



Tutorial

Violeta González Pérez

2022

Linux

- Linux es un sistema operativo con código abierto que permite hacer todo desde la línea de comandos (aunque puedes utilizar los iconos también).
- En general, en linux, el tabulador completará los nombres de comandos, rutas con directorios y ficheros.
- Si tu ratón tiene una ruleta central, puedes seleccionar texto con el ratón y pegarlo apretando la ruleta centra.
- Vamos a explorar comandos en la terminal. Puedes abrir una pulsando el icono o con `Ctrl-Alt-t`.

Explora los comandos anteriores

Abre una terminal de las dos formas indicadas. Copia una palabra de tu ordenador a la terminal de la máquina virtual. Comienza a escribir 'wh' y pulsa el tabulador, ¿qué opciones te aparecen?.

Comandos básicos en linux - I

pwd Muestra la ruta al directorio en el que estás.

ls Muestra lo que hay en el directorio en el que estás.

ls -al Muestra lo que hay en el directorio, incluyendo ficheros escondidos '.' e información detallada sobre los ficheros.

which NOMCOM Si el comando NOMCOM existe, esto nos mostrará la ruta hasta dicho comando.

mkdir DIRNAME Genera un directorio DIRNAME.

Explora los comandos anteriores

Comprueba que el comando `mkdir` existe y genera un directorio de prueba.

Comandos básicos en linux - II

cd DIRNAME Cambia al directorio DIRNAME.

cd .. Cambia al directorio *superior*.

cd Cambia a tu *home* (hace lo mismo el comando **cd ~/**).

emacs FICHERO & **emacs** es un editor de textos del que hablaremos en un momento, con este comando abrimos un fichero *de fondo*.

more FICHERO Muestra las primeras líneas de un fichero, para seguir hacia abajo pulsa espacio y para salir del comando pulsa q.

Explora los comandos anteriores

Abre un fichero con **emacs**, escribe algo y guárdalo. ¿Qué ocurre si quitas **&** del comando anterior? ¿Puedes correr **more FICHERO** en la terminal?

Comandos básicos en linux - III

cp FICHERO1 FICHERO2 Copia el fichero FICHERO1 en un nuevo fichero, FICHERO2.

mv FICHERO1 FICHERO2 Cambia el nombre del fichero de FICHERO1 a FICHERO2.

rm -i FICHERO Borra el FICHERO completamente, preguntándote si de verdad lo quieres borrar.

history Muestra una lista de los comandos utilizados en la terminal. Si sólo necesitas acceder a los últimos comandos, puedes utilizar las flechas arriba y abajo.

Explora los comandos anteriores

¿Qué pasa si borras un fichero sin la opción `-i`?

El comando grep de búsqueda

grep **STRING** **FICHERO** Busca 'STRING' en un FICHERO.

grep **STRING** ***.cpp** Busca 'STRING' en todos los ficheros con terminación '.cpp' que haya en este directorio.

grep -i **STRING** **../*.cpp** Busca 'STRING', sin importar que la palabra esté escrita en mayúsculas o minúsculas, en los ficheros con terminación '.cpp' que haya en el directorio superior.

grep -help Nos da información sobre el comando grep y esto sirve para cualquier comando.

man grep Abre la entrada sobre grep en el manual de linux. Esto sirve para cualquier comando. Para salir, pulsa q.

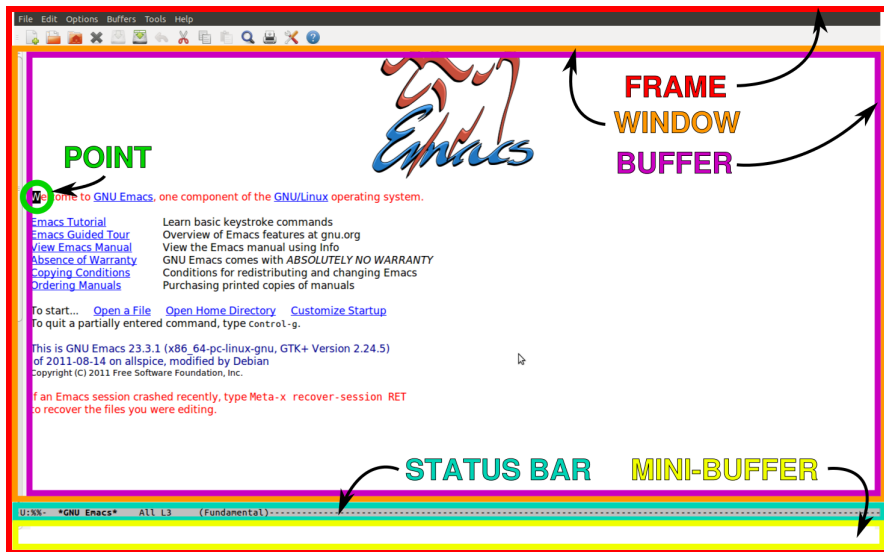
Explora los comandos anteriores

Utilizando emacs, genera varios ficheros '*.cpp' y utiliza grep para hacer búsquedas o bien de números o de letras y conjuntos de letras.

