Ken Thompson, Dennis Ritchie, et al.
CommercialUNIX
AT&T
UnixWare
Univel/SCO
Solaris 11.4
Sun/Oracle
System III & V family
HP-UX 11i v 3
AIX 7.2
IBM
IRIX 6.5.30
SGI



# Linux Distributions Timeline

Version 22.10

© Andreas Lundqvist, Donjan Rodic, Mohammed A. Mustafa
© Muhammad Herdiansyah, Fabio Loli
**https://github.com/FabioLolix/LinuxTimeline**
Original source: futurist.se/gldt
Published under the GNU Free Documentation License

Influence, developer switching
Rebasing, substantial code flow, project overtaking
Developer & code sharing, project merging

## The Shell

- A powerful way to perform work on a computer through a text interface
  - Run programs
  - Control how the programs work
- Ability to move around between different directories/folders on a computer
- Perform sequences of commands to achieve even more complex work
- There are many choices for which shell to use
  1. A popular shell is **bash** (bourne again shell)
  2. Recently MacOS moved to **zsh** (Z shell)

## The Shell

- If you don't use Linux or MacOS:
    - Try the Windows Subsystem for Linux (WSL)
        - run Linux commands directly under Windows
    - Alternatively try a virtual machine.
    - Additionally, recommmended: Get your mira login and try out the shell there.

## Basic Commands

- Where am I?
  - pwd (print working directory)
- What is here?
  - ls (list)

**Note: the ! is not needed in an actual shell (I'm running a jupyter notebook)**

## Basic Commands

- Where am I?
    - pwd (print working directory)
- What is here?
    - ls (list)

**Note: the ! is not needed in an actual shell (I'm running a jupyter notebook)**

In [6]:

```
!pwd
```

```
/Users/sjames/Teaching/COMP2221_2425_Lecture_Materials/lecture-slides/lecture2-UNIX1
```

## Basic Commands

- Where am I?
  - pwd (print working directory)
- What is here?
  - ls (list)

**Note: the ! is not needed in an actual shell (I'm running a jupyter notebook)**

In [6]:

```
!pwd
```

```
/Users/sjames/Teaching/COMP2221_2425_Lecture_Materials/lecture-slides/lecture2-UNIX1
```

