

En este capítulo vamos a estudiar los principales editores que Unix estándar y Linux ponen a disposición de los usuarios. Entre los primeros podemos citar al editor de líneas **ed**, al editor de pantallas **vi** y al también editor de pantallas **emacs**. Entre los que incorpora Linux veremos brevemente el editor **joe** que es muy parecido al *wordstar* y **jed** que es una versión reducida de *emacs*.

El editor básico de líneas **ed**.

El editor **ed** es un editor de líneas. Las operaciones del editor se realizan sobre líneas. Aunque algunos de sus comandos pueden hacer referencia a un conjunto de líneas, sin embargo, este editor no tiene la facilidad de movimiento entre líneas como en un editor de pantalla completa.

El editor ``ed'` tiene dos modos de operación:

- **Modo de Orden:** Se utiliza para dar órdenes, tales como escribir en un fichero, buscar cadenas de texto, etc.
- **Modo de Entrada:** Se utiliza para introducir texto. Estando en este modo para retornar al modo de orden después de haber introducido el texto se utiliza **^d**.

Para invocar a este editor se utiliza la orden: **ed fichero**, si existe fichero, copia el contenido del mismo en la memoria y muestra el tamaño del fichero en caracteres. Si el fichero no existe, señala que el buffer asignado al fichero está vacío. En ambos casos el editor queda en modo orden.

La orden **P** hace que se muestre el prompt del editor *ed* cuando se encuentre en modo de orden. Por omisión es el carácter *****.

Cuando se comete un error aparece el carácter **?**. La orden **H** activa una presentación de mensajes de error más explícita.

La mayoría de las operaciones en modo orden operan sobre la línea actual, o bien, sobre un rango de líneas delimitado por el número de línea inicial, una **,** (coma) y el número de línea final. En modo de orden la línea actual se denomina **.** (punto) y la última línea del fichero **\$** (dólar).

Comandos del editor **ed** y Función que realizan:

Comando	Función
[lin]a	Añadir texto a partir de la línea indicada, o en la actual si no se indica
[lin]i	Inserta texto antes de línea indicada
[lin]c	Cambiar la línea o líneas indicadas
[lin]d	Borra la línea o las líneas indicadas
[lin]p	Visualiza la línea o el rango de líneas indicadas.
[lin]n	Igual que el anterior, mostrando el número de la línea
[lin]md	Mover la línea o rango de líneas a partir de la línea d
[lin]td	Copiar la línea o rango de líneas indicado a partir de la línea d
e[fic]	Carga el fichero indicado. No actúa si el fichero actual no está grabado
E[fic]	Carga el fichero indicado sin comprobar si el fichero actual está grabado o no
f[nombre]	Fija el nombre del fichero actual
[lin]r [fic]	Lee el fichero indicado añadiendo a partir de la línea indicada o de la última si no se indica

[lin]w [fic]	Graba las líneas indicadas, todas por defecto, en el fichero especificado o en el fichero actual por defecto
q	Sale del editor, emite un error si el fichero actual no fue grabado
Q	Igual que el anterior, sin comprobar si el fichero actual fue grabado
P	Activa /Desactiva el prompt del editor
H	Activa/Desactiva los mensaje explicativos de error
u	Deshace el último comando que puede deshacerse. Por ejemplo una inserción.
!orden	Ejecuta la orden especificada llamando al intérprete de comandos sin salir del editor
+ <i>[n]</i>	Avanza <i>n</i> líneas o una por defecto desde la línea actual
- <i>[n]</i>	Retrocede <i>n</i> líneas o una por defecto desde la línea actual
^d	Vuelve al modo orden
[lin]g/expr/comando	Busca la expresión indicada (<i>expr</i>), en el rango de líneas indicado (todas por defecto) ejecutando en cada una de ellas el comando especificado, por ejemplo: g/hola/n
[lin]G/expr/	Se posiciona una a una en todas las líneas dónde ha encontrado <i>expr</i> , en el rango de líneas especificado (todas por defecto), a la espera de una orden o <intro> para continuar
[lin]v/expr/comando	Busca las líneas que no contengan la expresión indicada (<i>expr</i>), en el rango de líneas indicado (todas por defecto) ejecutando en cada una de ellas el comando especificado. Por ejemplo: v/hola/n
[lin]V/expr/	Se posiciona una a una en todas las líneas donde no ha encontrado <i>expr</i> en el rango de líneas indicado (todas por defecto) a espera de una orden o <intro> para continuar
[lin]s/expr/sust/[g]	Busca las líneas que contienen <i>expr</i> dentro del rango indicado o en la actual por defecto, sustituyendo <i>expr</i> por <i>sust</i> . Si se incluye <i>g</i> sustituye en todo el rango y no solo la primera concurrencia en ese rango

El editor de pantallas vi.

