



UNIDAD 8: LINUX. USO DEL SISTEMA OPERATIVO

**Sistemas Operativos Monopuesto
Curso 2012/2013**

UT8: LINUX. USO DEL SISTEMA OPERATIVO

- 1. Sistema de Archivos
- 2. Estructura del Sistema de Archivos
- 3. Gestión de Archivos y Directorios
- 4. Archivos Especiales
- 5. Ejercicios

1. SISTEMA DE FICHEROS

○ Sistema de Ficheros:

- Se encarga de almacenar, organizar, acceder y recuperar la información almacenada en forma de archivo
- Sistema de archivos ext2, o sus sucesores ext3 y ext4
- ReiserFS y Reiser4, a partir de este último han dejado de usarlo muchas distribuciones

1. SISTEMA DE FICHEROS

○ Sistema de Ficheros:

- Reconoce otros sistemas de archivos como NTFS, FAT16, FAT32, CDFS, UDF...
 - Sistemas de archivos de nuestros dispositivos:
 - Sistema – Administración – Monitor del sistema
- FUSEBLK: nombre con el que se monta el sistema de archivos NTFS
- VFAT: sistema de archivos compatible con los sistemas de archivos FAT de Windows

1. SISTEMA DE FICHEROS

○ Sistema de Ficheros:

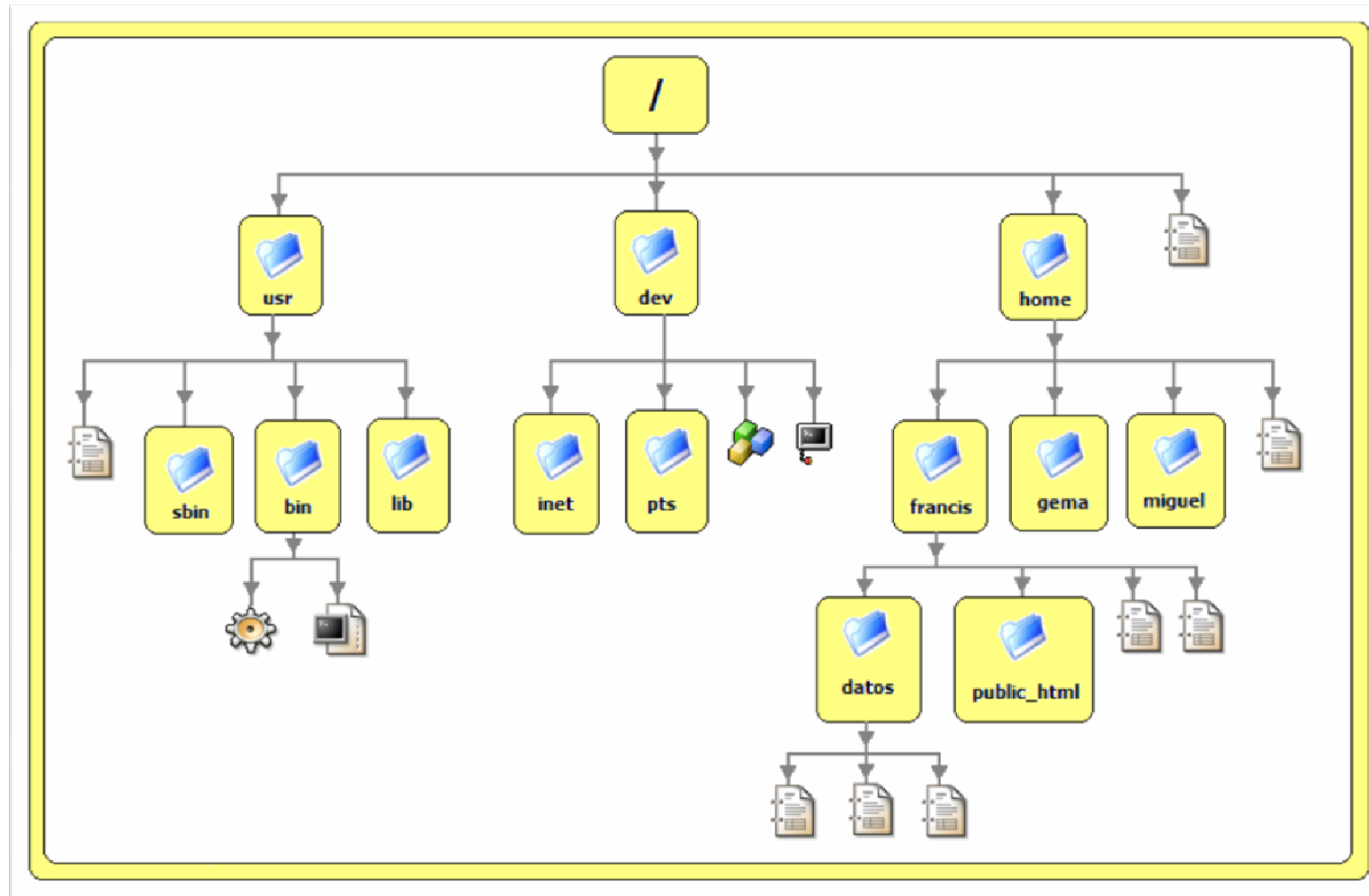
- Estructura básica jerárquica ó en forma de árbol
- La raíz del sistema de archivos se conoce como root y se denota con el símbolo /
- Un archivo siempre ocupará el nivel más bajo dentro del árbol, porque de un archivo no pueden depender otros
- En Linux todo es tratado como un fichero
 - Un fichero pertenece a un usuario, que forma parte de un grupo, y tendrá permisos de lectura, escritura y ejecución

