

Sesión 0: Presentación

Programación para Sistemas


Ángel Herranz

2019-2020

Universidad Politécnica de Madrid

¿Quién?

Ángel Herranz
D-2309

 aherranz@fi.upm.es

 <https://github.com/aherranz/pps>

 herranz: command not found

Tutoría fija: M y J 09:00 - 10:00

Tutoría cita previa: M 10:00 - 12:00

¿Qué?

Grado en Ingeniería Informática
32SM-B

Aula 6205
(10/09 ¡sólo hoy!)

Sala Los Verdes
(excepto anteriores y 9/12 y 16/12)

Sala Nerja
(9/12 y 16/12)

¿Por qué?

- Necesario para **Sistemas Operativos** y otras asignaturas
- Más **control** sobre la máquina
- Desarrolladores más **preparados**

Antes de continuar...

revisad la **guía docente** en Moodle UPM



[https://moodle.upm.es/titulaciones/oficiales/
course/view.php?id=6668](https://moodle.upm.es/titulaciones/oficiales/course/view.php?id=6668)

Resumen

C	Bash
$\frac{2}{3}$	$\frac{1}{3}$
10 semanas	5 semanas
1 práctica común ¹	
3 prácticas	1 práctica ²
1 examen escrito	
teoría + práctica	

¹Toma de contacto con el sistema de entrega: triqui.

²Las prácticas se guardan, consulta en Moodle.

Resumen

C	Bash
$\frac{2}{3}$	$\frac{1}{3}$
10 semanas	5 semanas
1 práctica común ¹	
3 prácticas	1 práctica ²
1 examen escrito	
teoría + práctica ⚠	

¹Toma de contacto con el sistema de entrega: triqui.

²Las prácticas se guardan, consulta en Moodle.

¿Quieres ser desarrollador?

¿Por qué usas Windows?

³Agosto 2019, y creciendo

¿Quieres ser desarrollador?

¿Por qué usas Windows?

Unix is used by 70.6 % of all the websites whose operating system we know.³

Web Technology Survey

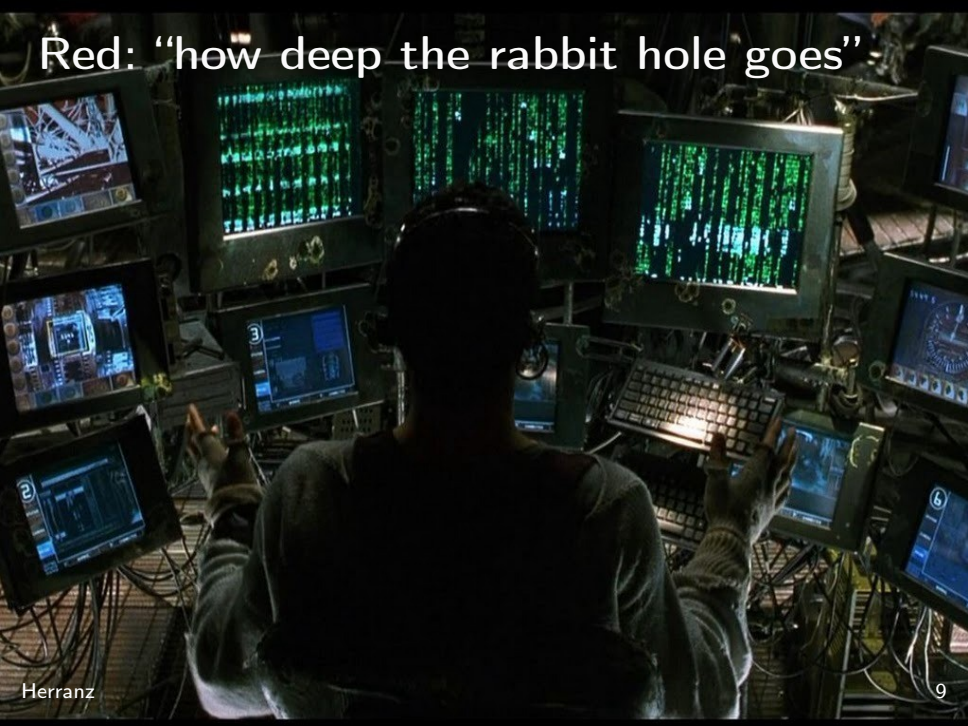
³Agosto 2019, y creciendo

Unix: “Do you want to know what is it?”



