

# *Guía del usuario de Unix*

**Ce.Ca.F.I.**  
**Facultade de Informática**  
**Universidade da Coruña**

## ÍNDICE

Pág.

1. Entrar y salir del sistema .....	3
1.1. Entrar en el sistema.....	3
1.2. Salir del sistema .....	3
2. Cómo cambiar tu password .....	3
3. Comandos .....	4
3.1. Introducir comandos .....	4
3.1.1 Corregir comandos .....	4
3.1.2 Repetir comandos .....	4
3.1.3. Redirección y piping .....	4
3.2. Obtener ayuda .....	5
3.3. Los shells.....	6
4. Ficheros y directorios .....	6
4.1. Conceptos fundamentales .....	6
4.2. Comandos de ficheros .....	6
4.2.1. Antes de empezar.....	6
4.2.2. Creando un fichero de prueba (touch) .....	7
4.2.3. Listando ficheros (ls) .....	7
4.2.4. Copiando ficheros (cp) .....	7
4.2.5. Moviendo y renombrando ficheros (mv) .....	7
4.2.6. Borrando ficheros (rm) .....	8
4.2.7. Mostrando contenido de ficheros (more, cat) .....	8
4.2.8. Mostrando el tipo de un fichero (file).....	8
4.2.9. Buscando en ficheros (grep).....	8
4.3. Directorios y jerarquía.....	9
4.3.1. Imprimiendo el directorio de trabajo (pwd) .....	9
4.3.2. Cambiar de directorio (cd) .....	9
4.3.3. Creando un directorio (mkdir) .....	10
4.3.4. Moviendo y renombrando directorios (mv) .....	10
4.3.5. Copiando directorios (cp -r) .....	10
4.3.6. Borrando directorios (rmdir) .....	10

4.4. Seguridad en ficheros y directorios .....	11
4.4.1. Mostrando los permisos .....	11
4.4.2. Listando ficheros ocultos .....	12
4.4.3. Cambiando permisos (chmod) .....	12
4.5. Cuotas de usuario .....	12
5. El editor de textos vi .....	13
5.1. Abrir un documento .....	13
5.2. Los modos de operación de vi .....	13
5.3. Salvar y salir .....	13
5.4. Comandos básicos .....	14
6. El Editor de textos pico .....	14
7. Uso del correo electrónico (mail) .....	15
7.1. Enviando mensajes .....	15
7.2. Recibiendo mensajes .....	15
7.3.- Utilizando PINE .....	16
7.4.- Listas de correo .....	17
8. Uso de la impresora .....	17
8.1. Imprimir un documento .....	17
8.1.1. Pedir aviso de fin de impresión .....	18
8.1.2. Copia de impresión .....	18
8.1.3. Opciones múltiples .....	18
8.2. Conocer el estado de la impresora .....	18
8.3. Cancelar una petición de impresión .....	19
9. Uso de la red .....	19
9.1. Establecer una conexión remota .....	19
9.2. Desconectarse de una maquina .....	19
9.3. Cómo saber quién eres .....	19
10. Variables de entorno .....	20
11. Compilación de programas .....	20

# MANEJO BÁSICO DEL SISTEMA OPERATIVO UNIX

## 1. ENTRAR Y SALIR DEL SISTEMA

### 1.1. Entrar en el sistema

Para entrar en el sistema debes tener una cuenta en él. Además debes conocer tu nombre de entrada (*login*) y tu clave (*password*).

Para entrar en el sistema debe aparecer en la pantalla :

*login:*

Introduce tu *login* y pulsa *Return* . Ahora el sistema te pedirá tu *password* :

*password:*

Introduce tu *password* y pulsa *Return* . Mientras introduces la *password* los caracteres no se ven en la pantalla para que nadie pueda verla.

Si te equivocas aparecerá el mensaje "*login incorrect* " y tendrás que volver a empezar desde el principio.

### 1.2. Salir del sistema

Para salir del sistema teclea *exit* y pulsa *Return* . **NO** se sale del sistema apagando el terminal así que **NO** lo apagues.

La única forma de cerrar tu sesión es mediante el comando *exit* , así que si no lo usas, aunque apagues el terminal tu sesión queda abierta y otro puede acceder fácilmente a ella.

**NO** apagues nunca un terminal ELC (son los que se llaman EO), pase lo que pase.

## 2. COMO CAMBIAR TU PASSWORD

Si deseas cambiar tu *password* , por seguridad o porque simplemente no consigues acordarte de la que tienes, puedes usar el comando *passwd* , en la máquina alba

*alba# passwd*

El sistema te pedirá tu clave antigua, dos veces la nueva y otra vez la antigua o solamente dos veces la nueva.

Es **IMPORTANTE** recordar que Unix distingue entre minúsculas y mayúsculas, así que clave no es lo mismo que ClaVE. Además se recomienda que por tu propia seguridad la clave sea larga e incluya letras minúsculas, mayúsculas y números.

## 3. COMANDOS

### 3.1. Introducir comandos

Para introducir un comando tecléalo y pulsa *Return* . Por ejemplo:

\$ *date* [Return]

\$ es el símbolo del sistema (*prompt* ), aunque como en DOS, puede ser distinto.