2. ESTRUCTURA DEL SISTEMA DE ARCHIVOS

○ Estructura de Directorios:

- Estructura jerárquica, en forma de árbol invertido. Se parte de un directorio principal **root** representado por el carácter **“/”**. Este directorio puede contener, al igual que en Windows, otros directorios o archivos que dependan de él
- La diferencia fundamental con Windows es que no existe un directorio raíz por cada unidad lógica de almacenamiento (C:, D:, etc.)

2. ESTRUCTURA DEL SISTEMA DE ARCHIVOS



2. ESTRUCTURA DEL SISTEMA DE ARCHIVOS

○ Directorios del sistema:

- /bin. Contiene ficheros de comandos ejecutables utilizables por todos los usuarios
- /sbin. Contiene ficheros de comandos ejecutables utilizados por el administrador
- /usr o /home. Contiene utilidades y programas generales de cada usuario
- /dev. Contiene archivos especiales de bloques y caracteres asociados a dispositivos hardware

2. ESTRUCTURA DEL SISTEMA DE ARCHIVOS

○ Directorios del sistema:

- **/lib.** Contiene librerías y compiladores del sistema
- **/etc.** Directorio con ficheros de configuración y utilidades para la administración
- **/var.** Contiene ficheros para el administrador
- **/mnt.** Directorio utilizado para montar dispositivos
- **/stand.** Contiene los ficheros de arranque del sistema y el núcleo de Unix/Linux
- **/opt.** Sirve de punto de montaje desde el que se instalan los paquetes de aplicaciones adicionales

2. ESTRUCTURA DEL SISTEMA DE ARCHIVOS

○ Directorios del sistema:

- **/bck.** Para montar un sistema de archivos para realizar copias de seguridad
- **/boot.** Contiene los archivos de configuración de arranque del sistema
- **/config.** Para configurar el arranque del sistema
- **/dgn.** Para almacenar los programas de diagnóstico
- **/export.** Para exportar sistemas de archivos

2. ESTRUCTURA DEL SISTEMA DE ARCHIVOS

○ Directorios del sistema:

- **/install.** Donde el administrador instala las utilidades de administración
- **/save.** Para guardar copias de ficheros de administración con el fin de realizar copias a disquetes
- **/tmp.** Para almacenar archivos temporales
- **/home.** Ubicación de los directorios propios de cada usuario. Contiene el directorio de presentación y el árbol de subdirectorios de cada usuario

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Estructura de Directorios:

- Caracteres para los **nombres de archivo**
 - Se puede utilizar cualquiera, a excepción del carácter /, ya que tiene un significado especial. Es conveniente no utilizar caracteres especiales, acentos, etc. No es conveniente emplear espacios en blanco ni caracteres como los siguientes en los nombres de archivo:
! # & () `” ; | > < @ \$ { } * ? \ Tab (tabulador) Spacebar (barra espaciadora) Backspace (borrado hacia atrás) + -
- Se destaca de nuevo que los nombres de archivo son diferentes *en mayúscula y en minúscula*. Se recomienda utilizar todos en **minúscula**:
 - PRACTICA, practica, Practica, serían nombres de archivos diferentes

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Sistema de Ficheros. Características:

