

2021-09-29

Anton-Bravo, Adolfo

Contents

1	Contenidos	1
1.1	Software esencial	1
1.1.1	Windows	2
1.1.2	Mac	3
1.2	Editores Markdown	3
1.3	Github	3
1.4	Notas	4
1.5	lynx	4
1.5.1	Error de certificado	4
1.6	Estructura de las sentencias en la línea de comandos	4
1.7	BASH	5
1.7.1	Variables	6
1.7.2	Variables de entorno	6
1.7.3	MacOSX: zsh vs bash	7
1.7.4	zsh vs bash	7
1.8	Atajos de teclado	7
1.9	Operadores vistos	8
1.10	ToDo	8
2	Pruebas	8

1 Contenidos

1.1 Software esencial

- Terminal CLI: para casi todo.

- Navegador: Firefox y/o Chrom*. Un compañero usa Opera porque le va bien y tiene integrada VPN (*Virtual Private Network* o red privada

virtual).

- Editor de textos
- Limpieza de datos: Open Refine
- Github: organización, documentación, trabajo colaborativo, publicación web.
- Bootstrap: framework web
- Pandoc: transformaciones de formatos de archivos estructurados.
- Datawrapper: jugar con la visualización de datos.

1.1.1 Windows

- 7-zip, <https://7-zip.org>
- Dr Hardware
- Cygwin, la emulación de la terminal.
- apt-cyg

Cygwin en 4 pasos

1. Descargo el programa de cygwin.com
2. Al instalarlo, incluir los programas `lynx` y `wget` que luego va a usar `apt-cyg`.

3. Instalar apt-cyg de <https://github.com/transcode-open/apt-cyg>
4. Instalar programas con apt-cyg, por ejemplo, git.

Recordad que este bloque son dos líneas, hay que ejecutar una a una
`language=bash,label=,caption=,captionpos=b,numbers=none lynx -source rawgit.com/transcode-open/apt-cyg/master/apt-cyg > apt-cyg install apt-cyg /bin`

Aunque en realidad podríamos hacerlo todo en una línea con otro operador que veremos el próximo día :smiling-imp:

Y recordad también que lo que hay antes de esas dos líneas en la página de github es un tutorial. Hay que leérselo.

1.1.2 Mac

- Xcode para la terminal.
- Brew

1.2 Editores Markdown

- El propio Github.
- Macdown, para Mac <https://macdown.uranusjr.com/>
- nano :-)

1.3 Github

A veces puede pasar que el repositorio remoto tenga algo que no tenga el local, sin embargo es el local el que sabemos que hemos actualizado. Si hacemos un `git push`, nos dará error por eso mismo.

Para pasar por alto este error y forzar a actualizar el repositorio remoto con el local:

```
language=bash,label= ,caption= ,captionpos=b,numbers=none git push -f  
<remote> <branch>
```

1.4 Notas

- También se puede instalar en Linux: <https://docs.brew.sh/Installation#linux-or-windows-10-subsystem-for-linux>
- De ahí descubro que te puedes instalar un subsistema Linux en Windows <https://docs.microsoft.com/en-us/windows/wsl/about>

1.5 lynx

Lynx es un navegador en línea de comandos.

Es muy útil para saber si la web está bien hecha porque debería mostrar la información textual y también alternativas textuales a informaciones gráficas o de vídeo.

En este curso lo hemos utilizado con una de sus opciones, esto es, para descargar el código fuente de una URL:

```
language=bash,label= ,caption= ,captionpos=b,numbers=none lynx -source  
https://github.com/transcode
```

1.5.1 Error de certificado

- En clase no he recordado este "problema" que también es un clásico.
- Se resuelve instalando `ca-cert`

1.6 Estructura de las sentencias en la línea de comandos

- A veces se pone el comando solo, como en el caso de `pwd` o si quisiéramos lanzar `lynx` lo mismo que abrimos Firefox, como un navegador.

- Otras veces se ponen argumentos, como cuando listamos un directorio: `ls uc3m-periodismo-datos`. El comando va seguido de los argumentos. En el caso de `lynx` se puede poner la URL como argumento, así: `lynx https://theguardian.com`
- Y otras tiene opciones, como en `lynx -source https://theguardian.com`. En este caso la opción dice de descargar el código fuente.
- ¿Por qué sale todo el resultado seguido? Porque este comando usa la pantalla como salida estándar.
- Quienes hayáis instalado Cygwin habréis visto que después de eso iba un símbolo de mayor que y luego `apt-cyg`. El símbolo de mayor que es un **operador** de bash para enviar la salida estándar de un comando a un archivo.
- Si os lía lo del operador pensar que son símbolos que en este lenguaje tienen un significado especial. Por ejemplo, el símbolo de + significa sumar en una operación matemática. Son lo mismo, operan sobre los elementos.
- Pero antes de seguir con los operadores, veamos una cosa de Bash.