Unix hace distinción entre letras mayúsculas y minúsculas: ***date*** es un comando válido, pero ***Date*** no.

#### 3.1.1. Corregir Comandos

Si te equivocas tecleando un comando puedes usar la tecla *Suprimir (Backspace)* para borrar el último carácter, o *CTRL-U* (pulsar las teclas Control y U simultáneamente) para borrar la línea completa.

Puedes utilizar la tecla *Backspace* para rectificar aunque es probable que no te funcione ( escribirá en la pantalla ^H o algo similar). En este caso, puedes usar el comando ***stty*** de la siguiente forma:

\$*stty erase* [Backspace]

A partir de ahí se puede utilizar de forma normal la tecla *Backspace*.

#### 3.1.2. Repetir comandos

Para repetir un comando, se puede utilizar el comando ***r***. Este comando repite el último comando ejecutado. Por ejemplo:

\$*date*

*Wed Sep 16 13:50:14 MET DST 1998*

\$*r*

*date*

*Wed Sep 16 13:50:14 MET DST 1998*

\$

Si le indicas una cadena de caracteres repetirá el último comando que comience con esa cadena de caracteres. Por ejemplo si ejecuto *r cc* repetirá el último comando ejecutado que comience con *cc*:

\$ *r cc*

*cc -o p1 practica1.c*

#### 3.1.3. Redirección y Piping

La salida de un comando puede usarse como entrada de otro fichero (***redirección*** ), o de otro comando

(*piping* ).

Por ejemplo, si quieres hacer un listado de tu directorio actual, y guardarlo en el fichero llamado *Directorio* debes poner:

```
$ ls > Directorio
```

Si el directorio es muy largo y no cabe en una pantalla, puedes hacer un listado (*ls* ) y usarlo como entrada del comando *more* , que coge la entrada y la saca página a página por la pantalla:

```
$ ls | more
```

### **3.2. Obtener ayuda**

Para obtener mas información sobre un comando teclea

```
$ man nombre-del-comando
```

lo que te dará la descripción completa del comando. Si simplemente quieres saber que hace teclea

```
$ whatis nombre-del-comando
```

Si no sabes el nombre del comando, pero tienes una idea de lo que quieres hacer, puedes usar

```
$ apropos palabra-a-buscar
```

que te dará una lista de todos los comandos que tienen algo que ver con esa palabra.

Debes tener en cuenta que hay versiones de Unix de diferentes fabricantes instaladas en las máquinas de la facultad. Dichas versiones son:

- **SunOS Unix** de **Sun Microsystems** en las máquinas riazor, deo y deza.
- **Solaris** de **Sun Microsystems** en la máquina alba.
- **DIGITAL Unix** de **Compaq** en la máquina dexo.
- **SGI Irix** de **Sillicon Graphics** en la máquina silgar.

La información que puedes obtener con los comandos especificados (**man**, **whatis** y **apropos**) dependerá de la máquina a la que estés conectado y la versión de Unix que utilice.

Para saber a qué máquina estás conectado puedes utilizar el comando **hostname**:

```
$ hostname  
alba
```

Para saber qué versión del sistema operativo se está usando:

```
$ uname -a
```

Los comandos que se especifican en este documento (comandos básicos de Unix en general) funcionan igual en todas las versiones salvo pequeños detalles que se indican. Sin embargo, al ser versiones diferentes de Unix algunos comandos no funcionan de la misma manera, por lo que si tienes alguna duda debes consultar con estos comandos en la máquina correspondiente.

NOTA: Los comandos *whatis* y *apropos* no se pueden utilizar en silgar.

### **3.3. Los shells**

Un **shell** es el equivalente Unix del COMMAND.COM del DOS. Es el programa que interpreta los comandos que teclea el usuario y se los pasa al sistema operativo para que los ejecute. En Unix existen varios distintos.

La mayoría de los comandos funcionan igual en unos que en otros, pero algunos comandos sólo funcionan en unos, y otros funcionan en todos pero las opciones se les pasan de forma distinta.

Los más conocidos son el **sh**, el **cs**h y el **ks**h. El que se usa normalmente es **ks**h.

Para saber cuál estás usando teclea:

```
$ echo $SHELL
```

Cuando consultas un comando usando *man*, al describirte el comando te indicarán a que shell se refieren en caso de que sean distintos. Por ejemplo:

```
$ man ls
```

## **4. FICHEROS Y DIRECTORIOS**

### **4.1. Conceptos fundamentales**

El fichero es la unidad básica del Sistema Operativo. Casi todo se trata como fichero, incluyendo:

#### **Documentos:**

Ficheros de texto, tales como cartas o informes, código fuente, o cualquier otra cosa que escribas y quieras grabar.

#### **Comandos:**

La mayoría de comandos son ficheros ejecutables. Por ejemplo, al ejecutar el comando *date* lo que estamos haciendo realmente es ejecutar un fichero con ese nombre y que nos proporciona como salida la hora actual.

#### **Dispositivos:**

El terminal, impresora, unidades de disco, etc., se tratan como ficheros.

