

El editor VI

Introducción

El editor VI es un editor de consola utilizado por la mayoría de los usuarios UNIX. Este editor tiene características extremadamente potentes para facilitar la tarea de los programadores, pero para muchos usuarios novatos se vuelve caótico acostumbrarse a sus funciones por sus diferencias con otros editores. El propósito de este documento es ayudar a los usuarios novatos a acostumbrarse al uso del VI, aunque algunas secciones podrán ser de ayuda para los usuarios cotidianos de VI.

A lo largo del texto se mostrarán ejemplos dado que la mejor forma de aprender el manejo de este editor es probando los ejemplos presentados y también pensar en ejemplos propios.

Iniciando el programa

El editor VI permite a un usuario crear nuevos archivos o editar archivos existentes. El comando para iniciar al VI es `vi`, ocasionalmente seguido del nombre del archivo a editar o crear. Por ejemplo para editar un archivo llamado `prueba` deberíamos ejecutar

```
vi prueba
```

Si iniciamos VI sin el nombre de archivo, entonces cuando deseemos guardar nuestro trabajo debemos indicarle al programa en qué nombre de archivo guardarlo.

Cuando el editor se inicia por primera vez, debería mostrar la pantalla llena de tildes (el tilde se visualiza así: `~`) sobre la izquierda de la misma. Cualquier línea en blanco más allá del final de archivo se muestra de esta forma. En la parte inferior de la pantalla se muestra el nombre de archivo, si es que se especificó un archivo existente, y el tamaño del mismo:

```
"nombre_archivo" 21 lines, 385 characters
```

Si el archivo especificado no existe, entonces se indicará que se trata de un nuevo archivo

```
"nombre_archivo" [New File]
```

Saliendo del programa

Ahora que sabemos como ingresar al VI, sería una buena idea saber como salir de él. El editor VI tiene **dos modos** y para poder salir de VI, debemos estar en el **modo comando**. Para pasar a este modo es necesario presionar la tecla **Escape**. Si el programa ya estaba en este modo cuando presionamos **Escape**, no hay problema, a lo sumo el sistema emitirá un beep.

El comando para salir del VI es `:q`. Una vez en modo **comando**, escribimos dos puntos, seguido de una `q` y presionamos **Enter**. Si el archivo ha sido modificado, el editor advertirá de este hecho al intentar salir sin grabar, impidiéndonos salir. Para ignorar este mensaje y salir del VI sin guardar los cambios, utilizamos el comando `:q!`.

Si queremos guardar los cambios debemos utilizar el comando `:w`. El comando de salida puede combinarse con el de guardado de la siguiente forma `:wq`. Es posible **guardar**

como" en VI, es decir, guardar con un nombre diferente un archivo editado. Para ello se debe anexar al comando **:w** el nombre del archivo. Entonces un ejemplo de ello sería **:w archivo2**.

Otra forma de guardar y salir de VI es con el comando **zz** (dos Z mayúsculas).

Los dos modos de vi

Lo primero que debe aprender un usuario es que el editor VI utiliza dos modos: comando e inserción. El modo comando permite el ingreso de comandos para manipular el texto. Estos comandos generalmente son de uno o dos caracteres de largo, y pueden ser ingresados presionando unas pocas teclas. El modo inserción pone todo lo que se tipea en el teclado dentro del archivo que se está editando, es decir es el modo en el que podemos escribir.

VI se inicia en modo comando. Existen varios comandos que ponen al editor en modo inserción. Los comandos más utilizados para llevar al VI a modo inserción son el comando **i** y **a**. Una vez en modo inserción, es posible salir al modo comando presionando la tecla **Escape**. Muchas veces es conveniente presionar más de una vez esta tecla hasta oír un beep, ya que de esta forma el sistema nos está avisando que estamos en modo comando.

Cómo tipear comandos en modo comando

El formato general que admiten los comandos del modo comando de VI es:

[**cant idad**] comando [**dónde**]

donde los corchetes indican que los parámetros son opcionales. El parámetro **cantidad** debe ser ingresado como un número comenzando con cualquier carácter entre 1 y 9. Por ejemplo, el comando **x** elimina el carácter debajo del cursor. Entonces tipeando **23x** estando en modo comando, se eliminarán 23 caracteres comenzando por el carácter debajo del cursor.

Algunos comandos utilizan el parámetro opcional **donde** para especificar por ejemplo cuantas líneas o a qué porción del documento el comando afectará. El parámetro **donde** también puede ser cualquier comando que mueva el cursor.

Algunos ejemplos simples de comandos VI

A continuación damos un conjunto de comandos que sirven de punto de partida para los usuarios novatos a VI.

