University of Warwick
Computing Society

Linux 101

An introduction to Linux

David Richardson

UWCS Tech Officer

tech@uwcs.co.uk

Alias: Tankski

In This Talk

In This Talk

- An introduction to Linux
- An introduction to the shell
- Questions
- Pizza

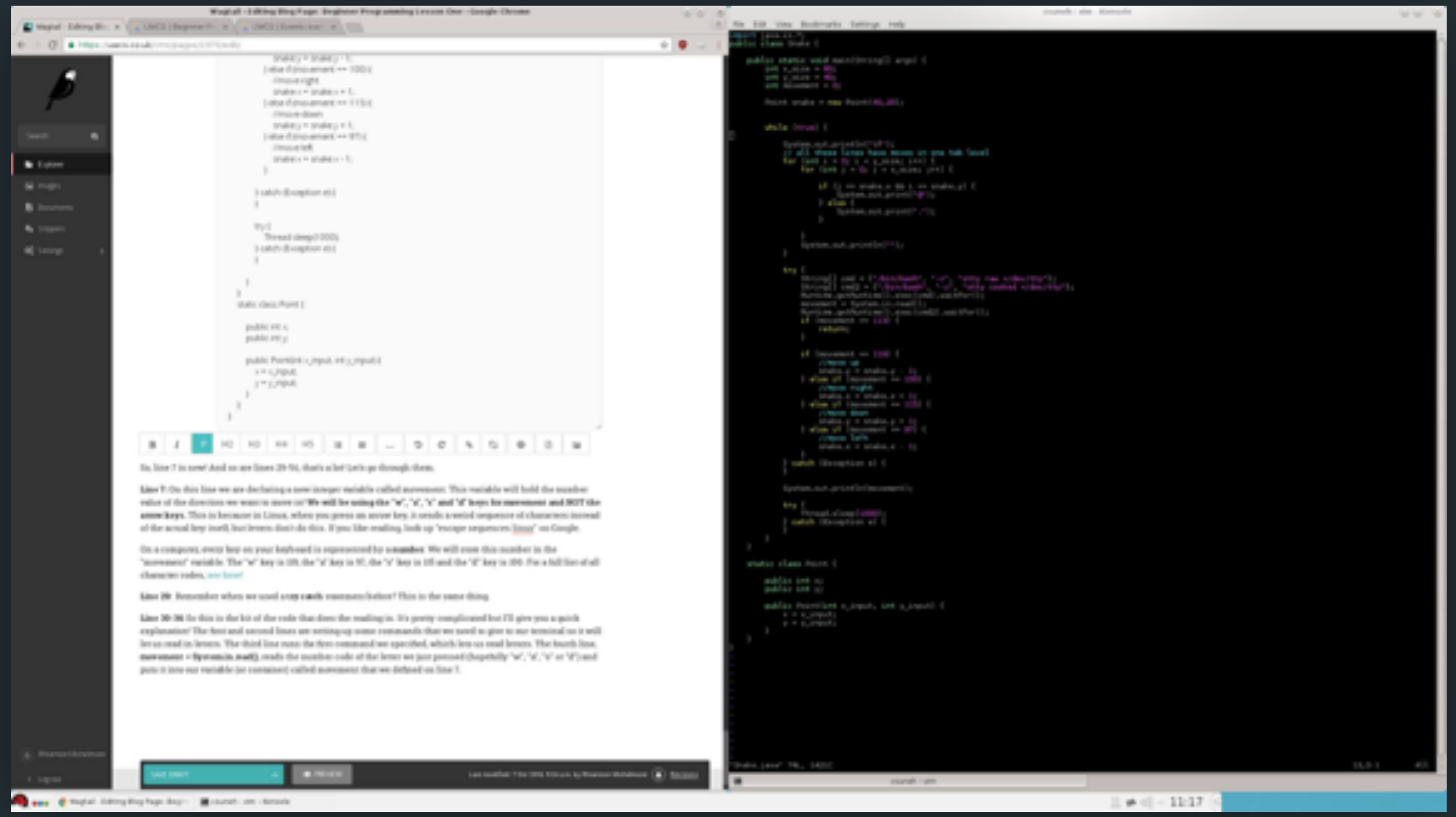
Introduction to Linux

