

Capítulo 7

Editor vi y vim

El editor **vi** es un estándar de UNIX, el **vi** se basa en ordenes, al contrario que la mayoría de los otros editores. El editor **vim** incorpora el manejo del mouse, sintaxis. El nombre **Vim** es una contracción de <Vi iMproved>, se podría traducir como **vi** mejorado. Estos editores son extremadamente eficientes y posibilita el máximo rendimiento con el mínimo esfuerzo. Lleva cierto tiempo aprender las funcionalidades del editor, pero una vez que uno aprendió veremos todo el potencial que tiene.

También tenemos el paquete **vim-gnome** que es el **vim** pero para entorno llamado **gvim**.

Tenemos un archivo de configuración general dentro de **/etc/vim/vimrc**

Hay que distinguir entre dos modos :

* **Modo inserción** : cualquier cosa que tecleemos aparecerá en pantalla y se grabará en el fichero si finalmente salvamos los cambios. Para salir de este modo y volver al modo órdenes, tenemos que pulsar la tecla **ESC**. Esta tecla cancela las acciones a medidas que tengamos. Es una buena costumbre pulsar ESC siempre justo después de añadir cualquier texto

* **Modo órdenes** : el **vi** estará esperando a que tecleemos órdenes, para interpretarlas y actuar en consecuencia.

Comenzando

```
# vim [opciones] [ficheros]
```

Opciones :

-x	<i>Permite encriptar un archivo :</i>
-n	<i>Va a la n linea.</i>
+/BUSCAR	<i>Editar el archivo ubicándose donde encuentre en la primer linea BUSCAR</i>

```
# vim -x mi_archivo.txt  
# vi mi_archivo.txt
```

* Si no se indica ningún archivo, **vi** o **vim** empezará con un buffer vacío.

* Si se indica un solo fichero , **vi** o **vim** lo leerá y lo cargará en un buffer que será el que muestre su pantalla al iniciarse.

* Si se indican varios ficheros, hay que separarlos por espacios en blanco. **Vim** asignará un buffer distinto a cada uno de ellos y al empezar mostrará en pantalla el buffer correspondiente al primer fichero. Para navegar entre los buffers hay que usar el comando **:next** y **:prev**

Si nosotros solamente escribimos **vim** nos mostrara una introducción, si nosotros especificamos un archivo por ejemplo : **vim mi_archivo** no nos aparece mas la introducción, para ver nuevamente la introducción ponemos :**intro**

Movimiento del cursor

k	<i>Arriba.</i>
j	<i>Abajo.</i>
h	<i>Derecha.</i>
l	<i>Izquierda.</i>
0	<i>Inicio de línea.</i>
\$	<i>Va al final de la línea.</i>
w	<i>Avanza por palabra (word).</i>
b	<i>Retrocede por palabra (back).</i>
e	<i>Final de la palabra (end).</i>
H	<i>Esquina sup. izq. de la ventana (Home).</i>
L	<i>Esquina inf. izq. de la ventana (Last).</i>
ctrl+u	<i>Subir ventana (Window up).</i>
ctrl+d	<i>Bajar ventana (Window down).</i>
ctrl+b	<i>Retroceder página (Page back).</i>
ctrl+f	<i>Avanzar página (Page forward).</i>
nG	<i>Salta a la línea n (Go).</i>
1G	<i>Va a la primera línea.</i>
\$G	<i>Va a la última línea.</i>
fc	<i>Buscar en la línea el carácter c (hacia delante).</i>
Fc	<i>Buscar en la línea el carácter c (hacia atrás).</i>

Insertar texto

i	<i>Insertar (delante del cursor).</i>
I	<i>Insertar al principio de la línea.</i>
a	<i>Añadir (detrás del cursor).</i>
A	<i>Añadir al final de la línea.</i>
o	<i>Insertar una línea debajo de la actual.</i>
O	<i>Insertar una línea encima de la actual.</i>

Borrar texto

x	<i>Borrar carácter actual.</i>
X	<i>Borrar carácter anterior.</i>
dd	<i>Borrar línea actual.</i>
D	<i>Borrar hasta final de línea.</i>
dw	<i>Borrar palabra.</i>

Cambiar carácter

rcar	<i>Reemplazar el carácter actual por car.</i>
R	<i>Reemplazar texto desde la posición del cursor.</i>
s	<i>Substituir el carácter actual por texto a insertar.</i>
S	<i>Substituir la línea actual.</i>
C	<i>Cambiar hasta el final de la línea.</i>

cw	<i>Cambiar palabra.</i>
J	<i>Unir a la línea actual la siguiente.</i>

Copiar y pegar

yy	<i>Copiar en el buffer la línea actual.</i>
nny	<i>Copiar en el buffer n líneas desde la actual.</i>
p	<i>Pega el buffer detrás del cursor.</i>
P	<i>Pega el buffer delante del cursor.</i>

