

El editor vi

1. Presentación

El editor de Unix por defecto se llama **vi** (*visual editor*). No es de los más ergonómicos en comparación con editores en modo gráfico, pero tiene la ventaja de estar disponible y de utilizar la misma sintaxis básica en todos los Unix. Cada Unix propone en general una sintaxis extendida más allá de la sintaxis básica. El editor vi en Linux se llama **vim**. Vim respeta toda la sintaxis de vi, pero no es recíproco. Vi es pequeño: ocupa poco espacio en disco, consume poca memoria. Algunas de las opciones disponibles sólo funcionan en vim, como por ejemplo la gestión de buffers de memoria múltiples.

vi [opciones] Archivo [Archivo2 ...]

vi no tiene menús, tampoco interfaz gráfica, y no es intuitivo. Requiere conocerse de memoria un determinado número de atajos de teclado para poder utilizarlo. Si bien el aprendizaje es un poco difícil, una vez dominado, vi es rápido y práctico, hasta tal punto de ir más rápido que con editores de texto gráficos.



El debate que enfrenta a los partidarios de emacs (o de otros editores) con los de vi no tiene razón de ser. Cualquier sistema Linux (y Unix) dispone pase lo que pase del editor vi, por lo que es ineludible. Si tiene la oportunidad, puede instalar el editor que le guste, aunque esto, en entornos empresariales, de servidores, etc., no siempre será posible. Este debate genera muchos troles en los foros y tribunas.

2. Funcionamiento

Existen tres modos de funcionamiento:

- ✓ Modo **comando**: las inserciones representan comandos. Se accede a ellos al pulsar [Esc]. Cada tecla o combinación de teclas activa una acción (supresión de líneas, inserciones, mover, copiar, pegar, etc.).
- ✓ Modo **inserción**: se trata de la inserción de texto clásico.
- ✓ Modo **línea de comandos**: una línea en la parte inferior de la pantalla permite insertar comandos especiales, validada con la tecla [Entrar]. Se accede a ella al presionar, en modo comando, la tecla «:».

Cuando ejecuta vi, por defecto está en modo comando. Para empezar a introducir texto hay que teclear un comando de añadido o inserción, lo que enerva rápidamente cuando uno no conoce vi: a o i. Por lo tanto, el comando [Esc] i permite insertar texto. Para salir, puede pasar por el modo línea de comandos. Teclee [Esc]: y luego q y [Entrar]. ¡Es sencillo!



Si ha usado un comando ya se encuentra en modo comando y no tiene que pulsar la tecla [Esc] de nuevo para enviar el siguiente. Por ejemplo, la «x» suprime un carácter. Si ya está en modo comando, puede pulsar 10 veces la «x» sin darle a [Esc] entre cada pulsación.

Por supuesto, hay un punto de ironía. Nadie conoce todos los atajos; el propio autor de estas líneas solamente recuerda unos treinta. Por lo tanto, además de las guías en la prensa especializada o de los libros especializados (a veces de hasta 500 páginas), aquí tiene la lista de los comandos más utilizados en forma de tablas. Cuando el «:» precede al comando, éste se debe insertar en línea de comandos.

Si se pierde, en cualquier caso y pase lo que pase, al pulsar la tecla [Esc] vuelve siempre en modo comando.

3. Los comandos

a. La introducción de datos

Se deben efectuar las acciones siguientes en modo comando. Antes se debe pulsar [Esc]: [Esc] a, [Esc] i, etc.

Comando	Acción
a	Añadir después del carácter actual.
A	Añadir texto al final de línea.
i	Insertar delante del carácter actual, como en un procesador de texto.
I	Insertar texto al principio de línea.
o	Añadir una línea debajo de la línea actual.
O	Insertar una línea encima de la línea actual.

b. Salir y guardar

Recuerde que «:» significa que el comando se introduce en la línea de comandos: [Esc] :, inserción del comando, luego [Entrar].

