

## Práctica 1. Complemento.

### Comandos de Linux y Editor vi.

#### **Objetivo:**

Aprender los comandos básicos para la manipulación de ficheros y directorios dentro de un determinado Sistema Operativo, en este caso Linux.

#### **Introducción.**

Para desenvolverse en cualquier entorno es imprescindible tener unas nociones básicas de la manipulación de ficheros y directorios que el Sistema Operativo con el que estamos trabajando nos ofrece.

Dentro de un entorno como Windows, es sencillo realizar todo este tipo de tareas cotidianas, ya que con los entornos de ventanas gráficas, las operaciones se limitan a simples acciones de ratón o teclado.

Linux también nos ofrece una interfaz gráfica que nos permite manipular todos los elementos a nuestra disposición, pero no siempre es así, con lo que resulta necesario conocer los comandos básicos para poder manipular nuestra información por medio del uso de la línea de comandos en modo texto.

#### **Arranque Linux.**

Vamos a arrancar nuestro ordenador y elegiremos la opción de **Linux** dentro del menú de opciones de Sistemas Operativos al inicio.

Una vez ya en este entorno, se nos pedirá un login y password para entrar. Este es:

**login:** ITTT-FT

**password:** ITTT-FT2004

Una vez aceptado el login/password entraremos al modo gráfico de Linux.

Dentro de este modo es posible manipular los ficheros y directorios y acceder a la información por medio de los gestores de ficheros y los editores en X-Windows. Pero nos interesa conocer como realizar todas estas operaciones en modo texto.

Pasaremos a modo texto mediante las siguientes opciones:

- *Cambio modo texto:* **CTRL+ALT+F1** (se abre una shell, identificada como shell 1)  
Tenemos disponibles 6 shells simplemente utilizando de **CTRL+ALT+[F1-F6]** para cambiar entre ellas.
- *Cambio modo gráfico:* **CTRL+ALT+F7**

#### **Comandos de la shell.**

Una vez en modo texto podemos probar los diferentes comandos básicos que nos proporciona la shells de linux:

- **man:** Muestra información del comando que se le pide.  
*Estructura:* man comando (ejemplo: man ls)  
Pulsando "q" saldremos del editor del man.
- **ls:** Listado de contenidos de directorios.  
*Estructura:* ls nombre directorio (solo muestra el nombre de los ficheros y directorios)  
*Opciones:* ls -l (descripción del contenido), ls -a (todo, incluso fichero ocultos o sistema)  
ls -t (orden por fecha modificacion), ls -S (orden por tamaño).

## Prácticas de Fundamentos de Telemática

### Práctica 1 (complemento) (Lucía Vera)

- **pwd:** Información donde nos encontramos, es decir de la ruta de directorios actual.
- **cp:** Copia ficheros/directorios en el destino especificado.  
*Estructura:* cp fichero fichero\_nuevo (o bien directorio/fichero\_nuevo)  
cp directorio directorio\_nuevo  
*Opciones:* cp -rfp directorio dir\_destino (copia recursivamente y manteniendo permisos)
- **rm:** Borrar ficheros/directorios.  
*Estructura:* rm fichero/directorio  
*Opciones:* rm -r directorio (permite borrar de manera recursiva el contenido del directorio)
- **mkdir:** Crea directorios.  
*Estructura:* mkdir nombre\_directorio
- **rmdir:** Borra un directorio, solo si está vacío.  
*Estructura:* rmdir nombre\_directorio
- **mv:** renombrar o mover ficheros/directorios.  
*Estructura:* mv directorio nuevo\_dir, mv fichero nuevo\_fichero, mv fichero directorio
- **more:** Muestra el contenido de un fichero con pausas cada 25 líneas.  
Con Enter iremos bajando línea a línea. Con el espacio pasamos bloques de 25 líneas.  
Con “q” saldremos de la edición del fichero.
- **grep:** Filtra el contenido de un fichero. Busca información contenida dentro de ficheros.  
*Estructura:* cat fichero | grep cadena  
grep cadena fichero (o \*.extension para buscar dentro de todos los ficheros con esa extensión)  
*Opciones:* grep -n cadena fichero (indica la línea en la que está la cadena en el fichero).
- **cat:** Muestra todo el contenido de un fichero sin pausa alguna.  
*Estructura:* cat fichero
- **chmod:** Cambia los permisos de lectura/escritura/ejecucion de fichero/directorios.  
*Estructura:* chmod +r fichero, chmod +w directorio, chmod +rw fichero, chmod -rw fichero  
*Opciones:* chmod [opciones][+][rwx] fichero  
chmod [opciones][+][rwx] directorio [-R] (recursivamente)
- **find:** Permite buscar ficheros/directorios dentro del sistema.  
*Estructura:* find [camino] [expresión]  
find / -name nombre\_buscado -print