#### **Directorios:**

Son *ficheros especiales* que contienen otros ficheros.

### **4.2. Comandos de fichero**

#### **4.2.1. Antes de empezar**

Antes de comenzar a experimentar los comandos, asegúrate de que estás en tu directorio *home*, es decir, aquel que te ha asignado el administrador a la hora de crearte la cuenta, ya que en otro directorio probablemente no tengas permisos de escritura.

Para asegurarte de que vas a tu directorio *home* teclea simplemente *cd* . Esto te moverá directamente al directorio *home*.  
Ahora introduce *pwd* para ver la localización de tu directorio en el sistema de ficheros:

```
$ cd
$ pwd
/export/home/username
```

#### **4.2.2. Creando un fichero de prueba (touch )**

Usa el comando **touch** para crear un fichero vacío. Si el fichero indicado no existe, se creará un fichero vacío, y si ya existe, se modifica la fecha del último acceso al fichero.

```
$ touch tempfile
$
```

#### **4.2.3. Listando ficheros (ls )**

Ahora usaremos el comando **ls** para verificar que lo hemos creado:

```
$ ls tempfile
tempfile
```

Cuando introduces simplemente *ls* obtendrás un listado de todos los ficheros del directorio actual y si lo usas seguido de un nombre de fichero, nos mostrara sólo el nombre de dicho fichero si este existe.

#### **4.2.4. Copiando ficheros (cp )**

Usa el comando **cp** para copiar el fichero *tempfile* a uno llamado *copyfile* :

```
$ cp tempfile copyfile
$
```

Ahora intentaremos visualizar o listar ambos ficheros. Date cuenta de que ambos ficheros terminan en "*file*". Puedes usar el comodín *\** como equivalente de cualquier cadena de caracteres. Así, el comando *ls \*file* nos debería listar ambos ficheros (y cualquier otro fichero del directorio cuyo nombre finalice con *file* ):

```
$ ls *file
copyfile tempfile
```

Fíjate que los ficheros se listan en orden alfabético (Mayúsculas y números preceden a las letras minúsculas).

#### **4.2.5. Moviendo y renombrando ficheros (mv )**

Puedes hacer ambas cosas con el comando **mv** (move). En este ejemplo, usamos el comando *mv* para renombrar *tempfile* a *emptyfile*:

```
$ mv tempfile emptyfile
$
```

Verifiquemos ahora el cambio:



```
$ ls *file
copyfile emptyfile
```

#### **4.2.6. Borrando ficheros (rm )**

Usa el comando **rm** (remove) para borrar *copyfile* y usa *ls* para verificar:

```
$ rm copyfile
$ ls *file
emptyfile
```

**NOTA:** Cualquier fichero borrado no podrá ser recuperado a menos que exista algún backup.

#### **4.2.7. Mostrando contenido de ficheros (more, cat )**

Usa el comando **more** para mostrar el contenido de un fichero. Teclea *more* seguido del nombre del fichero a mostrar. El contenido del fichero se mostrará por la pantalla y si su ocupa más de una pantalla, aparecerá un mensaje similar a:

```
--More--(nn%)
```

donde *nn* es el porcentaje del fichero mostrado. Si quieres cortar el listado por pantalla del fichero puedes usar *CTRL^Z*.

También puedes usar el comando **cat** pero muestra el contenido del fichero sin hacer pausa. A menudo se usa este comando para concatenar dos o más ficheros:

```
$ cat file1 file2 file3 > bigfile
$ ls *file
bigfile
file1
file2
file3
$
```

#### **4.2.8. Mostrando el tipo de un fichero (file )**

Algunos ficheros, tales como ficheros binarios o ejecutables, no son imprimibles y no podremos ver su contenido por pantalla. Cuando no estemos seguros del tipo del fichero podemos usar el comando **file** :

```
$ file copyfile
copyfile: ascii tex
```

NOTA: En el caso de ficheros ejecutables utilizando Digital Unix la respuesta que se obtiene es:

```
$ file ex
ex: Elf format data
```

#### **4.2.9. Buscando en ficheros (grep )**

Podemos usar el comando **grep** para buscar determinada cadena en un fichero. La sintaxis básica es:

```
$ grep cadena fichero
```

Dentro de la cadena puede haber espacios en blanco, puntuaciones e incluso caracteres de control invisibles. El comando permite muchas más opciones que se pueden consultar mediante el comando **man** (*man grep* ).

### **4.3. Directorios y jerarquía**

Los ficheros se agrupan en directorios, los cuales están organizados en una jerarquía. En la parte más alta de la jerarquía está el directorio "*root*" o raíz que se simboliza por */*.

Cada directorio del sistema de ficheros puede contener otros directorios. Por convención, distinguiremos entre los diferentes niveles de directorios con */*.

Especificaremos un directorio (y sus ficheros) incluyendo los nombres de los directorios que lo contienen. Esto se llama **path name** o camino.

En un subdirectorio, un nombre de fichero debe ser único pero podemos tener dos ficheros con el mismo nombre en directorios diferentes.