Comando	Acción
ZZ	Guarda el archivo y sale.
:q!	Salir sin guardar.
:q	Salir si no se modificó el archivo (en caso contrario, aparece un mensaje de error).
:w	Guardar el archivo. Puede especificar un nombre para seguir.
:wq o :x	Guardar y salir.
1,10w fic	Guardar las líneas de 1 a 10 en fic.

c. Desplazamiento

Quizá la idea le haga sonreír, pero existen todavía teclados sin teclas direccionales, así como programas e intérpretes de comando que no saben interpretarlos (ejemplo del shell ksh mal configurado en algunos Unix). En estas máquinas, se activa el modo de inserción de vi por defecto e, incluso en vi, es posible arreglárselas sin las teclas de dirección, en modo comando.

Comando	Acción
h	Ir a la izquierda.
l (l minúscula)	Ir a la derecha.

k	Ir hacia arriba.
j	Ir hacia abajo.
0 (cero)	Principio de línea.
:0	Principio de archivo (primera línea).
\$	Fin de línea.
:\$	Fin de archivo (última línea).
w	Ir a la palabra siguiente.
b	Ir a la palabra anterior.
f<c>	Saltar al carácter <c> siguiente.
Control+f	Avanzar una pantalla (Forward).
Control+b	Retroceder una pantalla (Backward).
G	Última línea del archivo.
<n>G	Salta a la línea «n» (p. ej.: 10G va a la décima línea).
:<n>	Ídem (:10 va a la décima línea).

d. La corrección

Comando	Acción
x	Borrar el carácter bajo el cursor.
X	Borrar el carácter delante del cursor.
r<c>	Sustituir el carácter bajo el cursor por el carácter <c>.
dw	Borrar desde el cursor hasta el final de la palabra.
d\$ o D	Borrar desde el cursor hasta el final de la línea.
dO	Borrar desde el principio de línea hasta el cursor.
df<c>	Borrar todo hasta el carácter <c>.
dG	Borrar todo hasta la última línea, incluida la línea actual.
d1G	Borrar todo hasta la primera línea, incluida la línea actual.
dd	Borrar la línea actual.
u	Undo. Cancelar la última acción.

Se pueden repetir estos comandos. 5dd suprime 5 líneas; 4dw, 4 palabras; 5x, 5 caracteres, etc.

e. Búsqueda en el texto

A diferencia de un editor de texto clásico, vi puede buscar otras cosas además de palabras

sencillas, y funciona con la ayuda de caracteres especiales y criterios. El comando de búsqueda es el carácter /. La búsqueda arranca en el carácter actual hasta el final del archivo. El carácter ? efectúa la búsqueda en sentido inverso. Primero se indica el criterio y luego se pulsa [Entrar].

/echo

busca la cadena 'echo' a continuación en el archivo. Cuando se encuentra la cadena, el cursor se para en el primer carácter de esta cadena.

El comando n permite seguir la búsqueda en el sentido indicado al principio. El comando N efectúa la búsqueda en sentido inverso.

Algunos criterios

- ˘ /[CcBb]ola: Cola, cola, Bola, bola.
- ˘ /[A-Z]e: todo lo que empieza con una mayúscula con una e en segunda posición.
- ˘ /[A-Za-Z0-9]: todo lo que empieza por una mayúscula, una minúscula o una cifra.
- ˘ /^[^a-z]: intervalo negativo: todo lo que no empieza con una minúscula.
- ˘ /co.a: el punto sustituye un carácter: coma, cola, cota...
- ˘ /Au*o: la estrella es un carácter de repetición, de 0 a n caracteres, Auo, Auto, Automoto...
- ˘ /.*: la estrella delante del punto, una cadena cualquiera de tamaño variable; el «.» representa un carácter.
- ˘ /[A-Z][A-Z]*: repetición del motivo entre [] de 0 a n veces, búsqueda de una palabra que tiene al menos una mayúscula (al principio de palabra).
- ˘ /^Auto: el ^ indica que la cadena buscada deberá estar al principio de línea.
- ˘ /Auto\$: el \$ indica que la cadena buscada deberá estar al final de línea.

f. Comandos de sustitución

Para sustituir texto, hay que colocarse al principio de la cadena que desea modificar y teclear uno de los comandos siguientes. A continuación, teclee simplemente su texto. El principio es sencillo. Es idéntico al de la supresión: se suprime la parte indicada y vi pasa

al modo de inserción para la introducción de datos.