## Editor de texto. El VI

En modo texto el editor más extendido y estándar es el **vi** (abreviatura de "visual"). Este editor provee capacidad de edición de texto básica. Hay tres aspectos de vi que lo hacen atractivo. Primero, vi es proporcionado con todos los sistemas UNIX. Segundo, vi usa una pequeña cantidad de memoria, lo cual permite una operación eficiente en cualquier momento. Tercero, porque usa teclas alfanuméricas estándares para los comandos, se puede usar en alguna terminal virtual o una estación de trabajo sin tener que preocuparse por mapeos de teclas inusuales.

### ➤ Arrancar VI

Para arrancar el VI simplemente abrir una shell de linux o bien en el modo texto de linux, teclear:

**# vi**

para que se abra el editor sin fichero asociado y sin nombre para el fichero a crear. Se le podrá dar cuando se salve desde dentro del editor.

**# vi nombre\_fichero**

para abrir un fichero existente o para crear un fichero con un nombre específico.

vi usa el cursor para indicar donde su próximo comando o inserción de texto tomará efecto. Al pie de la ventana, mantiene una línea de estado, llamada línea de modo. Esta información muestra la línea actual del archivo, el nombre del archivo, y el estado.

En muchos sistemas operativos basados el UNIX se puede usar una versión mejorada del vi, el vim (VI-IMProved, o vi mejorado). Nosotros aprenderemos el entorno básico del vi que es extensible a cualquiera de las versiones posteriores a él.

### ➤ Modo comando y Modo inserción

vi tiene dos modos, *modo comando* y *modo inserción*. En el modo comando, los caracteres que se escriban realizan funciones tales como movimiento del cursor, cortar o copiar texto, o buscar algún texto particular. En el modo inserción, se escribe para insertar o sobrescribir texto. Cuando arranca vi lo hace por defecto en modo comando.

Para cambiar desde modo comando a modo inserción, presione la tecla **"i"** (no es necesario presionar RETURN). vi le permite insertar texto comenzando desde la ubicación actual del cursor. Para volver a cambiar a modo comando, presione la tecla **ESC**. Puede también usar **ESC** para cancelar un comando incompleto en el modo comando.

Si el editor no muestra por defecto el modo en el que está es posible activar esta opción mediante la instrucción indicada en modo comando:

**:set showmode (+return)**

Normalmente se indica el modo de inserción solamente, el de comando no se especifica, ya que se considera por defecto.

### ➤ Insertando texto

En modo inserción, puede escribir texto normalmente. vi reconoce unas cuantas combinaciones de teclas especiales dentro de este modo.

<b>Backspace</b>	Borra el carácter previo
<b>Delete</b>	Borra el carácter corriente
<b>CTRL-W</b>	Borra la palabra previa
<b>CTRL-U</b>	Borra la línea actual
<b>RETURN</b>	Comienza una línea nueva

## Prácticas de Fundamentos de Telemática

### Práctica 1 (complemento) (Lucía Vera)

#### ➤ **Movimiento de cursor**

Podemos mover el cursor en modo comando. vi tiene algunos comandos diferentes para mover el cursor. Puede también usar la tecla backspace y barra espaciadora para mover a izquierda y derecha, respectivamente.

