

# Capítulo 9

## Comandos básicos IV

### Configuración del sistema

Con el comando **dpkg-reconfigure** veremos que podemos reconfigurar nuestro sistema.

Configuramos el idioma y teclado :

```
# apt-get install locales
# dpkg-reconfigure locales
```

Configuramos el teclado :

```
# apt-get install console-data
# dpkg-reconfigure console-data
```

### Ver historial de comandos

Todo los comandos que escribimos en nuestra consola se guarda en un archivo oculto llamado **.bash\_history** en nuestro **home**, guardara tantas cantidad de lineas que contenga la variable de entorno **HISTFILESIZE**.

```
# echo $HISTFILESIZE
500
```

Opciones :

<b>-c</b>	<i>Borra la lista de historia eliminando todas las entradas.</i>
<b>-d despl</b>	<i>Borra la entrada de la historia en la posición DESPL.</i>
<b>-a</b>	<i>Agrega las líneas de historia de esta sesión al fichero de historia.</i>
<b>-n</b>	<i>Lee todas las líneas de historia que no se han leído del fichero de historia.</i>
<b>-r</b>	<i>Lee el fichero de historia y agrega el contenido al fichero de historia.</i>
<b>-w</b>	<i>Escribe la historia actual en el fichero de historia.</i>
<b>-p</b>	<i>Hace expansión de historia en cada ARGumento y muestra el resultado, sin guardarlo en la lista de historia.</i>
<b>-s</b>	<i>Agrega los ARGumentos a la lista de historia como una sola entrada.</i>
<b>!n</b>	<i>Ejecuta el comando determinado por el número de línea.</i>
<b>!-n</b>	<i>Se referencia del comando corriente n líneas para atrás.</i>
<b>!!</b>	<i>Se referencia del comando corriente el anterior (-1).</i>
<b>!strings</b>	<i>Busca el comando dado por <b>strings</b> y lo ejecuta.</i>
<b>^string1^string2</b>	<i>Reemplaza todos los comandos de <b>string1</b> por <b>string2</b>.</i>
<b>^</b>	

Si la variable de entorno **HISTTIMEFORMAT** está definida y no es nula, se usa su valor como una cadena de formato strftime(3) para mostrar la marca de tiempo asociada con cada entrada de historia mostrada. No se muestra ninguna marca de tiempo de otra forma.

Opciones :

<b>%T</b>	<i>Reemplaza por el tiempo ( %H : %M : %S ).</i>
<b>%F</b>	<i>Equivale a %Y (año) - %m (mes) - %d (día).</i>

Ejemplo :

```
# export HISTTIMEFORMAT='%F %T '  
# history  
  
1 2010-05-18 10:33:01 ls  
2 2010-05-18 10:35:00 pwd  
3 2010-05-18 10:40:40 cd
```

Ejemplo :

```
# history  
  
1 ls  
2 pwd  
3 cd
```

Para ejecutar por ejemplo el comando **pwd** con número de línea 2 :

```
# !2
```

## Ver variables de entorno

Con los comandos **env** o **printenv** y **set** podremos ver las variables de nuestro entorno.

El comando **set** nos muestra tanto las variables locales como las globales, mientras que el comando **env** o **printenv** solo nos muestra las locales.

Opciones del comando **env** :

<b>-i, --ignore-environment</b>	<i>Comienza con las variables en blanco.</i>
<b>-u, --unset=NAME</b>	<i>Remueve una variable de entorno. Name es el nombre de la variable.</i>
<b>-h, --help</b>	<i>Ayuda.</i>
<b>--version</b>	<i>Version.</i>

Ejemplo :

```
# env
```

Opciones del comando **set** :

<b>-i, --ignore-environment</b>	<i>Comienza con las variables en blanco.</i>
<b>-u, --unset=NAME</b>	<i>Remueve una variable de entorno. Name es el nombre de la variable.</i>
<b>-h, --help</b>	<i>Ayuda.</i>
<b>--version</b>	<i>Version.</i>

Ejemplo :

```
# set
```

## Desactivar variables de entorno y funciones de shell

Con los comandos **unset** podemos desactivar valores y atributos de las variables y funciones de shell.

Opciones :

<b>-f</b>	<i>Trata cada NOMBRE como una función de shell.</i>
<b>-v</b>	<i>Trata cada NOMBRE como una variable de shel.</i>

Si no se especifica nada trata de borrar la variable, y si falla, trata de borrar la función.

Ejemplo :

```
# export HOLA='Hola Mundo !!!'
# echo $HOLA
  Hola Mundo !!!
# unset HOLA
# echo $HOLA
```

## Unir las líneas de varios ficheros

Con el comando **paste** las líneas en un archivo.

