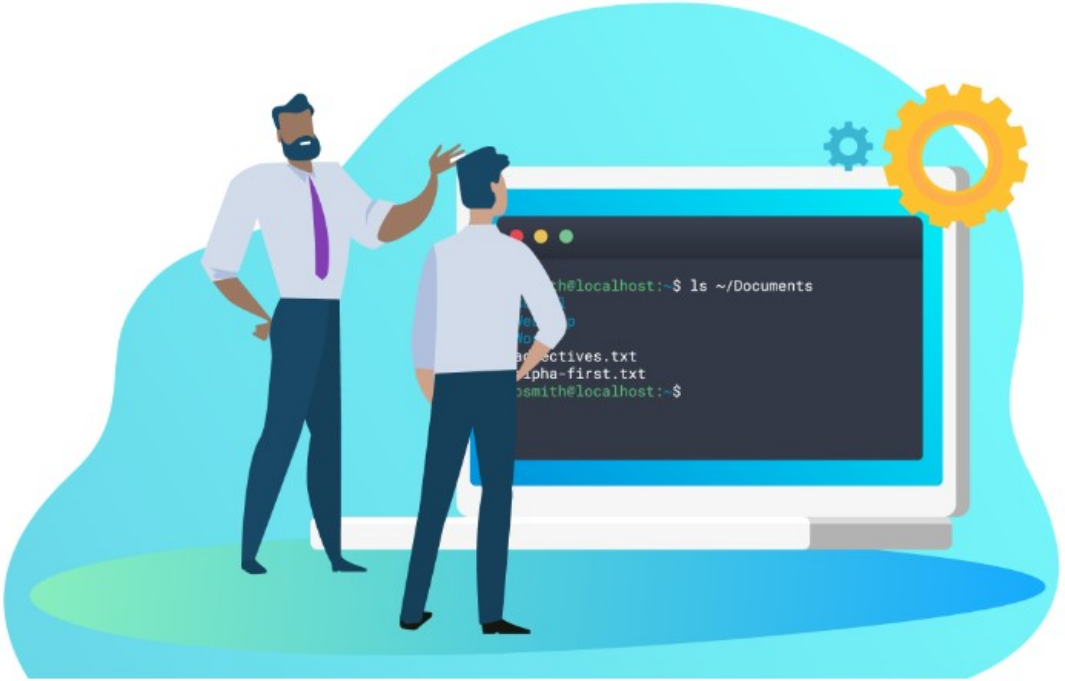

Editor de texto

El principal editor de texto para Linux y UNIX es un programa llamado `vi`. Si bien hay numerosos editores disponibles para Linux incluyendo desde el pequeño editor `nano` hasta el editor masivo `emacs`, el editor `vi` tiene varias ventajas:

- El editor `vi` está disponible en todas las distribuciones Linux del mundo. Esto no ocurre con ningún otro editor.

- El editor `vi` se puede ejecutar tanto en una CLI (interfaz de línea de comandos) como en una GUI (interfaz gráfica de usuario).
- Aunque se han añadido nuevas características al editor `vi`, las funciones principales han existido durante décadas. Esto significa que si alguien aprendió a usar el editor `vi` en la década de 1970, podrá usar la versión moderna sin ningún problema. Aunque eso pueda parecer trivial, puede que dentro de veinte años no sea tan trivial.

> **A tener en cuenta**
>
> La forma correcta de pronunciar el nombre del *editor vi* es *vi-ay*. Las letras vi representan visual, pero nunca se pronunció de esta manera. En su lugar se pronuncia la letra *v (vi)* seguida de la letra *i (ay)* en inglés.

En realidad, la mayoría de los sistemas Linux no incluyen el editor `vi` original, sino una versión mejorada del mismo conocida como `vim` (vi mejorada) (*vi improved*). Este hecho puede estar oculto en la mayoría de las distribuciones de Linux. En su mayor parte, `vim` funciona igual que `vi`, pero presenta funciones adicionales. Para los temas que se tratan en este curso, tanto `vi` como `vim` funcionarán perfectamente

Para comenzar a usar `vi`, simplemente escriba el comando seguido del nombre de ruta del archivo que quiere editar o crear:

```
sysadmin@localhost:~$ vi newfile.txt
```

Los tres modos utilizados en `vi` son los siguientes: modo de comando, modo de inserción y modo ex.

