

Sesión 11: Contacto Bash

Programación para Sistemas

Ángel Herranz

2019-2020

Universidad Politécnica de Madrid

Sesión 0: *Bash Crash Course*

Sesión 0: *Bash Crash Course*

★ <https://guide.bash.academy/>
¿Quién hizo esta guía?

En el capítulo de hoy...

- Sesión de programación en C para empezar a tocar...
 - **Conceptos** de sistemas operativos:

Ficheros, programas, procesos, variables de entorno, entrada/salida, invocación, estado de terminación, etc.
 - El juego de la **expansión**

En el capítulo de hoy...

- Sesión de programación en C para empezar a tocar...

- Conceptos de sistemas operativos:

Ficheros, programas, procesos, variables de entorno, entrada/salida, invocación, estado de terminación, etc.

- El juego de la expansión

 Cuaderno de UNIX en github

 <https://devhints.io/bash>

Conceptos

- Antes de *comenzar* con Bash necesitamos tener claros algunos **conceptos relacionados con el sistema operativo**
- **Ficheros, programas, procesos, variables de entorno, entrada/salida, invocación, estado de terminación, etc.**
- Lo vemos una tarea de C de **cursos pasados**: delreves¹

 **Aún no es una práctica de este año**

- delreves invierte cada línea
- ¡*Nos vamos a triqui3.fi.upm.es!*

¹Sólo hacemos el esqueleto ;).

delreves (a triqui3.fi.upm.es)

```
$ whoami
```

```
angel
```

```
$ hostname
```

```
T440
```

```
$ ssh aherranz@triqui3.fi.upm.es
```

```
Last login: Tue Nov 20 09:31:28 2018 from sri33.sri.fi.upm.es
```

```
$ whoami
```

```
aherranz
```

```
$ hostname
```

```
triqui3.fi.upm.es
```

Línea de comandos (*command line*), invocación, programa, proceso, salida estándar

delreves (~pps/pub/tarea-2.1.2018-2019.tar.gz)

❓ ¿Qué es ~pps/pub/tarea-2.1.2018-2019.tar.gz?

delreves (~pps/pub/tarea-2.1.2018-2019.tar.gz)

❓ ¿Qué es ~pps/pub/tarea-2.1.2018-2019.tar.gz?

Un nombre de un fichero

- Vamos a diseccionarlo

delreves (~pps/pub/tarea-2.1.2018-2019.tar.gz)

❓ ¿Qué es ~pps/pub/tarea-2.1.2018-2019.tar.gz?

Un nombre de un fichero

- Vamos a diseccionarlo

❓ ¿Qué es ~pps?

*Un nombre de un directorio, en concreto el directorio
home del usuario pps*

delreves (~pps/pub/tarea-2.1.2018-2019.tar.gz)

❓ ¿Qué es `~pps/pub/tarea-2.1.2018-2019.tar.gz`?

Un nombre de un fichero

- Vamos a diseccionarlo

❓ ¿Qué es `~pps`?

*Un nombre de un directorio, en concreto el directorio
home del usuario `pps`*

 **echo** `~pps`

delreves (~pps/pub/tarea-2.1.2018-2019.tar.gz)

❓ ¿Qué es `~pps/pub/tarea-2.1.2018-2019.tar.gz`?

Un nombre de un fichero

- Vamos a diseccionarlo

❓ ¿Qué es `~pps`?

*Un nombre de un directorio, en concreto el directorio
home del usuario pps*

💻 **echo** `~pps`

💬 ¿Por qué `/homefi/dep/pps`? ¿Qué es?

E*l nombre (absoluto) del directorio home del usuario
pps*

💬 ¿Qué es `~aherranz`? ¿Y `~y160215`? ¿Y `~`?

delreves (copiar a ~)

- Vamos a copiarnos ese fichero a **nuestra home**.

 A casa: ¿cómo?

delreves (copiar a ~)

- Vamos a copiarnos ese fichero a **nuestra home**.

 A casa: ¿cómo?

```
$ cd
```

```
$ pwd
```

```
/home/fi/personal/dlsiis/aherranz
```

delreves (copiar a ~)


- Vamos a copiarnos ese fichero a **nuestra home**.

 A casa: ¿cómo?

```
$ cd
```


```
$ pwd
```

```
/home/fi/personal/dlsiis/aherranz
```


 Haz un `ls -al` y entiende lo que dice

delreves (copiar a ~)

- Vamos a copiarnos ese fichero a **nuestra home**.

 A casa: ¿cómo?

```
$ cd  
$ pwd  
/home/fi/personal/dlsiis/aherranz
```

 Haz un `ls -al` y entiende lo que dice

 `cp_~pps/pub/tarea-2.1.2018-2019.tar.gz_.`


- Vamos a diseccionarlo (espacios resaltados con `..._...`)

delreves (copiar a ~)

- Vamos a copiarnos ese fichero a **nuestra home**.

