Introduction to working with UNIX and bash

•••

Argyris Zardillis and Paul Judge

With thanks to the developers of the Software Carpentry "Unix Shell" course

Why are you here?

Join the etherpad at http://board.net/p/UnixIntroFeb2020 :

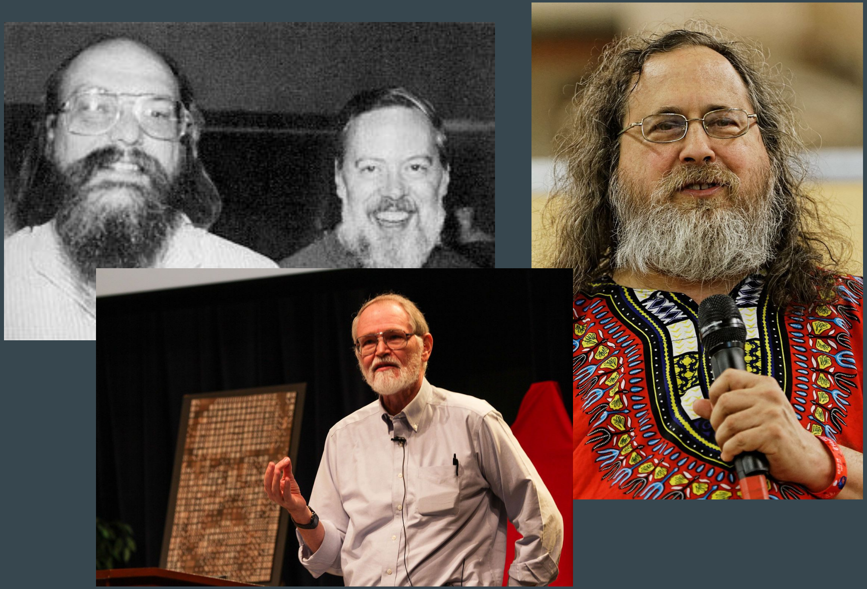
- Introduce yourself and say hello
- What's your background?
- What do you hope to get out of this course?
- Is there one question or topic you particularly want to cover today?

Why you should be here !

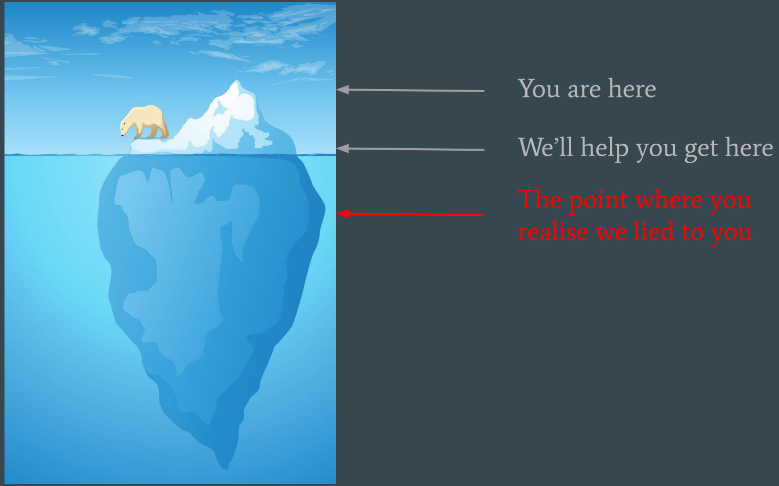
1. You probably can't avoid UNIX
2. UNIX is a free reproducible platform
3. UNIX is programmable from the ground up with bash you can record and reproduce every part of your analysis
4. Day off work with free coffee?