Modo de comando: Movimiento

Inicialmente, el programa empieza en modo de comando. El modo de comando se utiliza para escribir comandos, como los utilizados para desplazarse por un documento, manipular texto o acceder a los otros dos modos. Para volver al modo de comando en cualquier momento, presione la tecla **Esc**.

Una vez haya agregado texto a un documento, deberá presionar la tecla **Esc** para volver al modo de comando y realizar acciones como mover el cursor. Esto parece que sea mucho trabajo, pero recuerde que `vi` funciona en un entorno terminal en el cual un mouse es inservible.

Los comandos de movimiento en `vi` tienen dos aspectos, el movimiento (*motion*) y un prefijo numérico opcional (*count*) que indica cuántas veces se debe repetir ese movimiento. El formato general es el siguiente:

```
[número] movimento
```

En la siguiente tabla se resumen las teclas de movimiento disponibles:

Movimiento	Resultado
`h`	Un carácter a la izquierda
`j`	A la línea siguiente
`k`	A la línea anterior
`l`	Un carácter a la derecha
`w`	Una palabra adelante
`b`	Una palabra hacia atrás
`^`	Al principio de la línea
`$`	Al final de la línea

Nota

En la actualización `vim` también es posible usar las teclas de flecha `←` `↓` `↑` `→` en lugar de los caracteres `h` `j` `k` `l` respectivamente.

Estos movimientos se pueden anteponer con un número para indicar cuántas veces se debe realizar el movimiento. Por ejemplo, `5h` moverá el cursor cinco caracteres a la izquierda y `3w` moverá el cursor tres palabras a la derecha.

Para mover el cursor a un número de línea específico, escriba ese número de línea seguido del carácter `G`. Por ejemplo, para llegar a la quinta línea del archivo, escriba `5G`. Puede usar `1G` o `gg` para moverse a la primera línea del archivo, mientras que una `G` única le llevará a la última línea. Para averiguar en qué línea se encuentra el cursor, utilice **CTRL+G**.

Modo de comando: Acciones

La convención estándar para editar contenido con procesadores de texto es usando copiar, cortar y pegar. El programa vi no tiene ninguno de estos. En su lugar, vi utiliza los tres comandos siguientes:

Estándar	Vi	Significado
cortar	`d`	eliminar (*delete*)
copiar	`y`	sacar (*yank*)
pegar	`P` | `p`	poner (*put*)

Los movimientos aprendidos en la página anterior se utilizan para especificar dónde se llevará a cabo la acción, comenzando siempre con la ubicación actual del cursor. Cualquiera de los siguientes formatos generales es aceptable para comandos de acción:

```
acción [número] movimento
```

```
[número] acción movimento
```

Eliminar

Eliminar (*delete*) suprime el texto indicado de la página y lo guarda en el búfer, siendo el búfer el equivalente al “portapapeles” (*clipboard*) utilizado en Windows o Mac OSX. En la siguiente tabla se proporcionan algunos ejemplos de uso comunes:

Acción	Resultado
`dd`	Elimina la línea actual
`3dd`	Elimina las tres líneas siguientes
`dw`	Elimina la palabra actual
`d3w`	Elimina las tres palabras siguientes
`d4h`	Elimina cuatro caracteres hacia la izquierda

Cambiar

La función cambiar (*change*) es muy similar a la de eliminar; el texto se elimina y se guarda en el búfer. Sin embargo, el programa cambia a modo de inserción y permite la introducción de cambios inmediatos en el texto. En la siguiente tabla se proporcionan algunos ejemplos de uso comunes:

Acción	Resultado
`cc`	Cambiar la línea actual
`cw`	Cambiar la palabra actual
`c3w`	Cambiar las tres palabras siguientess
`c5h`	Cambiar cinco caracteres hacia la izquierda

Sacar

Sacar (*yank*) coloca el contenido en el búfer sin eliminarlo. En la siguiente tabla se proporcionan algunos ejemplos de uso comunes:

Acción	Resultado
`yy`	Sacar la línea actual
`3yy`	Sacar las tres líneas siguientes
`yw`	Sacar la palabra actual
`y$`	Sacar el fragmento desde el cursor hasta el final de la línea actual

Poner

Poner (*put*) coloca el texto guardado en el búfer antes o después de la posición del cursor. Tenga en cuenta que estas son las dos únicas opciones, poner no utiliza movimientos como los comandos de acción anteriores.