- No está ligado de una forma directa con la estructura del hardware
 - No se requiere que el conjunto de sistemas de archivos resida en un mismo dispositivo y es posible definir uno o varios sistemas “desmontables”, que residen físicamente en diversas unidades de disco
- Los directorios contienen información para cada archivo: su **nombre** y un número que el Kernel utiliza para manejar la estructura interna del sistema de archivos, conocido como el ***i-nodo***

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Sistema de Ficheros. Características:

- i-nodos:
 - Hay un i-nodo para cada archivo, con la información de su directorio en el disco, su longitud, los modos y las fechas de acceso, el autor, etc.
 - Existe, además, una tabla de descriptores de archivo, que es una estructura de datos residente en el disco magnético
- Cada vez que cambia un atributo del fichero, se reescribe la información en la tabla de i-nodos

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Archivos o ficheros:

- Operaciones:

- Crear
- Borrar
- Copiar
- Mover
- Modificar
- Seleccionar
- Renombrar
- Enlazar
- Abrir
- Cerrar

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Tipos de ficheros:

- Enlaces duros (Vínculos físicos):
 - No es específicamente una clase de archivo. Es un segundo nombre asignado a un archivo (tienen el mismo i-nodo)
 - Dos usuarios necesitan compartir la información de un mismo archivo. Si cada uno de ellos tiene una copia, la modificaciones que realice uno no podrán ser utilizadas por otro, solamente se modificará la copia de cada uno
 - Si en vez de tener una copia, cada uno utiliza un archivo tipo **enlace** al archivo original, cada vez que uno modifique su archivo lo que estará modificando es el archivo común
 - El vínculo sirve para localizar el fichero en su ubicación actual, pero no es el archivo real. Con tener un solo archivo real, éste se podrá utilizar por todos los usuarios que lo necesiten, sin tener que duplicarlo o triplicarlo

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Tipos de ficheros:

- Enlaces simbólicos (Vínculos simbólicos):
 - Se utilizan para asignar más de un nombre a un archivo. No sirven para directorios
 - Es un archivo que sólo contiene el nombre de otro archivo
 - Cuando el sistema operativo opera sobre un vínculo simbólico, éste se dirige al archivo al que apunta el vínculo simbólico
 - A diferencia de los enlaces físicos, que existen dentro de la estructura de archivos, los enlaces simbólicos solamente hacen referencia al nombre de otro archivo
 - Tienen distinto número de i-nodo

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Tipos de ficheros:

- Archivos especiales:

- Suelen representar dispositivos físicos como unidades de almacenamiento, impresoras, etc.
- Unix/Linux trata los archivos especiales como archivos ordinarios:
 - Un usuario puede abrir un archivo vinculado a una unidad de disquete, modificarlo, etc. Con ello consigue leer del disquete, escribir en el disquete, etc.
 - Unix/Linux transforma las operaciones básicas y órdenes básicas de archivos ordinarios o regulares en comandos que interactúan con el hardware a través de los archivos especiales

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Tipos de ficheros:

- Shell scripts:
 - Contienen órdenes o comandos
 - Algunos se ejecutan al arrancar el sistema y otros los crea el usuario
- Ficheros de configuración:
 - Permiten definir ciertos parámetros cuando se arranca el sistema para ajustarlo a las necesidades del usuario
 - Configuración del gestor de arranque GRUB:
 - /boot/grub/menu.lst ó
 - /boot/grub/grub.cfg (GRUB2)

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Ficheros de configuración:

- /etc/fstab
- /etc/apt/source.list
- /etc/X11/xorg.conf
- /etc/bashrc
- /etc/hosts: resolución local de nombres de equipos a direcciones IP
- /etc/profile: programas de arranque y configuración global del sistema. Si usamos shell bash:
 - /etc/bash.bashrc

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Ficheros de configuración:

- /etc/shells
- /etc/sudoers
- /etc/rc.local
- /etc/network/interfaces
- /etc/crontab
- Ficheros para configurar y administrar usuarios y grupos:
 - /etc/passwd
 - /etc/group
 - /etc/shadow
 - /etc/gshadow
 -

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Directorios o carpetas:

- Operaciones:

- Crear
- Borrar
- Copiar
- Mover
- Modificar
- Seleccionar
- Renombrar
- Enlazar
- Entrar
- Salir

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Estructura de Directorios:

- Los directorios en Unix/Linux son archivos especiales cuya función es la de almacenar archivos u otros directorios:
 - Tienen características especiales
 - Cada directorio, a excepción del raíz, consta de los directorios `.` y `..`, que indican *directorio actual* y *directorio padre*, respectivamente.
- Para referirnos a un fichero en el árbol de directorios es necesario incluir el camino o ruta de directorios (path) donde se encuentra el fichero. Si existen varios subdirectorios se utiliza el carácter `“/”` como separador