In [2]:

```
!ls -l
```

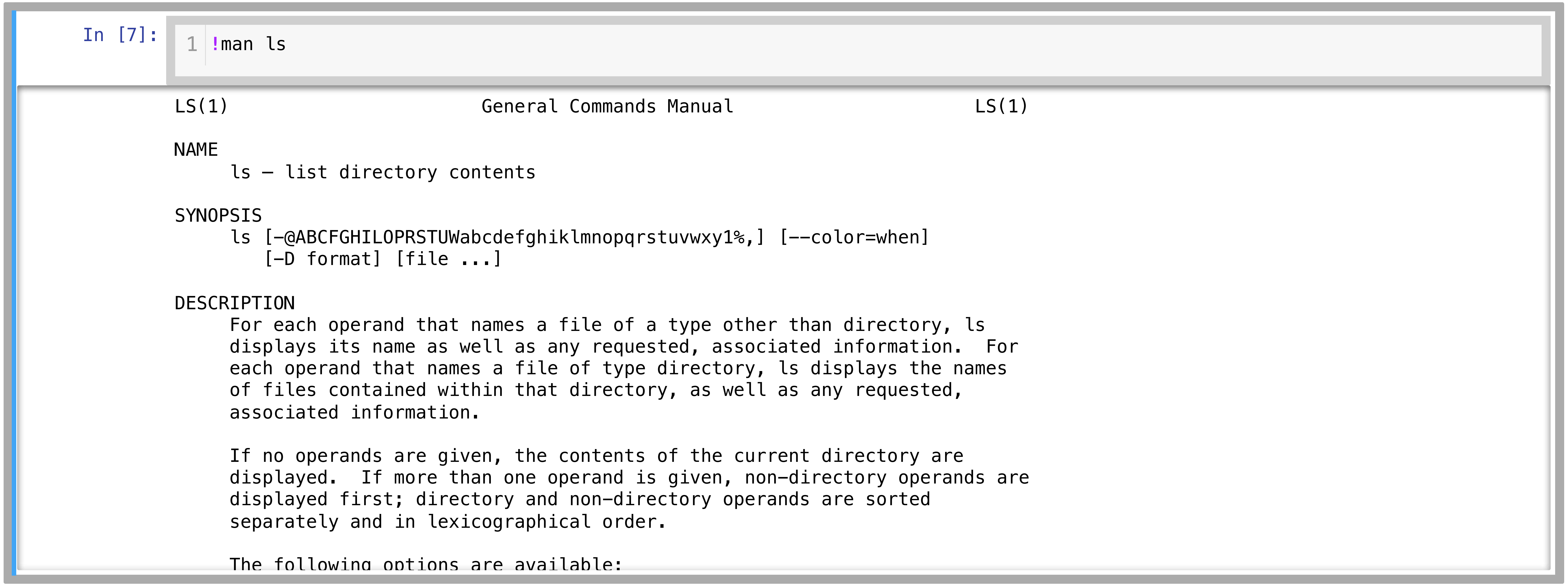
```
total 8928
drwxr-xr-x  5 sjames  staff      160  7 Sep 15:31 for-quiz
drwxr-xr-x  8 sjames  staff      256  9 Oct 19:19 images
-rw-r--r--@ 1 sjames  staff   256137  9 Oct 19:31 introunix-part1.ipynb
-rw-r--r--  1 sjames  staff   850428  7 Sep 15:31 introunix-part1.slides.html
-rw-r--r--@ 1 sjames  staff  3454594  7 Sep 15:31 lecture2-slides.pdf
-rw-r--r--  1 sjames  staff     2416  7 Sep 15:31 rise.css
```

## Basic Commands

- help?
    - man (manual)

## Basic Commands

- help?
  - man (manual)



```
In [7]: !man ls
```

```
LS(1)                  General Commands Manual                  LS(1)

NAME
     ls – list directory contents

SYNOPSIS
     ls [-@ABCFGHILOPRSTUWabcdefghiklmnopqrstuvwxy1%,] [--color=when]
        [-D format] [file ...]

DESCRIPTION
     For each operand that names a file of a type other than directory, ls
     displays its name as well as any requested, associated information.  For
     each operand that names a file of type directory, ls displays the names
     of files contained within that directory, as well as any requested,
     associated information.

     If no operands are given, the contents of the current directory are
     displayed.  If more than one operand is given, non-directory operands are
     displayed first; directory and non-directory operands are sorted
     separately and in lexicographical order.

     The following options are available:
```

## Basic Commands

Go somewhere else

- `cd` (change directory)
    - `.` = current directory
    - `~` = home folder
    - `..` = one folder up

**Note: the % is not needed in an actual shell (I'm running a jupyter notebook)**

## Basic Commands

### Go somewhere else

- `cd` (change directory)
  - `.` = current directory
  - `~` = home folder
  - `..` = one folder up

**Note: the % is not needed in an actual shell (I'm running a jupyter notebook)**

```
In [8]: %cd
        %pwd

/Users/sjames

Out[8]: '/Users/sjames'
```

## Basic Commands

### Go somewhere else

- `cd` (change directory)
    - `.` = current directory
    - `~` = home folder
    - `..` = one folder up

**Note: the % is not needed in an actual shell (I'm running a jupyter notebook)**

```
In [8]:  %cd
         %pwd

         /Users/sjames

Out[8]:  '/Users/sjames'
```