What this course isn't

History of computing

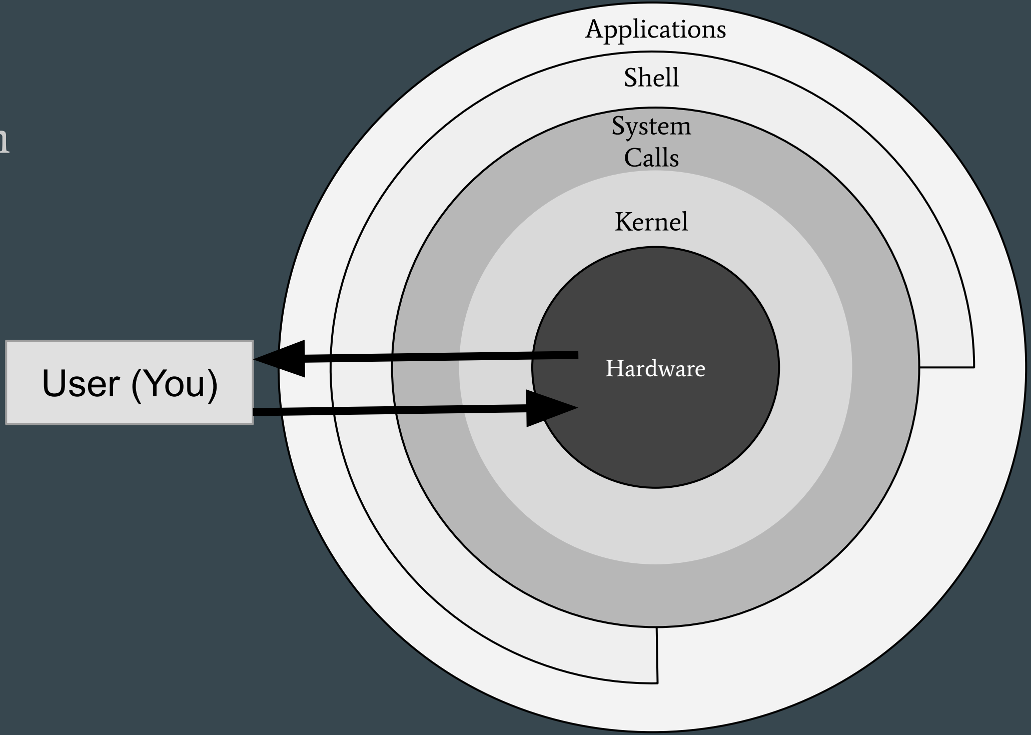


In depth
(but that's probably a good thing)



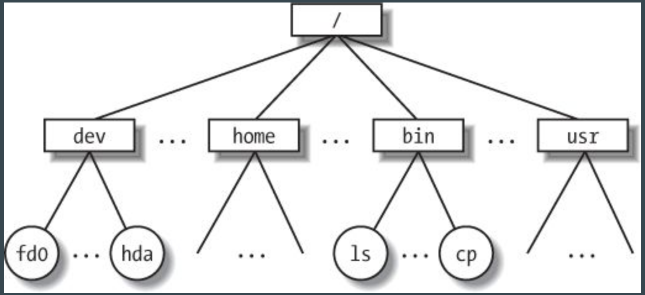
Unix uses lots of abstractions and terminology to make it accessible. What are we focussing on?

- The bash shell
- Files and the file system
- Processes
- Users

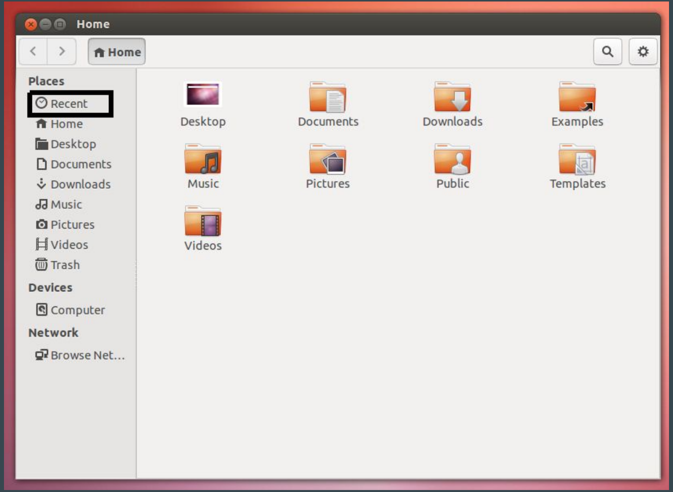


The Filesystem

Try typing “tree /”



