

GUIÓN 2

EL EDITOR VI

Bibliografía

Sarwar, S. M.; Koretsky, R.; Sarwar, S. A. *El libro de LINUX*. Addison Wesley, 2005.
Sobell, Mark G. *Manual práctico de Linux. Comandos, editores y programación Shell*. Anaya Multimedia, 2008
Sánchez Prieto, S. *UNIX y LINUX. Guía práctica (Tercera edición)*. Ra-Ma, D.L. 2004.

El programa **vi** es un editor de página o de pantalla. Este editor tiene una característica importante, su *simplicidad*, hace que su salida gráfica sea adecuada prácticamente para cualquier tipo de terminal y además es estándar. Para editar un fichero basta con invocar a este programa teniendo como argumento el nombre del fichero que se desea editar:

\$ vi nombre_fichero

El editor mostrará el contenido del fichero si este existe o una pantalla vacía en caso contrario, permitiendo la creación del mismo. Si el fichero no existe y se aborta la edición del fichero, este no se crea.

Este editor también puede ser invocado sin parámetro o con varios ficheros como parámetros, en cuyo caso los tratará uno a uno según el orden en que aparezcan en la lista de argumentos.

El editor **vi** comienza su edición en la primera línea del fichero y el cursor se posicionará en el primer carácter de dicha línea. Si se desea comenzar visualizando otra línea distinta, se debe indicar dicha línea de la siguiente forma:

\$ vi +número_línea nombre_fichero

Si se quiere comenzar en la última línea, esto se expresa como sigue:

\$ vi +\$ nombre_fichero

El editor **vi** tiene dos modos: modo *orden* y modo *edición*. Cuando se está en modo *orden*, no se puede escribir texto. La estructura de la mayoría de las órdenes de **vi** es la siguiente:

[número1] operador [número2] objeto

Número1 es opcional y representa el número de veces que debe aplicarse el operador. *Número2* es también opcional y representa el número de objetos sobre los cuales se aplicará el operador.

Los operadores son letras mayúsculas o minúsculas. Cada uno representa un mandato particular.

Las órdenes **vi** no tienen eco en la pantalla. En cualquier orden que requiera la introducción de texto,

e indica que se ha terminado de introducir dicho texto pulsando la tecla **ESC** (y se vuelve al modo orden). Si se empieza a escribir una orden **vi** y se cambia de idea antes de terminar, se cancela la orden parcial pulsando **ESC**.

Estando en modo orden se puede obtener ayuda mediante la orden `:help` (o `:h`). Si se acompaña a esta orden con el nombre de una de las funcionalidades del editor, se obtendrá información sobre dicha funcionalidad. Para cerrar la ventana de ayuda se usa el comando `:q!`.

Órdenes para desplazar el cursor:

<code>l</code>	un carácter a la derecha
<code>h</code>	un carácter a la izquierda
<code>j</code>	una línea abajo
<code>k</code>	una línea arriba
<code>\$</code>	fin de línea
<code>^^</code> ó <code>0</code>	comienzo de línea
<code>W</code>	palabra siguiente
<code>b</code>	principio de palabra
<code>e</code>	fin de palabra
<code>H</code>	inicio de pantalla
<code>M</code>	mitad de pantalla
<code>L</code>	parte inferior de la pantalla

La mayoría de estas órdenes aceptan un número como factor; por ejemplo, **6l** que desplazará el cursor 6 caracteres a la derecha.

Para realizar estos movimientos del cursor en algunos terminales también se pueden utilizar las flechas del cursor.

Órdenes para controlar la pantalla:

<code><Ctrl>d</code>	página adelante
<code><Ctrl>u</code>	página atrás
<code><Ctrl>f</code>	pantalla siguiente
<code><Ctrl>b</code>	pantalla anterior

Órdenes para realizar desplazamientos dentro de un fichero:

nG	ir a la línea <i>n</i>
G	última línea
1G o gg	primera línea

Órdenes para pasar al modo texto o de inserción:

i	delante del cursor
a	después del cursor
o	sobre la línea siguiente
I	al principio de la línea
A	al final de la línea
O	en la línea anterior

A partir del momento en que se pulsa la orden de inserción, se está en el modo texto. Para salir de este modo, se debe pulsar **ESC**, como se indicó anteriormente.

