

Sesión 12: Scripts

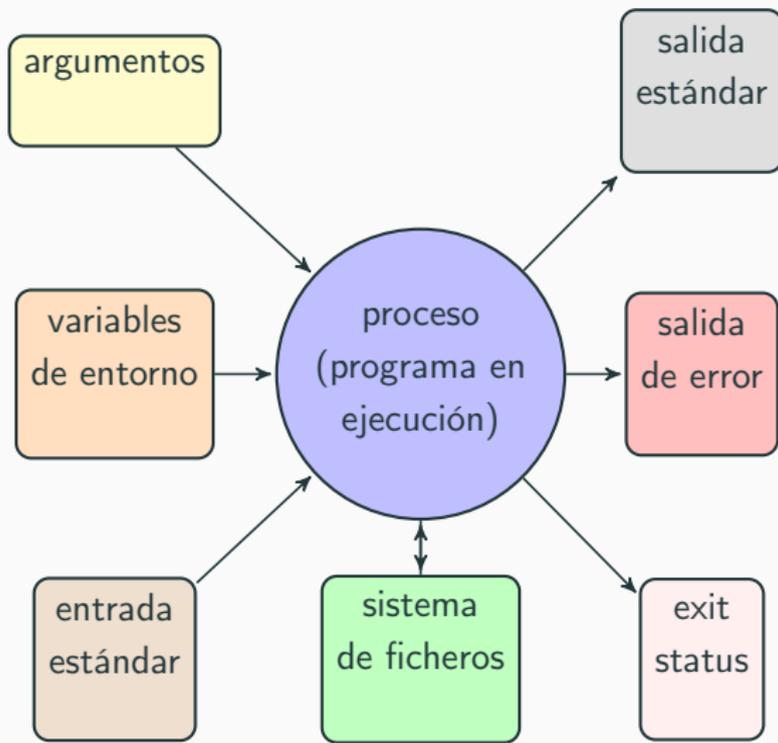
Programación para Sistemas

Ángel Herranz

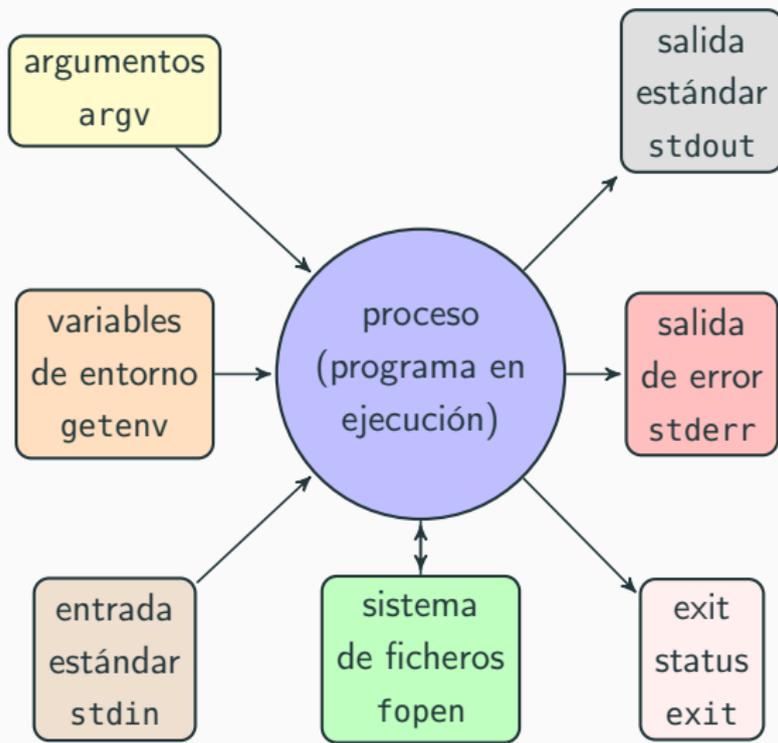
2020-2021

Universidad Politécnica de Madrid

Recordatorio



Recordatorio

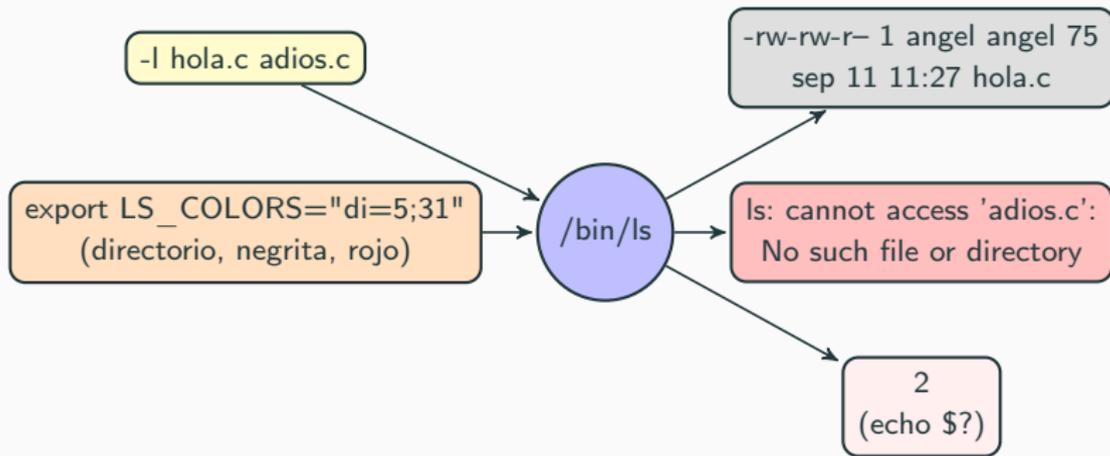


Ejemplo i

```
ls -l hola.c adios.c
```

Ejemplo i

```
ls -l hola.c adios.c
```

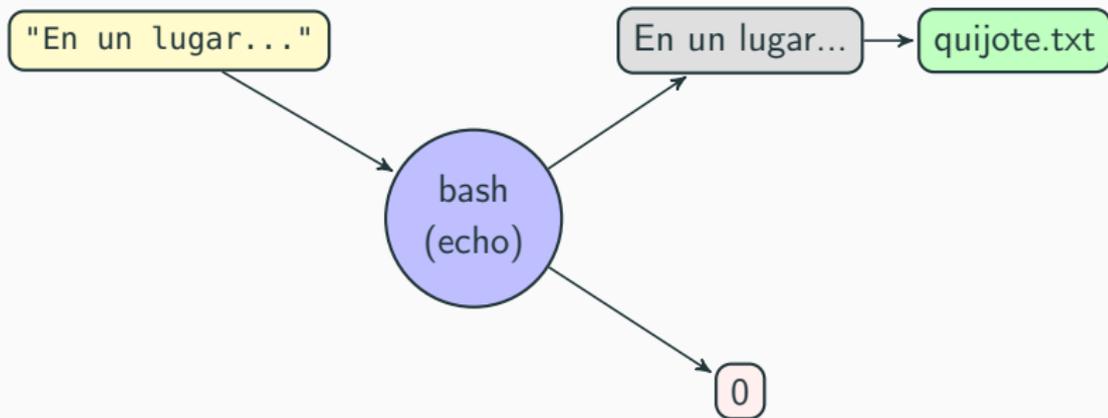


Ejemplo ii

```
echo "En un lugar de la mancha..." > quijote.txt
```

Ejemplo ii

```
echo "En un lugar de la mancha..." > quijote.txt
```