Blue: “belive whatever you want”

Red: "how deep the rabbit hole goes"



Mis recomendaciones

- Abrazad toda la tecnología:
Unix, Bash, make, gcc, gdb, etc.
- Instalad una distro de Linux: Ubuntu 18.04
- Usad Bash cada día
- Usad un buen editor de texto:
emacs⁴, vim, vscode, atom, sublime,
nano

⁴Meses de uso intenso para dominarlo, pero compensa.

Bash *crash course*

En Unix...

- **Ficheros:** *todo* son ficheros en Unix
- **Procesos:** programas en ejecución

En Unix...

- **Ficheros:** *todo* son ficheros en Unix
Directorios (carpetas), ficheros de texto, ficheros binarios, **teclado**, **pantalla**, ratón, etc.
- **Procesos:** programas en ejecución

En Unix...

- **Ficheros**: *todo* son ficheros en Unix
Directorios (carpetas), ficheros de texto, ficheros binarios, **teclado**, **pantalla**, ratón, etc.
- **Procesos**: programas en ejecución
Puesta en marcha, **argumentos** de la puesta en marcha, parada, **estado** de la parada, ...

Bash

- *Cáscara* que nos permite tener control sobre nuestro sistema operativo:

Ficheros y procesos

- Bash es un programa interactivo
- Lo ponemos en marcha desde el UI:
 - Lanzador de aplicaciones (term)
 - Ctrl-Alt-T (Ubuntu por defecto)
 - Cmd-Enter (i3)
 - etc.
- *Talk is easy, ...* (L. Torvalds)

En marcha



```
$ |
```

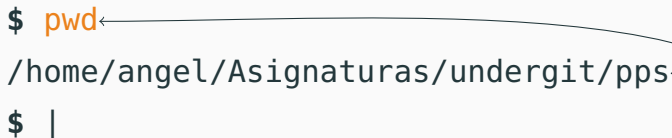
- **Prompt**: Bash está esperando órdenes
- **Cursor**: Bash es **interactivo**
- Bash es configurable, por ejemplo con un *prompt* **más informativo**



```
angel@T440p: /~ $ |
```

Mi primer mandato en Bash

```
$ pwd  
/home/angel/Asignaturas/undergit/pps  
$ |
```



- “Bash, busca y pon en marcha el mandato `pwd`⁵”
- Bash contesta escribiendo en la salida estándar el **directorio de trabajo actual**⁶ y espera un **nuevo mandato**

⁵*print working directory*

⁶Nombre absoluto

El segundo

```
$ ls
00-presenta 01-contacto_C ahbeamer.sty
$ ls -a -l
total 1068
drwxrwxr-x 5 angel angel 4096 sep 11 00:50 .
drwxrwxr-x 5 angel angel 4096 ago  5 09:24 ..
drwxrwxr-x 2 angel angel 4096 sep 11 01:36 00-presenta
drwxrwxr-x 3 angel angel 4096 sep 10 09:48 01-contacto_C
-rw-rw-r-- 1 angel angel  303 sep 11 00:50 ahbeamer.sty
```

- “Bash, busca y pon en marcha el mandato `ls`”
- “Bash, busca y pon en marcha el mandato `ls` con argumentos `-a` y `-l`”

¿Qué hace ls?

```
$ man ls
```

```
LS(1)
```

```
User Commands
```

```
LS(1)
```

```
NAME
```

```
ls - list directory contents
```

```
SYNOPSIS
```

```
ls [OPTION]... [FILE]...
```

```
DESCRIPTION
```

```
List information about the FILES (the current  
directory by default). Sort entries alphabetically  
if none of -cftuvSUX nor --sort is specified.
```

```
[...]
```

Ejemplos de mandatos

Editor emacs

\$ emacs

Editor vim

\$ vi

Editor nano

\$ nano

Nos lleva al directorio DIR

\$ cd DIR

Nos lleva de vuelta

al directorio home

\$ cd

Idem

\$ cd ~

Busca un fichero con

TEXT0 en el nombre

\$ locate TEXT0

Busca TEXT0 en el

fichero FICHERO

\$ grep TEXT0 FICHERO

Indica la versión

del compilador de C

\$ gcc --version

Se conecta a triqui2

como aherranz

\$ ssh aherranz@triqui2.fi.upm.es

Usad vuestro usuario

Y sobre todo...