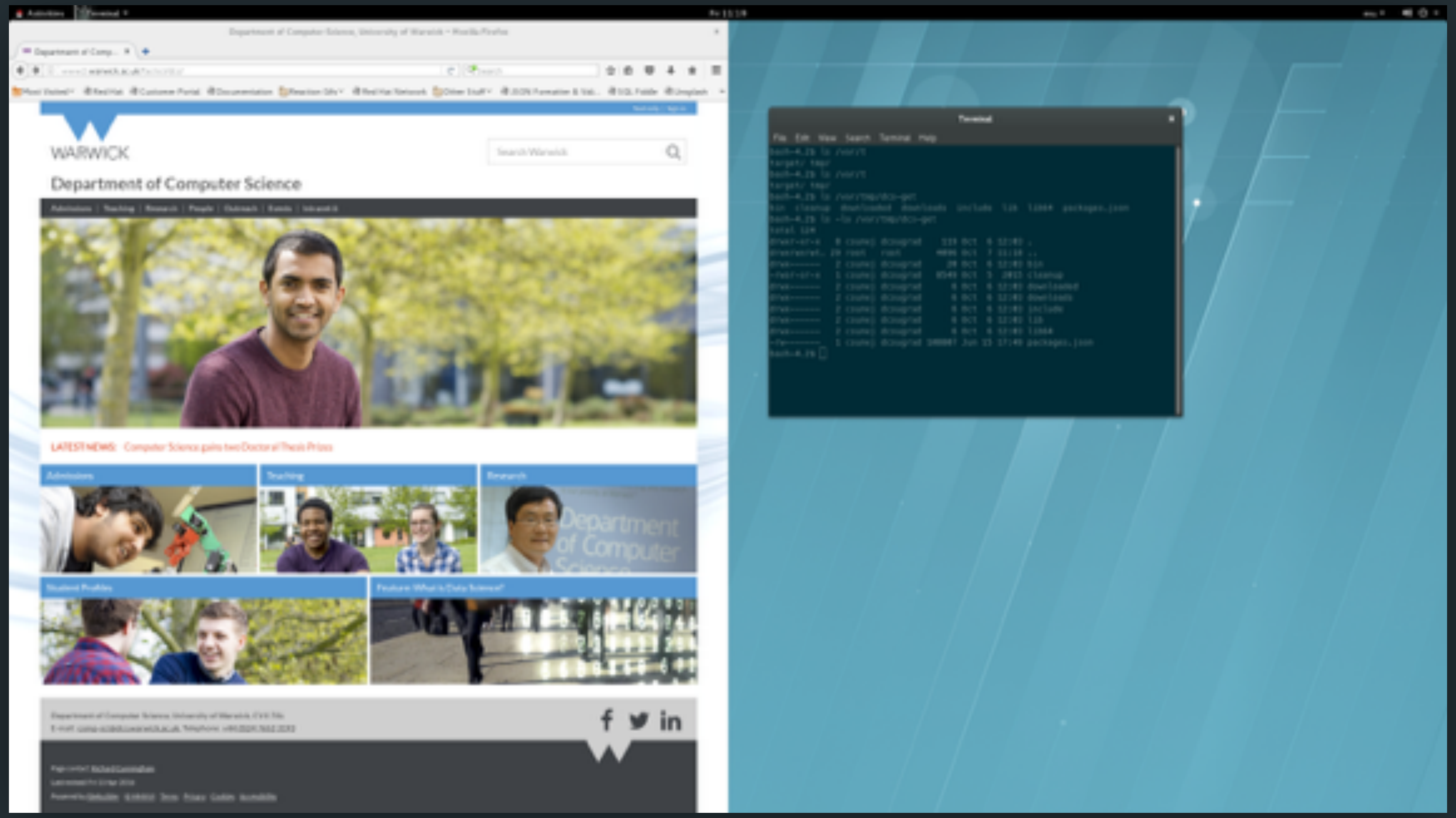
Introduction to Linux

- Linux is an operating system like Windows or macOS
  - Primarily a ‘kernel’ and some utility programs to interact with it
- Distributed freely with source code available online
- Maintained by an army of volunteers
- Most commonly found in web servers