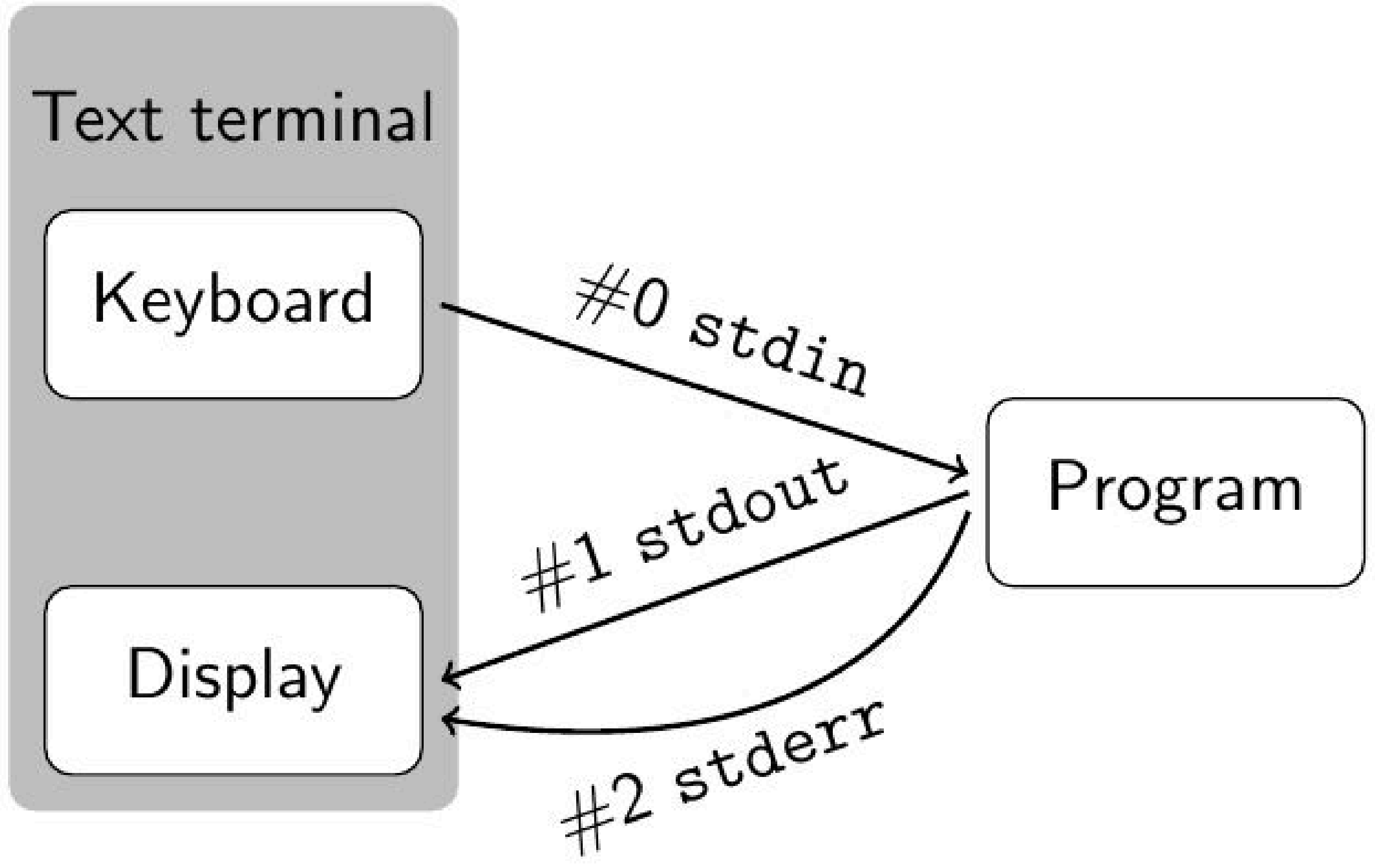
```
In [10]: %cd /Users/sjames/Teaching/COMP2221_2425_Lecture_Materials/lecture-slides/lecture2-UNIX1

         /Users/sjames/Teaching/COMP2221_2425_Lecture_Materials/lecture-slides/lecture2-UNIX1
```

## Ethos of the UNIX Shell

- Not one monolithic program
- Instead many small programs
- Allow user to combine these together to make new functionality
    - Using pipes
    - Using script files

## `stdin`, `stdout` and `stderr`

- Remove the need to worry about I/O devices
- Two types of output, each can be redirected



## Pipes

- The shell provides you with many small 'tools'
    - The power comes from composing them together.
    - Pipes provide a means to do this.
    - Each command takes input (from the keyboard).
    - Each command produces output (to the screen).

