PRÁCTICA DE LABORATORIO 5

EDITOR DE TEXTO NANO

En este tutorial vamos a conocer el editor de texto en línea de comandos Nano, atajos de teclado o accesos directos. Los editores de texto de la terminal son especialmente útiles cuando estamos conectados de manera remota a una terminal Linux como un servidor.

Ayuda del editor

En una terminal vamos a escribir el siguiente comando:

```
nano –help
```

La terminal nos mostrará la siguiente salida:

Con este comando conocerán todos los parámetros adicionales que podemos utilizar a la hora de trabajar con el editor. Los más destacados son:

Opciones al iniciar el editor

-E ó –tabstospaces

Esto convertirá todas las tabulaciones a espacios.

```
nano -E <archivo>
nano --tabstospaces <archivo>
```

-T ó –tabsize

Esta opción permite definir el número de columnas que conforman una tabulación. Como segundo parámetro recibe en número de columnas.

```
nano -T 2 <archivo>
nano --tabsize 2 <archivo>
```

-S ó –smooth

Con este parámetro se define el modo de desplazamiento vertical en archivos extensos. Por defecto el desplazamiento es de la mitad de la pantalla, con **smooth** se vuelve de línea por línea.

```
nano -S <archivo>
nano --smooth <archivo>
```

Combinemos múltiples opciones

```
nano -ES -T 2 <archivo>
nano --tabstospaces --tabsize 2 --smooth <archivo>
```

Opciones dentro del editor

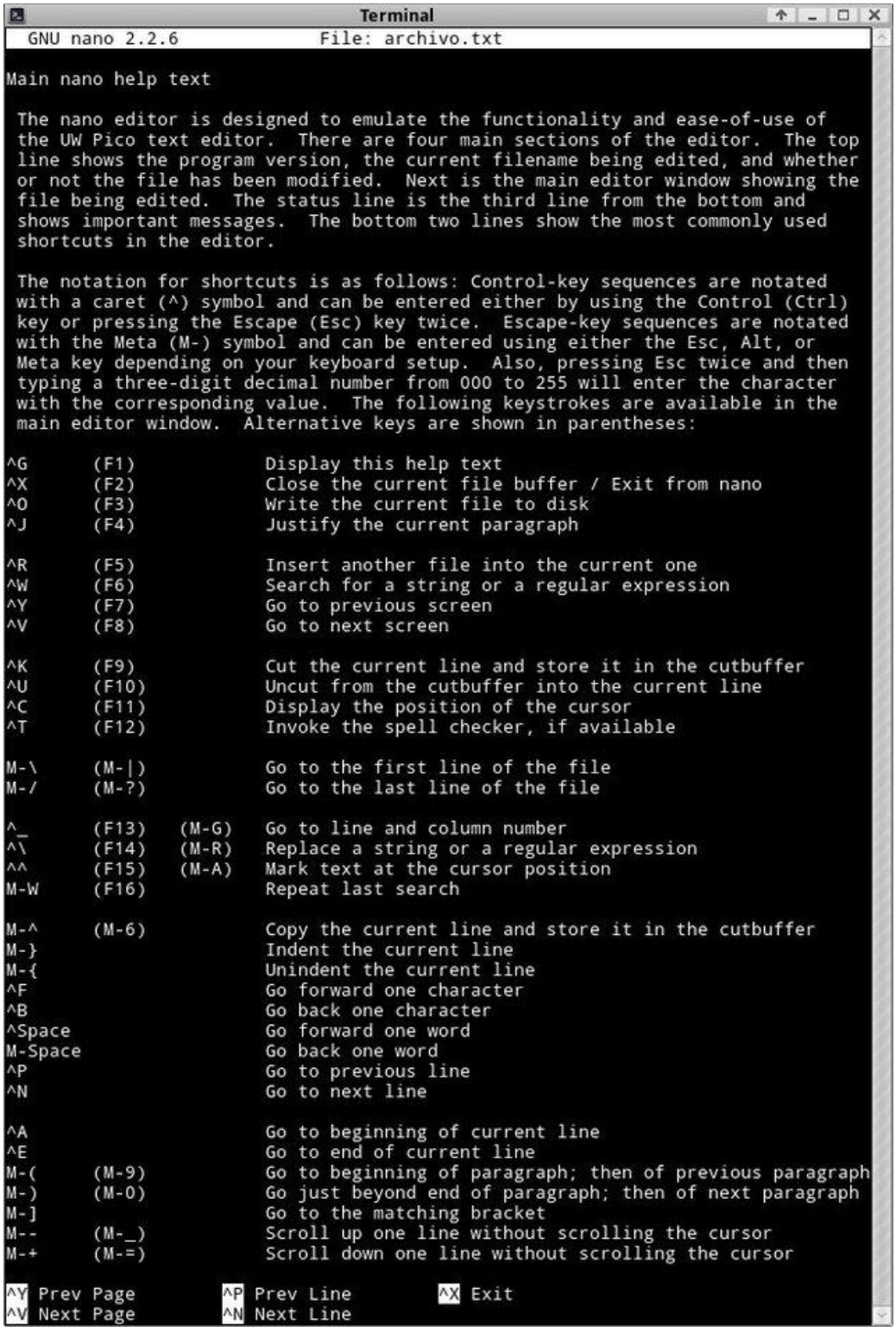
Dentro del editor tenemos muchas combinaciones de teclado disponibles para sus diferentes funciones.

