



# Curso Operador Linux

## Módulo 7

### USO DEL EDITOR VI



## **Presentación**

En esta unidad se introducirá a los participantes en el uso de los editores de texto más comunes. Se utilizarán los programas disponibles en modo consola



## **Objetivos**

Los participantes al finalizar la Unidad:

- Estarán familiarizados con la edición de texto
- Podrán utilizar en editor vi y su versión mejorada vim
- Conocerán editores de texto alternativos



## **Temario**

- 7.1 Introducción al editor vi
- 7.2 Como usar el editor vi
- 7.3 Otros editores



## **Actividad de aplicación del conocimiento**

Los participantes encontraran la actividad de aplicación en un archivo por separado en caso de ser requerida.



## **Examen**

Los participantes deberán rendir el examen online o presentar el material solicitado según corresponda.



## 7.1 Introducción al editor vi

La necesidad de modificar archivos de configuración hace que necesitemos utilizar editores de texto. Estos son programas que nos permiten crear y/o modificar archivos son denominados editores.

El vi es un editor muy poderoso, a pesar de ser muy poco amigable con el usuario. Es el único editor que siempre vamos a encontrar en cualquier distribución de Linux (así como en cualquier sistema Unix), es por ello que es de suma importancia aprender su uso.

Tiene dos modos, **modo edición** y **modo comando** . Se entiende estar en modo comando el momento en el cual se esta navegando dentro de los menús de cualquier editor tradicional de caracteres.

Cualquiera que haya utilizado un editor tradicional de caracteres sabrá que mientras se navega por los menús no se puede seguir escribiendo en el texto editado. El modo comando del vi tiene esa semejanza, mientras se este en el modo comando, no se podrá escribir sobre el texto, solo podremos darle ordenes al editor para que ejecute una u otra función predeterminada.

Para ejecutar las órdenes que se le da al editor en el modo comando se utilizan letras y combinaciones de ellas.

El modo edición por el contrario es el modo en el que podremos introducir texto al editor

## 7.2 Como usar el editor vi

### Modo Comando

Entraremos al modo comando tipeando en el shell cualquiera de las siguientes sentencias

```
# vi  
# vi [nombre de archivo existente]  
# vi [nombre de archivo a crear]
```



La siguiente es una tabla que detalla algunos de los comandos mas utilizados:

Comando	Descripción
a	Entra al modo edición, una posición después de la ubicación del cursor (diminutivo de "append")
A	Entra al modo edición, al final de la línea en la que estamos ubicados
i	Entra al modo edición, exactamente en la posición actual del cursor
l	Entra al modo edición, al principio de la línea
x	Borra el carácter que está debajo del cursor
X	Borra el carácter que está detrás del cursor
yy	Copia una línea entera en el buffer (copy)
p	Pega lo que tenga en el buffer en donde esta ubicado el cursor (paste)
dd	Borra el contenido de una línea y lo copia al buffer (cut)
Flechas	Movimiento dentro del archivo
h	Mueve el cursor hacia la izquierda
l	Mueve el cursor hacia la derecha
k	Mueve el cursor hacia arriba
j	Mueve el cursor hacia abajo
.	Repite la última función
cw	Nos permite reeditar caracteres hasta que se encuentre un ":" una "," un "." o un espacio
cW	Nos permite reeditar caracteres hasta que se encuentre un espacio (sin importar que caracteres haya hasta el espacio)
r	Reemplaza en carácter debajo del cursor
R	Reemplaza todos los caracteres hasta el final de la línea o finalización del modo "reemplazo"
ZZ	Grabar el archivo y salir del editor
u	Revertir el último comando ejecutado (o edición agregada)
G	Ir al final del archivo
número <comando>	El número representa la cantidad de líneas, el comando puede ser cualquiera de los descriptos que se aplique. Por ejemplo subir 10 líneas, podemos ejecutar "10k". Otro ejemplo sería, copiar 20 líneas, lo haríamos con "20yy"
o	Agrega una línea debajo de la que estamos ubicados y cambia a modo edición
O	Agrega una línea arriba de la que estamos ubicados y cambia a modo edición
Enter	
(	Mueve el cursor al principio de la frase
)	Mueve el cursor al fin de la frase
{	Mueve el cursor al principio del párrafo
}	Mueve el cursor al fin del párrafo



### Comandos combinados

Además de los comandos citados en la tabla anterior, existen muchos otros comandos o funciones que se pueden ejecutar desde el vi utilizando el comando simple «:». Cuando se tecléa este símbolo, el editor lo lleva a la última línea de la pantalla donde podrá ejecutar diferentes funciones.