#### **4.3.1. Imprimiendo el directorio de trabajo (*pwd* )**

El comando **pwd** nos dice en que lugar estamos en la jerarquía del sistema de ficheros:

```
$ pwd
/home/x1/x1001
```

#### **4.3.2. Cambiar de directorio (*cd* )**

El comando **cd** nos permite movernos por la jerarquía del sistema de ficheros:

```
$ cd /tmp
$ pwd
/tmp
```

Si tecleas sólo el comando **cd** , volverás a tu directorio *home*

```
$ cd
$ pwd
/home/x1/x1001
```

En algunos shells se usa *~* como abreviatura de tu directorio *home* . Por ejemplo, si quieres cambiar al directorio *music* que cuelga de tu directorio *home* :

```
$ pwd
/tmp
$ cd ~/practicas
$ pwd
/home/x1/x1001/practicas
```

El directorio situado inmediatamente "abajo" de un subdirectorio se llama directorio padre. En los ejemplos anteriores, */home* es el directorio padre de */home/user1*. El símbolo *..* (punto-punto) representa al directorio padre. Así, si se escribe *cd ..* cambia el directorio de trabajo al padre:

```
$ pwd
/home/x1/x1001
$ cd ..
$ pwd
/home/x1
```

#### **4.3.3. Creando un directorio (*mkdir* )**

Lo haremos con el comando **mkdir** seguido del nombre del nuevo directorio:

```
$ mkdir veggies
$ cd veggies
$ mkdir broccoli
$ cd broccoli
$ pwd
/home/x1/x1001/veggies/broccoli
```

#### **4.3.4. Moviendo y renombrando directorios (*mv* )**

Para renombrar un directorio hay que moverlo a un nombre diferente. Para ello se usa el comando **mv** :

```
$ pwd
/home/x1/x1001/veggies
$ ls
broccoli
mv broccoli carrots
$ ls
carrots
```

También podremos usar el comando **mv** para mover el directorio a otro lugar:

```
$ pwd
/home/x1/x1001/veggies
$ ls
carrots
$ mv carrots ../veggies2
$ ls ../veggies2
carrots
```

En este ejemplo, movemos el directorio *carrots* de *veggies* a *veggies2*.

#### **4.3.5. Copiando directorios (*cp -r*)**

Utiliza *cp -r* para copiar directorios y los ficheros que contienen:

```
$ cp -r veggies veggies3
$
```

Con esto copiamos todos los ficheros y subdirectorios del directorio *veggies* a un nuevo directorio llamado *veggies3*. Es una copia recursiva, tal como indica la opción *-r*. Un intento de copia de un directorio sin esta opción nos dará un mensaje de error.

#### **4.3.6. Borrando directorios (*rmdir* )**

Para borrar directorios vacíos usa el comando **rmdir** :

```
$ rmdir nombre_directorio
$
```

El comando fallará en caso de que el subdirectorio no esté vacío. En este caso podremos usar *rm -r* que

borrará de forma recursiva un directorio y todos sus ficheros/subdirectorios:

```
$ rm -r nombre_directorio
$
```

#### **4.4. Seguridad en ficheros y directorios**

Los permisos de fichero nos ayudan a proteger nuestros ficheros y directorios de lecturas o escrituras no autorizadas. A menudo tendrás ficheros que quieras que otros puedan leer pero no cambiar, o querrás que puedan ejecutar determinados ejecutables y otros no, etc. Todo esto lo podremos hacer con los permisos.

Estos son los tipos permisos básicos:

**r :**

permiso de lectura (*read*): Permite que un fichero pueda ser examinado o copiado, y que se pueda listar el contenido de un directorio.

**w :**

permiso de escritura (*write*): Permite que un fichero pueda ser modificado, borrado o renombrado, y que se pueda añadir o borrar ficheros de un directorio.

**x :**

permiso de ejecución (*execute*): Permite que un fichero se pueda ejecutar, y que se pueda acceder a un directorio a sus directorios.

Hay tres categorías de usuarios, cada uno con sus permisos:

##### **Propietario:**

El dueño del archivo.

##### **Grupo:**

Otros usuarios del mismo grupo del propietario.

##### **Otros:**

Cualquier otro usuario.

#### **4.4.1. Mostrando los permisos (ls -l)**

Ya habrás usado el comando **ls** para listar ficheros. Lo usaremos ahora con la opción **-l** para obtener mas información:

```
$ pwd
/home/x1/x1001
$ ls -l
total 8
drwxr-xr-x  2 x1001  1024 Feb  9 14:22 directorio1
-rw-r--r--  1 x1001    0 Feb 10 10:20 fich_vacio
-rw-r--r--  1 x1001 104357 Feb  5 08:20 fich_grande
drwxr-xr-x  3 x1001  1024 Feb 10 11:13 veggies2
```

Veamos ahora el significado de cada línea:

- El primer carácter en la línea indica el tipo de fichero. Un guión - indica que es un fichero normal, una *d* indica que es un directorio, y otros caracteres indicarán que es un tipo especial de fichero.