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Estructura de Directorios:

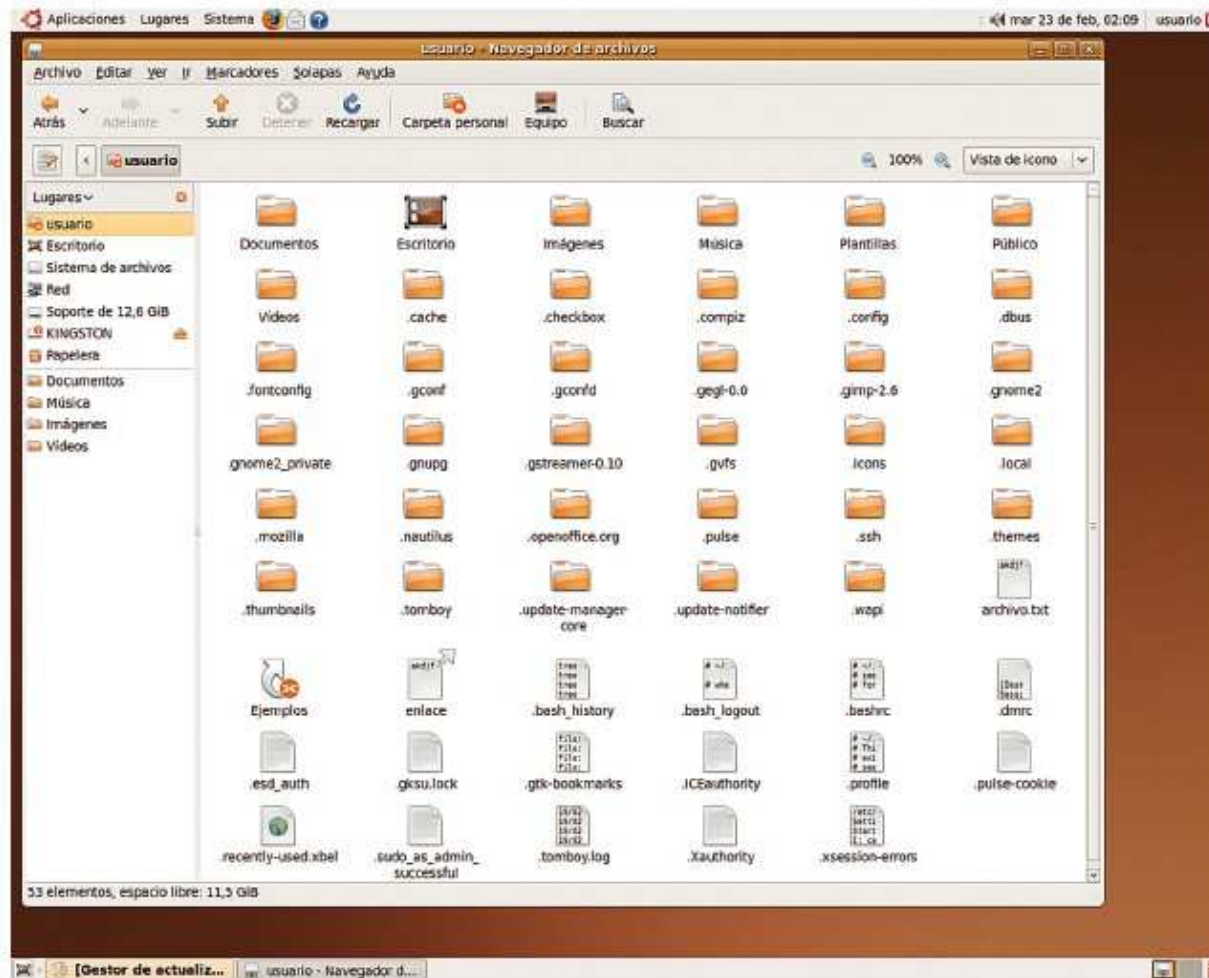
- El camino indica la ruta que hay que recorrer para encontrar el fichero
- El nombre de camino completo, partiendo desde el directorio raíz (/) hasta el directorio o fichero al que se desea llegar, se conoce como camino absoluto y comienza siempre por el carácter “/”
- La posición en la que se está situado en el árbol en un momento dado se conoce con el nombre de directorio actual.
- Se denomina camino relativo al que hay que seguir a partir del directorio actual hasta el fichero o directorio al que se quiere llegar. Los caminos relativos no comienzan por el carácter “/”