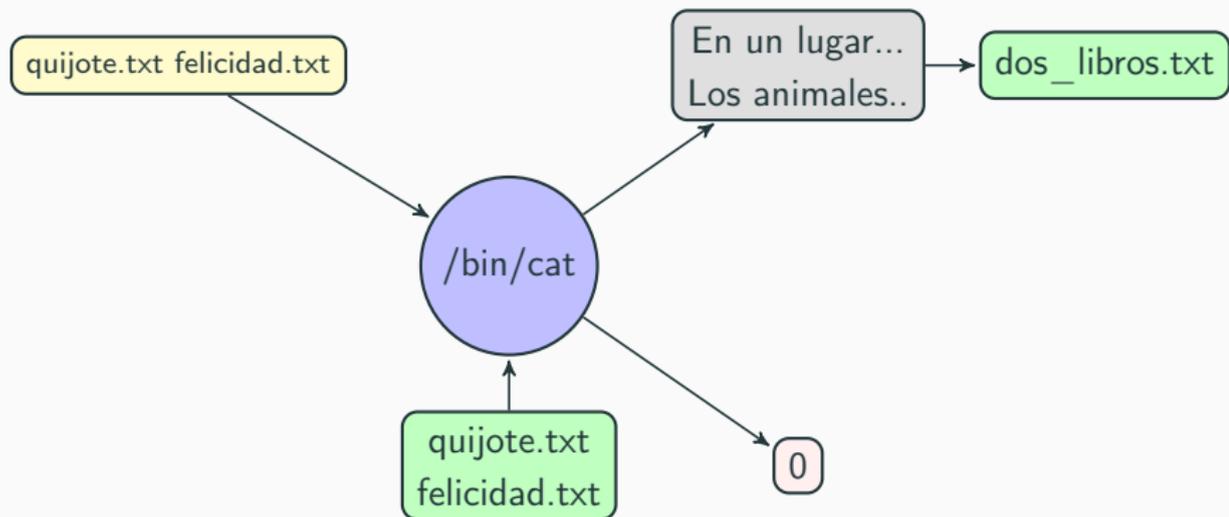


Ejemplo iii

```
cat quijote.txt felicidad.txt > dos_libros.txt
```

Ejemplo iii

```
cat quijote.txt felicidad.txt > dos_libros.txt
```



En el capítulo de hoy. . .

- Combinando programas con *pipes* (`|`): ej. *Crawling*
- *Scripts*

- ¿Crawling?

¹Instalado en triqui, instalable en Ubuntu con `apt-get install curl`

Crawling i

- ¿Crawling?
- Usaremos el programa **cURL**¹: `man curl`

```
$ curl http://www.fi.upm.es
```

```
$ curl http://www.fi.upm.es | grep href
```

¹Instalado en triqui, instalable en Ubuntu con `apt-get install curl`

Crawling i

- ¿Crawling?
- Usaremos el programa **cURL**¹: `man curl`

```
$ curl http://www.fi.upm.es
```

```
$ curl http://www.fi.upm.es | grep href
```

 ¿Qué es grep?

¹Instalado en triqui, instalable en Ubuntu con `apt-get install curl`

Crawling i

- ¿Crawling?
- Usaremos el programa **cURL**¹: `man curl`

```
$ curl http://www.fi.upm.es
```

```
$ curl http://www.fi.upm.es | grep href
```

Q ¿Qué es grep?

- *Crawling*:

```
$ curl -s http://www.fi.upm.es | grep -Po '(?<=href=")[^"]*(?=")'
```