Acción	Resultado
`p`	Poner o pegar después del cursor
`P`	Poner antes del cursor

Buscar en vi

Otra función estándar que ofrecen los procesadores de texto es la función de búsqueda (*find*). A menudo, las personas usan **CTRL+F** o miran en el menú de edición. El programa `vi` utiliza la búsqueda. La función de búsqueda es más potente que la función *find* porque admite patrones de texto literales y expresiones regulares.

Para buscar hacia adelante desde la posición actual del cursor, use la `/` para iniciar la búsqueda, escriba un término de búsqueda y, a continuación, presione la tecla **Enter** para iniciar la búsqueda. El cursor se moverá al primer resultado que coincida con su término de búsqueda.

Para proceder al siguiente resultado coincidente usando el mismo patrón, presione la tecla `n`. Para volver al resultado anterior, presione la tecla `N`. Si se alcanza el final o el comienzo del documento, la búsqueda se ajustará automáticamente para continuar con el resto del documento.

Para empezar a buscar desde la posición del cursor hacia atrás, empiece por escribir `?`, entonces escriba el patrón de búsqueda y presione la tecla **Enter**.

Modo Insertar

El modo Insertar se utiliza para agregar texto a un documento. Hay algunas maneras de entrar en el modo de inserción desde el modo de comando, cada una diferenciada por donde comienza la inserción de texto. La siguiente tabla presenta los más comunes:

Entrada	Función
`a`	Comenzar a insertar justo después del cursor
`A`	Comenzar a insertar al final de la línea
`I`	Comenzar a insertar justo antes del cursor
`I`	Comenzar a insertar al principio de la línea
`o`	Comenzar a insertar en una nueva línea después del cursor
`O`	Comenzar a insertar en una nueva línea antes del cursor

Modo Ex

Originalmente, el editor `vi` se llamaba editor `ex`. El nombre `vi` era la abreviatura del comando *visual* en el editor `ex` que cambiaba el editor al modo “visual”.

En el modo normal original, el editor `ex` sólo permitía a los usuarios ver y modificar una línea cada vez. En el modo visual, los usuarios podían ver la mayor parte del documento que podía caber en la pantalla. Dado que la mayoría de los usuarios preferían el modo visual al modo de edición por línea, el archivo de programa `ex` se vinculó a un archivo `vi`. De este modo los usuarios podían iniciar `ex` directamente en modo visual al ejecutar el enlace `vi`.

Eventualmente, el archivo de programa fue renombrado `vi` y el editor `ex` se convirtió en un enlace que apuntaba al editor `vi`.

Cuando se utiliza el modo ex del editor `vi`, es posible ver o cambiar su configuración, así como ejecutar comandos de archivo como abrir, guardar o cancelar cambios en un documento. Para acceder al modo ex, escriba el carácter `:` en el modo de comando. En la tabla siguiente se enumeran algunas acciones comunes realizadas en modo ex:

Entrada	Función
`:w`	Escribir el documento actual al sistema de archivos
`:w nombre_del_archivo`	Guardar una copia del documento actual bajo el nombre *nombre_del_archivo*
`:w!`	Forzar escritura al documento actual
`:1`	Ir a la primera línea (o otra línea indicada por el número)
`:e nombre_del_archivo`	Abrir *nombre_del_archivo*
`:q`	Suspender (salir) (*quit*) si no se han realizado cambios al documento
`:q!`	Suspender sin guardar los cambios realizados al documento

Un análisis rápido de la tabla anterior revela que cuando un signo de exclamación, `!` , se agrega a un comando, se intentará forzar la operación. Por ejemplo, imagine que realiza cambios a un archivo en el editor `vi` y luego intenta salir usando `:q`, solo para descubrir que el comando falla. En este caso, el editor `vi` no quiere salir del documento sin guardar los cambios realizados, pero usted puede forzar la salida con el comando ex `:q!` .

A tener en cuenta

Aunque el modo ex ofrece varias maneras de guardar y salir, también está disponible el comando `ZZ`; éste es el equivalente a `:wq`. Hay muchas más funciones que se solapan entre el modo ex y el modo de comando. Por ejemplo, el modo ex se puede utilizar para navegar a cualquier línea del documento escribiendo `:` seguido del número de línea, mientras que `G` se puede utilizar en modo de comando como se ha demostrado anteriormente.

Siga leyendo

Si tiene un archivo de texto abierto, salga ejecutando el comando `:q!`. Esto lo cerrará sin guardar cambios.