

Edici3n de ficheros: vi

El editor `vi` est1 disponible en cualquier distribuci3n de UNIX. Para empezar conviene decir que es un editor pensado hace bastantes a1os y no es muy amigable comparado con editores de texto que estamos acostumbrados a usar, sin embargo resulta conveniente conocer su manejo ya que est1 muy difundido y no es seguro poder encontrar cualquier otro editor de textos en todas las instalaciones UNIX.

`Vi` posee b1asicamente dos modos de trabajo: modo inserci3n y modo comando. En el modo inserci3n el editor permite a1adir texto a partir de una posici3n determinada (b1asicamente ofrece la funcionalidad de una m1quina de escribir); en modo comando se pueden hacer el resto de funciones t1picas de un editor (grabar, buscar una palabra, realizar sustituciones, salir,...).

Al entrar al editor se est1 en modo comando. Una posible forma de pasar a modo inserci3n es apretar la tecla `i` min1scula; a partir de ese momento todos los caracteres que tecleemos se insertaran en la posici3n anterior del cursor.

Para pasar de modo inserci3n a modo comando se debe apretar la tecla `ESC` o apretar las teclas `CTRL` y `3` simult1neamente.

Un resumen de los comandos simples m1s usuales es:

Teclas del cursor	Movimiento del cursor derecha/izquierda/arriba/abajo
<code>^b</code>	Avanza (<u>b</u> ackward) una p1gina (hay que apretar a la vez la tecla <code>ctrl</code> y la tecla <code>b</code>)
<code>^f</code>	Retrocede (<u>f</u> oreward) una p1gina
<code>dd</code>	Borra (<u>d</u> elete) la l1nea sobre la que est1 situado el cursor
<code>x</code>	Borra el car1cter sobre el que est1 situado el cursor
<code>u</code>	Deshace (<u>u</u> ndo) el efecto del 1ltimo comando
<code>yy</code>	Copia la l1nea actual en el buffer de cut-and-paste
<code>p</code>	Copia el buffer de cut-and-paste sobre la posici3n del cursor (<u>p</u> aste)
<code>^g</code>	Nos dice en qu1 l1nea del fichero nos encontramos

A continuaci3n presentamos una serie de comandos que precisan ser terminados con `RETURN`:

<code>:w</code>	Actualiza (<u>w</u> rite) los cambios realizados a disco sin salir del editor
<code>:wq</code>	Actualiza (<u>w</u> rite) los cambios y sale (<u>q</u> uit) del editor
<code>:q!</code>	Sale del editor sin actualizar los cambios
<code>:r fichero</code>	Lee (<u>r</u> ead) el fichero indicado y lo a1ade al fichero en edici3n
<code>:n</code>	Va a la l1nea n1mero <code>n</code> del fichero
<code>:\$</code>	Sit1a el cursor en la 1ltima l1nea del fichero
<code>/palabra</code>	Busca una palabra en el fichero
<code>:set number</code>	Visualiza los n1meros de l1nea del fichero
<code>:set showmode</code>	Muestra en qu1 modo se encuentra el editor

(comando/inserción)


A los principiantes en el uso de vi les resulta de gran ayuda saber en qué modo se encuentran en cada momento. El último comando muestra en la línea inferior de la pantalla dicha información.

Para acabar diremos que el editor vi permite adaptar algunas características de su entorno a los gustos del usuario. Para ello se deberá editar el fichero .exrc en el “home directory” del usuario e insertar en él los comandos de tipo set que interesen.

Por ejemplo *set showmode* y *set number*.

Guía rápida para el uso del editor Vi

Ejemplos de comandos

	mueven el cursor.
h j k l	hacen lo mismo que las flechas.
itexto<Esc>	inserta texto.
cwnueva<Esc>	cambia la actual palabra por nueva.
x	borra un carácter.
dw	borra una palabra.
dd	borra una línea.
3dd	borra tres líneas.
u	deshace el cambio previo.
:wq	sale de vi guardando los cambios.
:q!	sale de vi sin guardar los cambios.
/texto	busca <i>texto</i> .
^U ^D	desplaza una página arriba o abajo.

Interrupción, cancelación

<Esc>	finaliza la inserción o un comando incompleto.
^L	reimprime la pantalla.

Manipulación de ficheros

:w	escribe los cambios efectuados en el fichero.
:q	abandona la edición.
:q!	abandona la edición descartando los cambios.
:e fichero	edita el fichero.
:w fichero	reescribe el <i>fichero</i> .
:sh	ejecuta un shell y vuelve a editar en el punto actual.
!: comando	ejecuta <i>comando</i> del shell y vuelve a editar en el punto actual.
^G	muestra el fichero y línea actuales.
:r fichero	añade fichero al fichero en edición

Posicionamiento

^F	muestra la siguiente página.
^B	muestra la página anterior.
^D	muestra la media página siguiente.
^U	muestra la media página anterior.
: #	se mueve a la línea especificada.
H	se mueve a la primera línea de la pantalla.
L	se mueve a la última línea de la pantalla.
M	se mueve a la línea de en medio de la pantalla.
RETURN	siguiente línea, en el primer carácter no blanco.
-	línea anterior, en el primer carácter no blanco.
← ó h	mueve el cursor a la izquierda.
↓ ó j	mueve el cursor abajo.
↑ ó k	mueve el cursor arriba.
→ ó l ó espacio	mueve el cursor a la derecha
0	principio de la línea.
\$	final de la línea.
/patrón	busca el <i>patrón</i> hacia adelante.
?patrón	busca el <i>patrón</i> hacia atrás.
n	repite el último comando de búsqueda.
N	invierte el sentido del último comando de búsqueda.
(principio de la frase.
)	final de la frase.
{	principio del párrafo.
}	final del párrafo.

Correcciones durante la inserción

^H o erase	borra el último carácter.
^W	borra la última palabra.
Kill	la tecla definida con la función kill borra la línea actual.
<Esc>	acaba la inserción.

Inserción y reemplazo

a	añade tras el cursor.
i	inserta antes del cursor.
A	añade al final de la línea.
I	inserta antes del primer carácter diferente de blanco.
o	abre una línea debajo de la actual.
O	abre una línea encima de la actual.
rx	reemplaza el carácter bajo el cursor por x.
R <i>texto</i> <Esc>	reemplaza a partir del carácter bajo el cursor por <i>texto</i> .
J	junta líneas.
y	copia línea al buffer de "cut and paste".
: <i>línea</i> , <i>lineas</i> y	copia el rango de líneas especificado al buffer de 'cut and paste'.
p	pon texto del buffer de "cut and paste" tras el cursor.
P	pon texto del buffer de "cut and paste" antes del cursor.
: <i>línea</i> , <i>lineas</i> / <i>texto_existente</i> / <i>texto_nuevo</i> /gc	

sustituye el texto existente por texto nuevo entre las *líneas* que se indican. El punto (.) equivale a la línea actual y el dollar (\$) a la última línea del fichero. La opción g al final del comando indica que si aparece el mismo texto varias veces en una misma línea se reemplace siempre, en caso contrario solo se reemplazará la primera ocurrencia. La opción c indica que se desea confirmar cada cambio. A continuación mostramos un ejemplo:

```
:1,.s/error/escriure_errorlg
```

Substituye todas las apariciones entre la primera línea y la línea actual del string **error** por **escriure_error**.