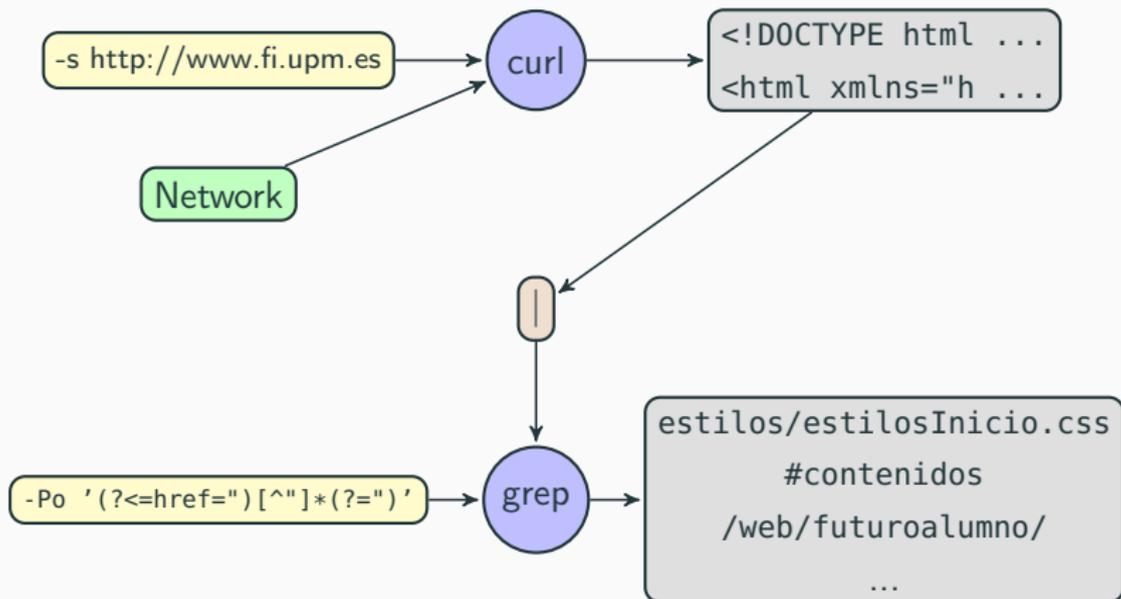
¹Instalado en triqui, instalable en Ubuntu con `apt-get install curl`

Crawling ii

```
curl -s http://www.fi.upm.es | grep -Po '(?<=href=")[^"]*(?=")'
```

Crawling ii

```
curl -s http://www.fi.upm.es | grep -Po '(?<=href=")[^"]*(?=")'
```



- *Almost magic:*

```
$ curl -s http://www.fi.upm.es |  
  grep -Po '(?<=href=")[^"]*(?=")' |  
  sort |  
  uniq
```

🔍 man sort y man uniq

Scripts

Algo más *útil*: MP3 de youtube i

- <https://www.youtube.com/watch?v=ukKQw578Lm8>
- Necesitaremos dos programas
- *youtube-dl* - *download videos from youtube.com or other video platforms*
- *ffmpeg* - *ffmpeg video converter*
- Necesitaremos la versión más actual de *youtube-dl*:

```
$ cd tmp
$ curl -L https://yt-dl.org/downloads/latest/youtube-dl -o youtube-dl
$ chmod +x youtube-dl
$ ls -l youtube-dl
```

Algo más útil: MP3 de youtube ii

- Descargamos el video

```
$ ./youtube-dl https://www.youtube.com/watch?v=ukKQw578Lm8
```

- Eso ha generado el fichero

```
TheLogicalSongporRogerHodgson-Letra.-ukKQw578Lm8.webm
```

- Guardamos el nombre sin extensión en una variable:

```
$ VIDEO="The Logical Song por Roger Hodgson -Letra.-ukKQw578Lm8"
```

- Lo pasamos a MP3

```
$ ffmpeg -i "$VIDEO.webm" -vn -ab 128k -ar 44100 -y "$VIDEO.mp3"
```

¿Por qué tanto entusiasmo?

Se puede
automatizar

¿Por qué tanto entusiasmo?

Se puede
automatizar
en
Scripts

C es un lenguaje compilado

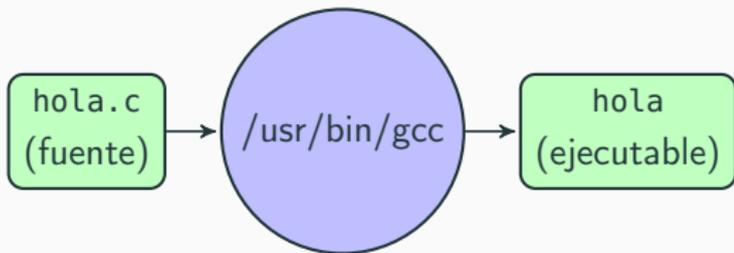
1. Compilación

```
$ gcc -o hola hola.c
```