<b>k</b>	mueve una línea arriba (similar a la tecla ↑)
<b>h</b>	mueve un carácter a la izquierda sobre la línea (similar a la tecla ←)
<b>l</b>	mueve un carácter a la derecha sobre la línea (similar a la tecla →)
<b>j</b>	mueve una línea abajo (similar a la tecla ↓)

En las últimas versiones del VI (en el caso del VIM, (vi mejorado)) es posible desplazarse por el texto por medio de los cursores.

Otros posibles comandos:

<b>b</b>	Comienzo de la palabra previa
<b>w</b>	Comienzo de la próxima palabra
<b>e</b>	final de la corriente/próxima palabra
<b>0</b>	(cero) o ^ comienzo de la línea
<b>\$</b>	fin de la línea
<b>(</b>	comienzo de la corriente/previa sentencia
<b>)</b>	comienzo de la próxima sentencia
<b>{</b>	Comienzo del corriente/previo párrafo
<b>}</b>	fin del corriente párrafo
<b>H</b>	línea del tope de pantalla
<b>M</b>	línea del medio de la pantalla
<b>L</b>	última línea de la pantalla

#### ➤ **Borrar texto.**

<b>x</b>	borra solo el carácter actual.
<b>D</b>	borra hasta el final de la línea.
<b>db</b>	borra desde el carácter actual hasta el comienzo de la palabra actual.
<b>de</b>	borra desde el carácter actual hasta el final de la palabra actual.
<b>dd</b>	borra la línea actual.
<b>dw</b>	borra desde el carácter actual hasta el comienzo de la próxima palabra.

#### ➤ **Deshacer acciones.**

Para deshacer el último cambio realizado, se debe teclear:

<b>u</b>	deshacer la última acción realizada.
----------	--------------------------------------

#### ➤ **Grabar y salir del editor.**

vi tiene varias maneras de grabar los cambios y de salir del editor. Además de guardar su trabajo antes de salir, también es buena idea guardar su trabajo periódicamente. Para grabar el archivo actual se debe teclear en modo comando ("w" de "write"):

**:w** (return)

También se puede indicar el nombre del archivo con el que se quiere guardar el documento:

**:w nombre\_fichero** (return)

De manera similar, para salir de vi se usa el comando:

**:q** (return)

## Prácticas de Fundamentos de Telemática

### Práctica 1 (complemento) (Lucía Vera)

Si lo que se quiere es grabar el fichero y salir se usa:

**:wq** (return) o **:wq nombre\_fichero** (return)

En algunos casos se quiere salir sin guardar los cambios, con lo que se usa

**:q!** (return)

#### ➤ **Otras opciones.**

- *Mostrar número de línea:*  
**:set number** (return)
- *Mueve (corta) la línea número, line1, a la línea que se indica luego en destline j:*  
**:line1mdestline**
- *Mueve (corta) líneas entre las líneas line1 y line2, hasta el número de línea, destline:*  
**:line1,line2mdestline**
- *Transfiere (copia) la línea numero, line1, a la línea que se indica en, destline:*  
**:line1tdestline**
- *Transfiere (copia) líneas entre las líneas line1 y line2, en el número de línea, destline:*  
**:line1,line2tdestline**
- *Buscar:*  
Buscar hacia delante con la barra (/) o hacia atrás con el signo de interrogación (?).  
Se escribe el comando luego el string seguido de RETURN. Para cancelar la búsqueda, presione ESC en vez de RETURN.  
Se puede hacer una nueva búsqueda escribiendo n (hacia delante) o N (hacia atrás).  
También, cuando vi alcanza el final del texto, continua buscando desde el comienzo.