Introduction to Linux

- The Linux desktop is very similar to those of Windows or macOS
- Departmental machines have 2 desktops:




WARWICK
Department of Computer Science
Search Warwick
LATEST NEWS: Computer Science gains two Doctoral Thesis Prizes
Admissions
Teaching
Research
Student Profiles
Feature: What is Data Science?
Terminal
File Edit View Search Terminal Help

- Functionally the same, possible differences in utility program names

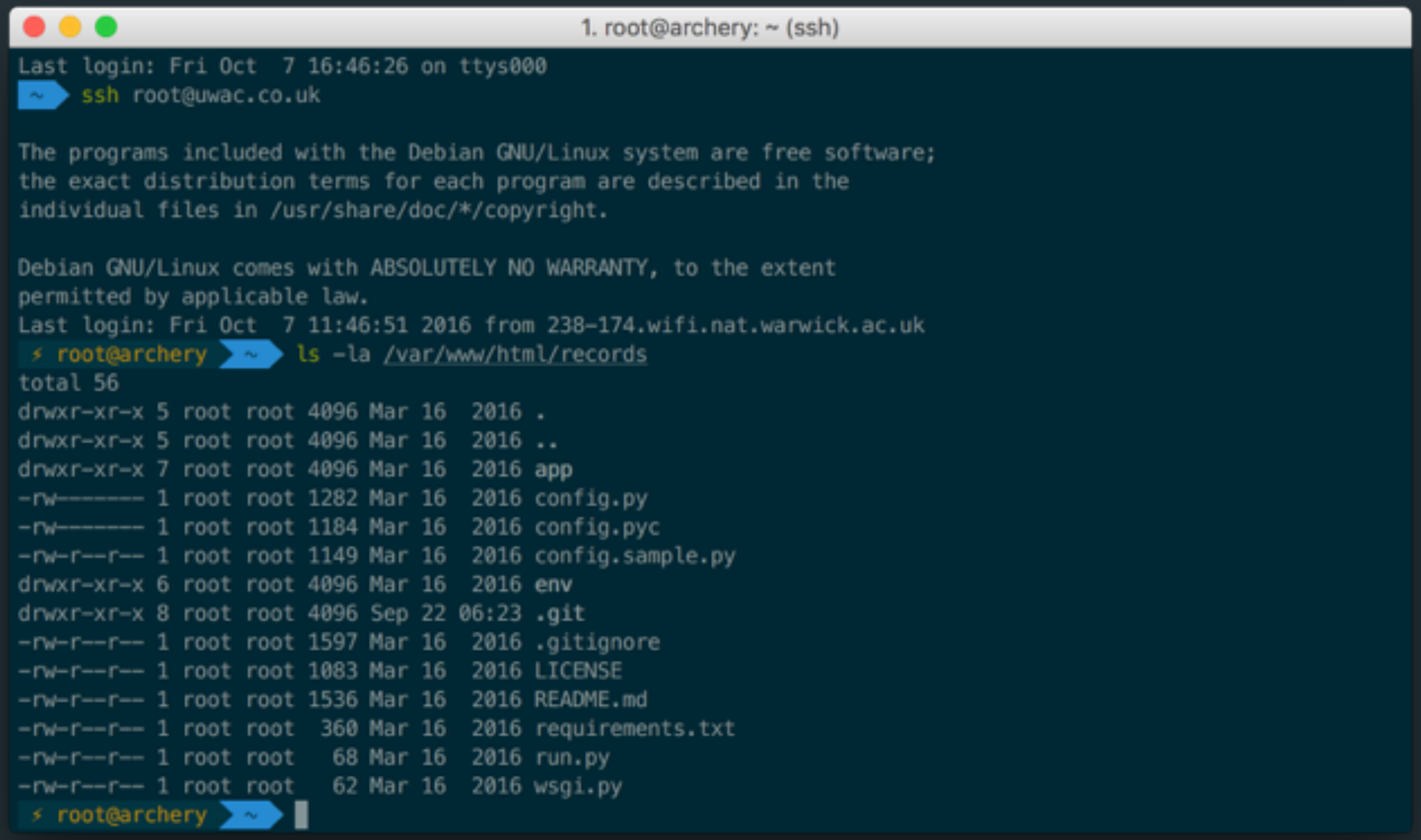
Introduction to Linux

- DCS machines have various extra software installed already:
  - Browsers: Chrome, Firefox
  - Text editors: Kate, Gedit, Vim, Emacs
  - PDF viewers: Okular, Evince
- Other notable software is on the cheat sheet

The Shell

The Shell

- A convenient way to interact with the underlying Linux OS
- Can be used to launch programs or explore the file system
- Different flavours exist but they all do the same thing