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Gestión de archivos y directorios en entorno gráfico:
 - GNOME: Administrador de archivos Nautilus
 - Lugares – Carpeta personal
 - Para ver archivos y carpetas ocultos:
 - Ctrl + H
 - Ver – Ficheros ocultos
 - KDE: Administrador de archivos Konqueror (que también es navegador Web)

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Gestión de archivos y directorios en entorno gráfico:



3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Operaciones en entorno gráfico: Menú contextual



3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Operaciones en entorno gráfico: Propiedades



3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Operaciones en entorno gráfico: Propiedades
 - Emblemas: imágenes que se colocan sobre las carpetas y directorios para ayudarnos a clasificarlos



● Operaciones en entorno gráfico: Compartir carpetas



3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Asociar aplicaciones a tipos de ficheros:
 - Propiedades – Abrir con...
 - Menú contextual – Abrir con otra aplicación



3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Gestión de archivos y directorios en modo texto:
 - Comodines

CARACTER	ACCIÓN
?	Sustituye a un solo carácter
*	Sustituye a cualquier carácter o conjunto de caracteres
[]	Cuando se utilizan como parte de nombres de archivo o directorio, representan un solo carácter de los incluidos entre los corchetes, que se sustituirán al carácter en el nombre de archivo o directorio en la posición en la que estén estos corchetes. Pueden incluir rangos separados por un guión
!	Permite la negación o exclusión de caracteres
{a,b}	Busca coincidencia con uno u otro valor (pueden usarse cadenas)

- Caracteres de escape: \

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **ls.** Permite ver el contenido de un directorio; muestra todos los archivos del directorio. Puede ocurrir que un directorio no esté disponible para la lectura, por lo que no se podrá ver su contenido hasta que no sea desprotegido. Su sintaxis es:

ls [opciones] [directorio]

- Sin argumentos: visualiza el contenido del directorio actual
- Para visualizar el contenido de un directorio concreto se incluirá su camino como argumento: **\$ ls mail**
- *Nota: Cuando se utiliza la orden ls sin opciones sólo se visualizan los nombres de los ficheros y/o subdirectorios que contiene el directorio*

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

ls [opciones] [directorio]

- Opciones más importantes:
 - -l listado en formato largo incluyendo permisos, usuario, grupo, etc.
 - -n muestra los mismos datos que -l pero con GID e UID en lugar de nombres de grupo y usuario
 - -t ordenado por fecha y hora de modificación
 - -c ordenado por fecha y hora de creación
 - -a lista todos los ficheros, incluso los que empiezan por punto, ordenados alfabéticamente
 - -s tamaño de los ficheros en bloques
 - -i informa del número de inodo de cada fichero
 - -R listado recursivo

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Ejemplo:

\$ ls -l

- Suele incluir siete columnas de información:
 - (1) Contiene diez caracteres. El primer carácter indica si el archivo es de tipo ordinario (-) o directorio (d). Los nueve caracteres restantes dan una representación simbólica de los derechos de acceso actuales que tiene el archivo
 - Estos nueve caracteres se dividen en tres grupos con tres caracteres cada uno. El primero representa al usuario, el segundo al grupo de trabajo al que pertenece el usuario, y el tercero al resto de los usuarios del sistema. Cada grupo muestra los permisos

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Ejemplo:

\$ ls -l

- Suele incluir siete columnas de información:
 - (2) Contiene el número de enlaces en el sistema de archivos a este archivo
 - (3) Muestra el propietario del archivo (el usuario que creó inicialmente el archivo)
 - (4) Muestra el grupo al que pertenecía el usuario cuando creó el fichero
 - (5) Visualiza el tamaño en caracteres (o bytes)
 - (6) Muestra la fecha en la que el archivo fue modificado por última vez
 - (7) Muestra la hora en la que el archivo fue modificado por última vez
 - (8) Contiene el nombre del archivo

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **pwd.** Presenta por pantalla el camino absoluto del directorio actual
- **mkdir.** Permite crear uno o varios directorios:
`mkdir [opciones] [camino]/ nombre/s_directorio/s`
- **-v:** Muestra un mensaje por cada directorio creado
- **rmdir.** Borra uno o varios directorios. Para poder borrar un directorio tiene que estar vacío:
`rmdir [camino]/ nombre/s_directorio/s`
- Si intentamos borrar un directorio que no está vacío nos aparecerá un mensaje de error y no se borrará el directorio