- Los siguientes 9 caracteres indican los permisos para el fichero o directorio. Consisten en 3 grupos de 3, mostrando los permisos para el dueño, el grupo y otros respectivamente. Por ejemplo, en *fich\_vacio* son *rw-r--r--*, indicando que el propietario puede leer y escribir en este fichero, todos pueden leerlo y nadie ejecutarlo. Los permisos para el directorio *veggies2* son *rwxr-xr-x*, indicando que todos tienen permiso de ejecución pero sólo el propietario puede escribir en él.
- El siguiente campo es el número de *links* de este fichero/directorio.
- Nombre del propietario (*x1001* en este caso)
- Longitud del fichero.
- Fecha y hora de la última modificación del fichero/directorio.
- Nombre del fichero/directorio.

#### **4.4.2. Listando ficheros ocultos (*ls -a*)**

Los ficheros cuyo nombre comienza por punto, son ocultos y sólo se mostrarán si se usa el comando **ls -a**

```
$ ls -a
.
..
.cshr
.login
.profile
fich_vacio
```

Hay dos ficheros especiales en esta lista: el fichero *.* que es referencia al directorio actual y el *..* que referencia al directorio padre.

#### **4.4.3. Cambiando permisos (*chmod*)**

Usa el comando **chmod** para cambiar los permisos de un fichero o directorio. Para ello debes ser el propietario del fichero o directorio (o tener acceso *root*). La forma de hacerlo es:

```
$ chmod permisos nombre
```

donde *permisos* indica los permisos a cambiar y *nombre* el fichero o directorio afectado.

Los permisos se indicarán como un número de 3 cifras, uno para el propietario, otro para el grupo y otro para todos. Para ello se pondrá a 1 cada permiso que se quiera activar. Por ejemplo, supongamos que queremos poner los permisos *rw-r--r--*, es decir, 111 100 100, con lo que el número será 744. Para poner todos los permisos será 777 y para poner solo permisos para el propietario será 700.

#### **4.5. Cuotas de usuario**

Cada usuario dispone de un espacio de disco limitado (hay un número máximo de ficheros que se puede tener y un límite de capacidad). Para comprobar del que se dispone se emplea el comando **quota**:

```
$quota -v
```

NOTA: Para más información se puede utilizar *man quota* o ver el tablón de notas de la sala de laboratorios de la planta 1.

## 5. EL EDITOR DE TEXTOS VI

### 5.1. Abrir un documento

Cuando quieras crear o editar un documento usando *vi* teclea

*\$ vi nombre-del-documento*

Esto crea un nuevo fichero si no hay ninguno con ese nombre, o lo abre en caso de que lo haya.

### 5.2. Los modos de operacion de vi

En *vi* hay dos modos de operación, el modo de entrada y el modo de comando.

Cuando estás en modo de entrada puedes introducir caracteres y pulsar *Return* para obtener saltos de línea. En este modo no puedes usar los cursores para moverte a través del texto.

Cuando estás en modo comando puedes introducir los comandos de *vi*, como inserción y borrado; los movimientos del cursor también se realizan en este modo.

No hay ninguna forma de saber en que modo estás, así que si dudas pulsa la tecla *Esc*. *Esc* siempre te pone en el modo comando estés en él o no.

Cuando inicias *vi* está siempre en modo comando.

Cuando introduzcas comandos asegúrate de no estar en mayúsculas, ya que ni para *Unix* ni para *vi* significan lo mismo que minúsculas.

### 5.3. Salvar y salir

Cuando tecleas un documento, *v* no lo guarda en el fichero a menos que le digas que lo haga. Es decir, si se interrumpe la sesión con *vi* sin haber guardado los cambios, habrás perdido el trabajo.

Cada vez que quieras guardar lo que estas tecleando entra en el modo comando (pulsando *Esc*) y teclea

*:w*

Esto es, pulsa los dos puntos y la *w* minúscula, después pulsa *Return*.

Si ya has terminado la sesión y quieres guardar el fichero y terminar *vi* teclea

*:wq*

siempre desde el modo comando.

Para salir de *vi*

*:q*

te saca de *vi* siempre y cuando no hayas introducido nada de la última vez que salvaste el documento. Si no quieres guardar los cambios teclea:

*:q!*

## **5.4. Comandos básicos**

### **Inserción de texto**

Cada vez que entras en *vi* estás en modo comando. Para comenzar a introducir texto usa el comando de inserción, que se obtiene pulsando la tecla *i* . Al contrario que los comandos que empiezan por *:* no es necesario pulsar *Return* al final del comando.

Cuando quieras introducir texto en un determinado lugar, pon el cursor en él y usa el comando *i*.

### **Moverse a través del texto**

Para moverte a través del texto que teclees, entra en el modo comando y usa los cursores. Usar los cursores desde el modo de entrada sólo sirve para hacer aparecer caracteres raros por la pantalla. Si el teclado que usas no tiene cursores o no funcionan, las teclas equivalentes son: *h,j,k,l*.

### **Borrado de texto**

Para borrar un carácter posiciona el cursor sobre él y pulsa la tecla *x* .

Para borrar una palabra pon el cursor al principio de ella y teclea *dw* .

Para borrar una línea pon el cursor en cualquier lugar de ella y teclea *dd* .

Alguna de estos comandos son de más de una letra. Si te equivocas en la primera o si ya no quieres ese comando simplemente vuelve a pulsar *Esc*.

