

Sesión 14: Scripts

Programación para Sistemas

Ángel Herranz

Otoño 2018

Universidad Politécnica de Madrid

En capítulos anteriores. . .

Sesión 11: Contacto Bash

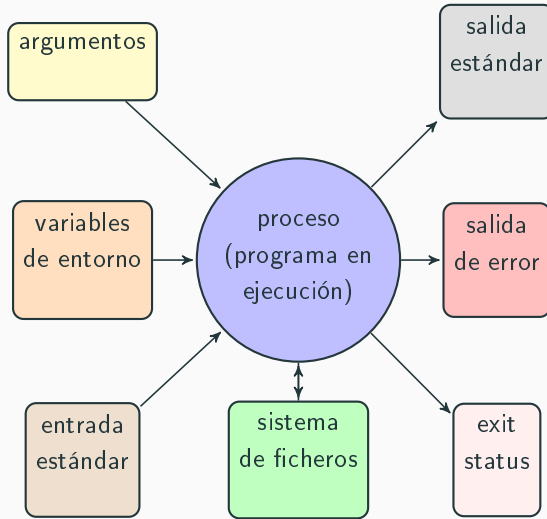
Sesión 12: Practicando Bash

Sesión 12: Mandatos, argumentos, variables y un poquito de redirección

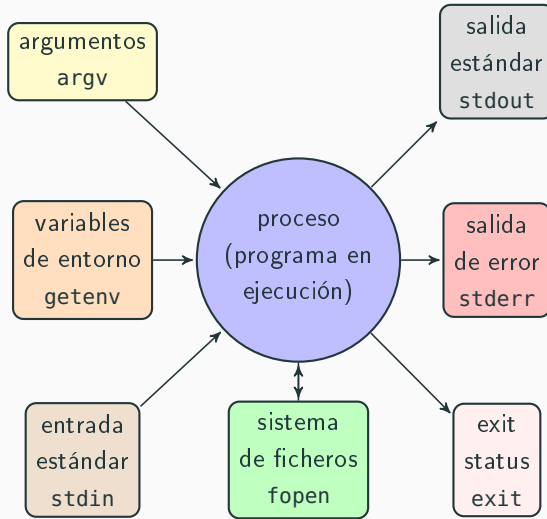
En el capítulo de hoy...