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **cd.** Permite cambiar de un directorio a otro:
cd [camino]
- El camino o ruta puede ser absoluto o relativo.
 - \$ cd /home/i82fecoe/mail: Nos sitúa en el directorio mail utilizando un camino absoluto
- Existen algunos atajos:
 - \$ cd ó cd ~: Nos sitúa en el directorio \$HOME
 - \$ cd ..: Nos sitúa en el directorio padre del directorio actual
 - \$ cd -: Nos cambia al último directorio donde estuvimos

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **rm.** Se utiliza para borrar ficheros. Su sintaxis es:
rm [opciones] fichero/s
- Sus opciones:
 - -r borra recursivamente. Se emplea para borrar directorios
 - -f borra todos los ficheros sin confirmación
 - -i pide confirmación para borrar cada fichero
 - -d borra el directorio, aunque no esté vacío
 - -v muestra un mensaje por cada directorio o fichero borrado

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **cp.** Se utiliza para copiar ficheros. Su sintaxis es:
`cp [opciones] fichero1 fichero2`
- Copia el contenido de “fichero1” en “fichero2”. Se puede especificar el camino absoluto o relativo de cada fichero
- Las opciones más importantes son:
 - -i pedirá confirmación si el fichero destino ya existe antes de sobrescribirlo
 - -r copia un directorio junto con sus ficheros y subdirectorios en otro directorio. Si el directorio destino no existe, lo crea
 - -f fuerza la copia

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **mv.** Cambia un fichero de un directorio a otro:
 - a) mv [opciones] fichero1 fichero2
 - b) mv [opciones] directorio1 nuevo_directorio
 - c) mv [opciones] fichero/s directorio
 - d) mv [opciones] directorio1 directorio2
- a) y b) cambia el nombre de un fichero o un directorio
- c) y d) cambia la localización
- Las opciones más importantes son:
 - -i si el fichero destino existe, pide confirmación antes de sobrescribirlo
 - -f es la opción por defecto. Si el fichero existe, lo sobrescribe sin pedir confirmación
 - -u Mueve solo si el destino no existe o es anterior
 - -v Muestra un mensaje por cada fichero movido

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **file.** Muestra el tipo de fichero

file [ruta/]nombre

- **du.** Muestra el espacio que ocupa el fichero o directorio:

du [opciones] [argumentos]

- Opciones:
 - -b: muestra el tamaño en bytes
 - -h: muestra el tamaño en la unidad de medida mayor, para que se pueda entender mejor
- Como argumento se puede pasar uno o varios ficheros o directorios

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **df.** Muestra el espacio libre en los dispositivos de almacenamiento y en las particiones montadas:

df [opciones]

- Opciones:
 - -h: Muestra el tamaño en la unidad de media mayor
 - -k: Muestra el tamaño en bloques de 1K
 - -a: Muestra todos, incluso si tienen tamaño 0

- **cat.** Visualiza el contenido los ficheros especificados como parámetros.

cat [opciones] archivos

- Opciones:
 - -n: Numera todas las líneas

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **head.** Obtiene en la salida estándar las primeras líneas de un fichero, por defecto las 10 primeras líneas. Su sintaxis es:

head [opciones] [fichero(s)]

- - n indica las n primeras líneas

- **tail.** Visualiza el final de un fichero, por defecto visualiza las diez últimas líneas. Su sintaxis es:

tail [opciones] [fichero]

- Donde las opciones pueden ser:
 - - n empieza a visualizar a partir de n líneas del final del fichero

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **wc.** Cuenta el número de caracteres, palabras y líneas de uno o más fichero(s), incluye espacios en blanco y caracteres de salto de línea.
- Cuando se indica el nombre de más de un fichero proporciona los contadores individuales y sus totales. Si no se especifica un fichero, leerá de la entrada estándar:
wc [opciones] [fichero(s)]
- -m visualiza sólo el número de caracteres del fichero
- -l visualiza sólo el número de líneas del fichero
- -w visualiza sólo el número de palabras del fichero
- -c visualiza el número de bytes
- -L muestra el tamaño de la línea más larga del fichero

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **more.** Muestra el contenido de un archivo paginando la salida por pantalla. Su sintaxis es:

more fichero1

- Pulsando <Enter> aparece una nueva línea
- Pulsando barra espaciadora aparece una nueva pantalla
- Pulsando q finalizamos la visualización