La tecla **Control**, que vamos a definir con el caracter **"^"** nos será útil para cualquiera de estos atajos de teclado o accesos directos.

En la parte inferior aparece una barra donde se muestran las funciones más utilizadas. Vamos a dar un repaso en algunas de ellas.

Ayuda o ^G

Este comando nos muestra todas las funciones y su acceso directo mediante el teclado.
Para salir de esta pantalla ejecutaremos la combinación de teclado ^X



Combinaciones más utilizadas

```
^X          Salir del editor
^C          Mostrar la posicion actual (linea/columna)
```

^O	Guardar
^W	Buscar texto
^\	Buscar y reemplazar
^/	Ir a línea, columna
^K	Cortar la línea actual
^U	Pegar en la línea actual
^ALT+K	Cortar múltiples líneas
^Y	Subir
^V	Bajar

Es importante saber que siempre contamos con la barra inferior. En algunas funciones, puede que se habiliten nuevas opciones en ella.

MANUAL PARA APRENDER A UTILIZAR VIM

Vim es un editor de ficheros de textos muy versátil, que dispone de una gran flexibilidad a la hora de escribir scripts, modificar ficheros de texto, etc. pero sobretodo, a la hora de **programar**.

Las ventajas, son múltiples. VIM ocupa muy poco y existe en prácticamente todos los Linux o Unix disponible. Al ser un programa que se ejecuta en entorno de texto es útil para accesos remotos y edición vía terminal.

En este tutorial se verá el uso básico desde cero y comprobar lo útil que es aprender a usarlo.

VIM: Introducción

No todas las distribuciones de Linux vienen con Vim instalado. Para comprobarlo debemos ejecutar el siguiente comando:

```
vim <archivo>
```

Si no tenemos el programa instalado debemos ejecutar el siguiente comando:

```
sudo apt-get install vim
```

Una vez instalado, arrancar el vim es muy sencillo. Sólo hay que escribir en una terminal **vim**, seguido del nombre del fichero a editar. Nos aparecerá una ventana en negro, donde nos aparecerá el contenido del fichero (*o en negro si está vacío*). En la parte inferior, nos aparecerán los mensajes o comandos que escribamos para manejar el editor, así como la línea en la que estamos, porcentaje del fichero, etc.

