

# Sesión 12: Practicando Bash

Programación para Sistemas

---

Ángel Herranz

Otoño 2018

Universidad Politécnica de Madrid

En capítulos anteriores. . .

## Sesión 11: Contacto Bash

# En el capítulo de hoy...

- Mail en triqui
- ¿Cómo nos comunicamos con nuestros programas?
- Empezamos en clase el tutorial de 2012 y ...

# En el capítulo de hoy...

- Mail en triqui
- ¿Cómo nos comunicamos con nuestros programas?
- Empezamos en clase el tutorial de 2012 y ...
- lo terminamos en casa

# Mail en triqui

- ¿Cómo funciona el correo electrónico?

MUA, MSA, MTA, MDA

- En triqui teneis una cuenta de correo (MSA + MDA)  
aherranz@triqui3.fi.upm.es
- Pero tienen limitaciones de uso

# Mail en triqui

- ¿Cómo funciona el correo electrónico?

MUA, MSA, MTA, MDA

- En triqui teneis una cuenta de correo (MSA + MDA)

aherranz@triqui3.fi.upm.es

- Pero tienen **limitaciones** de uso
- **Podeis** enviar correo **desde** triqui
- **Podeis** enviar correo **a** triqui desde triqui

MUAs: mail o mutt

# Mail en triqui

- ¿Cómo funciona el correo electrónico?

MUA, MSA, MTA, MDA

- En triqui teneis una cuenta de correo (MSA + MDA)

aherranz@triqui3.fi.upm.es

- Pero tienen **limitaciones** de uso
- **Podeis** enviar correo **desde** triqui
- **Podeis** enviar correo **a** triqui desde triqui

MUAs: mail o mutt

- **No podeis** enviar correo **a** triqui desde fuera de triqui

# Mail en triqui

- ¿Cómo funciona el correo electrónico?

MUA, MSA, MTA, MDA

- En triqui teneis una cuenta de correo (MSA + MDA)

aherranz@triqui3.fi.upm.es

- Pero tienen **limitaciones** de uso
- **Podeis** enviar correo **desde** triqui
- **Podeis** enviar correo **a** triqui desde triqui

MUAs: mail o mutt

- **No podeis** enviar correo **a** triqui desde fuera de triqui
- Los correctores se ejecutan en triqui y envían correo a vuestras cuentas en triqui



# mutt en triqui i

- Recomendación de MUA: mutt
- Configuración: ~/.muttrc

```
set mbox_type=Maildir
set folder=~/.Maildir
set mbox=~/.Maildir
set record=~/.Maildir/.Sent
set postponed=~/.Maildir/.Drafts
set spoolfile=~/.Maildir
set editor=nano
```

- ¿Problemas? Borra el directorio ~/.Maildir<sup>1</sup>:



```
$ cd
```

```
$ rm -rf Maildir
```


---

<sup>1</sup>Vas a perder todos los correos que tuvieras

# mutt en triqui ii

-  Envía varios correos a tu compañero de prácticas
-  Lee los correos que has recibido y contesta a alguno

# ¿Cómo nos comunicamos con los procesos?

 ¿Cómo ponemos en marcha nuestros programas<sup>2</sup>?:

---

<sup>2</sup>programa en ejecución = proceso

# ¿Cómo nos comunicamos con los procesos?

💬 ¿Cómo ponemos en marcha nuestros programas<sup>2</sup>?:

línea de comandos

💬 ¿Cómo podemos **enviar** información a nuestro programa?

---

<sup>2</sup>programa en ejecución = proceso

# ¿Cómo nos comunicamos con los procesos?

💬 ¿Cómo ponemos en marcha nuestros programas<sup>2</sup>?:

línea de comandos

💬 ¿Cómo podemos **enviar** información a nuestro programa?

- Parámetros en la línea de comandos

---

<sup>2</sup>programa en ejecución = proceso

# ¿Cómo nos comunicamos con los procesos?

💬 ¿Cómo ponemos en marcha nuestros programas<sup>2</sup>?:

línea de comandos

💬 ¿Cómo podemos **enviar** información a nuestro programa?

- Parámetros en la línea de comandos
- Entrada estándar: **stdin**

💬 ¿Cómo podemos **recibir** información de nuestro programa?

---

<sup>2</sup>programa en ejecución = proceso

# ¿Cómo nos comunicamos con los procesos?

💬 ¿Cómo ponemos en marcha nuestros programas<sup>2</sup>?:

línea de comandos

💬 ¿Cómo podemos **enviar** información a nuestro programa?

- Parámetros en la línea de comandos
- Entrada estándar: **stdin**

💬 ¿Cómo podemos **recibir** información de nuestro programa?

- Salida estándar: **stdout**

---

<sup>2</sup>programa en ejecución = proceso

# ¿Cómo nos comunicamos con los procesos?

💬 ¿Cómo ponemos en marcha nuestros programas<sup>2</sup>?:

línea de comandos

💬 ¿Cómo podemos **enviar** información a nuestro programa?

- Parámetros en la línea de comandos
- Entrada estándar: **stdin**

💬 ¿Cómo podemos **recibir** información de nuestro programa?

- Salida estándar: **stdout**
- Salida de error: **stderr**

---

<sup>2</sup>programa en ejecución = proceso



# ¿Cómo nos comunicamos con los procesos?

💬 ¿Cómo ponemos en marcha nuestros programas<sup>2</sup>?:

línea de comandos

💬 ¿Cómo podemos **enviar** información a nuestro programa?

- Parámetros en la línea de comandos
- Entrada estándar: **stdin**

💬 ¿Cómo podemos **recibir** información de nuestro programa?

- Salida estándar: **stdout**
- Salida de error: **stderr**
- Estado de terminación: **exit** o **return** en **main**

---

<sup>2</sup>programa en ejecución = proceso

# Estado de terminación (*Exit Status*)

- ✎ Escribir un programa que termine **mal** (`mal.c`)
- ✎ Ejecutar el programa y comprobar **cómo de mal ha terminado**

# Estado de terminación (*Exit Status*)

- 📄 Escribir un programa que termine **mal** (mal.c)
- 📄 Ejecutar el programa y comprobar **cómo de mal ha terminado**

```
$ ./mal
$ echo $?
-1
$ echo $?
0
$ if ./mal; then echo BIEN; else echo MAL; fi
MAL
```

💬 ¿Puedes explicar los resultados?

# Tutorial Bash (2012)

- `tutorial_bash_2012.pdf`
- Disponible en <https://github.com/aherranz/pps>
- Elaborado originalmente por Francisco Rosales<sup>3</sup>
- Empezamos en clase y ...

---

<sup>3</sup>Profesor del DATSI

# Tutorial Bash (2012)

- `tutorial_bash_2012.pdf`
- Disponible en <https://github.com/aherranz/pps>
- Elaborado originalmente por Francisco Rosales<sup>3</sup>
- Empezamos en clase y ...
- ¡lo terminamos en casa!
- ❓ Estad preparados: `man bash`

---

<sup>3</sup>Profesor del DATSI