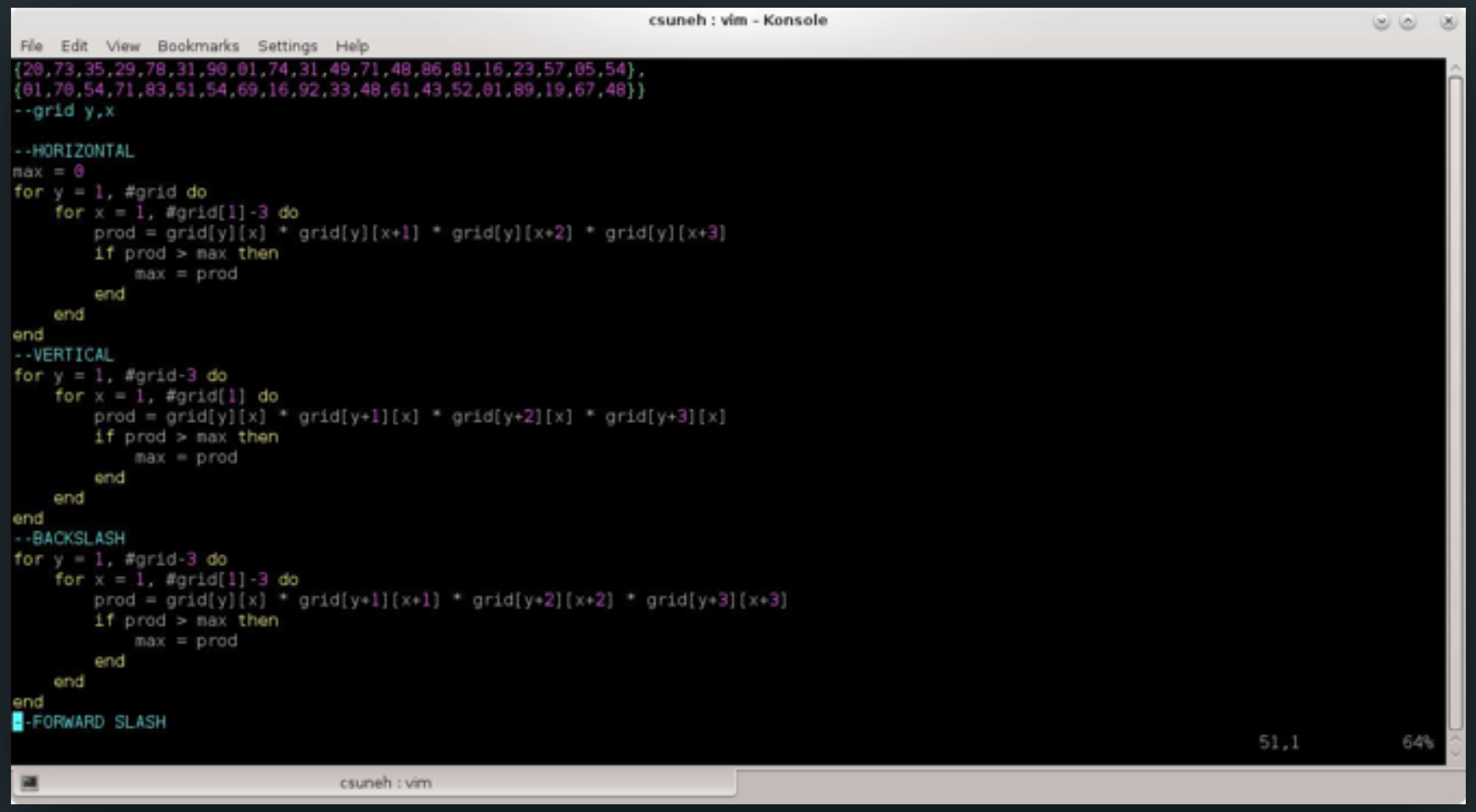


1. root@archery: ~ (ssh)

```
Last login: Fri Oct  7 16:46:26 on ttys000
 ~  ssh root@uwac.co.uk

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct  7 11:46:51 2016 from 238-174.wifi.nat.warwick.ac.uk
 ⚡ root@archery  ~  ls -la /var/www/html/records
total 56
drwxr-xr-x 5 root root 4096 Mar 16  2016 .
drwxr-xr-x 5 root root 4096 Mar 16  2016 ..
drwxr-xr-x 7 root root 4096 Mar 16  2016 app
-rw------- 1 root root 1282 Mar 16  2016 config.py
-rw------- 1 root root 1184 Mar 16  2016 config.pyc
-rw-r--r-- 1 root root 1149 Mar 16  2016 config.sample.py
drwxr-xr-x 6 root root 4096 Mar 16  2016 env
drwxr-xr-x 8 root root 4096 Sep 22 06:23 .git
-rw-r--r-- 1 root root 1597 Mar 16  2016 .gitignore
-rw-r--r-- 1 root root 1083 Mar 16  2016 LICENSE
-rw-r--r-- 1 root root 1536 Mar 16  2016 README.md
-rw-r--r-- 1 root root  360 Mar 16  2016 requirements.txt
-rw-r--r-- 1 root root   68 Mar 16  2016 run.py
-rw-r--r-- 1 root root   62 Mar 16  2016 wsgi.py
 ⚡ root@archery  ~  
```