Lo primero que hay que aprender de **Vim** (*muy importante*) es que tiene varios modos de uso:

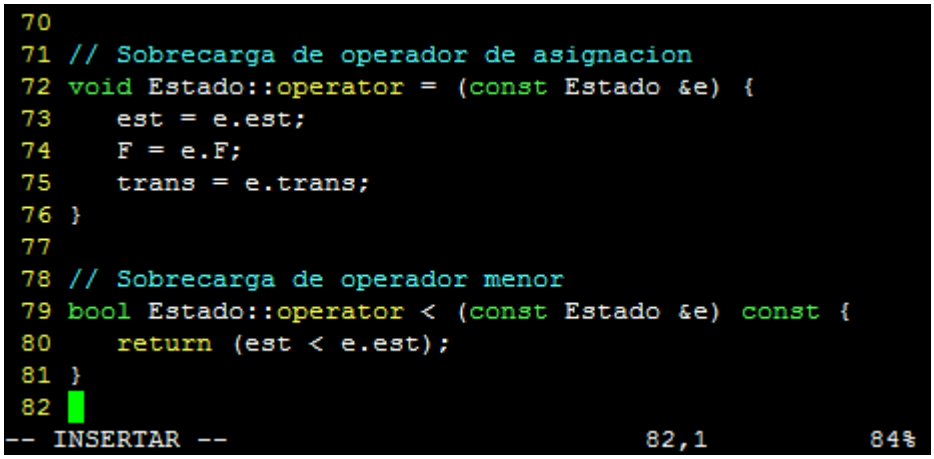
- Nada más entrar en vim a editar un fichero, estamos en el **modo normal**, en el que podremos introducir **atajos** para realizar operaciones (*borrar línea, deshacer, etc.*). **IMPORTANTE**: En este modo no podemos escribir en el fichero. Las teclas que pulsemos probablemente estarán asociadas a una operación determinada. Muchos de estos comandos (*no todos*) comenzarán por **:**.
- Para escribir texto en el fichero tendremos que entrar en el **modo edición**, que es tan fácil como pulsar la tecla **insert** (*o* **i**). Sabrás que has entrado en este modo porque abajo aparecerá el texto **-- INSERTAR --**. Ahora todo lo que tecleemos se estará escribiendo en el fichero de texto. Para volver al **modo normal** sólo hay que pulsar la tecla **ESC** (*Escape*).

Todo esto puede parecer muy lioso al principio, pero conforme comiences a utilizarlo con frecuencia, verás que resulta cómodo y lo haces de forma automática.

Primeras impresiones

Vim reconoce automáticamente por la extensión del fichero, el lenguaje en el que estamos programando (*.C, .sql, .pl, .latex, .php...*), por lo tanto nos hará un **resaltado de sintaxis** con colores, que nos resultará bastante agradable.

Esta opción puede no estar disponible en algunos Linux con versiones minimalistas de **Vim**. Sólo tenemos que instalar la versión completa de vim con **apt-get install vim-common** y escribir (*en el modo normal del Vim*) **:syntax on**.



Sin duda, el resaltado de sintaxis es algo muy valioso para el programador.

Operaciones básicas del editor

Una vez tengamos nuestro texto escrito, necesitaremos saber cómo realizar algunas operaciones como **guardar fichero**, **salir del editor**, etc.

Como hemos dicho antes, para realizar operaciones que no son de escribir en el fichero, necesitamos entrar en el **modo normal** (*pulsando ESC si estamos en el modo edición*) y a continuación los atajos que queramos:

Secuencia	Significado	¡Mnemotécnica!
:q	Salir del editor sin guardar	quit
:q!	Salir del editor sin guardar ni pedir confirmación	quit ya!
:wq!	Salir del editor guardando sin pedir confirmación	write & quit ya!
:w *f2.txt*	Guardar en un fichero llamado **f2.txt** y seguir	write en f2.txt
:e *f1.txt*	Cierra el fichero actual y abre **f1.txt**	edit f1.txt

Operaciones básicas de texto

En **Vim** como en cualquier editor, necesitaremos manipular rápidamente texto, y algo que enseguida se echa en falta en VIM, son las famosas opciones **Cortar, Copiar y Pegar**.

Con los cursores nos desplazamos por el contenido del fichero hasta llegar al inicio de la zona que queremos copiar. Pulsamos **ESC** (*si estamos en el modo edición*) y la tecla **V** para entrar en el **modo visual** y nos desplazamos hacia el final de la zona que queramos copiar. Se verá que se remarca en otro color la zona seleccionada.