*vi* tiene muchos más comandos, que puedes ver haciendo

*\$ man vi*

desde el *prompt* del sistema.

## **6.-EL EDITOR DE TEXTOS PICO**

Para editar ficheros de texto está disponible también otro editor de texto en todas las máquinas (excepto en silgar), el editor pico. Para utilizarlo sólo tienes que teclear el comando:

*\$pico nombre*

y creará un nuevo documento con ese nombre o, si existe, lo abrirá. Aparecerá la pantalla de edición y ya puedes teclear el texto que quieras almacenar en el fichero. Para moverte por el texto puedes utilizar las teclas del cursor y utilizar la tecla *Backspace* para borrar caracteres. En la parte inferior de la pantalla aparecen una serie de comandos del editor. Para utilizarlos deberás pulsar las teclas resaltadas a su izquierda, por ejemplo, para obtener la ayuda, *^G*, debes pulsar las teclas de Control y G simultáneamente. Algunos de estos comandos son:

Get Help (*CTRL-G*): Muestra la información de ayuda para utilizar el editor de texto.

WriteOut (*CTRL-O*): Permite actualizar el fichero en disco con los cambios que has realizado.

Cut Text (*CTRL-K*): Elimina la línea del texto donde se encuentra situado el cursor.

UnCut Text (*CTRL-U*): Inserta, donde se encuentra situado el cursor, la línea (o líneas) eliminada con el comando anterior.

Exit (*CTRL-X*): Termina la ejecución del programa. Antes de salir te preguntará si guarda los cambios realizados en el fichero, si pulsas la tecla Y actualizará el fichero, pulsando N el fichero quedará sin cambios. Después aparecerá

el nombre del fichero, si quieres grabar el fichero con ese nombre pulsa *Return* y, si quieres grabarlo con otro, escribe el nuevo (borrando antes el nombre antiguo) y pulsa *Return*.

---

## 7.-USO DEL CORREO ELECTRÓNICO (mail)

Puedes utilizar el correo electrónico para mandar mensajes a otros usuarios, avisos, etc.. Para ello se utilizan los comandos **mail** / **mailx** y el programa **pine**.

### 7.1. Enviando mensajes

Para enviar mensajes tienes que utilizar los comandos **mail/mailx** de la siguiente forma:

```
$mail username
Hola, ¡Que tal!
¿Como estas?
.
$
```

*Username* es el login del usuario al que le quieres enviar el mensaje. La última línea contiene sólo un punto para indicar que se ha terminado de escribir el mensaje, al pulsar *Return* aparece el *prompt* del sistema y el mensaje es enviado.

Dependiendo del comando que utilices y de la máquina a la que estés conectado el sistema te pedirá un sujeto para el mensaje o no.

También puedes escribir el mensaje en un fichero de texto y enviarlo. Por ejemplo, si el mensaje está en el fichero *mensaje.txt* para enviarlo sólo hay que redireccionar la entrada del comando:

```
$mail username < mensaje.txt
$
```

Si al intentar enviar un mensaje existe algún problema y el mensaje no es enviado éste queda almacenado en el fichero de texto *dead.letter* para que no tengas que reescribirlo cuando intentes enviarlo de nuevo.

### 7.2. Recibiendo mensajes

Para leer los mensajes que te han enviado se utilizan los mismos comandos (**mail/mailx**) de la forma:

```
$mailx
```

A continuación aparecerá el último mensaje que te han enviado seguido del *prompt* ?. A partir de aquí puedes usar los siguientes comandos:

- *Return* va mostrando los mensajes y al mostrar el último abandona mail.
- *p* imprime por pantalla de nuevo el mensaje que ha mostrado.
- *d* borra el mensaje que acaba de mostrar y muestra el siguiente.
- *s* graba el mensaje en un fichero de texto (*mbox* si no se especifica otro) y muestra el siguiente.
- *q* abandona mail. (En riazor graba todos los mensajes a fichero antes de abandonar mail. Si utilizas *x* solo abandonará mail.)
- *?* facilita una lista de comandos de mail.



### **7.3. Utilizando PINE**

Para enviar y recibir mensajes también puedes utilizar el programa **PINE** instalado en todas las máquinas (excepto en silgar, que ya tiene una aplicación para gestionar el correo electrónico en su entorno de ventanas).

Para utilizarlo teclea:

*\$pine*

A continuación se cargará el programa y aparecerá el menú principal del mismo. El menú principal consta de varias opciones que puedes elegir utilizando las teclas de cursor y pulsando *Return* o pulsando en el teclado el carácter que aparece a la izquierda de cada una. Algunas de estas opciones son:

? HELP

Proporciona información de ayuda para utilizar el programa.