- **pg.** Se utiliza para visualizar el contenido de un fichero página a página. Su sintaxis es:

pg [opciones] [fichero(s)]

- -n obtiene las n primeras líneas

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **less.** Muestra el contenido de un archivo de la misma forma que more, pero con la diferencia de que podemos movernos por ellos usando también las flechas del cursor. Su sintaxis es:

```
less fichero1 ....
```

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **sort.** Se utiliza para ordenar líneas, por orden alfabético o numérico teniendo en cuenta que cada línea del fichero es un registro compuesto por varios campos, los cuales están separados por un carácter denominado separador de campo (tabulador, espacio en blanco, dos puntos...). Su sintaxis es:

`sort [opciones] [-knumero] [fichero/s]`

- `-knumero` cuando se quiere ordenar por un campo determinado del registro, se numeran a partir de 1. Por defecto un campo es cualquier secuencia de caracteres separados por espacios en blanco. Para parar la ordenación en otro campo se pone `-knº` de campo, por ejemplo: `-k1`
- ficheros nombres de los ficheros a ordenar

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Opciones:
 - -f ignora mayúsculas y minúsculas
 - -n ordena campos numéricos por valor no por caracteres
 - -r orden inverso
 - -u suprime líneas repetidas en el fichero de salida
 - -d ignora caracteres especiales (solo letras y números)
 - -t indica el delimitador de campos, por defecto el espacio en blanco
 - -o para que el fichero ordenado se almacene en el propio fichero. Funciona como una redirección
 - -m mezcla ficheros ya ordenados
 - -c comprueba si está ordenado (no ordena)

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **ln.** Un enlace es un mecanismo que permite referirse con distinto nombre de fichero a un mismo fichero en disco. Existen dos tipos de links: físico,(duro o hardlink) y simbólico (blando o softlink)
- Enlace duro: Crea un enlace, asociando uno o mas nombres de fichero con el mismo inodo (un fichero aparece bajo varios nombres). Se realizan mediante la orden ln. Su sintaxis es:

ln archivo destino

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **ln.** Enlace simbólico: Asocia un nombre a un nuevo inodo, que indica la posición en disco donde se guarda la información del path donde esta el fichero. Se realizan mediante la orden ln con el parámetro -s. Su sintaxis es:
ln -s archivo destino
- Cuando se hace un enlace simbólico se le asigna un nº de inodo nuevo. En ningún caso se incrementa el nº de enlaces. Sí se puede hacer de directorios. Los enlaces simbólicos tienen su propio tipo de archivo que se representa por una l y ocupan mucho menos que el archivo original porque solamente almacenan el nombre de la ruta de otro archivo

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **cut.** Muestra solo ciertas líneas verticales de los ficheros que se pasen como argumento. Su sintaxis:

cut [opciones] [fichero(s)]

- Donde las opciones pueden ser:
 - -b: muestra solo los bytes que se le especifiquen
 - -c: muestra solo los caracteres que se le especifiquen
 - -d: usa como delimitador el que se especifique, en lugar del tabulador
 - -f: muestra solo los campos que se le indiquen en la lista
 - -s: no muestra la líneas que no contengan el delimitador

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **grep.** Permite buscar cadenas de caracteres (patrón) en los archivos que se le indica y muestra la línea del fichero que contiene el patrón. Su sintaxis:

grep [opciones] patrón [fichero(s)]

- Donde las opciones pueden ser:
 - -c: visualiza el nº total de líneas donde se da el patrón
 - -i: elimina la diferencia entre mayúsculas y minúsculas
 - -l: visualiza sólo el nombre de los ficheros donde se localiza el patrón
 - -n: visualiza el nº de línea donde aparece el patrón, así como la línea completa
 - -v: visualiza las líneas del fichero donde no aparece el patrón
- Nota: Los comodines ? Y * no son válidos en este comando si no se escriben entre comilla simple, se recomienda utilizar el .