Una vez tengamos la zona a copiar seleccionada, sólo tenemos que pulsar C (para cortar) o Y (para copiar). Nos aparecerá abajo un mensaje X lineas copiadas.

Ahora sólo tenemos que desplazarnos a donde queramos pegar ese fragmento y pulsar (*como siempre, en el modo normal, no en el modo edición*) la tecla **P** (*pegar*).

Veamos más operaciones de texto:

Secuencia	**Significado**	**¡Mnemotécnica!**
dd	Suprimir línea actual al buffer (*p para pegar*)	delete
u	Deshacer el último cambio en el fichero	undo
CTRL+R	Rehacer el último cambio en el fichero	redo
guu	Convertir a minúsculas la línea actual	lowercase
gUU	Convertir a mayúsculas la línea actual	UPPERCASE
:*num*	Posicionarse en la línea *num* del fichero	
gg	Posicionarse al principio del fichero	
G	Posicionarse al final del fichero	

ga	Muestra código ASCII, hex y octal del caracter actual

Operaciones de búsqueda y sustitución

Otra función que solemos echar de menos enseguida es la de buscar algún texto, reemplazar, etc. En **vim** no puede faltar esa opción, con sus respectivas mejoras y añadidos:

Para buscar un texto, escribimos (*en modo normal, pulsando antes* **ESC** *si estamos en modo edición*) la secuencia */palabra*. Veremos que se resalta la palabra encontrada (*o nos avisa de que no existe*). Entonces podemos seguir buscando la próxima coincidencia pulsando **n** o buscarla hacia detrás pulsando **N**.

Para sustituir un texto debemos escribir la secuencia **:%s/***texto1***/***texto2***/g**, donde **texto1** es el texto a buscar y **texto2** el texto que será reemplazado. Si incluimos la **g** final (*global*), sustituirá todas las coincidencias que encuentre, sino sólo la primera que encuentre.

ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"

EJERCICIOS DE SISTEMAS DE ARCHIVOS EN UBUNTU

1. Suponiendo que acabamos de ingresar a una terminal y que el directorio donde nos encontramos (~) está vacío. Escriba los comandos que deben ser ejecutados en la terminal para realizar lo siguiente:
   a) Crear un archivo llamado "`practica5`" en el directorio `~/esfot/2016B/materias/SO1/`
   b) Abrir, editar y guardar el archivo "`practica5`" con el contenido "`practica5 Bimestre 1`"
   c) Crear otro fichero llamado "`planificación`" dentro del directorio "`2016B`"
   d) Mover el fichero "`practica5`" al directorio `~/esfot/2016B/materias/SO1/bimestre1/`
   e) Suponer que estamos ubicados en el directorio `~/esfot/2016A/materias/SO1/bimestre1/` crear un directorio llamado "`poo`" en el directorio `~/esfot/2016B/materias/` (utilizar rutas relativas).
   f) Mostrar el contenido que tiene el directorio `esfot`, en forma de lista y en forma de árbol de directorios.
   g) Mostrar todos los contenidos que tiene el directorio `/etc` incluir los directorio y ficheros ocultos y todos los subdirectorios que hay dentro de cada directorio de manera recursiva.
   h) Mostar una lista detallada de los contenidos del directorio "`materias`" ordenados por tamaño de mayor a menor. Mostrar los permisos, propietario, fecha de creación, nombre, etc.
   i) ¿Qué sucede si intenta crear un directorio dentro del directorio raíz? ¿por qué sucede esto?
   j) Ubicarse en el directorio personal (~). Crear la siguiente estructura de directorios:

```
~
|__ gastos
      |__ noviembre
            |__ casa.txt
            |__ salidas.txt
      |__ diciembre
            |__ casa.txt
            |__ salidas.txt
```

   - ¿Qué sucede si intenta crear el directorio "`diciembre`" antes del directorio "`gastos`"?
   - ¿Qué comando debe utilizar si se desea crear la estructura de directorios en una sola línea de comandos?
   - Liste 3 opciones para crear los directorios `casa.txt` y `salidas.txt`.