El editor **vi** es un editor de pantalla completa.

De forma similar a **ed**, el editor **vi** siempre trabaja con una copia del fichero que se edita, que él mismo se encarga de mantener en un buffer. Los cambios realizados sobre el fichero solo se graban cuando se invoca algún comando de grabación.

Para invocar el editor **vi** se hace mediante: **vi fichero**. Si el fichero existe lo carga en el buffer de trabajo.

Además, la orden para invocar el editor **vi** tiene una serie de opciones, las más importantes son:

r Cada cierto tiempo `vi' salva el buffer de edición en disco. De forma que si se produce cualquier fallo en el sistema, siempre se encontrará una copia reciente del fichero que se editaba.

R Indica que solo se editará el archivo para consulta. No permitiéndose realizar cambio alguno sobre la información.

wn Normalmente `vi' utiliza la pantalla completa como ventana de edición. Mediante esta opción se podrá alterar el tamaño de la ventana de edición a un número de líneas determinado por *n*.

Podemos utilizar la opción + , que permite entrar en el **vi** en la línea que le indiquemos. Por ejemplo:

Vi +7 fichero Editará el fichero indicado y tomará como línea actual la 7.

Vi +\$ fichero Editará el fichero indicado y tomará como línea actual la última línea del fichero.

Vi +/-cadena fichero Editará el fichero indicado y tomará como línea actual la primera línea que contenga la cadena indicada.

El editor **vi** tiene tres modos de trabajo diferente:

- **Modo comando:** Es el modo en el que queda el editor al entrar. En él cada pulsación de una tecla se interpreta como un comando, no siendo posible la introducción de texto.
- **Modo texto:** Cada pulsación de una tecla se interpreta como un carácter a introducir al texto editado. Es el modo indicado para añadir información al fichero que se está editando.
- **Modo ex:** Se podrán utilizar los comandos del intérprete **ex**, que en muchos casos son iguales que los comandos **ed**. Para pasar a este modo se debe estar en modo comando.

A continuación estudiaremos un esquema para conocer cómo se cambia de un modo de trabajo a otro:

Una característica interesante del editor **vi** es que permite la edición múltiple de ficheros. Es decir, la posibilidad de editar conjuntamente varios ficheros. Esto puede hacerse de dos formas:

- Al entrar mediante línea de comandos: **vi fichero1 fichero2 fichero3**. Inicialmente se editará *fichero1*, pero podemos editar el siguiente mediante el comando **n**, es necesario escribir los cambios del fichero actual antes de editar el siguiente. Para volver a editar el primer fichero de la lista utilizar el comando **rew**.
- Estando ya trabajando con **vi** mediante el comando **e fichero2**. Si se está editando un fichero, puede llamarse a otro fichero para que se edite conjuntamente con el actual. Para editar el fichero anterior puede utilizarse el comando **e#**.

Al entrar en **vi** pasa directamente al Modo Comando.

Comandos del editor **vi** y Función que realizan

MODO TEXTO. Cuando estamos en modo de entrada, todo el texto que introduzcamos formará parte del fichero. Los NEWLINE marcan el final de línea, y habilitan la siguiente. El BACKSPACE suprime el carácter de la izquierda, así hasta el principio de la línea, pero no se permite ir a la línea anterior. La tecla ESC permite pasar de modo entrada a modo comando.

^u + <ESC> Cancela la última inserción realizada.

^w + <ESC> Cancela la última palabra introducida.

<ESC> Vuelve al modo comando.

MODO COMANDO Al entrar en **vi** estamos en modo comando. Si estamos en modo de entrada y queremos pasar a modo comando pulsaremos la tecla ESC.

Para comenzar a ejecutar un comando primero debemos situarnos en el punto en que queremos modificar el texto.