Buscar y sustituir

%	<i>Busca el caracter delimitador () [] { } que balancea el actual (Dentro de un entorno salta al delimitador inicial).</i>
/ExpReg	<i>Busca hacia delante la expresión regular ExpReg.</i>
?ExpReg	<i>Busca hacia atrás la expresión regular ExpReg.</i>
n	<i>Repite la última búsqueda.</i>
N	<i>Repite la última búsqueda en el sentido contrario.</i>
:s/txt/txt2	<i>Substituye el texto txt por txt2 la primera vez que aparece en la línea.</i>
:s/txt/txt2/g	<i>Substituye todas las apariciones de txt por txt2 en la línea.</i>
:m,n s/txt/txt2/g	<i>Substituye en el rango de líneas [m,n].</i>

Repetir y deshacer

.	<i>Repetir último comando de actualización (Borrado/Inserción/Cambio).</i>
u	<i>Deshacer último comando de actualización.</i>
U	<i>Deshacer todos los cambios en la línea actual.</i>

Comandos de Shell

:sh	<i>Invoca un nuevo shell. Al salir continua la edición.</i>
!:CmdShell	<i>Ejecuta un comando del sistema operativo.</i>
:r!CmdShell	<i>Ejecuta un comando del S.O. e inserta su salida en la posición del cursor.</i>
!!:	<i>Ejecuta el ultimo comando.</i>

Operaciones con archivos

:w	<i>Graba las modificaciones efectuadas en el archivo.</i>
:q	<i>Salir (si no hay cambios).</i>
:q!	<i>Salir (sin grabar si hay cambios).</i>
:wq	<i>Guardar cambios y salir.</i>
:x	<i>Guardar cambios y salir.</i>
ZZ	<i>Guardar cambios y salir.</i>

Estadísticas de archivos

:=	<i>Muestra el número total de líneas del archivo.</i>
:.=	<i>Muestra el número de línea actual.</i>
ctrl+G	<i>Muestra el nombre del archivo, línea actual, número total de</i>

líneas y porcentaje recorrido del archivo.

Opciones de entorno

:set opción	<i>Activa la opción de vi correspondiente.</i>
:set noopción	<i>Desactiva la opción de vi correspondiente.</i>
:Texplore	<i>Abre un explorador para moverse en los directorios y abrir un archivo en una nueva columna.</i>
:set ic o :set ignorecase	<i>Ignore Case (ignora entre mayúscula y minúsculas en las búsqueda).</i>
:set noic o :set noignorecase	<i>Realiza lo contrario, diferencia entre minúsculas y mayúsculas.</i>
:set mouse=a	<i>Habilita el mouse para todas las tareas.</i>
:set number o :set num	<i>Muestra el número de línea.</i>
:set ai	
:set hlsearch	<i>Pinta todas las coincidencia que buscamos.</i>
:set nohlsearch	<i>Lo contrario.</i>

Opciones de entorno opción :

all	<i>Muestra todas las opciones y sus valores.</i>
number	<i>Muestra numeración de líneas.</i>
list	<i>Muestra caracteres de control.</i>
ic	<i>Ignora mayúsculas/minúsculas en las búsqueda.</i>

Para utilizar las sintaxis tenemos que instalar el siguiente paquete :

```
# apt-get install vim-puppet
```

Y para utilizarlo **:syntax on**

En el *home* del usuario tenemos nuestra propia configuración de **vim** llamado **.vimrc**

```
# vi .vimrc

syntax on
set tabstop=3
set textwidth=98
set hlsearch
set shiftwidth=4 softtabstop=4
set expandtab

map <F3> :call CompileGcc()<CR>
func! CompileGcc()
  exec "w"
  exec "!gcc % -o %<"
endfunc
```

También como vimos en el ejemplo anterior podemos mapear teclas.

Para abrir una nueva pestaña o solapa deberemos usar el siguiente comando

:tabnew [archivo]

esto creara la pestaña y el contenido del archivo definido **[archivo]** en ella.

Para abrir múltiples pestañas con múltiples archivos al iniciar el editor de textos emplearemos el comando

vim -p archivo1 archivo2 archivo3

Esto abrirá el archivo 1 en la primera pestaña, el archivo 2 en la segunda y así sucesivamente, por defecto **Vim** soporta hasta 10 pestañas en simultaneo, pero esto lo podremos modificar editando el archivo

.vimrc

que esta en nuestro directorio principal y agregando la siguiente entrada

set tabpagemax=[valor]

donde en valor colocaremos la cantidad de pestañas que queremos que tenga como máximo. Para movernos entre pestañas utilizaremos los comandos

:tabn y :tabp

Con el primero avanzaremos hacia la siguiente pestaña y con la siguiente retrocederemos pestañas, de todas formas podremos elegir mediante un Click del Mouse al igual que lo hacemos en el Firefox.