C es un lenguaje compilado

1. Compilación

```
$ gcc -o hola hola.c
```



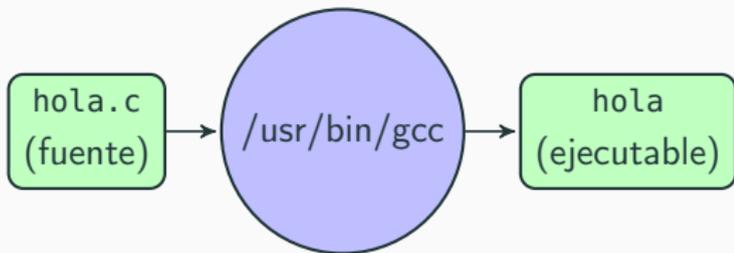
2. Ejecución

```
$ ./hola
```

C es un lenguaje compilado

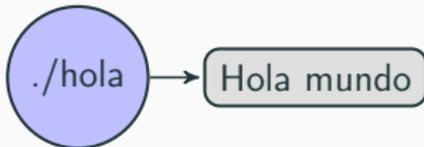
1. Compilación

```
$ gcc -o hola hola.c
```



2. Ejecución

```
$ ./hola
```



Bash es un lenguaje interpretado

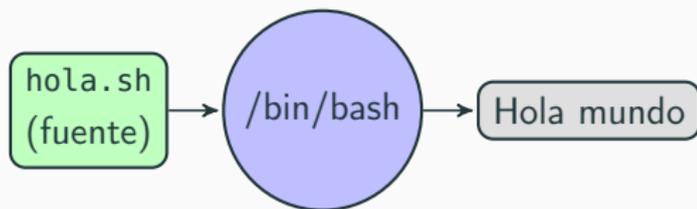
1. Ejecución

```
$ ./hola.sh
```

Bash es un lenguaje interpretado

1. Ejecución

```
$ ./ho!a.sh
```

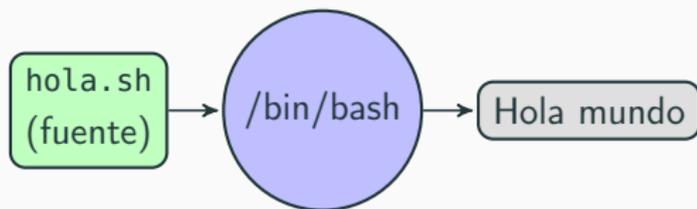


- ¡Y no hay más pasos!
- Vamos a aprender a hacer el ho!a.sh
- Otros lenguajes interpretados:

Bash es un lenguaje interpretado

1. Ejecución

```
$ ./ho!a.sh
```

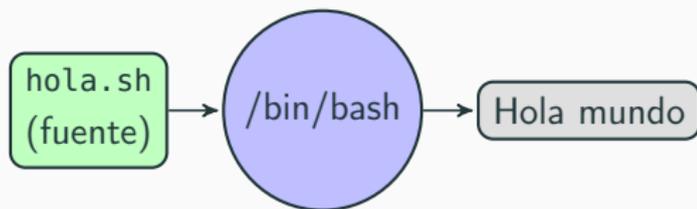


- ¡Y no hay más pasos!
- Vamos a aprender a hacer el ho!a.sh
- Otros lenguajes interpretados: PHP (php),

Bash es un lenguaje interpretado

1. Ejecución

```
$ ./ho!a.sh
```

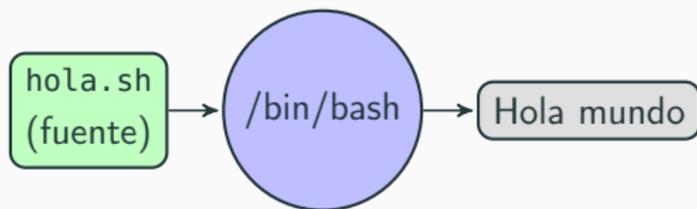


- ¡Y no hay más pasos!
- Vamos a aprender a hacer el ho!a.sh
- Otros lenguajes interpretados: PHP (php), Python (python),

Bash es un lenguaje interpretado

1. Ejecución

```
$ ./ho!a.sh
```



- ¡Y no hay más pasos!
- Vamos a aprender a hacer el ho!a.sh
- Otros lenguajes interpretados: PHP (php), Python (python), Javascript (node), etc.