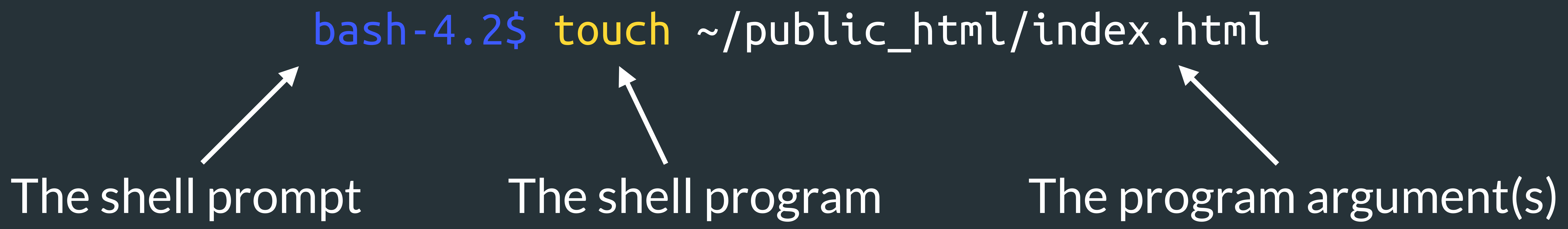
csuneh : vim - Konsole

```
{20,73,35,29,78,31,90,01,74,31,49,71,48,86,81,16,23,57,05,54},
{01,70,54,71,83,51,54,69,16,92,33,48,61,43,52,01,89,19,67,48}}
--grid y,x

--HORIZONTAL
max = 0
for y = 1, #grid do
    for x = 1, #grid[1]-3 do
        prod = grid[y][x] * grid[y][x+1] * grid[y][x+2] * grid[y][x+3]
        if prod > max then
            max = prod
        end
    end
end
--VERTICAL
for y = 1, #grid-3 do
    for x = 1, #grid[1] do
        prod = grid[y][x] * grid[y+1][x] * grid[y+2][x] * grid[y+3][x]
        if prod > max then
            max = prod
        end
    end
end
--BACKSLASH
for y = 1, #grid-3 do
    for x = 1, #grid[1]-3 do
        prod = grid[y][x] * grid[y+1][x+1] * grid[y+2][x+2] * grid[y+3][x+3]
        if prod > max then
            max = prod
        end
    end
end
--FORWARD SLASH
                                                                51,1          64%
```

The Shell

- There a bunch of programs accessible from the shell by default
  - `cd, ls, rm, mkdir, touch,` …
- Functionality includes creating, reading, updating and deleting files
- A typical line in the shell may look like:



```
bash-4.2$ touch ~/public_html/index.html
```

The shell prompt

The shell program

The program argument(s)

Directories and Paths

- When you load up the shell you are placed in your home directory
- To move around you use the `cd` command
- You give `cd` a path to the directory you want to move to
- Paths are **case sensitive** and shouldn't contain spaces
- Paths can be **relative** to the current directory or **absolute**

`../../foobar/baz`

Relative path

`/var/www/html/`

Absolute path

Directories and Paths

- There are some special relative path values in Linux
  - . (dot) represents the current directory
  - . . (dot dot) represents the parent directory
  - ~ (tilde) represents your home directory
- Example:

```
bash-4.2$ cd ~/foobar/Baz/../../blat
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |   |--min
   |   \--dev
   |
   |--perl
   \--ogg
       \--mp3
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |   |--min
   |   \--dev
   |
   |--perl
   \--ogg
       \--mp3  ←  Starting here
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |    |--min
   |    \--dev
   |
   |--perl
   \--ogg
        \--mp3  ←—— Starting here
```

```
bash-4.2$ cd ./
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |   |--min
   |   \--dev
   |
   |--perl
   \--ogg
       \--mp3  ←  You're now here
```

```
bash-4.2$ cd ./
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |   |--min
   |   \--dev
   |
   |--perl
   \--ogg
       \--mp3  ←  You’re now here
```

```
bash-4.2$ cd ../
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |    |--min
   |    \--dev
   |
   |--perl
   \--ogg ←——— You’re now here
        \--mp3
```

```
bash-4.2$ cd ../
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |      |--min
   |      \--dev
   |
   |--perl
   \--ogg  ←——  You’re now here
       \--mp3
```

```
bash-4.2$ cd css/min/
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |    |--min
   |    \--dev
   |
   |--perl
   \--ogg  ←  You’re still here
        \--mp3
```

```
bash-4.2$ cd css/min/

   no such file or directory: css/min/
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |   |--min
   |   \--dev
   |
   |--perl
   \--ogg  ←  You’re now here
       \--mp3
```

```
bash-4.2$ cd ../css/min/
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |    |--min
   |    \--dev
   |
   |--perl
   \--ogg
        \--mp3
```

You’re now here

```
bash-4.2$ cd ../css/min/
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |     |--min  ←——— You’re now here
   |     \--dev
   |
   |--perl  ←——— Get to here using tilde
   \--ogg
         \--mp3
```

```
bash-4.2$ cd ?
```