Abre el manual del programa PROGRAMA

\$ man PROGRAMA

Una vez en el manual:

- 'q' para salir

- '/' para buscar texto (como en vi)

Ctrl-C para terminar cualquier programa

Ctrl-D (fin de fichero) para cerrar Bash

Recursos para aprender

- ★ <https://guide.bash.academy/>
- ★ <https://devhints.io/bash>
- <https://learnxinyminutes.com/docs/bash/>
- <https://tldr.sh/>
- <https://github.com/LeCoupa/awesome-cheatsheets/blob/master/languages/bash.sh>
- <https://www.git-tower.com/learn/cheat-sheets/cli>

Mandatos: referencia rápida i

cat	Lists a file or files sequentially.
cd	Change directories.
chmod ugo+rw	Set read, write and execute permissions for user, group and others.
chmod a-rwx	Remove read, write and execute permissions from all.
chmod 755	Set user write and universal read-execute permissions
chmod 644	set user write and universal read permissions.
cp	Copy files.
cp -r	Copy files and directories recursively.
expr 2 + 2	Add 2 + 2.
fgrep	Search for string match.

Mandatos: referencia rápida ii

<code>grep</code>	Search for string pattern matches.
<code>grep -v</code>	Search for no match.
<code>grep -n</code>	List line numbers of matches.
<code>grep -i</code>	Ignore case.
<code>grep -l</code>	Only list file names for a match.
<code>head -n5 source.txt</code>	List first 5 lines.
<code>less</code>	View a text file one screen at a time; can scroll both ways.
<code>ll</code>	Give a listing of files with file details.
<code>ls</code>	Give a simple listing of files.
<code>mkdir</code>	Make a directory.
<code>more</code>	Displays a file a screenfull at a time.
<code>mv</code>	Move or rename files.
<code>paste f1 f2</code>	Paste files by columns.

Mandatos: referencia rápida iii

<code>pg</code>	Variant on "more".
<code>pwd</code>	Print working directory.
<code>rm</code>	Remove files.
<code>rm -r</code>	Remove entire directory subtree.
<code>rmdir</code>	Remove an empty directory.
<code>sed 's/txt/TXT/g'</code>	Scan and replace text.
<code>sed 's/txt/d'</code>	Scan and delete text.
<code>sed '/txt/q'</code>	Scan and then quit.
<code>sort</code>	Sort input.
<code>sort +1</code>	Skip first field in sorting.
<code>sort -n</code>	Sort numbers.
<code>sort -r</code>	Sort in reverse order.
<code>sort -u</code>	Eliminate redundant lines in output.
<code>tail -5 source.txt</code>	List last 5 lines.

Mandatos: referencia rápida iv

<code>tail +5 source.txt</code>	List all lines after line 5.
<code>tr '[A-Z]' '[a-z]'</code>	Translate to lowercase.
<code>tr '[a-z]' '[A-Z]'</code>	Translate to uppercase.
<code>tr -d '_'</code>	Delete underscores.
<code>uniq</code>	Find unique lines.
<code>wc</code>	Word count (characters, words, lines).
<code>wc -c</code>	Characters count only.
<code>wc -w</code>	Word count only.
<code>wc -l</code>	Line count only.

<https://wikis.nyu.edu/display/ADRC/Bash+commands+quick+reference>

Agradecimientos

- Sergio Valverde, se ha revisado el material, incluso ha hecho algún cambio a las mismas
- Varios estudiantes que han detectado y comunicado typos y errores
- No dejéis de enviarme un correo si veis algún error o teneis alguna sugerencia