

## ARTÍCULO

### VI: EDITOR POR CAPAS

vi (pronunciado a menudo como vee-eye, es una abreviatura de "visual") provee capacidad de edición de texto básicas. Hay tres aspectos de vi que lo hacen atractivo. Primero, vi es proporcionado con todos los sistemas UNIX, siendo el editor más estándar en estos sistemas. Se puede usar vi en cualquier lugar donde exista un sistema operativo UNIX, ya sea en universidades o en empresas. Segundo, vi usa una pequeña cantidad de memoria, lo cual permite una operación eficiente cuando la red está muy ocupada. Tercero, porque vi usa teclas alfanuméricas estándares para los comandos, se puede usarlo en alguna terminal virtual o una estación de trabajo sin tener que preocuparse por mapeos de teclas inusuales.

El vi es un editor multicapa. El primer editor estándar era el ed, como UNIX fue difundido en las universidades norteamericanas se desarrollo sobre ed otro que se llamaba ex. Este editor tenía más comandos que el ed pero continuaba siendo de línea con todos sus defectos ya que, por los años 70, las terminales eran máquinas de escribir con rodillo y papel que se denominaban typewriter, por eso las líneas de terminales de UNIX se llaman tty, el hecho de contar con un terminal de tubos de rayos catódicos implicaba mucha inversión además, estos, no respondían a secuencias de escape para el manejo del cursor, sino que imitaban a las maquinas de escribir.

Con los modernismos y gracias a cierta marca de terminales que popularizo las conversaciones host y terminal se empezaron a ver terminales que respondían a ciertas cadenas de caracteres que le permitían posicionar cursor, limpiar pantalla, limpiar línea, etc. En Berkeley nació el vi, que es otra capa más sobre ex, pero que ofrecía trabajar a plena pantalla. Más tarde un alumno de Berkeley, tuvo la idea de remplazar las secuencias de escape incluidas dentro del vi por pseudo comandos utilizando un fichero anexo que contenía la traducción de dichos pseudo comandos a secuencias de escape y

secuencias de escape a pseudo comandos, la librería que creo se llama termio y el fichero termcap, con esto se consiguió la independencia del terminal que se usa.

Muchos novatos ven a vi como un editor no intuitivo, aburrido o difícil, ya que es fácil confundirse y sentirse perdido al intentar, por primera vez, aprenderse las características de este editor. A menudo hay varias formas de hacer la misma operación con efectos ligeramente diferente. Se necesita, tan sólo, pocas teclas para indicar a vi tareas complejas.

### Modos de edición de orden, de entrada y de línea

Los editores usan el teclado para dos operaciones muy diferentes: para especificar las órdenes de edición y para recibir entrada de caracteres. Consideradas como órdenes de edición, ciertas teclas borrarán texto, otras realizarán modificaciones y otras movimientos del cursor. Consideradas como entrada de caracteres, las teclas representan a los caracteres que pueden introducirse en un archivo que se esté editando. En muchos editores para PC, estas dos funciones diferentes están repartidas entre diferentes teclas del teclado; las teclas alfabéticas se reservan para entrada de caracteres y las teclas de función y de control que especifican órdenes de edición tales como el borrado de texto o el desplazamiento del cursor.

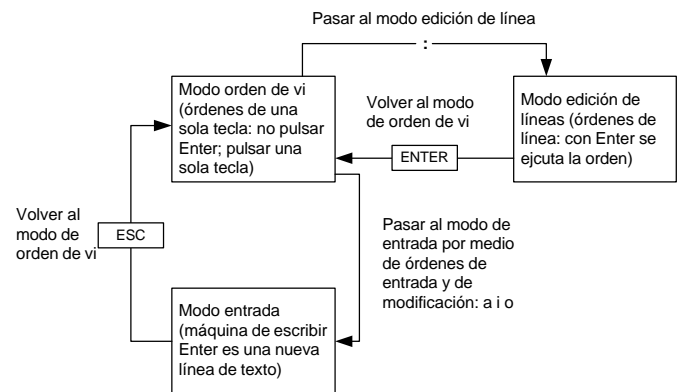
Los editores en Unix fueron diseñados asumiendo un teclado mínimo con caracteres alfabéticos, algunos caracteres de control y las teclas Esc o Enter. En lugar de repartir las funciones de órdenes y de entrada entre diferentes teclas, el editor vi tiene dos modos de operación independientes para el teclado: el modo orden y el modo entrada. En modo orden, todas las teclas del teclado se convierten en órdenes de edición. En modo entrada, las teclas del teclado se convierten en caracteres de entrada.

