

Trabajando con el editor de texto VIM

PONLE ALGO DE VIM



Vim se siente en casa con cualquier distribución Linux. Este editor en modo texto, que no requiere para nada el uso del ratón, es de una ayuda imprescindible para los fans de la línea de comandos. **POR HEIKE JURZIK**

El editor de texto Vim [1] y su predecesor Vi, (implementado típicamente como enlace a Vim o como un alias para Vim) nos exigen acostumbrarnos a ellos. Además, la (nada intuitiva) interfaz de usuario que posee esta práctica herramienta espanta a muchos usuarios que evitan probarla. Pero una vez que se consiguen conocer los distintos modos de

operación, probablemente no se desee hacer nada sin los útiles servicios que ofrece este ligero y potente editor.

Una Cuestión de Status

La mayoría de los usuarios tiene alguna dificultad para comprender el concepto de los distintos modos de operación suministrados por Vim. Des-

pues de todo, dispone de no menos que de cuatro de ellos.

Normal mode es el modo central desde el que puede cambiarse a los otros restantes. Es el que presenta Vim cuando arranca; también conocido como *command mode*. Distintos atajos de teclado permiten a los usuarios eliminar, copiar y mover texto. Las entradas siguen siempre el mismo patrón:

se comienza diciendo la frecuencia con la que se desea ejecutar el comando (deja de presentarse exactamente una vez) y luego se escribe el comando. Para volver al modo comando desde uno de los otros modos se pulsa [Esc] una vez o múltiples veces, hasta que se oiga un pitido avisando de que se ha vuelto a modo normal.

Para introducir texto se pulsa la tecla [I] (de insertar) para pasar a *insert mode*. Vim presenta un mensaje – INSERT – en la parte inferior izquierda de la pantalla. Otros métodos para cambiarse al modo insertar son pulsando [A] (de “append” o adjuntar) para insertar texto directamente detrás del cursor, [Shift] + [I] que lleva hasta el comienzo de la línea actual y cambia al modo insertar, y [Shift] + [A] que lleva hasta el final de la línea y cambia al modo insertar. También puede escribirse [O] en modo normal para abrir una línea debajo del cursor, o [Shift] + [O] para insertar una línea por encima de éste.

Para cambiar al modo *command line* pueden escribirse dos puntos (:) en modo normal. Una vez hecho, se pueden introducir comandos en el prompt de los dos puntos para buscar y reemplazar, por ejemplo. Para ejecutar los comandos de la shell se escribe una exclamación (!) después de los dos puntos.

El *Visual mode* permite seleccionar áreas carácter por carácter, línea por línea o bloque por bloque para copiar o cortar posteriormente la selección. Pulsando el atajo de teclado [Shift] + [V] lleva hasta el modo visual. Vim presenta una etiqueta – VISUAL – en la parte inferior izquierda de la pantalla para decir en qué modo se encuentra, y cualquier texto que se seleccione será resaltado.

Primeros Pasos

El editor se arranca introduciendo *Vim* en la línea de comando; opcionalmente puede suministrarse el nombre del fichero que se desea editar:

```
vim linux-user 2
/2006/08/ vim.txt
```

Para abandonar el editor se cambia de nuevo al modo normal y se escribe :q!

para abandonar Vim y descartar los cambios que se hayan hecho. Alternativamente, puede escribirse :wq si se desean salvar las modificaciones antes de abandonar Vim. Si lo que se quiere es salvar un paso se escribe :x, o en su lugar, [Shift] + [Z] dos veces.

Para guardar los cambios sin salir de Vim se introducirá el comando :w. También puede adjuntarse un nombre de fichero al comando para guardar los cambios en un nuevo fichero:

```
:w nuevo_fichero.txt
```

La manera en la que arranca el editor depende de varios parámetros de la línea de comandos. Si desea abrirse un fichero de sólo lectura, se hará añadiendo la opción -R. Vim posee también una opción de recuperación práctica que permite restaurar una sesión después de una caída del sistema. Para ello, se escribe *vim -r fichero* en la línea de comandos.

Pulsando [U] es posible deshacer un paso. Cuando se trate de varios, pulsaremos esta misma tecla varias veces. Para descartar todas las modificaciones en la línea actual se pulsa [Shift] + [U].

Para obtener una perspectiva de todos los parámetros de la línea de comandos, con cortas explicaciones, se escribe en la shell *Vim -help*.