Opciones :

<b>-d, --delimiters=LIS</b>	<i>Trata cada NOMBRE como una función de shell.</i>
<b>T</b>	
<b>-s, --serial</b>	<i>Pegar un archivo a la vez en lugar de paralelo.</i>
<b>--help</b>	<i>Ayuda.</i>
<b>--version</b>	<i>Versión.</i>

Ejemplo :

```
# echo "Dato1|Dato2|" > archivo_01.txt
# echo "Marcos|36|" > archivo_02.txt
# paste -s archivo_01.txt archivo_02.txt
```

```
Dato1|Dato2|  
Marcos|36|
```

```
# paste -d '|' archivo_01.txt archivo_02.txt
```

```
Dato1|Dato2| Marcos|36|
```

## Leer una entrada y escribir la salida

Con el comando **tee** lee la entrada estándar por defecto y escribe su salida a la salida estándar y al archivo especificado.

Opciones :

<b>-a, --append</b>	<i>Añade su salida al archivo en lugar de sobrescribirlo.</i>
<b>-i, --ignore-interrupt</b>	<i>Ignora las señales de interrupciones.</i>
<b>--help</b>	<i>Ayuda.</i>
<b>--version</b>	<i>Versión.</i>

Ejemplo :

Lo muestra por pantalla y también lo graba en un archivo.

```
# ls -l | tee listado.txt
```

## Leer una entrada y escribir la salida con substitución o borrado

Con el comando **tr** lee la entrada estándar por defecto y escribe su salida siendo modificada.

Opciones :

<b>-d</b>	<i>Borrar los caracteres indicados en SET1.</i>
<b>-s</b>	<i>Elimina o reemplaza los caracteres repetidos indicados en SET1.</i>
<b>-c</b>	<i>Todos los caracteres que no sean los indicados en SET1 los convierte en SET2.</i>
<b>-t</b>	<i>Trunca SET1 a la longitud de SET2.</i>

Clases de caracteres:

<b>alnum</b>	<i>Caracteres alfanuméricos.</i>
<b>alpha</b>	<i>Caracteres alfabéticos.</i>
<b>cntrl:</b>	<i>Control.</i>
<b>digit</b>	<i>Dígitos.</i>
<b>graph</b>	<i>Caracteres gráficos.</i>
<b>lower</b>	<i>Minúsculas.</i>
<b>print</b>	<i>Caracteres imprimibles.</i>
<b>punct</b>	<i>Caracteres de puntuación.</i>
<b>space</b>	<i>Espacio.</i>

<b>upper</b>	<i>Caracteres mayúsculas.</i>
<b>xdigi</b>	<i>Caracteres hexadecimal.</i>

Ejemplo :

```
# echo murcielago | tr -d ae
# echo murcielago | tr '[a-z]' [A-Z]'
# echo "Hola Mundo" | tr -d ' '
# echo 'naci el 13-12-1966' | tr -c '0123456789' '-'
# cat /etc/passwd | tr '[:lower:]' '[:upper:]'
```

## Mostrar las líneas numeradas

Con el comando **nl** podemos numerar las líneas de un archivo.

Ejemplo :

```
# nl /etc/passwd
# nl -v=10 /etc/passwd
```

## Cortar sección de archivos

Con el comando **cut** nos permite buscar y/o seleccionar columnas o campos dentro de un archivo estructurado.

Opciones :

<b>-b, --bytes=LIST</b>	<i>Selecciona solamente bytes.</i>
<b>-c,</b>	<i>Selecciona solamente caracteres.</i>
<b>--characters=LIS</b>	
<b>T</b>	
<b>-d,</b>	<i>Usa DELIM como delimitador.</i>
<b>--delimiter=DEL</b>	
<b>IM</b>	
<b>-f, --fields=LIST</b>	<i>Selecciona solamente una columna para mostrar.</i>
<b>-s, --only-delimited</b>	<i>No muestra las líneas que no contienen delimitadores.</i>
<b>--help</b>	<i>Ayuda.</i>
<b>--version</b>	<i>Versión.</i>

Rangos

<b>N</b>	<i>Sólo ese byte,carácter o columna. O varios separados por comas.</i>
<b>N-</b>	<i>Desde N hasta el final de la línea.</i>
<b>N-M</b>	<i>Desde N hasta M (incluído).</i>
<b>-M</b>	<i>Desde el primero hasta M (incluído).</i>

Ejemplo :

```
# echo "Dato1|Dato2" > archivo_01.txt
```

```
# echo "Marcos|36|" > archivo_02.txt
# cut -d '|' -f2 archivo_02.txt
36

# echo "Marcos|36|Avellaneda|" > archivo_02.txt
# cut -d '|' -f2- archivo_02.txt
36|Avellaneda|

# cut -c1-6 archivo_02.txt
Marcos
```