Editores de texto en Linux

Linux suele venir con varios editores de texto básico por defecto, como `nedit` y `gedit`. Sin embargo, para escribir código, es mejor utilizar editores más potentes como `vi` o `emacs`. También podrías utilizar un IDE (entorno de desarrollo integrado) para C++, pero esto lo tendrías que instalar cada vez si estuviésemos en las aulas de informática de la facultad (y puede que no tengas permiso para ello).

Emacs



Comandos básicos de emacs

Abre un fichero con emacs y después de escribir algo, prueba los siguientes comandos:

Ctrl-x Ctrl-s Guardar.

Ctrl-x Ctrl-c Cerrar.

Ctrl-x u Deshacer.

Esc < Ir al inicio del fichero.

Esc > Ir al final del fichero.

Ctrl-a Ir comienzo de la línea en la que está el cursor.

Ctrl-k Corta la línea en la que está el cursor.

Ctrl-w Corta la selección.

Ctrl-y Pega lo que se haya cortado anteriormente.

Ctrl-x 1 En el caso de tener varias 'pantallas' de emacs abiertas, cierra las 'pantallas' en las que no está el cursor.

Ctrl-g Deshabilitar el comando que se haya empezado a escribir.

Buscar y reemplazar en emacs

Para buscar:

1. **Ctrl-s** para buscar hacia adelante/ **Ctrl-r** para buscar hacia atrás.
2. Escribe lo que quieras buscar en el 'mini-buffer'.
3. Pulsa **Ctrl-s** (o **Ctrl-r**) para ir a la siguiente entrada encontrada.
4. Sal pulsando las flechas arriba o abajo.

Para reemplazar (esto no funciona en VirtualBox pero se puede hacer desde 'Edit'):

1. **Esc x**
2. En el 'mini-buffer' escribe 'replace-string' (puedes utilizar el tabulador para autocompletar).
3. Escribe en el 'mini-buffer' lo que quieras cambiar y pulsa **Enter**.
4. Escribe en el 'mini-buffer' el nuevo contenido y pulsa **Enter**.

Explora los comandos anteriores

En un fichero con texto busca las , ¿qué ocurre si no hay ninguna?, y cambia las b por v.

Cambiar o borrar un rectángulo en emacs

1. Con el cursor en el vértice superior izquierdo del rectángulo a modificar, pulsa `Ctrl-Space`.
2. Con el cursor en el siguiente espacio al vértice inferior derecho del rectángulo a modificar, pulsa
 - `Ctrl-r k` para borrar
 - `Ctrl-r t` para cambiar todo el rectángulo por lo que escribas en el 'mini-buffer' (pulsa `Enter` cuando hayas terminado de escribir).

Explora los comandos anteriores

En un fichero con al menos 4 filas, modifica la primera columna de manera que contenga '%'.
'%'

Buenas prácticas programando

1. Documenta tu código explicando el concepto de cada parte. En el caso de funciones, es recomendable explicar qué son las variables de entrada y salida.
2. Realiza copias de seguridad al final de cada sesión. De forma ideal, esto se hace con un código de **control de versiones** como **git**.

Introducción rápida a git

git es una herramienta para **controlar las versiones** de tu código o de tus documentos. **git** te permite trabajar con distintas versiones de un mismo código, facilitando el desarrollo colaborativo, y puedes realizar copias directas de seguridad en los servidores de **Bitbucket** y **GitHub** (estos servidores tienen limitaciones, guarda sólo lo importante). Estos son los comandos básicos que necesitarás en tu día a día:

1. Copia localmente un repositorio que ya exista en un servidor remoto:

```
$ git clone git@github.com:[gituser]/[repo_name].git  
$ cd [repo_name]
```

2. Añade ficheros a un repositorio (equivalente a un directorio):

```
$ git add [file]
```

3. Guarda tus cambios localmente:

```
$ git commit -am "[Description of changes]"
```

4. Guarda tus cambios en un servidor remoto:

```
$ git push
```