Pizarra en blanco

A diferencia de la versión antigua del editor Vi, las teclas [Del] y [Backspace] suprimen texto en el modo de inserción de Vim. Sin embargo, las funciones de borrado y modificación en el modo comando son mucho más prácticas. Se pulsa [X] para borrar el carácter bajo el cursor, añadir un número para borrar un número específico de caracteres: por ejemplo, si se pulsa 5x se borrarán exactamente cinco caracteres.

Para eliminar la palabra bajo el cursor se mueve éste hasta la primera letra de la palabra que se desea borrar y se escribe dw. Si se desea reemplazar una palabra, entonces se escribe cw. Vim no solamente sustituirá la palabra bajo el cursor, sino que cambiará automáticamente al modo insertar y podrá introducirse el texto de sustitución.

Para borrar una frase se escribe la

combinación d), y para eliminar el texto desde la posición actual del cursor hasta el final de la línea se escribe d\$. De nuevo, puede escribirse c en lugar de d para cambiar directamente a modo insertar después de borrar el texto. Cuando haya que eliminar una línea completa escribiremos dd y, al modo Vim típico, cuando lo que se quieran eliminar sean tres líneas de una sola vez, se añadirá el número, como en 3dd; 3cc borrará tres líneas y, como se habrá adivinado, cambiará al modo insertar.

También son posibles combinaciones con otros comandos. Si se desea eliminar desde la línea actual hasta el final del fichero, se escribe dG, o bien dgg para cuando sea desde la línea actual hasta el comienzo del fichero.

Copia Artística

Vim ofrece la posibilidad de insertar texto eliminado mediante uno de los comandos anteriores en una posición diferente en el documento. Para ello, se pulsa [P] para insertar el contenido del portapapeles delante del cursor; o pulsar [Shift] + [P] para insertar el texto detrás del cursor.

Evidentemente no es necesario borrar texto para guardarlo en el portapapeles y luego insertarlo en otra posición. Vim también ofrece un número determinado de funciones de copia. Para copiar una línea se escribe yy y luego se pulsa [P] o [Shift] + [P]. De nuevo pueden establecerse combinaciones de comandos: puede escribirse un número para decir el número de líneas que se desean copiar. Por ejemplo, si se escribe 5yy copiará la línea actual y las cuatro siguientes.

Existen muchos métodos para copiar una selección definida con mayor precisión: se pulsa [Esc] + [V] luego se selecciona una región de texto con las teclas de flecha y se pulsa [Y] para copiar el texto al portapapeles. A continuación pueden pulsarse [P] o [Shift] + [P] para insertarlo en una posición distinta.

Búsqueda y Sustitución

Para encontrar una cadena de texto se pulsa /, se escribe la clave de

CONSEJO

Pulsando [.] se repetirá el último comando de Vim.

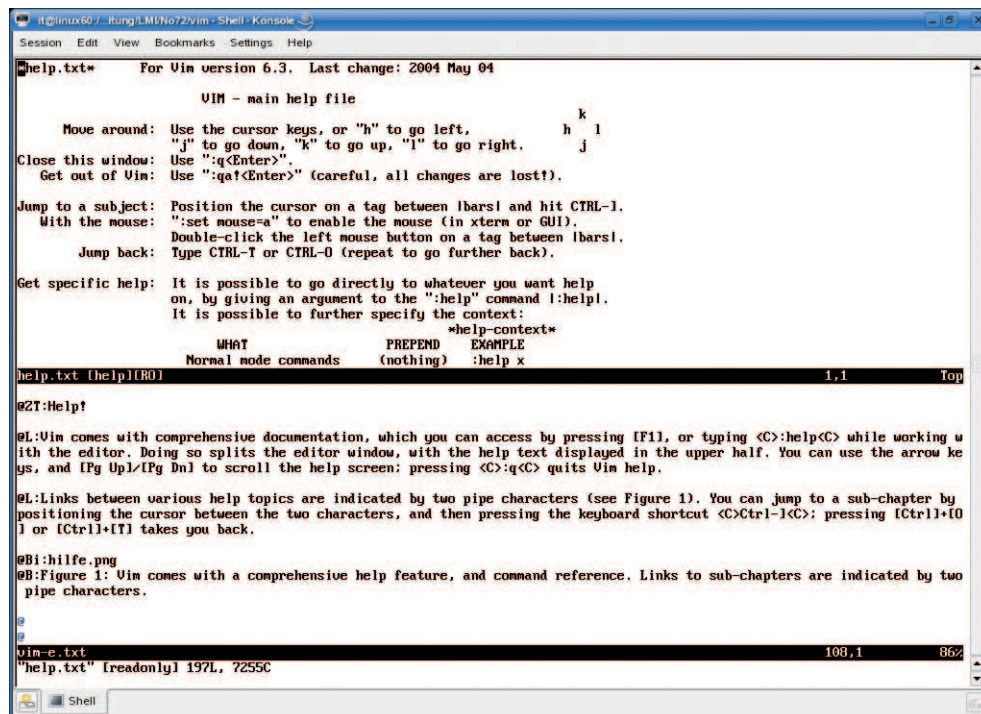


Figura 1: Vim viene con una característica de ayuda extensa y una referencia a comandos. Los enlaces a los sub-capítulos se indican mediante dos caracteres tubería.