1.7 BASH

- El lenguaje de la terminal se denomina bash. Es un lenguaje de programación en sí pero también un entorno sobre el que trabajamos incluso sin programar.
- Bash es uno de los lenguajes posibles de la Shell, que es como se denominaría a la terminal. Es el más popular. Sin embargo, hay otros también muy extendidos como `zsh`.
- Podéis ver las que tenéis disponibles con `cat /etc/shells`

- No os diría esto si no fuera porque de un tiempo a esta parte MacOSX ha decidido usar zsh y eso produce algunos "problemas" curiosos. Básicamente, lo que ocurre es que aunque hemos instalado brew siguiendo las instrucciones, no parece existir. ¿Por qué? Porque no sabe dónde está brew o más bien, no se lo hemos dicho.
- En el caso de Mac podemos hacer dos cosas: o seguir usando zsh, y entonces configurarlo bien para que sepa dónde este brew; o cambiar a bash. (leer más adelante)
- Por cierto, esto es algo que nos lleva a explicar lo que son "las variables de entorno" y lo que son las variables.

1.7.1 Variables

Una de las cosas que tienen los lenguajes de programación son las variables, sirven para definir los datos. Por ejemplo, $x = 3$ lo que dice es que la variable x tiene el valor de 3. En el uso de los programas es similar y se llaman "de entorno" porque configuran la relación del programa con su entorno.

1.7.2 Variables de entorno

- Podemos ver las variables que tiene Bash escribiendo `env`. Vaya, salen muchas cosas en la pantalla y no podemos leerlas todas, sería mejor "paginar" el resultado con `less`. Tanto `less` como `more` son paginadores de texto, visores.
- Antes hemos visto cómo enviar la salida de un comando a un archivo con el operador `>`.
- Ahora vamos a ver cómo enviar la salida de un comando a otro comando. Es decir, queremos que el resultado de `env` lo utilice el paginador `less` como datos de entrada: `env | less`. El operador es la barra vertical `|` que se escribe con `Alt Gr + 1`.

- También podemos ver las variables, si sabemos su nombre, con `echo`. En este caso me interesa saber el valor de la variable `PATH`, así que `echo $PATH`, donde el símbolo del dólar indica que se trata de una variable.

1.7.3 MacOSX: zsh vs bash

- Para usar `bash` hay que escribir `chsh -s /bin/bash`, y con ello cambiamos la shell de `zsh` a `bash`.
- Para seguir usando `zsh` y tener `brew` hay que editar el archivo `.zshrc` y poner `export PATH"/opt/homebrew/bin:$PATH"`. O bien podemos conocer *otro operador de bash* y usarlo inteligentemente: `~export PATH"/opt/homebrew/bin:$PATH" » ~/.zshrc~`*. El operador `»` o "doble mayor que" envía el resultado del comando al final del archivo de destino. Es decir, no lo sobrescribe sino que lo añade.

Completitions <https://docs.brew.sh/Shell-Completion>

1.7.4 zsh vs bash

- Cosas chulas con `zsh`: <https://ohmyz.sh/>
- Cosas chulas con `bash`: <https://ohmybash.nntoan.com/> <https://github.com/ohmybash/oh-my-bash>

1.8 Atajos de teclado

- Es muy habitual usar atajos para ir más rápido en la relación con el ordenador.
- Los atajos se veían guay hasta que llegaron los *mousepad*.
- La convivencia es posible, hay veces que usar el *mousepad* está bien pero no os perdáis la potencia de los atajos.

1.9 Operadores vistos

- `>`, envía la salida de un comando a un archivo. Si este existiera ya, lo sobrescribe; si no existe, lo crea.
- `»`, envía la salida de un comando a un archivo. Si este existiera ya, lo añade al final; si no existe, lo crea.
- `|`, envía la salida de un comando como entrada de otro comando.

1.10 ToDo

- Otros operadores de bash.
- RTFM.
- Interfaces: Chatbots: Conversational Interfaces <https://www.youtube.com/watch?v=Kb01bWAvUK8> (2017)
- NICAR posters: https://www.dropbox.com/sh/yi1y0i6ka24498h/AAD_0c8rbYTKXGKnqJ69z8c4a?dl=0

2 Pruebas

- Compara el software que considero esencial con el que consideras esencial.
- ¿Cuál es la estructura de las sentencias/instrucciones en la línea de comandos?
- Si quisieras ver la web theguardian.com, ¿cómo lo harías desde la línea de comandos?

- ¿Cómo te descargarías la web theguardian.com desde la línea de comandos?
- ¿Cuál es la versión de Shell que utilizas?
- ¿Cómo verías las variable de entorno de tu shell "PATH"? Escribe su valor también.