MOVIMIENTO DEL CURSOR	
Comando	Función
<ESPACIO> , ! , l	Avanza una posición a la derecha

<BS> , ! , h	Retrocede una posición a la izquierda
^n , ! , j	Avanza una línea
^p , ! , k	Retrocede una línea
+ , <intro>	Avanza al principio de la línea siguiente
–	Retrocede al principio de la línea anterior
0 (cero)	Retrocede al principio de la línea actual
\$	Avanza al final de la línea actual
[n]G	Va a la línea <i>n</i> o a la última por defecto
[n]	Va a la columna <i>n</i> de la línea actual o a la primera por defecto
w, W	Va al principio de la palabra siguiente
b, B	Va al principio de la palabra anterior
e, E	Va al final de la palabra actual
)	Va al final del fichero, al final de la última línea del texto
H	Va a la esquina superior izquierda de la pantalla
M	Va a la línea central de la pantalla

MOVIMIENTO DE LA PANTALLA

Comando	Función
[lin]^U	Desplaza hacia arriba media pantalla o el número de líneas indicado por <i>lin</i>
[lin]^D	Desplaza hacia abajo media pantalla o el número de líneas indicado por <i>lin</i>
^F	Avanza una pantalla completa
^B	Retrocede una pantalla completa
^R, ^L	Redibuja la pantalla

INTRODUCCIÓN DE TEXTO (PASO A MODO TEXTO)

Comando	Función
i	Inserta a partir del carácter actual
I	Inserta a partir del principio de la línea actual
a	Añade a partir del carácter actual
A	Añade a partir del final de la línea actual
o	Abre una línea bajo la actual y pasa a añadir al principio de la misma
O	Abre una línea sobre la actual y pasa a añadir al principio de la misma

BORRADO DE TEXTO

Comando	Función
[n]x	Borra el carácter sobre la posición actual del cursor, o bien, <i>n</i> caracteres
[n]X	Borra el carácter anterior a la posición actual del cursor, o bien, <i>n</i> caracteres
d[movimiento cursor]	Borra el texto existente entre la posición actual del cursor y la posición que se alcanzará mediante el comando de movimiento del cursor especificado, por ejemplo: dw borra hasta el principio palabra siguiente d) borra hasta el final del fichero
[n]dd	Borra la línea actual, o bien, <i>n</i> líneas a partir de la actual

D	Borra desde la posición actual del cursor hasta el final de la línea actual
---	-----------------------------------------------------------------------------

Nota: El texto borrado se almacena en un buffer para poder ser recuperado.

MODIFICACIÓN DE TEXTO	
Comando	Función
u	Deshace el último cambio sobre el texto
U	Deshace todos los cambios sobre la línea actual
r	Sustituye el carácter sobre el que está el cursor, por el que se pulse a continuación
R	Permite reescribir texto sobre la actual posición del cursor
s	Sustituye el carácter actual sobre el texto introducido a continuación
S	Sustituye la línea actual por el texto introducido a continuación
J	Une la línea siguiente al final de la actual
cw	Cambia la palabra actual desde la posición del cursor por un nuevo texto
c)	Cambia el texto, desde la posición actual del cursor hasta el final del fichero

CONTROL DE BUFFERS (BLOQUES)	
Comando	Función
[letra][n]yy	Almacena en el buffer indicado por letra, o en el buffer sin nombre, las <i>n</i> líneas siguientes a la actual, o la línea actual.
[letra] y [movim_cursor]	Almacena en el buffer indicado por letra, o en el buffer sin nombre, las líneas existentes entre la posición actual y la indicada por la orden de movim_cursor
[letra]p	Recupera el texto del buffer letra, o del buffer sin nombre, insertándolo a partir de la línea actual
[letra]P	Recupera el texto del buffer letra, o del buffer sin nombre, insertándolo antes de la línea actual

BÚSQUEDA	
Comando	Función
/patrón	Busca el patrón indicado desde la posición actual del cursor hasta el final del fichero
?patrón	Busca el patrón indicado desde la posición actual del cursor hasta el principio del fichero
n	Repite la última búsqueda
N	Repite la última búsqueda en el sentido opuesto al establecido
f[carácter]	Busca carácter dentro de la línea actual hasta el final de la misma
F[carácter]	Busca carácter dentro de la línea actual hasta el principio de la misma
; (punto y coma)	Repite la última búsqueda de carácter
, (coma)	Repite la última búsqueda de carácter en sentido opuesto
ZZ	Sale al intérprete de comandos almacenando el fichero actual si es necesario

ULTIMO COMANDO

. (punto) Repite el último comando.

MODO INTERPRETE `ex'

Todos los comandos de **ex** comienzan con el carácter : (dos puntos). Tenemos que estar en modo orden.