## Pipes

- We can redirect input or output
  - `<` take input from a file
  - `>` write output to a file
  - `|` take the output of one command and use as input to next

**Example: output pipes**

- Add "`>`" or "`>>`" and the name of a file after your command before you hit "Enter/Return" -- e.g. "

`ps > file.txt`"

- If the file doesn't exist already, it will be created for you in the directory in which you are working
- "`>>`" appends, "`>`" overwrites -- so be careful when using "`>`"!!

## Using pipes in practice

- How many files in a directory?

## Using pipes in practice

- How many files in a directory?

```
In [11]: !ls
```

```
directoryList.txt           introunix-part1.slides.html
for-quiz                    lecture2-slides.pdf
images                      rise.css
introunix-part1.ipynb       test.txt
introunix-part1.pdf
```

## Using pipes in practice

- How many files in a directory?

In [11]:
```
!ls
```

```
directoryList.txt            introunix-part1.slides.html
for-quiz                     lecture2-slides.pdf
images                       rise.css
introunix-part1.ipynb        test.txt
introunix-part1.pdf
```

In [12]:
```
!ls | wc -l
```

```
       9
```

## Using pipes in practice

- How many files in a directory?

```
In [11]: !ls
```

```
directoryList.txt            introunix-part1.slides.html
for-quiz                     lecture2-slides.pdf
images                       rise.css
introunix-part1.ipynb        test.txt
introunix-part1.pdf
```

```
In [12]: !ls | wc -l
```

```
       9
```

- Write the list of files in a directory to a file

## Using pipes in practice

- How many files in a directory?

In [11]:
```
!ls
```

```
directoryList.txt           introunix-part1.slides.html
for-quiz                    lecture2-slides.pdf
images                      rise.css
introunix-part1.ipynb       test.txt
introunix-part1.pdf
```

In [12]:
```
!ls | wc -l
```

```
       9
```

- Write the list of files in a directory to a file

In [13]:
```
!ls > directoryList.txt
!cat directoryList.txt
```

```
directoryList.txt
for-quiz
images
introunix-part1.ipynb
introunix-part1.pdf
introunix-part1.slides.html
lecture2-slides.pdf
```

## Using pipes in practice

- How many files in a directory?

In [11]:
```
!ls
```

```
directoryList.txt           introunix-part1.slides.html
for-quiz                    lecture2-slides.pdf
images                      rise.css
introunix-part1.ipynb       test.txt
introunix-part1.pdf
```

In [12]:
```
!ls | wc -l
```

```
       9
```

- Write the list of files in a directory to a file

In [13]:
```
!ls > directoryList.txt
!cat directoryList.txt
```

```
directoryList.txt
for-quiz
images
introunix-part1.ipynb
introunix-part1.pdf
introunix-part1.slides.html
lecture2-slides.pdf
```

## Using pipes in practice

- How many files in a directory?

In [11]:
```
!ls
```

```
directoryList.txt             introunix-part1.slides.html
for-quiz                      lecture2-slides.pdf
images                        rise.css
introunix-part1.ipynb         test.txt
introunix-part1.pdf
```

In [12]:
```
!ls | wc -l
```

```
       9
```

- Write the list of files in a directory to a file

In [13]:
```
!ls > directoryList.txt
!cat directoryList.txt
```

```
directoryList.txt
for-quiz
images
introunix-part1.ipynb
introunix-part1.pdf
introunix-part1.slides.html
lecture2-slides.pdf
```