Órdenes para borrar texto:

dw	borra la palabra que sigue
dd	borra línea actual (por ejemplo: 7dd borra siete líneas desde la actual)
D	borra hasta el final de la línea
x	borra el carácter donde está el cursor

Órdenes para modificar texto:

cw	cambia la palabra que sigue
cc	cambia la línea
C	cambia hasta el fin de la línea
r	sustituye el carácter donde está el cursor
R	sustituye desde el cursor. Salida con esc
J	junta la línea siguiente a la actual

Mover líneas con *d* (delete) y *p* (put)

Si se desea mover 5 líneas desde la actual, se escribe antes **5dd**, para suprimirlas. Estas líneas no se han perdido, pues van a parar a un buffer específico de donde puede recuperarlas con el mandato **p**, para

ello se coloca el cursor en la línea tras la cual se desean escribir las cinco líneas borradas y se pulsa la tecla **p**.

Copiar líneas con **y** (*yank*) y **p** (*put*)

Si se desea copiar cinco líneas entonces se coloca el cursor en la primera línea que se desea copiar y se pulsa **5yy**, posteriormente se coloca el cursor en el lugar en que se desea copiar dichas líneas y se pulsa **p** (las líneas se recuperan del buffer donde han quedado depositadas).

Órdenes para buscar texto:

/ccc	busca la primera ocurrencia de <i>ccc</i> hacia delante
?ccc	busca la primera ocurrencia de <i>ccc</i> hacia atrás
n	próxima ocurrencia hacia delante
N	próxima ocurrencia hacia atrás

Las cadenas de búsqueda son expresiones regulares, por lo que ciertos caracteres tienen un significado especial dentro de una cadena de búsqueda.

- Cuando el primer carácter de una cadena de búsqueda es el carácter **^** (circunflejo), se busca la línea siguiente/anterior que comienza por la cadena de búsqueda. Ejemplos:
 /^hola busca la siguiente línea que comienza por la palabra hola
 ?^hola busca la anterior línea que comienza por la palabra hola
- Cuando la cadena de búsqueda termina en el carácter **\$** (dólar), se busca la siguiente/anterior línea que termina con la cadena de búsqueda. Ejemplos:
 /hola\$ busca la siguiente línea que termina por la palabra hola
 ?hola\$ busca la anterior línea que termina por la palabra hola
- Cuando se usa el carácter **.** (punto) en cualquier parte de la cadena de búsqueda, se busca la siguiente/anterior cadena que coincida con la cadena de búsqueda sustituyendo el citado carácter punto por cualquier carácter. Ejemplos:
 /h.a busca la siguiente línea que contenga una palabra que comienza por h seguida de un carácter y termine en a
 ?h..a busca la anterior línea que contenga una palabra que comienza por h seguida de dos caracteres y termine en a
- Cuando la cadena de búsqueda finaliza con el par de caracteres **\>** se busca en la siguiente/anterior línea, palabras que terminan con dicha cadena de búsqueda. Ejemplos:
 /a\> busca en la siguiente línea una palabra que termine en a
 ?a\> busca en la anterior línea una palabra que termine en a
- Cuando la cadena de búsqueda comienza con el par de caracteres **\<** se busca en la siguiente/anterior línea, palabras que comienzan con dicha cadena de búsqueda. Ejemplos:
 /\<a busca en la siguiente línea una palabra que comience por a
 ?\<a busca en la anterior línea una palabra que comience por a

- Si se usa el carácter * (asterisco) en la cadena de búsqueda, se busca en la siguiente/anterior línea cadenas con cero o más ocurrencias del carácter que precede inmediatamente al carácter *. Ejemplos:

/ha*s	busca en la siguiente línea una cadena que comience por h y que contenga cero o más caracteres a seguidos de una s.
?ha*s	busca en la anterior línea una cadena que comience por h y que contenga cero o más caracteres a seguidos de una s.

Se puede anular el significado especial de los tres primeros caracteres (^, \$ y .) dentro de una cadena de búsqueda anteponiéndole una barra invertida (\).

Orden para sustituir texto:

La sintaxis del comando de sustitución es:

: [g][dirección]s/cadena de búsqueda/cadena por la que sustituir[/opción]

La s indica que es el comando de sustitución que debe ir acompañado por un delimitador (/) marcando el principio de la cadena de búsqueda. Otro delimitador (/) marca el final de la cadena de búsqueda y el comienzo de la cadena de reemplazamiento. Esta cadena de reemplazamiento debería ir acompañada del carácter delimitador, pero se puede omitir si no existen opciones. Este comando sólo sustituye la primera aparición de la cadena de búsqueda en la línea.