CARGA Y GRABACIÓN	
Comando	Función
:w[fichero]	Graba el contenido del buffer del editor en el fichero indicado, o en el fichero por defecto
:w >>[fichero]	Graba el contenido del buffer del editor añadiéndolo al contenido del fichero indicado, o en el fichero por defecto
:r[fichero]	Carga el contenido de fichero, o del fichero por defecto, añadiéndolo a partir de la línea actual
:wq	Grabar el fichero actual y salir del editor
:q	Abandona el editor volviendo al intérprete de comandos
:q!	Abandona el editor volviendo al intérprete de comandos, sin realizar comprobaciones de grabación
:x	Guarda el archivo si es necesario y sale al intérprete de comandos

EJECUCIÓN DE ÓRDENES DEL SISTEMA	
Comando	Función
:r!orden	Ejecuta la orden indicada, añadiendo la salida de la misma a partir de la línea actual
:!orden	Ejecuta la orden especificada

RECUPERACIÓN ANTE FALLOS EN EL SISTEMA	
Comando	Función
:preserve	Almacena el fichero como si se hubiese producido un fallo en el sistema
:recover [fichero]	Recupera el fichero indicado o el actual

EDICIÓN COMPARTIDA DE FICHEROS	
Comando	Función
:n	Edita el siguiente fichero de la lista con la que se llamó al editor
:n!	Edita el siguiente fichero de la lista con la que se llamó al editor, sin realizar comprobaciones de grabación
:rew	Vuelve a editar el primer fichero de la lista con la que se llamó al editor
:rew!	Vuelve a editar el primer fichero de la lista con la que se llamó al editor, sin realizar comprobaciones de grabación
:e[fichero]	Edita el fichero especificado, o el fichero por defecto
:e![fichero]	Edita el fichero especificado, o el fichero por defecto, sin realizar comprobaciones de grabación
:e[#]	Vuelve a editar el texto que se editó antes del actual, o el actual por defecto

INFORMACIÓN DEL FICHERO ACTUAL	
Comando	Función
:args	Muestra la línea de llamada al editor en la línea de estatus
:f	Muestra el nombre del fichero editado
:f nombre	Fija el nombre del archivo por defecto

ABREVIATURAS	
Comando	Función
:abbr a b	

	<p>Establece que la cadena <i>b</i> es la expansión de la abreviatura <i>a</i>, cada vez que se escriba la cadena <i>a</i> se expandirá el texto según <i>b</i>, ejemplo:</p> <p>abbr uc UNIDAD DE CONTROL</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

BÚSQUEDA Y SUSTITUCIÓN	
Comando	Función
: [lin]s/expr/sust/g	<p>Busca el texto <i>expr</i> dentro del rango indicado, sustituyéndolo por <i>sust</i>, donde:</p> <p>; Indica que es un comando de ex</p> <p>[lin] Marca el rango, expresado en números de línea, dónde se realizará la búsqueda y posterior sustitución, según el siguiente formato: [lin_inicial,lin_final] y dónde cada número de línea puede sustituirse por los caracteres \$ (línea final), . (línea actual), % (referencia a todo el fichero)</p> <p>s Es el comando de sustitución</p> <p>expr Es el texto a buscar</p> <p>sust Es el texto a cambiar</p> <p>g Indica que deben sustituirse todas las apariciones del texto <i>expr</i> dentro del rango especificado</p> <p>Ejemplos: %s/volverán/vendrán/g</p> <p>:1,\$s/volverán/vendrán/g</p> <p>:3,./s/volverán/vendrán/g</p>

COMANDOS DE CONFIGURACIÓN DE OPCIONES	
Comando	Función
:set autoindent	Activa el modo de indentación automática, mediante el cual cada vez que se pulsa <return> para terminar una línea, el cursor se sitúa en la línea de abajo pero no en la primera columna, sino en la columna dónde empezó a escribir en la línea anterior
:set noautoindent	Desactiva el modo de indentación automática
:set autowrite	Activa la grabación automática en la edición compartida de ficheros, mediante la cual, al editar otro fichero de la lista se graba el actual
:set noautowrite	Desactiva la grabación automática
:set ignorecase	Activa la no distinción entre mayúsculas y minúsculas al realizar búsquedas
:set noignorecase	Desactiva la no distinción entre mayúsculas y minúsculas al realizar búsquedas, se buscará el texto que coincida exactamente con el tecleado
:set number	Activa la numeración de líneas
:set nonumber	Desactiva la numeración de líneas.
:set all	Visualiza la configuración actual
:set showmode	Muestra cualquier tipo de modo en el que se esté. Al activar esta opción en la parte inferior derecha aparecen mensajes del tipo: INPUT modo texto, COMMAND modo comando, etc