## `grep`

- `grep` is a search tool
  - can search through files
  - can also search through the results obtained by running a command - e.g., `ps`

## `grep`

- `grep` is a search tool
  - can search through files
  - can also search through the results obtained by running a command - e.g., `ps`

In [15]:

```
!grep -ri "hello"
```

```
./test.txt:hello this is an example
./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      "./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:
\"./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      \\\"grep: hello: No such file or directory\\\\n\\\"\\n\",\n",
./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      "./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:
\"./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:    \\\"!grep . -ri \\\\\\\"hello\\\\\\\"\\\"\\n\",\n",
./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      "./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:
\"./introunix-part1.ipynb:      \\\"grep: hello: No such file or directory\\\\n\\\"\\n\",\n",
./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      "./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:
\"./introunix-part1.ipynb:    \\\"!grep . -ri \\\\\\\"hello\\\\\\\"\\\"\\n\",\n",
./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      "./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:
\"./hello.txt:hello\\n\",\n",
./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      "./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:
\"./introunix-part1.slides.html:Makefile     README.md    helloworld.c power.c      power.h\\n\",\n",
./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      "./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:
\"./introunix-part1.slides.html:<span class=\\\"o\\\">!</span>mv<span class=\\\"w\\\"> </span>helloworld.c<span class=\\\"w
\\\"> </span>src\\n\",\n",
./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      "./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:
\"./introunix-part1.slides.html:\\t<span class=\\\"ansi-red-fg\\\">deleted:    helloworld.c</span>\\n\",\n",
./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      "./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:
\"./introunix-part1.slides.html:<div class=\\\" highlight hl-ipython3\\\"><pre><span></span><span class=\\\"o\\\">!</span>gi
t<span class=\\\"w\\\"> </span>rm<span class=\\\"w\\\"> </span>helloworld.c\\n\",\n",
./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:      "./.ipynb_checkpoints/introunix-part1-checkpoint.ipynb:
```

## `grep`

- Uses regular expressions for matching
  - `grep "help" file.txt`
    - lists all lines in `file.txt` containing the word '`help`'

## Regular Expressions

- A concise way to match different strings
    - `.` - matches any single character (except a newline character)
    - `*` - matches any number of the proceeding character
    - `?` - matches one of the proceeding character
    - `+` - matches one or more of the proceeding character
    - `[ABC]` - class as single character
    - `[A-Z]` - all the upper case characters `A` to `Z`
- e.g. `[A-Za-z]*[0-9].txt`

## Regular Expressions

- A concise way to match different strings
  - `.` - matches any single character (except a newline character)
  - `*` - matches any number of the proceeding character
  - `?` - matches one of the proceeding character
  - `+` - matches one or more of the proceeding character
  - `[ABC]` - class as single character
  - `[A-Z]` - all the upper case characters `A` to `Z`
- e.g. `[A-Za-z]*[0-9].txt`

In [23]:

```
!ls */* | grep '.*\.jpg\|png'
```

```
for-quiz/PXL_20230106_104415973.MP.jpg
for-quiz/cube_p.png
for-quiz/logo-gcc.png
images/durhamlogo.png
images/quiz-qr.png
images/stdoutinerr.jpg
images/t-and-r.jpg
images/tar.png
```

## Getting help other than the manual... AI Prompts

- How would you prompt a terminal to find and count all files in all folders under the current

## Getting help other than the manual... AI Prompts

- How would you prompt a terminal to find and count all files in all folders under the current

```
In [25]: !ls -R | grep -v '^d' | wc -l
```

```
20
```

## Getting help other than the manual... AI Prompts

- How would you prompt a terminal to find and count all files in all folders under the current

In [25]:
```
!ls -R | grep -v '^d' | wc -l
```

```
20
```

In [24]:
```
!find . -type f | wc -l
```

```
18
```

## Summary

- UNIX and its history
- The Shell
- Piping
- grep

## Next Lecture

- More on the Shell
- Basics of Git