Relative Path Examples

```
~/public_html
   |--js
   |--img
   |--css
   |   |--min  ←  You’re now here
   |   \--dev
   |
   |--perl  ←  Get to here using tilde
   \--ogg
       \--mp3
```

```
bash-4.2$ cd ~/public_html/perl/
```

Relative Path Examples

```
~/public_html
    |--js
    |--img
    |--css
    |    |--min
    |    \--dev
    |
    |--perl  ←  You’re now here
    \--ogg
         \--mp3
```

```
bash-4.2$ cd ~/public_html/perl/
```

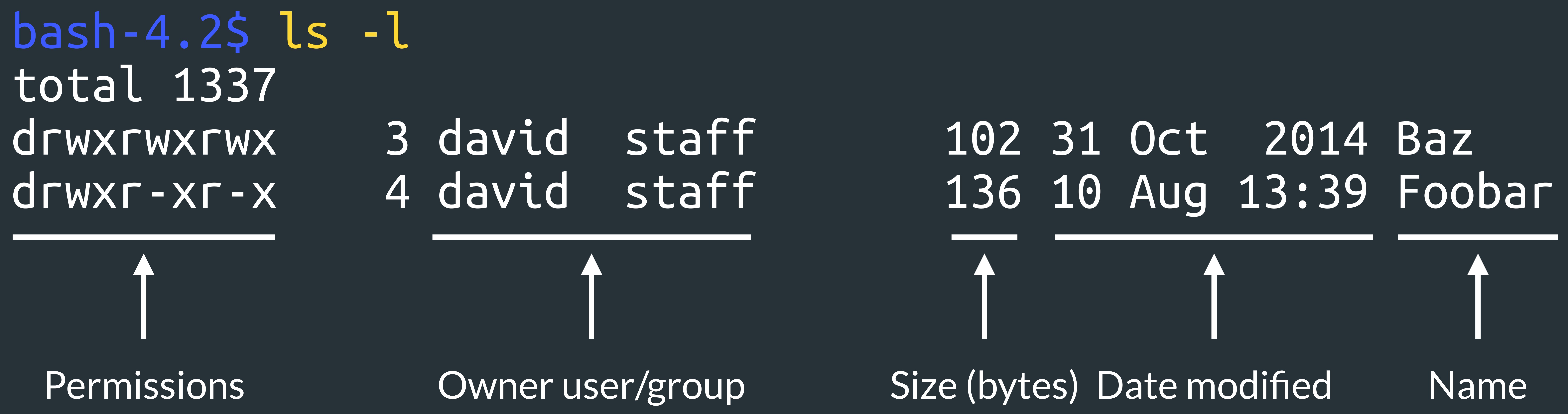
Listing Files and Folders

- Listing files and folders in the shell is straight forward
- Using the `ls` command lists files and folders in the current directory
- You can also give `ls` a path for a specific directory to list
- Output of `ls` can be formatted using common options
- Example:

```
bash-4.2$ ls ~/
Adlm Qt jagex_cl_runescape_LIVE.dat
AndroidStudioProjects Samsung …
```

Listing Files and Folders

- `ls -l` formats the output into a detailed list



```
bash-4.2$ ls -l
total 1337
drwxrwxrwx    3 david  staff        102 31 Oct  2014 Baz
drwxr-xr-x    4 david  staff        136 10 Aug 13:39 Foobar
```

Permissions

Owner user/group

Size (bytes)