La siguiente es una tabla con las funciones más utilizadas, todas estas funciones se pueden ejecutar, únicamente, después de teclear los ":".

Es importante destacar que se pueden combinar estas funciones, por ejemplo, podríamos ejecutar «:wq» grabando el trabajo y saliendo del editor.

Función	Descripción
w [nombre de archivo]	Grabar archivo (si le agregamos el símbolo de admiración "!", se fuerza la grabación).
q	Salir del editor (si le agregamos el símbolo de admiración "!", se sale del editor sin grabar).
n [nombre de archivo]	Editar el archivo pasado como argumento, manteniendo el anterior en memoria.
n#	Volver al archivo anterior.
r<nombre de archivo>	Insertar el archivo, pasado como argumento, dentro del archivo actual.
!<comando>	Ejecutar "<comando>" (del Shell) y mostrar su salida.
!!<comando>	Ejecutar "<comando>" (del Shell) e insertar la salida del mismo dentro del archivo en edición.
x	Grabar y salir del editor (sinónimo de ":wq")
sh	Abrir una pseudo terminal (se sale con exit)
<comando vi>!	Forzar el comando, por ejemplo :q! forzar la salida

### Modo Edición

Para ingresar al modo edición se pueden utilizar varios comandos, la mayoría de ellos están descriptos en la tabla de comandos simples (ej.: «a», «i», «A», «I»).

Existe una única tecla para salir del modo edición y es la tecla escape <Esc>.

Una vez que se ingresó al modo edición, simplemente tendrá que escribir el texto deseado



## 7.3 Otros editores

### nano

Se encuentra en muchas distribuciones, especialmente en las derivadas de Debian. Funciona como cualquier editor pero su particularidad es que los comandos se llaman mediante la tecla Ctrl,

```
GNU nano 0.9.9pre3 File: nano.c
$Id: nano.c,v 1.136 2001/02/16 04:48:38 astjanax Exp $ */
*****
* nano.c
*
* Copyright (C) 1999 Chris Allegretta
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 1, or (at your option)
* any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
*
[ Read 2583 lines ]
Get Help WriteOut Replace Prev Page Cut Text Cur Pos
Exit Read File Where Is Next Page UnCut Text To Spell
```

### mcedit

Encontrada especialmente en las derivaciones de Red Hat. Los comandos se invocan mediante las teclas de función (F1, F2, etc)

```
logfile.list [----] 0 L:[ 1+ 0 1/ 29 ]*(0 /1172b)= # 35 0x23
# A tab separated list of service directories and their log files.
SMTP Server /var/log/qmail/smtpd/current
Mail Server /var/log/qmail/current
IMAP Server /var/log/courier/imapd/current
POP3 Server /var/log/courier/pop3d/current
Fetchmail /var/log/fetchmail/current
Spam Control /var/log/spamd/current
AntiVirus /var/log/clamav/current
LDAP Server /var/log/slapd/current
NSCD Service /var/log/nscd/current
DNS Caching Server /var/log/dnscache/current
NetBIOS Name Server /var/log/samba/nmbd/current
Windows File Sharing /var/log/samba/smbd/current
SSH Service /var/log/sshd/current
FTP Server /var/log/proftpd/main/current
Apache Web Server /var/log/apache2/main/current
System Scheduler /root/.uschedule/log/current
Easypush Access /var/log/apache2/easypush-access.log
Easypush Error /var/log/apache2/easypush-error.log
Webmail Access /var/log/apache2/access.log
Webmail SSL Access /var/log/apache2/ssl-access.log
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```



Linux ofrece muchas más alternativas en cuanto a editores de texto, se pueden citar

gim: poderoso y pequeño editor patrimonio del BSD.

emacs: el editor definitivo patrimonio del GNU

xemacs: Emacs: La Próxima Generación, originaria de Lucid

mcedit: editor GNU para principiantes. Idéntico al editor interno del mc.

ae: pequeño editor que trae por defecto Potato. Evítelo.

nano: pequeño editor GNU que trae por defecto Woody. Emula a pico.

joe: para los nostálgicos usuarios del WordStar o TurboPascal.

jed: editor rápido, con múltiples funciones, basado en menús y con las combinaciones de teclas de emacs.

jove: editor muy pequeño con las combinaciones de teclas de emacs.

nvi: el nuevo vi. Totalmente compatible con el vi original.

Editores de texto en X

gvim: Vim con GUI (paquetes vim y vim-gtk)

emacs: el Único y Verdadero Emacs (autodetecta X).

xemacs: Emacs de la próxima generación (autodetecta X)