Comando	Descripción
a	Ingresa al modo inserción. Los caracteres tipeados se insertarán detrás de la posición actual del cursor
h	mueve el cursor un carácter a la izquierda
i	Ingresa al modo inserción. Los caracteres tipeados se insertarán delante de la posición actual del cursor
j	mueve el cursor un carácter hacia abajo
k	mueve el cursor un carácter hacia arriba
l	mueve el cursor un carácter hacia la derecha
r	reemplaza un carácter, en particular el que está debajo del cursor
u	Deshace el último cambio sobre el archivo. Topeando nuevamente u se rehace el cambio
x	elimina un carácter, en particular el que está debajo del cursor

Buffers de texto en VI

El editor VI utiliza 36 buffers para almacenar pedazos de texto, además de un buffer de propósito general. Cada vez que una porción de texto es eliminada o copiada (el termino para copiada utilizado por vi es yanked) del archivo, es ubicada en un buffer de propósito general. La mayoría de los usuarios de VI raramente utilizan el resto de los buffers, ya que es posible arreglárselas con solo este buffer. Los bloques de texto también pueden ubicarse en otros buffers si es que se especifica. Para especificar un buffer se utiliza el comando " (comilla doble). Por ejemplo el comando "mdd utiliza el buffer **m**, y los dos siguientes caracteres indican borrar la línea actual, es decir, borra la línea actual posicionándola en el buffer general y en el buffer **m**.

Luego el texto ubicado en un buffer puede ser **pegado** con el comando **p** o **P**. Por ejemplo el comando "mp **pega** el contenido del buffer **m** delante de la posición actual del cursor.

Cortando y Copiando

El comando usado para cortar es **d**. Este comando borra texto del archivo. El comando es precedido opcionalmente por una **cantidad** y seguido por una indicación de movimiento. A continuación se muestran algunas combinaciones:

Comando	Descripción
dd	elimina la línea actual
d^	elimina desde la posición actual del cursor hasta el inicio de la línea
d\$	elimina desde la posición actual del cursor hasta el fin de la línea
dw	elimina desde la posición actual del cursor hasta el final de la palabra
3dd	elimina tres líneas desde la posición actual del cursor hacia delante

El comando **y** funciona como el comando **d** en el sentido que copia al buffer general, o también a otro si es que se lo especifica, pero se diferencian en que el comando **y** no borra el texto del archivo.

Pegando el texto de los buffers

Los comandos para **pegar** son **p** y **P**. la única diferencia entre ellos es la forma en que pegan el texto en relación a la posición del cursor. **p** pega el texto delante del cursor, mientras que **P** lo hace detrás del cursor. Especificando una **cantidad** es posible pegar más de una vez el buffer.

Utilidades para los programadores: indentación y balanceo

La indentación es una buena costumbre de los programadores organizados. Para ello VI provee dos comandos.

Comando	Descripción
>>	desplaza la línea actual hacia la derecha según indica shift width
<<	desplaza la línea actual hacia la izquierda según indica shift width

Respecto al balanceo de paréntesis y llaves utilizadas en estructuras de control, funciones, etc. es muy común confundirse al escribir un paréntesis de menos o de más. El comando % posiciona el cursor en el paréntesis que abre y cierra algún bloque, y en caso de no existir lo indica con un beep.

Búsquedas en VI

Para la búsqueda de cadena de caracteres, se utilizan los comandos / y ?. Cuando se tipea alguno de los dos, inmediatamente se muestra en la última línea de la pantalla la cadena que ingresamos para buscar. Estos dos comandos difieren en la dirección en la que buscan. El comando / busca hacia delante, es decir, desde el cursor hacia el final del archivo, mientras que el comando ? busca hacia atrás. Los comandos n y N repiten la última búsqueda en la misma dirección u opuesta, respectivamente. Algunos caracteres tienen significado especial en VI y por ello deben **escapearse** utilizando la barra invertida (\) con el fin de considerar dichos caracteres en la búsqueda. Los caracteres especiales son:

^	inicio de línea (al inicio de una expresión de búsqueda)
.	machea un carácter cualquiera
*	machea con cero o más del carácter anterior
\$	fin de línea (al final de la expresión de búsqueda)
[]	comienza un conjunto de expresiones de mach o no. Por ejemplo: /f[aei]n machea fan, fen y fin. Otro ejemplo: /a[^bcd] machea con a y a seguido de cualquier letra diferente de b, c o d.
<>	puesto en una expresión escapeado con barra invertida sirve para encontrar el comienzo o fin de una palabra. Por ejemplo /\<disc> debería machear disc pero no disco o minidisc

Ejercicios

- 1) Genere un fichero llamado **fich-1**. Introduzca 10 líneas de texto.
 - a) Sitúese al final del texto.
`<esc> L`
 - b) Colóquese al principio del texto
`<esc> H`
 - c) Sitúese al final de la quinta línea
`<esc> 5 $`
 - d) Vuelva al principio de la misma línea.
`<esc> 0`
 - e) En la situación actual, muévase dos palabras hacia la derecha. Vuelva ahora a la posición anterior.
`<esc> 2 W`
`<esc> 2 B`
 - f) Guarde los cambios y salga