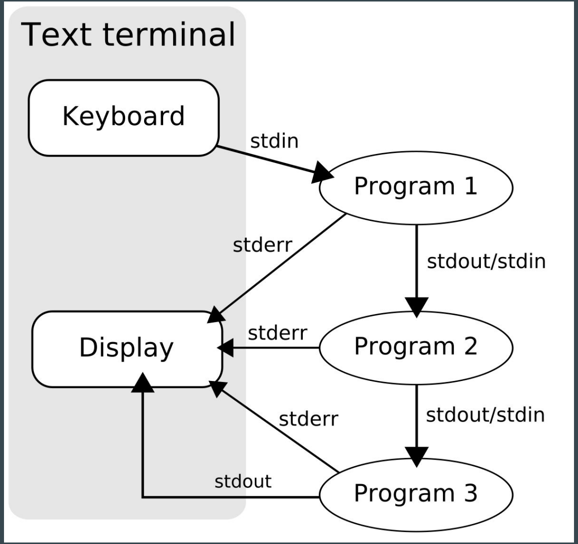
All representations of the same thing



/home/participant/Desktop/data-shell

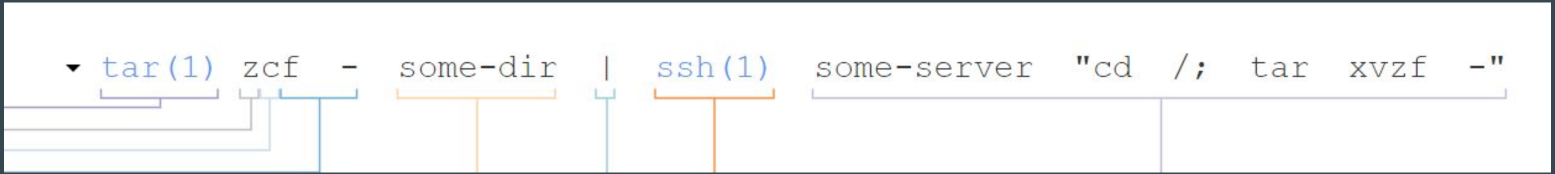
Processes

- Try typing “top”
- Press ‘q’ to get out
- Almost everything you do in bash creates a new process
- Shared interface and management system allows us to connect simple processes together to perform complex tasks



The bash terminal

- Bash is the name of the shell, there are others but don't worry about it
- Bash is the language
- Bash is also the tool, the terminal in front of you is connected to a bash process
- It's not always pretty but it's flexible enough to do anything the operating system can do
- It brings together all of the other abstractions in UNIX
- But it's not always the answer



```
tar(1) zcf - some-dir | ssh(1) some-server "cd /; tar xvzf -"
```

Look familiar?

cd

ls

rm

chown

mkdir

cp

cat

chmod

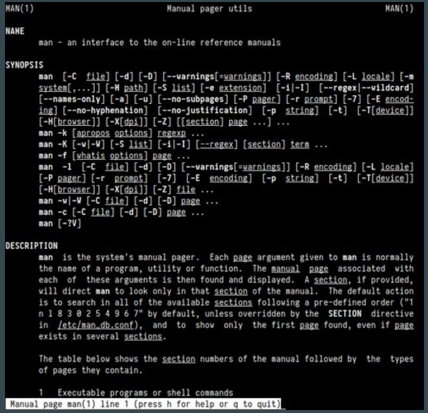
mv

grep

wc

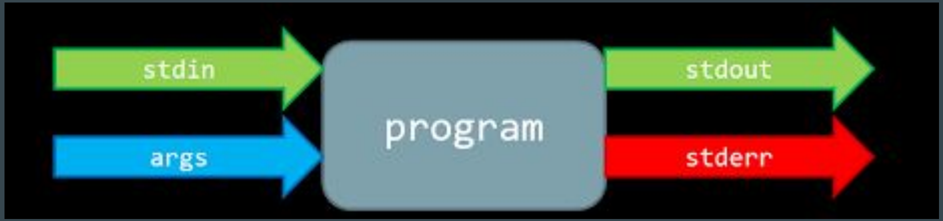
What should the take-aways be?

- Structures
- Abstractions
- Not specifics
- Not manual pages
- Not commands … except a couple



```
MAN(1)                        Manual pager utils                        MAN(1)

NAME
       man - an interface to the on-line reference manuals

SYNOPSIS
       man  [-C  file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale] [-m
       system[,...]] [-M path] [-S list] [-e extension]  [-i|-I]  [--regex|--wildcard]
       [--names-only] [-a] [-u] [--no-subpages] [-P pager] [-r prompt] [-7] [-E encod-
       ing] [--no-hyphenation]  [--no-justification]  [-p  string]  [-t]  [-T[device]]
       [-H[browser]] [-X[dpi]] [-Z] [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
       man -f [whatis options] page ...
       man  -l  [-C  file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale]
       [-P pager] [-r  prompt]  [-7]  [-E  encoding]  [-p  string]  [-t]  [-T[device]]
       [-H[browser]] [-X[dpi]] [-Z] file ...
       man -w|-W [-C file] [-d] [-D] page ...
       man -c [-C file] [-d] [-D] page ...
       man [-?V]

DESCRIPTION
       man  is the system's manual pager.  Each page argument given to man is normally
       the name of a program, utility or function.  The manual  page  associated  with
       each  of  these arguments is then found and displayed.  A section, if provided,
       will direct man to look only in that section of the manual.  The default action
       is to search in all of the available sections following a pre-defined order ("1
       n 1 8 3 0 2 5 4 9 6 7" by default, unless overridden by the  SECTION  directive
       in  /etc/man.db.conf),  and  to  show  only  the first page found, even if page
       exists in several sections.

       The table below shows the section numbers of the manual followed by  the  types
       of pages they contain.

       1   Executable programs or shell commands
 Manual page man(1) line 1 (press h for help or q to quit)
```

<command name> <arg 1> <arg2> …



Before we begin

- Use the post-its
- Don’t rely on tricks ... but here are some tricks
  - Use the arrow keys to see your previous commands
  - Tap tab once to let bash try and complete your command (good for long file paths)
  - Tap tab twice to have bash list all the possible ways you could complete your command
  - If things goes wrong Ctrl + C will kill whatever the active process is in your terminal and get you back to the prompt