Al cambiar de modo, cambia la funcionalidad del editor. Diferente a muchos editores, el modo orden es el modo inicial o modo por defecto; cada tecla será una orden de edición de modo que, al pulsar una tecla, se ejecutará

una orden determinada. Por ejemplo, al pulsar la tecla x se borrará un carácter de texto. Al pulsar la tecla l, se desplazará el cursor un carácter hacia la derecha. Algunas de estas órdenes de edición, tal como a (apend), i (insert) u o (open), hacen pasar al modo entrada.

Una vez en modo entrada, el teclado cambia nuevamente su funcionalidad. Cada tecla representa un carácter que se puede introducir en el texto. El teclado pasa a ser como una máquina de escribir. Al pulsar una tecla, su carácter correspondiente es añadido al texto. Por ejemplo, al pulsar la tecla x, simplemente se añade la x al texto. Una excepción es al pulsar la tecla Esc, que automáticamente se regresa al modo orden y las teclas se convierten en órdenes de edición. Después se puede fácilmente volver al modo entrada mediante cualquiera de las órdenes de edición de entrada. A medida que se vaya editando texto se encontrará en un constante cambio de modo.

Aunque el modo orden de vi se ocupe de la mayoría de las operaciones de edición, hay algunas, como guardar archivos o realizar sustituciones globales, que no puede efectuar. Para llevar a cabo dichas operaciones se necesita recurrir a órdenes de edición de línea. Para entrar al modo edición de línea se usa la orden dos puntos (:) de vi. Los dos puntos representan una orden especial que le permite realizar una operación de edición de línea. Al pulsar dos puntos aparecerá una línea en la parte inferior de la pantalla con un cursor situado al principio, éste es el modo edición de línea, en el cual se puede introducir órdenes de edición de una línea, pulsar Enter y así ejecutarlas. Cabe destacar que este modo es temporal, al ejecutar la instrucción automáticamente se vuelve al modo orden de vi y el cursor regresa a su posición anterior en la pantalla. La siguiente figura ilustra los modos de operación de vi.



Uno o dos caracteres son usados para las órdenes básicas. Así, usando letras como órdenes se puede editar un texto a gran velocidad y no se necesita memorizar teclas de función o estirar los dedos para alcanzar complicadas combinaciones de teclas. Muchos de los comandos pueden ser recordados por la letra que realiza la acción y casi todos los comandos siguen patrones similares.

Se puede usar vi para editar cualquier fichero de texto. Para realizar esta tarea, vi copia el archivo a ser editado dentro de un buffer. Además vi edita archivos de texto regular, no siendo así, los archivos de bloque, de carácter, binarios, directorios y no ascii (American Standard Code for Information Interchange).

Si quieres aprender a usar un buen editor de textos sobre casi cualquier sistema operativo existente en el mundo (Unix/Linux, AmigaOS, AtariMiNT, BeOS, DOS, MacOS, MachTen, OS/2, RiscOS, VMS y Windows) usa vi. Aunque quizás para proyectos sea conveniente usar algún entorno integrado; vi siempre es una muy buena opción multiplataforma.

Y como con cualquier habilidad, una vez que uses la totalidad de la funcionalidad de este editor no querrás regresar a otro editor "más simple".

#### Referencias:

The VIM (Vi IMproved) Home Page

<http://www.vim.org/>

Linux: Manual de referencia

Richard Petersen, 1999 McGraw-Hill

Vim Color Editor HOW-TO

<http://www.linuxdoc.org/HOWTO/Vim-HOWTO.html>

Por: Jesús Mena Chalco

j.mena@usp.edu.pe