C COMPOSE MESSAGE

Permite crear y enviar mensajes. Para ello aparece una pantalla en la que especificarás las características del mensaje. Lo más importante es especificar el destinatario ( *To :* ) y escribir el mensaje a enviar ( *Message Text* ). También puedes incluirle un sujeto al mensaje ( *Subject :* ), aparecerá con la cabecera del mensaje al recibirlo, o realizar un Attachment ( *Attchmnt :* ). Ésta última opción permite incluir ficheros en el mensaje indicándole el nombre del mismo (si está en \$HOME) o el path (si está en otro subdirectorio). Una vez escrito el mensaje puedes enviarlo pulsando *CTRL-X* o cancelar el envío pulsando *CTRL-C*. Tanto si cancelas el envío como si hay algún problema con el mensaje éste quedará almacenado en el fichero *dead.letter*.

I FOLDER INDEX

Permite visualizar los mensajes en el directorio actual. El directorio actual por defecto es *INBOX* (contiene los mensajes recibidos). Al elegir esta opción aparece en pantalla un índice con las cabeceras de los mensajes y se elige con las teclas del cursor y pulsando *Return* el que se quiere leer. Después puedes utilizar las teclas P y N para mostrar los mensajes anterior y posterior respectivamente, y M para volver al menú principal. También puedes utilizar la tecla D para borrar los mensajes, en ese caso al salir del programa te pedirá la confirmación del borrado.

L FOLDER LIST

Permite ver los mensajes almacenados en cualquier directorio, p. ej. el directorio de mensajes recibidos (*INBOX*) o el directorio que contiene los mensajes enviados (*SENT-MAIL*). El directorio que eligas pasará a ser el directorio actual. Su funcionamiento es igual al de la anterior opción.

Q QUIT

Permite abandonar el programa. Al salir te pedirá la confirmación de la salida, pulsando y saldrás del programa. Además, en el caso de que hayas borrado algún mensaje, te preguntará si borra los mensajes del directorio correspondiente, pulsando y borrará dichos mensajes.

Además de las opciones que tiene el programa en el menú principal, en la parte inferior de la pantalla se muestran una serie de comandos que se pueden ejecutar. Para ejecutarlos sólo tienes que pulsar las teclas que aparecen resaltadas a la izquierda del comando en cuestión ( recuerda que si aparece ^X tienes que pulsar las teclas Control y X al mismo tiempo para ejecutar ese comando). Algunos de estos comandos son:

- ? muestra la ayuda específica para cada parte del programa.
- Q termina el programa.
- M vuelve al menú principal.
- O muestra más comandos (en el caso de que los haya).

## **7.4. Listas de correo**

La red Unix de la facultad cuenta con un servicio de listas de correo. Éste servicio permite que todos los usuarios que estén suscritos a una lista puedan expresar sus opiniones sobre el tema de la lista y, al mismo tiempo, conocer las opiniones de los demás usuarios suscritos. Para ello los usuarios envían un mensaje a una dirección determinada y el sistema se encarga de reenviar el mensaje a todos los usuarios suscritos a la misma lista.

Para utilizar el sistema primero debes pedir la ayuda de usuario del gestor de Listas (Majordomo), para ello debes enviar el mensaje:

*\$ mailx majordomo@cecafi*

*Subject:*

*help*

El campo *Subject* y otros campos opcionales que pueda haber deben quedar vacíos.

Una vez enviado recibirás por mail la ayuda, información general sobre el sistema de listas así como la forma de suscribirte a las listas y comandos relacionados con las mismas.

Para suscribirte a una lista tienes que enviar el siguiente mensaje:

*\$ mail majordomo@cecafi*

*Subject:*

*subscribe NOMBRE\_DE\_LISTA*

Si todo está bien recibirás un mensaje del gestor (*majordomo*) comunicándote que has sido dado de alta en la lista.

Si hay algún problema recibirás un mensaje indicándote el error y lo que tienes que hacer para rectificar.

Para dejar tus opiniones deberás enviarlas por mail a la dirección:

*NOMBRE\_DE\_LISTA@cecafi.fi.udc.es*

Las listas disponibles son:

*cecafi-admin*: Esta lista está destinada a comunicar y resolver dudas que puedan surgir en relación con el uso de los sistemas del CECAFI.

*cecafi-info*: Esta lista está destinada a la información sobre actos, publicaciones, actividades y otros temas de interés general para los miembros de la Facultad de Informática.

*cecafi-alt*: Esta lista está destinada a debatir temas de interés para los usuarios de los sistemas del CECAFI y que guarden relación con la informática.

Estas listas sólo permiten la suscripción de cuentas del CECAFI, no de INTERNET ni de otro lado. Para leer los mensajes podrás utilizar cualquiera de los equipos del CECAFI.

Has de tener en cuenta también que NO se puede utilizar para enviar mensajes sistemas con SunOS 4.X, es decir, no se puede utilizar las estaciones eo, ni conexiones a riazor o a deo.

## **8. USO DE LA IMPRESORA**

### **8.1. Imprimir un documento**

Para imprimir un documento desde la línea de comandos se usa el comando **lp** :

`$ lp nombre-del-fichero`

Esto envía el documento a la impresora por defecto. Si hay otros documentos esperando para imprimirse se guarda en la cola de impresora hasta que le llegue el turno.