**Prácticas de Fundamentos de Telemática**  
**Práctica 1 (complemento) (Lucía Vera)**

**Comandos básicos de la shell**

<b>Comando/Sintaxis</b>	<b>Descripción</b>	<b>Ejemplos</b>
cat <i>fich1</i> [ <i>..fichN</i> ]	Concatena y muestra un archivos	cat /etc/passwd
	archivos	cat dict1 dict2 dict
cd [ <i>dir</i> ]	Cambia de directorio	cd /tmp
chmod <i>permisos fich</i>	Cambia los permisos de un archivo	chmod +x miscript
chown <i>usuario:grupo fich</i>	Cambia el dueño un archivo	chown nobody miscript
cp <i>fich1...fichN dir</i>	Copia archivos	cp foo foo.backup
diff [- <i>e</i> ] <i>arch1 arch2</i>	Encuentra diferencia entre archivos	diff foo.c newfoo.c
du [- <i>sabr</i> ] <i>fich</i>	Indica el tamaño del directorio	du -s /home/
echo <i>cadena</i> / <i>\$variable</i>	Muestra texto o valores de variables	echo \$HOME
file <i>arch</i>	Muestra el tipo de un archivo	file arc_desconocido
find <i>dir test acción</i>	Encuentra archivos.	find . -name ``.bak" -print
grep [- <i>clnv</i> ] <i>expr archivos</i>	Busca patrones en archivos	grep mike /etc/passwd
head - <i>count fich</i>	Muestra el inicio de un archivo	head prog1.c
ls	Lista el contenido de directorios	ls
man <i>comando</i>	Ayuda del comando especificado	man gcc
mkdir <i>dir</i>	Crea un directorio.	mkdir temp
mv <i>fich1 ...fichN dir</i>	Mueve un archivo(s) a un directorio	mv a.out prog1
mv <i>fich1 fich2</i>	Renombra un archivo.	mv .c prog_dir
less / more <i>fich(s)</i>	Visualiza página a página un archivo.	more muy_largo.c
	less acepta comandos vi.	less muy_largo.c
ln [- <i>s</i> ] <i>fich acceso</i>	Crea un acceso directo a un archivo	ln -s /users/mike/.profile .
ls	Lista el contenido del directorio	ls -l /usr/bin
pwd	Muestra la ruta del directorio actual	pwd
rm <i>fich</i>	Borra un fichero.	rm foo.c
rm - <i>r dir</i>	Borra un todo un directorio	rm -rf prog_dir
rmdir <i>dir</i>	Borra un directorio vacío	rmdir prog_dir
tail - <i>count fich</i>	Muestra el final de un archivo	tail prog1.c
vi <i>fich</i>	Edita un archivo.	vi .profile

**Prácticas de Fundamentos de Telemática**  
**Práctica 1 (complemento) (Lucía Vera)**

**Comandos básicos del editor vi**

Comando	Descripción
<ESC>	Mientras se teclea un comando, lo cancela
i	Inserta en la posición del cursor (pasa a modo comando)
a	Inserta tras la posición del cursor (pasa a modo comando)
I	Inserta al inicio de la línea (pasa a modo comando)
A	Inserta al final de la línea (pasa a modo comando)
x	Borra un carácter
r	Reemplaza un carácter
u	Deshace la última operación realizada (se puede repetir para deshacer varias operaciones)
U	Deshace los cambios efectuados sobre la línea actual
:q	Salir del editor
:x	Salir del editor <b>guardando</b> el fichero
:q!	Salir del editor <b>sin guardar</b> el fichero
:w	Guardar el fichero
:w nombrefichero	Guardar el fichero con nombre <b>nombrefichero</b>
nG	Ir a la línea n
\$G	Ir al final del fichero
v	Activa el modo de selección (utiliza los cursores para seleccionar texto)
Y	Copia en memoria (buffer) el texto seleccionado
d	Borra el texto seleccionado, y lo copia en memoria (buffer)
p	Pega el texto copiado en memoria, tras la posición del cursor
P	Pega el texto copiado en memoria, en la posición del cursor
:syntax on	Activa el coloreado de sintaxis
:syntax off	Desactiva el coloreado de sintaxis
/palabra	Busca la cadena <code>palabra</code> hacia adelante
?palabra	Busca la cadena <code>palabra</code> hacia atrás
n	Muestra la siguiente coincidencia de la última búsqueda