Si no se indica una dirección de reemplazo, el comando busca sólo en la línea actual la primera ocurrencia de la cadena de búsqueda. Si se indica un solo número de línea, el comando busca la primera ocurrencia en dicha línea. Si la dirección tiene dos números separados por una coma, el comando busca en aquellas líneas y en las que están en medio. Ejemplos:

:s/hola/adios	busca en la línea actual la primera ocurrencia de la cadena hola y la sustituye por la cadena adios
:3s/e/s	busca en la tercera línea del fichero la primera ocurrencia del carácter e y lo sustituye por el carácter s
:3,10s/es/y	busca en las líneas comprendidas entre la tercera y décima del fichero, la primera ocurrencia de la cadena es y la sustituye por el carácter y

Si la dirección es una cadena encerrada entre barras inclinadas (/), el editor realiza la búsqueda y el reemplazo en la siguiente línea que coincide con la cadena de direcciones. Si se antepone a la primera barra inclinada de la cadena de direcciones la opción g (global), el editor utiliza todas las líneas del fichero que contienen la cadena de direcciones para realizar la búsqueda y el reemplazo. Ejemplos:

:/hola/s/x/y	busca la siguiente línea que contenga la cadena hola y sustituye la primera ocurrencia del carácter x por el carácter y
:g/hola/s/con/fin	busca todas las líneas del fichero que contenga la cadena hola y sustituye la primera ocurrencia de la cadena con por la cadena fin
:g//s/con/fin	
:%s/con/fin	ambas órdenes realizan la misma sustitución. Se busca en todas las líneas del fichero y se sustituye la primera ocurrencia de la cadena con por la cadena fin. El patrón de búsqueda es la cadena vacía, por lo que

se buscan en todas las líneas del fichero. Como se observa el carácter % es sinónimo de la opción g (global) y el patrón de búsqueda vacío.

Si se desea reemplazar todas las ocurrencias que coinciden con la cadena de búsqueda en una línea, es necesario añadir la opción **g** después del delimitador que termina con la cadena de reemplazo. La opción **c** (comprobación) hace que el editor pregunte si se hace la sustitución una vez localizada la cadena de búsqueda. Ante esta pregunta existen varias opciones: “y” (yes) reemplaza la cadena de búsqueda, “l” (last) se hacen los reemplazamientos y se sale, “q” (quit) se termina el comando, “a” (all) se hacen el resto de reemplazos, y con “n” (no) se sigue la búsqueda sin hacer reemplazo alguno. Ejemplo:

:s/hola/adios/g busca en la línea actual todas las ocurrencias de la cadena hola y las sustituye por la cadena adios
:s/hola/adios/gc busca en la línea actual todas las ocurrencias de la cadena hola y pregunta si se desea sustituir por la cadena adios
:g/s/con/fin/g
:%s/con/fin/g ambas órdenes realizan la misma sustitución. Se busca en todas las líneas del fichero y se sustituye todas las ocurrencias de la cadena con por la cadena fin

Órdenes diversas:

ZZ	guarda el fichero y sale de vi
u	Anula el mandato anterior
<Ctrl>r	rehace el mandato anulado
y	coloca la línea actual en el buffer
p	escribe el contenido del buffer después de la línea actual
P	escribe el contenido del buffer antes de la línea actual
:f	muestra el nombre del archivo que se está editando junto con información de su estado
:w	guarda el texto
:q	se abandona el editor si se han guardado los cambios, si esto no es así avisa y no abandona el editor
:q!	abandona el editor tanto si se han guardado los cambios como si no
:wq	guarda los cambios efectuados en el fichero y abandona el editor
:sh	abre un comando desde el editor. Para salir teclear exit.
:set number	muestra el número de las líneas
:set nonumber	elimina el número de las líneas

Salida inesperada del editor vi

Existen varias causas por las que es posible que el editor **vi** termine de ejecutarse de forma inesperada

cuando se está editando un fichero. En este caso el editor genera un fichero de intercambio, que es un fichero oculto (comienza por punto) seguido del nombre del fichero cuya edición se ha cancelado y con la extensión *swp*. Para poder continuar editando el fichero mediante el editor **vi** se puede realizar uno de los siguientes procedimientos:

a. El primero conlleva los siguientes pasos:

1. Editar el fichero lanzando el editor **vi** con la opción **r** (recuperar):

```
$ vi -r nombre_fichero
```

2. Salir del editor **vi**:

```
:wq
```

3. Eliminar el fichero de intercambio:

```
$ rm .nombre_fichero.swp
```

b. Los pasos a seguir del segundo procedimiento son:

1. Editar el fichero lanzando el editor **vi**:

```
$ vi nombre_fichero
```

2. Al iniciarse el editor **vi**, éste avisa de que ha localizado un fichero de intercambio permitiendo hacer una de las siguientes opciones: [A]brir para lectura,[E]ditar de todas formas,[R]ecuperar, ...

De estas opciones es aconsejable usar [R] que recupera el fichero.

3. Al salir del editor **vi**, se debe borrar el fichero de intercambio:

```
$ rm .nombre_fichero.swp
```