Comando	Acción
cw	Sustituir la palabra corriente.
c\$	Sustituir hasta el final de la línea.
c0 (cero)	Sustituir hasta el principio de la línea.
cf<x>	Sustituir hasta el próximo carácter <x>.
c/<rech>	Sustituir hasta la próxima coincidencia de la cadena <rech>.

g. Copiar - Pegar

El comando **v** permite una selección visual. Se subraya el texto y se puede desplazar el cursor para seleccionar el texto. Luego utilice uno de los comandos siguientes:

- ˆ el comando **d** corresponde a cortar (mover);
- ˆ la **c** hace casi lo mismo, pero **vi** se queda en modo edición;
- ˆ para pegar el texto en el lugar elegido, se utiliza el comando **p** (detrás del carácter) o **P** (delante del carácter). Si se ha copiado una línea entera, se colocará debajo de la línea activa.

Las acciones siguientes son posibles en modo comando:

- ˆ Para copiar una línea: **yy**.
- ˆ Para copiar cinco líneas: **5yy**.
- ˆ Para colocar las líneas copiadas en un sitio determinado: **p**.
- ˆ El editor **vi** dispone de 26 búferes de memoria para almacenar en ellos los datos que se pueden nombrar como se desee. Para ello, se utiliza el **"**.

- ~ Para copiar cinco palabras en la memoria m1: "m1y5w.
- ~ Para pegar el contenido de la memoria m1 en un lugar determinado: "m1p.

Otro truco, que podemos utilizar en un entorno gráfico, consiste en capturar el texto que se ha de copiar con el ratón, y luego pasar a modo inserción y hacer clic en el botón central.

h. Sustitución

La sustitución permite reemplazar de manera automática varias coincidencias por otra cadena.

:**[1ª línea, última línea]**s/Modelo/Sustitución/[gc]

Los números de líneas son opcionales. En este caso, la sustitución se hace en la línea corriente únicamente. Como sustitución de los números de línea, **.** determina la línea corriente, **1** la primera línea, **\$** la última línea.

El modelo es un criterio de búsqueda presentado en estas últimas páginas. Sustitución es una cadena cualquiera que sustituirá el modelo.

Por defecto sólo se sustituye la primera coincidencia. La letra **g** indica que hay que sustituir todas las coincidencias. Con **c**, **vi** requiere una confirmación para cada una de las coincidencias. El siguiente ejemplo sustituye todas las coincidencias de Unix o unix en UNIX.

:1,\$**s**/[Uu]nix/UNIX/g

i. Otros

Edición avanzada

Presentamos algunos comandos prácticos.

:r fic	Insertar el contenido de fic desde el lugar actual.
:! cmd	Ejecutar el comando. Pulse [Entrar] para volver a vi.
:r! cmd	Insertar el resultado del comando en el lugar actual.
:e fic	Cargar el archivo fic para edición.
:e#	Conmutar entre los diferentes archivos abiertos.

Comando set

El comando **set** permite configurar el editor y acceder a sus opciones:

- ˆ set all: muestra el conjunto de las opciones posibles.
- ˆ set number (o nu) / nonumber (o nonu): muestra/suprime los números de líneas.
- ˆ set autoindent / noautoindent: se conserva el sangrado durante un retorno a la línea.
- ˆ set showmatch / noshowmatch: durante la introducción de una llave o de una paréntesis de cierre, se visualiza el de apertura durante un instante muy corto, luego el editor vuelve al carácter corriente.
- ˆ set showmode / noshowmode: vi mostrará una línea de estado (INPUT MODE).
- ˆ set tabstop=x: define el número de caracteres para una tabulación.

Vim propone a menudo la coloración sintáctica. Detecta el tipo de archivo cargado y colorea las líneas, las contraseñas, etc. Puede desactivar la coloración:

- ˆ syntax off para desactivar;
- ˆ syntax on para activar.