- Recordatorio
- Más sobre variables
- *Scripts*

Leyenda



Leyenda

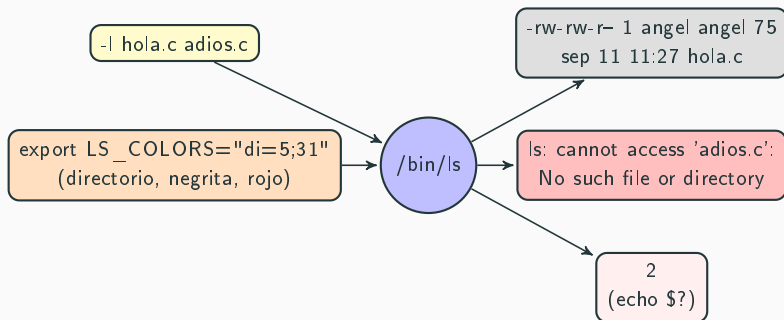


Ejemplo i

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

Ejemplo i

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

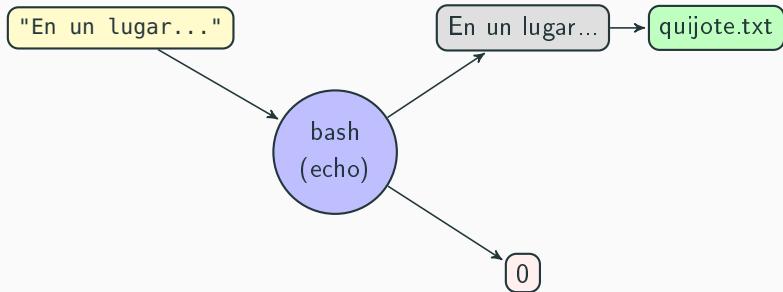


Ejemplo ii

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


Ejemplo ii

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

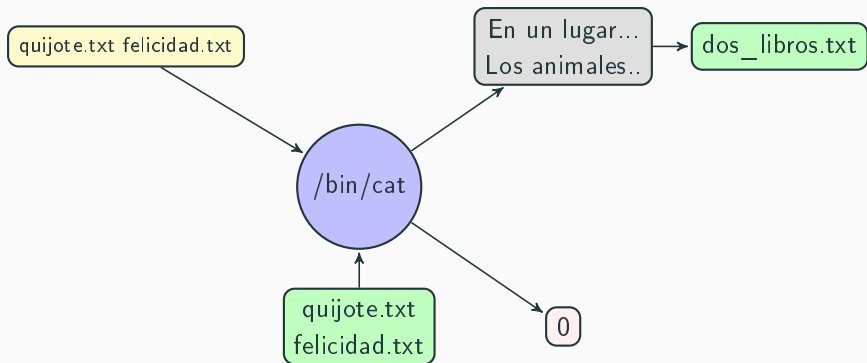


Ejemplo iii

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

Ejemplo iii

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

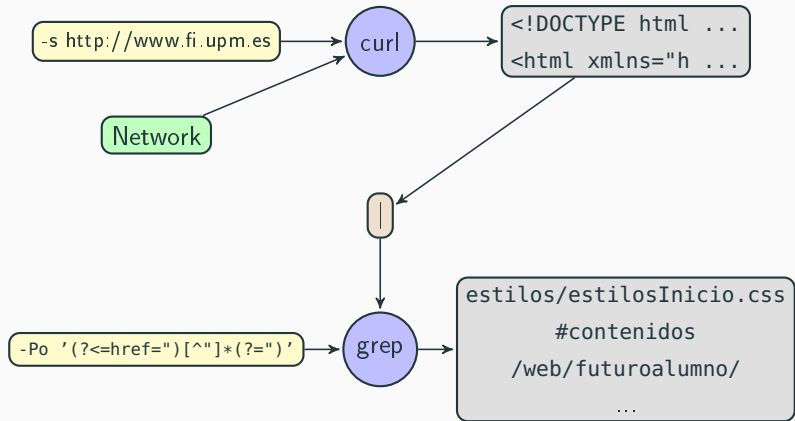


Ejemplo iv

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

Ejemplo iv

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



Scripts

C es un lenguaje compilado

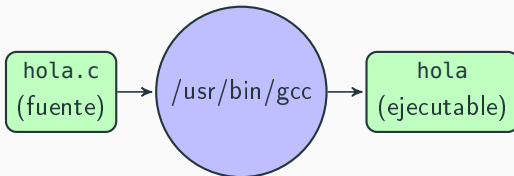
1. Compilación

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

C es un lenguaje compilado

1. Compilación

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



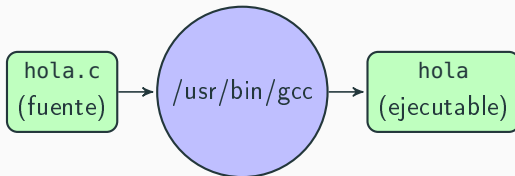
2. Ejecución

```
$ ./hola
```


C es un lenguaje compilado

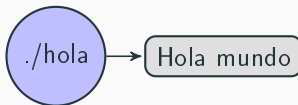
1. Compilación

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



2. Ejecución

```
$ ./hola
```



Bash es un lenguaje interpretado

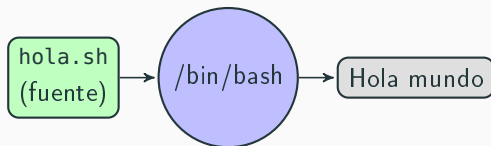
1. Ejecución

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

Bash es un lenguaje interpretado

1. Ejecución

\$./hola.sh

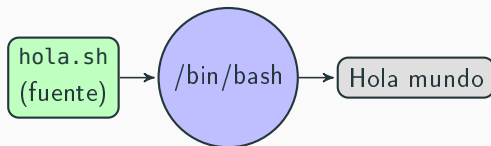


- ¡Y no hay más pasos!
- Vamos a aprender a hacer el `hola.sh`
- Otros lenguajes interpretados:

Bash es un lenguaje interpretado

1. Ejecución

\$./ho!a.sh

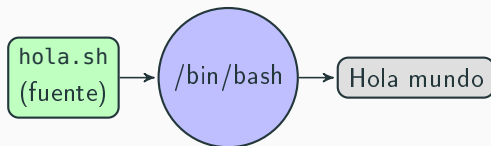


- ¡Y no hay más pasos!
- Vamos a aprender a hacer el ho!a.sh
- Otros lenguajes interpretados: PHP (php),

Bash es un lenguaje interpretado

1. Ejecución

\$./ho!a.sh

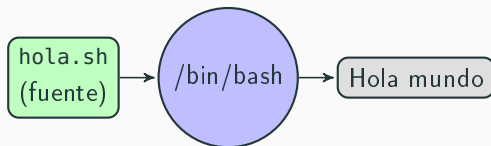


- ¡Y no hay más pasos!
- Vamos a aprender a hacer el ho!a.sh
- Otros lenguajes interpretados: PHP (php), Python (python),

Bash es un lenguaje interpretado

1. Ejecución

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




- ¡Y no hay más pasos!
- Vamos a aprender a hacer el `hola.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

Convección en UNIX

Cuando se va a ejecutar cualquier fichero de texto,

Convección en UNIX

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

Convección 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

Convección 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),

Convección 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

Convección 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*