<code>:set noshowmode</code>	Desactiva la opción anterior <code>showmode</code>
------------------------------	----------------------------------------------------

CREACIÓN DE MACROS

Comando	Función
<code>:map nombre comando</code>	Crea la macro llamada <code>nombre</code> , que realiza las acciones que especifiquen los comandos. Esta orden mapea cierta pulsación en el teclado para convertirla en un comando. Para introducir una secuencia de escape, es necesario pulsar anteriormente <code>^V</code> . Otra de la secuencia necesaria para crear macros es <code>^M</code> que sustituye a <code><return></code>
<code>:map! nombre comando</code>	Indica que la macro operará en modo texto
<code>:unmap nombre</code>	Deshace la macro especificada

Ejemplos de macros:

`:map ^F:!date^M` Para que `^F` y `^M` surtan efecto antes pulsar `^`

`:map! LL ^[:q!^M` La nueva orden `LL` sale sin grabar desde modo texto. Para que `<esc>` (`^[]`) surta efecto antes pulsar `^V`.

`:map Q :q!^M` La nueva orden `Q` sale sin grabar desde modo comando.

`:map! ZZ ^[:wq^M` La orden `ZZ`, ya implementada, abandona el editor grabando.

OTROS COMANDOS

Comando	Función
<code>:w>>fichero</code>	Añade lo editado al fichero indicado
<code>:>n</code>	Corre a la derecha un tabulador el número de líneas que le indiquemos, a partir de la actual
<code>:<n</code>	Corre a la izquierda un tabulador el número de líneas que le indiquemos, a partir de la actual
<code>:>>n</code>	Corre dos tabuladores a la derecha
<code>:<<n</code>	Corre dos tabuladores a la izquierda
<code>:n</code>	Desplaza el cursor a la línea que le indicamos
<code>:sh</code>	Escapa al shell, aparece el prompt <code>PS1</code> . Para volver a continuar con el editor pulsar <code>Ctrl d</code> o teclear <code>exit</code>

OPCIONES POR DEFECTO

Comando	Función
<code>.exrc</code>	Si se desea trabajar con una configuración personalizada de opciones de <code>vi</code> , se puede crear el fichero <code>.exrc</code> en el directorio raíz del usuario. El usuario al invocar a <code>vi</code> y existir el fichero, configurará el editor con todas las opciones y comandos que contenga. Es posible que algunos comandos de configuración no funcionen correctamente por problemas de emulación

Ejemplos:

`:5,30w! otro.fichero` Graba en otro.fichero el contenidos de las líneas indicadas, borrando lo que hubiera. Si no existe lo crea.

`!!:ls -l` Escapa al shell, lista el directorio actual, y vuelve al `vi`.

:3,56!comando Suprimirá las líneas especificadas por el resultado de la ejecución del comando.

:l,\$s /pepe /juan Desde la línea 1 hasta el final, cambia pepe, por juan.

.,,\$s /pepe /juan Lo mismo desde la línea donde está el cursor hasta el final.

:abbr URSS Unión de Repúblicas Socialistas Soviéticas Cuando escribamos en el texto URSS se sustituirá automáticamente por el texto indicado.

:1,5 copy 20 Copia el contenido de las líneas de la 1 a la 5 en la línea 20.

:move 5 Mueve la línea actual a la línea indicada, la primera línea es la 0.

:move -5 Mueve la línea actual 5 posiciones al .contrario

:/expresión/ Busca hacia adelante la expresión indicada.

:?expresión Busca hacia atrás la expresión indicada.

:r nombre_programa Llama a otro programa añadiéndolo

El carácter **0** (cero) indica primera de línea, mientras que el carácter **\$** indica última línea.

Si hemos introducido **vi** con una lista de ficheros, la orden **:n** cerrará el fichero actual y pasará al siguiente fichero de la lista de argumentos. Para acabar sin grabar escribiremos **:n!** o **:q!**

Para pasar a modo de ordenes pulsar la tecla ESC.

El editor emacs.

El nombre **emacs** significa Editor MACroS, que nació como una sustitución de un editor de textos anterior llamado **teco**. Actualmente **emacs** es uno de los editores más usados y ampliamente instalados disponible en el mundo Linux. Hay versiones **emacs** disponibles en casi todas las plataformas informáticas conocidas en el sector, desde Linux a MS-Windows.