Tu primer script: `hola.sh`

- 📄 Crea el fichero `hola.sh` con tu editor favorito:

```
#!/bin/bash  
echo Hola mundo
```

- 📄 Ejecuta tu *script*

Tu primer script: hola.sh

- 📄 Crea el fichero `hola.sh` con tu editor favorito:

```
#!/bin/bash  
echo Hola mundo
```

- 📄 Ejecuta tu *script*

```
$ ./hola.sh  
bash: ./hola.sh: Permission denied
```

- 💬 ¿Qué ocurre?

```
$ ls -l hola.sh  
-rw-rw-r-- 1 angel angel 28 dic 11 10:09 hola.sh
```

- 📄 Añadir permisos de ejecución con `chmod` y ejecutar

Convención en UNIX

Cuando se va a ejecutar cualquier fichero de texto,

Convención en UNIX

Cuando se va a ejecutar cualquier fichero de texto, UNIX mira la primera línea y busca #!,

Convención en UNIX

Cuando se va a ejecutar cualquier fichero de texto, UNIX mira la primera línea y busca `#!`, extrae el nombre del programa especificado detrás

Convención en UNIX

Cuando se va a ejecutar cualquier fichero de texto, UNIX mira la primera línea y busca #!, extrae el nombre del programa especificado detrás (en general suele ser un interprete),

Convención en UNIX

Cuando se va a ejecutar cualquier fichero de texto, UNIX mira la primera línea y busca `#!`, extrae el nombre del programa especificado detrás (en general suele ser un interprete), y entonces, coge todo el fichero y se lo pasa a dicho programa por la entrada estándar

Convención en UNIX

Cuando se va a ejecutar cualquier fichero de texto, UNIX mira la primera línea y busca `#!`, extrae el nombre del programa especificado detrás (en general suele ser un interprete), y entonces, coge todo el fichero y se lo pasa a dicho programa por la entrada estándar



¿Qué va a hacer este programa `hola.cat`?

```
#!/bin/cat
```

```
Hola mundo
```

¡Descarga Youtube a MP3! ¡

 Crear un *script* `yourmp3.sh` con la siguiente funcionalidad:

- admita como argumentos **URLS de videos** de Youtube,
- **descargue** los videos (usar `youtube-dl`),
- los **pase a MP3** (usar `ffmpeg`),
- **borre** los videos,
- saque por la salida estándar los **nombres de los MP3**,
- saque por la **salida error** cualquier problema encontrado, y
- de un código de terminación (*exit status*) apropiado

¡Descarga Youtube a MP3! ii

- Como siempre, empezamos por un programa que no hace nada

```
#!/bin/bash  
exit 0
```

¡Descarga Youtube a MP3! ii

- Como siempre, empezamos por un programa que no hace nada

```
#!/bin/bash  
exit 0
```

- Bash es un lenguaje **muy diferente a Java o a C**
- las variables **no se declaran**,
- y se *expanden* con **\$** (ej. `${PATH}`),
- no tiene tipos,
- no tiene módulos o paquetes,

¡Descarga Youtube a MP3! ii

- Como siempre, empezamos por un programa que no hace nada

```
#!/bin/bash  
exit 0
```

- Bash es un lenguaje **muy diferente a Java o a C**
- las variables **no se declaran**,
- y se *expanden* con **\$** (ej. `${PATH}`),
- no tiene tipos,
- no tiene módulos o paquetes,
- *Be water, my friend*