- 2) Abra el archivo **fich-1**
 - a) Inserte una nueva línea delante de la primera.
`<esc> H`
`<esc> :i`
`[línea]`
 - b) Sitúese en la segunda palabra de la tercera línea e inserte una nueva palabra delante de ella.
`<esc> 3 G`
`<esc> 2 W`
`<esc> i`
 - c) Añada texto al final de la línea actual.
`<esc> $`
`<esc> a`
 - d) Borre los cinco primeros caracteres de la línea actual.
`<esc> 0`
`<esc> 5 x`
 - e) Elimine la línea actual y la siguiente utilizando una sola orden.
`<esc> 2 S`
 - f) Sitúese en la segunda palabra de la línea actual y borre el resto de la línea.
`<esc> 0`
`<esc> 2 W`
`<esc> D`
 - g) Guarde y salga del editor

- 3) Pruebe las siguientes instrucciones determinando qué es lo que hace cada una
 - a) `<esc> : set nu`
 - b) Ingrese algún texto de tal forma de exceder el ancho de pantalla. Luego escriba el siguiente comando `<esc> : set wm=65`

- 4) Vuelva a abrir el archivo "fich-1"
 - a) Copie las líneas desde la tercera a la quinta (incluida) y sitúe la copia a partir de la séptima línea.
`<esc> H`
`<esc> 3 G`
`<esc> 3 Y`
`<esc> H`
`<esc> 7G`
`<esc> P`
 - b) Mueva las dos últimas líneas al principio del texto.
`<esc> L`

```
<esc> k  
<esc> 2Y  
<esc> H  
<esc> P
```

c) Añada un fichero de texto que ya exista, al final del fichero editado.

```
<esc> L  
<esc> $  
<esc> :r [fichero]
```

5) Crear un archivo llamado **numero.linea** que contenga 50 líneas cada una con el texto "línea n". Ejemplo:

```
línea 1  
línea 2  
...  
línea 49  
línea 50
```

a) Mover el cursor a:

- la primera línea del archivo
- la última línea del archivo
- línea 15
- desplazarse de a un carácter en todos los sentidos

b) Borrar la palabra línea del renglón 10

c) Insertar la palabra número entre línea y 15

d) Borrar "ero" de la palabra número insertada en c)

e) Borrar las líneas 3,4 y 5. Reinsertarlas luego de la línea 12.

f) Copiar la línea 9 después de la línea 2

g) Mover la línea 10 luego de la 13.

h) Grabar el archivo y salir.

6) Ingresar el siguiente texto en el archivo **trabalenguas: dicho he dicho**

a) Insertar al principio de la línea: *Un*

b) Agregar al final de la línea: *yo*

c) Posicionarse al final de la primera aparición de la palabra "*dicho*" y agregar: *que*

d) Ingresar la palabra: "*Maxima:*" en la línea anterior a la ingresada anteriormente.

e) Unir ambas líneas.

f) Grabar el archivo y salir.

Resumen de comandos

Movimientos básicos:

h	Un espacio a la izquierda.
j	Una línea hacia abajo.
k	Una línea hacia arriba.
l	Un espacio a la derecha.
\$	Mueve al final de la línea actual.
+	Comienzo de la línea siguiente.
-	Comienzo de la línea anterior.
0	Comienzo de la línea actual.

Control de despliegue:

CTRL-d	Deslizar hacia adelante.
CTRL-u	Deslizar hacia atrás.
CTRL-f	Pantalla siguiente.
CTRL-b	Pantalla anterior
CTRL-l	Redibuja la pantalla.
G	Moverse al final del archivo.

Comandos para agregar textos:

a	Después del cursor.
A	A final de la línea.
i	Antes del cursor.
I	Al principio de la línea.
O	Abre una línea por encima.
o	Abre una línea por debajo.

Comandos para eliminar y cambiar texto:

dd	Elimina línea.
cc	Cambiar línea.
D	Elimina hasta el final de línea.
C	Cambiar hasta final de línea
x	Elimina el carácter del cursor.
r	Cambia el carácter del cursor.
J	Une línea actual con siguiente

Comandos sobre palabras:

cw	Cambiar palabra.
dw	Elimina palabra.
e	Mueve cursor a final de palabra.
w	Mueve cursor a siguiente palabra.

Comandos de búsqueda:

/	Búsqueda hacia delante.
?	Búsqueda hacia atrás.
n	Siguiente ocurrencia.
N	Posterior ocurrencia

Comandos de edición avanzada:

u	Deshacer cambio más reciente.
p	Coloca el contenido del buffer a la derecha del cursor.
P	Coloca el contenido del buffer a la izquierda del cursor.
y	Copia los caracteres comenzando en el cursor en el buffer.
Y	Copia la línea actual completa en el buffer.

Comandos de control:

:w	Guarda sin salir.
:wq	Guarda y sale.
:q	Abandona si no hay modificaciones.
:q!	Sale sin guardar los cambios.
:número	Ir a la línea número.
ZZ	Guarda y sale.