 A casa: ¿cómo?

```
$ cd  
$ pwd  
/home/fi/personal/dlsiis/aherranz
```

 Haz un `ls -al` y entiende lo que dice

 `cp_~pps/pub/tarea-2.1.2018-2019.tar.gz_.`

- Vamos a diseccionarlo (espacios resaltados con `..._...`)
- ¿Qué es `cp`? ¿Y `.`?
- Haz un `ls -al` ahora

delreves (descomprimir en ~)

- Deberías **estar en tu home** (**cd** y **ls** si no estás seguro) y **no tener un directorio pps** (puedes renombrarlo)

delreves (descomprimir en ~)

- Deberías estar en tu home (`cd` y `ls` si no estás seguro) y no tener un directorio `pps` (puedes renombrarlo)
- ❓ `tar_xvfz_tarea-2.1.2018-2019.tar.gz`
- ¿Qué es `tar`? ¿Y `xvfz` y `tarea-2.1.2018-2019.tar.gz`?

delreves (descomprimir en ~)

- Deberías estar en tu home (`cd` y `ls` si no estás seguro) y no tener un directorio `pps` (puedes renombrarlo)

❓ `tar_xvfz_tarea-2.1.2018-2019.tar.gz`

- ¿Qué es `tar`? ¿Y `xvfz` y `tarea-2.1.2018-2019.tar.gz`?

```
$ tar xvfz tarea-2.1.2018-2019.tar.gz
pps/tarea-2.1.2018-2019/
pps/tarea-2.1.2018-2019/auxiliar.c
pps/tarea-2.1.2018-2019/autores.txt
pps/tarea-2.1.2018-2019/bitacora.txt
pps/tarea-2.1.2018-2019/auxiliar.h
pps/tarea-2.1.2018-2019/delreves.c
pps/tarea-2.1.2018-2019/ejemplo.c
pps/tarea-2.1.2018-2019/Makefile
pps/tarea-2.1.2018-2019/pruebas.sh
```

delreves (¡a programar!)

- Vamos hasta el directorio `~/pps/tarea-2.1.2018-2019`

 ¿Diferencias?

`~pps/pub/tarea-2.1.2018-2019.tar.gz`

vs.

`~/pps/tarea-2.1.2018-2019`

delreves (¡a programar!)

- Vamos hasta el directorio `~/pps/tarea-2.1.2018-2019`

💬 ¿Diferencias?

`~pps/pub/tarea-2.1.2018-2019.tar.gz`

vs.

`~/pps/tarea-2.1.2018-2019`

- Haz un `ls`
- ¡Tenemos un Makefile!

💻 **cat** Makefile

💻 **make**

- ¡No compila porque `delreves.c` está vacío!

delreves.c i

- Un programa que no hace nada, salvo **terminar bien**:

```
int main() {  
    return 0;  
}
```

 make + ejecución:

```
$ make  
Autores:  
[...]  
gcc -Wall -g delreves.c -o delreves auxiliar.o -lm  
$ ./delreves  
$ |
```


delreves.c i

- Un programa que no hace nada, salvo **terminar bien**:

```
int main() {  
    return 0;  
}
```

 make + ejecución:

```
$ make  
Autores:  
[...]  
gcc -Wall -g delreves.c -o delreves auxiliar.o -lm  
$ ./delreves  
$ |
```

 echo \$?

delreves.c ii

- Un programa que da **error** cuando el número de **parámetros no es el apropiado**:

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    if (argc > 2) {
        fprintf(stderr, "ERROR\n");
        return -1;
    }
    return 0;
}
```

 Prueba estas invocaciones (**echo** \$?):

```
$ ./delreves
$ ./delreves -h
$ ./delreves a b
```

delreves.c iii

- Un programa que da un **mensaje de ayuda** cuando se **invoca con -h** (y termina bien):

```
#include <string.h>
...
}
if (argc == 2 && strcmp(argv[1], "-h") == 0) {
    printf("Uso: %s [FICHERO]\n", argv[0]);
    return 0;
}
return 0;
}
```



Prueba las mismas invocaciones:

```
$ ./delreves
$ ./delreves -h
$ ./delreves a b
```

- Un programa que abre el fichero adecuado² y da un error si no se pudo abrir:


```
...
int main(int argc, char *argv[]) {
    FILE *fichero;
    ...
}
if (argc == 2)
    fichero = fopen(argv[1], "r");
else
    fichero = stdin;
if (fichero == NULL) {
    fprintf(stderr, "El fichero no se pudo abrir\n");
    return -1;
}
return 0;
}
```

²man fopen

 Prueba las invocaciones:

```
$ ./delreves  
$ ./delreves -h  
$ ./delreves a b  
$ ./delreves supercalifrasgilisticoepialidoso  
$ ./delreves delreves.c
```

- Un programa que lee líneas y las escribe.

 `man stdin`, `man fopen` y `man fgets`

- Un programa que lee líneas y las escribe.



man stdin, man fopen y man fgets

```
extern FILE *stdin;
```

```
FILE *fopen(const char *pathname, const char *mode);
```

```
char *fgets(char *s, int size, FILE *stream);
```

- Recordemos

```
FILE *fichero;
```

```
...
```

```
if (argc == 2)
```

```
    fichero = fopen(argv[1], "r");
```

```
else
```

```
    fichero = stdin;
```

 Un programa que lee líneas y las escribe.

 Un programa que lee líneas y las escribe.

```
char linea[2049];  
...  
while (fgets(linea, 2048, fichero) != NULL) {  
    printf("%s", linea);  
}
```

delreves.c vi

- Aprendemos a usar la biblioteca auxiliar.h:

```
...  
#include "auxiliar.h"  
int main(int argc, char *argv[]) {  
    FILE *fichero;  
    /* argv0 es una variable global de auxiliar */  
    argv0 = argv[0];  
    if (argc > 2) {  
        Error(EX_USAGE, "Error de uso");  
    }  
    ...  
    return 0;  
}
```



Prueba ahora:

```
$ ./delreves a b  
./delreves: Error(EX_USAGE), uso incorrecto del mandato. "Success"  
./delreves+ Error de uso  
$ |
```

- 🔍 Explora la biblioteca *auxiliar*: *auxiliar.h* y *auxiliar.c*
- 🔍 Busca *sysexits.h*
- 📄 En Bash: `locate sysexits.h`
- 📄 Modificar *delreves.c* para que el mensaje de error cuando el fichero no se puede abrir sea más adecuado (utilizar la biblioteca *auxiliar*)