Una versión completa de **emacs** es muy grande, ocupando varios megas en disco. Es un editor funcionalmente completo, muy potente y ampliado con funciones que van más allá de la edición de textos. En algunas instalaciones se puede usar para editar archivos, mantener un calendario, trabajar con e-mail, gestionar archivos, leer USENET o noticias de la red, usarlo como calculadora, etc. De alguna forma, **emacs** es un entorno de trabajo que contiene un editor de textos. Una versión popular de **emacs** se distribuye por medio de la licencia GNU. Esta versión fue creada por Richard Stallman que es uno de los patriarcas de la Free software Foundation y del proyecto GNU.

El editor **emacs** no dispone de los dos modos básicos que tiene **vi**. Esto significa que todo lo que escriba se coloca en la memoria intermedia. Para dar comandos al editor, para guardar archivos, buscar textos, borrarlos, etc. debe usar otras teclas. En **emacs** se utiliza la tecla **<ctrl>** y **<esc>** para ejecutar los distintos comandos.

INICIO, SALIDA Y AYUDA	
Comando	Función
emacs	Inicia el editor sin fijar ningún fichero
emacs fichero	Comienza la edición con el fichero especificado

<Ctrl-x><Ctrl-c>	Sale de emacs. Si no se ha guardado el fichero, pedirá confirmación
<Ctrl-h>	Llama a la ayuda de emacs
ESC	Retrocede. Salida de cualquier sitio
<F10>	Acceso al menú

Copiar, cortar y pegar.

Cuando se suprime o corta caracteres estos se guardan en una memoria intermedia. Para pegar este texto se utiliza <Ctrl-y>. Este comando también llamado comando **yank**, pega el objeto a la derecha del cursor. Para realizar la operación de cortar y pegar de un bloque, hay que seguir los siguientes pasos:

- Situar el cursor al principio del texto a marcar.
- Pulsar <Ctrl-barra espaciadora> para fijar la marca.
- Mover el cursor al final del bloque de texto. Así se crea lo que emacs llama una región.
- Suprimir el texto con <Ctrl-w>.
- Si lo que se pretende es copiar el texto antes de mover el cursor hay que volver a pulsar <Ctrl-y>. Si lo que se desea es moverlo seguir con el siguiente paso.
- Situar el cursor en el punto del texto donde se quiera copiar el mismo.
- Pulsar <Ctrl-y> para pegar el bloque por debajo de la línea donde está el cursor.

COMANDOS BÁSICOS	
Comando	Función
<Ctrl-x><Ctrl-s>	Guarda la memoria intermedia actual (texto editado) en disco
<Ctrl-x><Ctrl-w>	Escribe la memoria intermedia actual en disco, pregunta por un nuevo nombre de archivo
<Ctrl-x><n>	Cambia el nombre del archivo de la memoria intermedia actual
<Esc><z>	Borra desde la posición actual hasta el carácter tecleado
<Ctrl-x><Ctrl-f>	Encuentra un archivo, lo lee en una nueva memoria intermedia creada con el nombre del archivo
<Ctrl-x><Ctrl-r>	Lee el archivo dentro de la memoria intermedia actual, borrando el contenido previo
<Ctrl-x><Ctrl-i>	Inserta el archivo dentro de la memoria intermedia actual en la posición del cursor
<Ctrl-f>	Mueve el cursor un carácter hacia adelante
<Ctrl-b>	Mueve el cursor un carácter hacia atrás
<Ctrl-a>	Va al principio de la línea actual.
<Ctrl-e>	Va la final de la línea actual
<Ctrl-n>	Va a la línea siguiente
<Ctrl-p>	Va a la línea anterior
<Esc><f>	Avanza una palabra
<Esc>	Retrocede una palabra
<Esc><Shift -. >	Va al principio de la memoria intermedia
<Esc><Shift -, >	Va al final de la memoria intermedia
<Ctrl-d>	Suprime el carácter donde se encuentra
<Ctrl-c>	Inserta un espacio
<Esc><d>	Suprime la palabra siguiente
<Ctrl-k>	