búsqueda y se pulsa [Enter]. [N] le dice a Vim que busque en la dirección actual, y [Shift] + [N] invierte la dirección. También puede optarse por la búsqueda hacia atrás escribiendo `?search_key`. Vim ofrece también dos prácticos atajos: con `*` buscará hacia adelante a partir de la palabra bajo el cursor, y con `#` lo hará hacia atrás.

Posee además otra práctica característica que ayudará a los programadores y fans de LaTeX a buscar por parejas de corchetes. Es preciso ir hasta un corchete abierto, tal como `(`, `{` o `[`, y se escribe `%`. Automáticamente Vim irá al corchete cerrado; pulsando otra vez `%` se volverá hasta el corchete abierto.

Para buscar y sustituir es preciso cambiar a la línea de comandos. Para sustituir un string de Vi con Vim se escribiría `:s/Vi/Vim`. Vim sustituirá solamente el siguiente caso de la clave de búsqueda. Es posible sustituir todas las claves de búsqueda en una línea escribiendo `s/Vi/Vim/g`, y todas las claves de búsqueda en el fichero completo introduciendo `%s/Vi/Vim/g`.

Por Siempre Jamás

Cuando Vim arranca busca un fichero de configuración, que por defecto es `~/ .vimrc` de su propio directorio home. Si

se necesita una buena plantilla para que se ajuste a sus necesidades recomen-

```

Tu Propio .vimrc
01 "Activar resaltado sintáctico:
02 syntax on
03
04 "Mostrar números de línea:
05 set number
06
07 "Ir a texto coincidente en
   búsqueda:
08 set incsearch
09
10 "Desactivar sangrado
   automático:
11 set nosmartindent
12
13 "Mapear F2 a comando del shell
   fmt:
14 map <F2> !}fmt<CR>
15
16 "Llamar a parámetros de ispell
   cuando se pulsa F3 :
17 map <F3> :w!<CR>:!ispell -T
18 latin1 %<CR>:e! %<CR>
    
```

mos echar un vistazo a `etc/vimrc` o `etc/vim/vimrc`.

El fichero de configuración para el editor (`~/ .vimrc`) no solamente da alcance para las propias preferencias personales, sino también para macros, resaltado sintáctico individual y muchas otras cosas. El cuadro titulado "Tu Propio .vimrc" ofrece un par de ejemplos sencillos comentados (los comentarios en Vim se indican con el carácter `"`).

¡Ayuda!

Vim viene con una extensa documentación a la que puede accederse pulsando [F1] o escribiendo `:help` en el editor. De este modo se abre la ventana del editor con el texto de ayuda presentado en la mitad superior. Las teclas [AvPág] y [RePág] permiten desplazarse por la pantalla de ayuda; con `:q` se sale de la ayuda de Vim.

Los enlaces entre los temas de ayuda se indican por dos tuberías (ver Figura 1). Es posible cambiar a un sub-capítulo situando el cursor entre los dos caracteres, y luego pulsando el atajo de teclado [Ctrl-]; con [Ctrl] + [O] o [Ctrl] + [T] se vuelve hacia atrás.

Para buscar el texto de ayuda relativo a palabras clave, se escribe `:help topic`; pueden usarse comodines como `*` (para cualquier número de caracteres), y `?` (para un único carácter). Si Vim no encuentra su clave de búsqueda, el editor ofrecerá algo que se le parezca. Puede usarse un truco para presentar distintos temas y luego seleccionar uno de los que se estaba buscando. Se escribe `:help` seguido de las primeras letras del tema en el que se está interesado, por ejemplo, `help syn` y luego se pulsa [Ctrl] + [D]. Vim dará una selección de temas de ayuda que comienzan con la cadena "syn".

El tutorial que viene con la distribución Vim es otra útil fuente de información. Se arrancará introduciendo `vimtutor` en la línea de comandos. ■

RECURSOS

[1] Página home de Vim: <http://www.vim.org>