Si la impresora a la que deseas mandar el documento no es la predeterminada, se usa la opción `-d`

`$ lp -d nombre-impresora nombre-fichero`

### **8.1.1. Pedir aviso de fin de impresión**

Si quieres saber cuando se ha terminado la impresión, para ir a buscar el documento:

`$ lp -w nombre-fichero`

### **8.1.2. Copia de impresión**

Para crear una copia del fichero a imprimir puedes utilizar la opción:

`$ lp -c nombre-fichero`

Con esta opción el sistema crea una copia del fichero que es la que al final se imprimirá. Si no la especificas entonces cualquier modificación que hagas sobre el fichero original después de dar la orden de impresión pero antes de que se imprima, aparecerá reflejada en el documento impreso.

### **8.1.2. Opciones múltiples**

Por supuesto todas las opciones pueden combinarse al gusto :

`$ lp -d Impresora -m -fichero`

Envía una copia de *fichero* a *Impresora* y pide notificación de fin de impresión por ventana.

## **8.2. Conocer el estado de la impresora**

Para saber si la impresora funciona, o está ocupada o disponible :

`$ lpstat -p nombre-impresora`

El comando **lpstat** sin opciones te informa del estado de los documentos que has enviado a impresión.

El comando **lpq** nos permite conocer los trabajos que están en cola de impresión.

## **8.3. Cancelar una petición de impresión**

Si has enviado un documento, pero ya no te interesa, puedes cancelarlo mientras aún siga en la cola de impresión.

*\$ cancel petition-id*

donde *petition-id* es el primer campo que aparece al hacer un *lpstat*.

Estos comandos tienen más opciones, que puedes ver con *man*.

## 9. USO DE LA RED

En un entorno en red, los ficheros y recursos están repartidos entre distintas máquinas y para poder usarlos hay que conectarse a ellas. Por ejemplo, si quieres usar un compilador debes estar conectado a la máquina que lo tiene.

### 9.1. Establecer una conexión remota

Cuando entras en el sistema lo haces a través de una máquina determinada. Si necesitas entrar en otra debes hacer un *login remoto*. Hay dos posibilidades:

- Comando **rlogin**

*\$ rlogin nombre-maquina*

El sistema te pide el *password* para dejarte entrar.

- Comando **telnet**

*\$ telnet nombre-maquina*

En este caso, se te solicita tanto el login como el password.

### 9.2. Desconectarse de una máquina

Para salir de una máquina puedes hacerlo de dos formas: tecleando el comando **logout** al final de la sesión.

Si usas el comando **exit** vuelves a la máquina desde la que te conectaste.

### 9.3. Cómo saber quien eres

Si te has conectado a varias máquinas, puede que con distintos nombres, y no estás seguro de con cual estás en ese momento teclea

*\$ whoami*

Si sólo tecleas el comando **who** obtienes la lista de todos los usuarios que están conectados en ese momento.

*\$ who*

## 10. VARIABLES DE ENTORNO

Las variables de entorno indican al sistema aspectos particulares sobre cómo debe tratarte. Por ejemplo la variable *PATH* indica las rutas a las que tienes acceso, la variable *TERM* el tipo de terminal que estás usando, la variable *HOME* cual es tu directorio origen,...

Para ver todas las variables definidas teclea:

```
$ env
```

Las variables también pueden cambiarse, y para ello existen los comandos **set** y **setenv**. Estos comandos dependen mucho del *shell* que estés usando así que consulta los con *man* antes de usarlos.

Un ejemplo de la necesidad de cambiar las variables son los terminales *WYSE*, que son los terminales de texto monocromos. A veces la variable de entorno *TERM* está mal configurada y no puedes usar los editores de texto porque el sistema no sabe como manejar tu terminal. Esto puede arreglarse tecleando:

```
$ TERM=vt100
$ export TERM
```

para establecer como tipo de terminal el standard *VT100*.

## 11. COMPILACIÓN DE PROGRAMAS

Antes de ejecutar un programa, deberá ser compilado. La compilación es el proceso mediante el cual se lleva a cabo la traducción del programa escrito en un lenguaje de alto nivel, a un formato que permite su ejecución en un ordenador particular.

En muchos sistemas, el compilador genera un fichero objeto. Este fichero deberá ser lincado (asociándolo a rutinas de ciertas librerías). Concluida esta operación, el fichero se considera ejecutable y estará listo para su ejecución.

El compilador de Pascal de Unix se denomina *pc*. La forma de compilar un programa en Pascal es la siguiente:

```
$pc nombre-del-fichero.pas
```

El compilador de C de Unix se llama *cc*. La forma de compilar un programa en C es la siguiente:

```
$cc nombre-del-fichero.c
```

Si no hay ningún error de compilación, se genera un fichero ejecutable. Si no se especifica ningún nombre, se le asignará uno por defecto: *a.out*. En otro caso, teclear:

```
$pc -o nombre-ejecutable fichero.pas    (para Pascal )
```

o

```
$cc -o nombre-ejecutable fichero.c      (para C)
```

NOTA: Ten en cuenta que hay diferentes versiones de Unix instaladas en los servidores. Dependiendo de la versión que utilices el compilador utilizará librerías diferentes por lo que un programa que compiles en un servidor puede que no te funcione en otro con otra versión de Unix (sobre todo si utilizas las librerías).