	Suprime hasta el final de la línea actual. Si el cursor está al principio de la misma, la suprime entera
<intro>	Inserta una nueva línea
<Ctrl-j>	Inserta una nueva línea y sangra.
<Ctrl-o>	Abre una nueva línea (no está claro cuando funciona, a veces funciona como un intro y otras como se indica)
<Ctrl-w>	Suprime la región entre la marca y el cursor
<Esc><w>	Copia la región a la memoria intermedia de eliminación
<Ctrl-x><Ctrl-o>	Suprime la línea que está a los lados del cursor
<Ctrl-s>	Busca hacia adelante desde la posición actual.
<Ctrl-r>	Busca hacia atrás desde la posición actual
<Ctrl-s>	Repite búsqueda hacia delante
<Ctrl-r>	Repite búsqueda hacia atrás.
<Esc><Ctrl-r>	<p>Pregunta antes de realizar una sustitución, se responde con:</p> <p><Ctrl-g> Cancelar la operación</p> <p><!> Sustituir el resto</p> <p><?> Obtener una lista de opciones</p> <p>< . > Sustituir y salir donde se inició el comando</p> <p><y> Sustituir y continuar con la operación de sustituir</p> <p><n> No sustituir, pero continuar con la operación</p>
<Ctrl-barra espaciadora>	Fija una marca en la posición actual del cursor
<Ctrl-x><Ctrl-x>	Intercambia la marca y el cursor
<Ctrl-w>	Suprime la región marcada
<Esc-w>	Copia la región marcada en la memoria intermedia de eliminación
<Ctrl-y>	Inserta la memoria intermedia de eliminación en la posición actual del cursor
<Ctrl-x>	Conmuta a otra memoria intermedia
<Ctrl-x><x>	Conmuta a la próxima memoria intermedia en la lista de memorias intermedias
<Esc><Ctrl-n>	Va al final del fichero
<Ctrl-x><k>	Suprime una memoria intermedia que no se muestra
<Ctrl-x><o>	Cuando tenemos la pantalla dividida, cambia de una a otra
<Ctrl-x><n>	Divide la pantalla en n ventanas. Ventanas con las que podemos trabajar máximo 2. Si ponemos un 3 la ventana actual se divide en dos verticalmente y podemos ir haciendo más subdivisiones

Veamos a continuación como podemos solucionar problemas con emacs. Es posible que determinados caracteres de control no funcionen en los terminales. Esto es debido a la emulación, es posible que no se haya configurado la variable TERM. Para ello hay que realizar los siguientes pasos: TERM= tipo de emulación. La emulación que se utiliza es vt100, pero da algunos problemas, la mejor es la vt200 o ansi, aunque el telnet de Windows 3.11 no la incorpora.

Si la solución no llega por la emulación entonces podemos recurrir a dar las órdenes, no con caracteres de control sino que con los comandos emacs. Para ello hay que realizar las siguientes acciones:

<esc-x> nombredeorden Esta es la sintaxis para dar órdenes de emacs. Pero ¿Qué órdenes?.

A continuación veamos las dos que sustituyen a los caracteres de control que más problemas dan:

set-mark Establece una marca en la posición del cursor.(<Ctrl–barra espaciadora>)

yank Inserta el contenido del kill–buffer a partir de la posición del cursor(<Ctrl–y>).

Podríamos seguir poniendo toda la lista de comandos emacs, pero es mejor explicar como consultarlos en la ayuda de emacs:

- Pulsar <Ctrl–h><Ctrl–h>
- <Ctrl–v> Muestra más pantalla de la ayuda.
- <q> Sale de la ayuda.
- <i> muestra el índice de comandos.
- <a> pregunta por una palabra, por ejemplo yank, y muestra todos los comandos que tienen relación con esta palabra.

El editor joe.

Este editor es exclusivo de Linux. Es una de las múltiples aportaciones que se han hecho por voluntarios a este Sistema Operativo. Es una imitación de Wordstart o de editor de entornos Borland, como por ejemplo el editor del Turbo Pascal.

El inicio de **joe** es: **joe fichero**.

Para invocar a la pantalla de ayuda se pulsa ^KH. La pantalla de ayuda es la siguiente:

CURSOR	GO TO	BLOCK	DELETE	MISC	EXIT
^B left ^F right	^U prev. screen	^KB begin	^D char	^KJ reformas	^KX save
^P up ^N down	^V next screen	^KK end	^Y line	^T options	^C abort
^Z previous word	^A beg. of line	^KM move	^W > word	^R refresh	^KZ shell
^X next word	^E end of line	^KC copy	^O word <	^@ insert	FILE
SEARCH	^KU top of file	^KW file	^J >line	SPELL	^KE edit
^KF find text	^KV end of file	^KY delete	^– undo	^[N word	^KR insert
^L find next	^KL to line No	^K/ filter	^^redo	^[L file	^KD save