Date modified

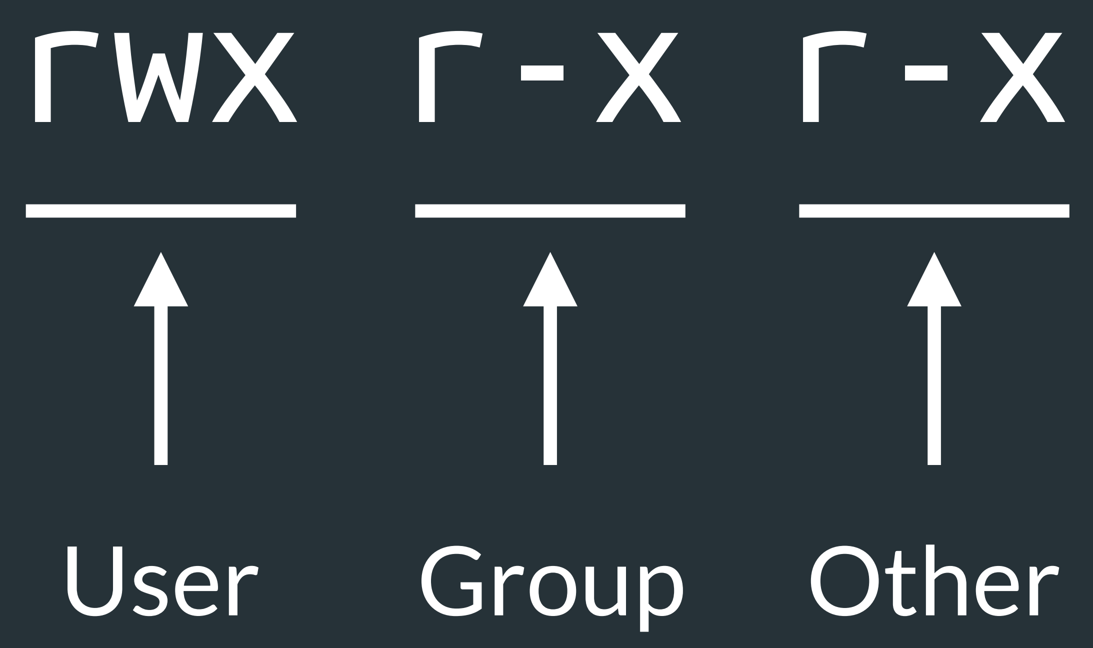
Name

- `ls -a` includes hidden files/folders (beginning with a dot)

- Options can be combined together (`ls -la`) — see the cheat sheet

Permissions

- Files and folders in Linux are owned by a user and a user group
- Users not in these categories are classed as ‘other’



rwx r-x r-x

User Group Other

Permission	Char	Meaning
Read	r	Can view the contents
Write	w	Can change the contents
Execute	x	Can run as a program

Making Files and Folders

- Files and folders are created using `touch` and `mkdir` respectively
- Write permissions are needed in the parent folder to create items
- `touch <file path>` creates a file
- `mkdir <directory path>` creates a directory
- Examples:

```
bash-4.2$ touch foo.bar
```

```
bash-4.2$ mkdir -p ~/baz/blat/
```

Removing Files and Folders

- `rm <path>` remove files on the given path
- Write permissions are needed to remove files and folders
- Using `rm` is typically irreversible — there is no trash can
- `rm -r` recursively removes files **and** folders
- Example:

```
bash-4.2$ rm -f ~/public_html/dank_memes.html
```

Command Manuals/Help

- `man <command>` or `<command> --help` displays help menus
- These menus have in-depth information about commands
  - Optional arguments, command operation, history, etc
- A lot of the information in these slides can be found in `man` pages
- Example:

```
bash-4.2$ man man
```

Handy Shortcuts

- Clearing the shell can be done with `clear` or pressing `Ctrl+L`
- Exiting a shell program is done using the keys `Ctrl+C`
- Where `Ctrl+C` doesn’t work, try use `Ctrl+D`
- Failing `Ctrl+D`, pressing ‘q’ or typing ‘:q’ (colon q) might work
- A last resort may involve closing the terminal window

A Practical Example

A Practical Example

- Let’s create a Java program in this folder:
  - `~/uwcs/shelltalk/java/`
- This’ll demonstrate various previously seen programs
- Over to the shell!

Questions?

Pizza? :D