Resumen de comandos joe:

MOVIMIENTO DEL CURSOR	
Comando	Función
^B	Mueve el cursor una posición a la izquierda
^F	Mueve el cursor una posición a la derecha
^P	Mueve el cursor una posición hacia arriba
^N	Mueve el cursor una posición hacia abajo
^Z	Mueve el cursor una palabra hacia la izquierda
^X	Mueve el cursor una palabra hacia la derecha
^U	Muestra la pantalla anterior a la actual

^V	Muestra la pantalla siguiente a la actual
^A	Mueve el cursor al principio de la línea
^E	Mueve el cursor al final de la línea
^KU	Mueve el cursor al principio del fichero
^KV	Mueve el cursor al final del fichero
^KL	Mueve el cursor a un número de línea concreto

BÚSQUEDA Y SUSTITUCIÓN

Comando	Función
^KF	Busca el texto especificado
^L	Repite búsqueda

BLOQUES

Comando	Función
^KB	Marca el inicio de un bloque
^KK	Marca el final de un bloque
^KC	Copia en la posición actual del cursor el bloque marcado
^KM	Mueve a la posición actual del cursor el bloque marcado
^KW	Graba el bloque marcado en un fichero

BORRADO

Comando	Función
^KY	Borra el bloque marcado
^D	Borra el carácter de la posición del cursor
^Y	Borra la línea actual del cursor
^W	Borra la palabra de la derecha
^O	Borra la palabra de la izquierda
^J	Borra desde la posición actual del cursor hasta el final de la línea

SALIDA Y GRABACIÓN

Comando	Función
^KX	Abandona la edición grabando
^C	Abandona el editor. Pregunta si abandona los cambios al fichero. Contestar `Y' para no grabar
^KZ	Abandona la edición sin grabar y sin preguntar
^KE	Edita otro fichero, al pulsar <^C> vuelve al fichero original
^KR	Lee otro fichero y lo inserta en la posición actual del cursor
^KD	Graba el fichero y continúa con la edición

OTRAS ACCIONES

Comando	Función
^K/	Ejecuta un comando unix
^T	Muestra una lista de opciones del editor. Están a OFF/ON, pulsando <intro> o <barra espaciadora> se activan o desactivan. Con flechas de desplazamiento se recorre la lista. Con

	<^C> se abandona la lista de opciones. Esta lista contiene: activar inserción, autoindentación, tabulación, márgenes, etc
--	---------------------------------------------------------------------------------------------------------------------------

El editor jed.

Este editor es una incorporación de Linux, lo realizó John E. Davis entre 1992 y 1993. Fue escrito en un principio en cc, posteriormente en gcc y hay versiones en BCC 3.0 para PC

jed como editor es sencillo, mas adelante veremos sus principales comandos, pero **jed** es más que un editor.

jed puede usarse como asistente para ayuda en muy diversas tareas, por ejemplo:

- Programar en c, c++, smalltalk, lisp, etc.
- Uso de emacs.
- Usos de los diferentes shell's de Linux.
- Instrucciones sobre como contribuir al proyecto GNU.
- Explica como jed puede emular a emacs y wordstar.

Este editor se invoca poniendo: **jed fichero**

USO DEL EDITOR	
Comando	Función
^X^C	Salir de jed
^V	Página arriba
<esc>V	Página abajo
^D	Borra el carácter bajo el cursor
^X^W	Graba el fichero
^X^F	Abre un nuevo fichero
^A	Va al principio de la línea
^E	Va al final de línea

Como hemos dicho antes, **jed** tiene un uso más interesante que el de editor. En concreto es utilizarlo como asistente para ayuda. A continuación se explica cómo moverse por la ayuda:

ASISTENTE PARA AYUDA	
Comando	Función
^H	Invoca a este asistente para ayuda. Aparece un pequeño menú
I	Es la opción del menú que muestra el asistente. Para elegir una de las opciones hay que situarse con el cursor en la línea elegida y pulsar <intro>
Q	Sale de la ayuda y vuelve a la edición, desde el menú principal
D	Vuelve al menú principal.
H	Muestra un acerca de la ayuda

18

Tema 3 Edición de textos Página:3/17

10

MODO EX

MODO COMANDO

MODO TEXTO

: /

a, A i,I o,O

<ESC>

<ESC>

MODO TEXTO

MODO COMANDO

MODO EX