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Patrón:

- Expresión regular que define a un conjunto de cadenas de caracteres

CARACTER	ACCIÓN
^	La cadena buscada debe estar al principio de la línea
\$	La cadena buscada debe estar al final de la línea
.	Se corresponde con cualquier carácter, pero sólo uno y en la posición marcada por el
[a...z]	Coincide con uno cualquiera carácter de los que están incluidos en la lista
[x-y]	Sustituyen a un carácter y puede ser cualquiera de los comprendidos en el rango definido entre [].
[^]	Indica exclusión de los caracteres incluidos entre corchetes.
	Dentro de un patrón permite elegir entre uno u otro (aunque al ser un carácter especial suele ser necesario escaparlo con \ y usarlo con comilla simple)

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Patrón:

- Expresión regular que define a un conjunto de cadenas de caracteres

CARACTER	ACCIÓN
*	Cadena que se repite 0 o más veces
+	Cadena que se repite 1 o más veces
{n}	El carácter que le precede se repite n veces
?	El carácter que le precede es opcional

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **which.** Muestra la ruta absoluta del archivo del comando o de los comandos que se pasen como argumentos. Su sintaxis:

which argumento/s

- **locate.** Busca archivos dentro del sistema de archivos. Solo puede hacer búsquedas por nombre de archivo. Es muy rápido porque busca en una base de datos propia que se va actualizando periódicamente. Su sintaxis:

locate fichero

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **find.** Permite efectuar búsquedas de archivos. Busca dentro de los directorios indicados y en sus subdirectorios archivos que cumplan con el criterio:
find directorio [condiciones] acción
- Donde:
 - directorio es a partir del cual comienza la búsqueda
 - condiciones se puede substituir por:
 - -name nombre_fichero: Busca por nombre de fichero
 - -perm NNN: Búsqueda por la máscara de permisos en octal
 - -user nombre_usuario: Búsqueda por el nombre de usuario
 - -size n: Fichero con tamaño n bloques
 - -mtime n: Ficheros modificados hace n días
 - -atime: n Ficheros cuyo último acceso fue hace n días
 - acción indica la acción a realizar con los ficheros encontrados

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Ejemplos:

`$ find informes -name lunes`

- Se busca en el directorio informes el fichero lunes

`$ find programas -name '*.c' -print`

- Se busca en el directorio programas ficheros que tengan extensión .c y se muestra la ruta de los ficheros

- Se pueden usar los caracteres comodín del shell como parte del patrón del criterio de búsqueda de archivos. El carácter comodín debe ir delimitado con el fin de evitar su evaluación por el shell

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Cada vez que se inicia un intérprete de órdenes, se abren automáticamente tres archivos.
 - Recuerda que para LINUX todos los dispositivos físicos son archivos. Estos archivos son los siguientes:
 - **stdin**: archivo estándar de entrada, es de donde los programas leen su entrada (generalmente el teclado)
 - **stdout**: archivo estándar de salida, es a donde los programas envían sus resultados (pantalla)
 - **stderr**: archivo estándar de error, es a donde los programas envían sus salidas de error (pantalla)

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Redireccionamiento de Entrada (<)

- A cualquier orden que lea su entrada en **stdin** se le puede decir que que tome dicha entrada de otro archivo. No produce ningún cambio en el archivo de entrada

```
$ mail pepe < listado2
```

- Se envía el fichero listado2 por correo electrónico al usuario pepe

○ Redireccionamiento de Salida (>)

- La salida de cualquier comando se envía a un determinado archivo en vez de a la salida estándar stdout:
 - Si el archivo se creará automáticamente
 - Si ya existía se sobrescribirá la información, “machacando” el contenido original

```
$ date > prueba
```

- Se almacena la fecha actual del sistema en el archivo prueba

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

○ Redireccionamiento de Adición a un Fichero (>>)

- Si añadir información a un archivo sin destruir su contenido anterior

`$ ls >> prueba`

- Se almacena al final del fichero prueba el contenido del directorio activo

○ Redireccionamiento de Errores (2> y 2>>)

- Se utiliza para almacenar los errores producidos en la ejecución de un comando en un archivo específico, en vez de usar la pantalla: 2> y 2>>

`cat fichero>fichero1 2>errores`

- Si el archivo fichero no existe, o se da cualquier otro error, se crea un fichero llamado errores que contiene el mensaje del error que se hubiera mostrado por pantalla.

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- Permiten ejecutar un comando, y utilizar su resultado como para otro, que a su vez puede servir de entrada a un tercero

`$ comando1 | comando2 | comando3`

- La salida proporcionada por el comando1 se tomará como entrada para el comando2 y
 - la salida del comando2 es tomada como entrada para el comando3
- Estos filtros son more, tail, wc, grep, sort, head...

3. GESTIÓN DE ARCHIVOS Y DIRECTORIOS

- **tee.** En ciertas ocasiones puede interesar además de redireccionar la salida de una orden a un archivo, visualizar los resultados obtenidos. Su sintaxis es:

tee [opciones] fichero

- **Opciones:**
 - -a: escribe al final del contenido del fichero, no lo sobrescribe
 - -i: ignora las señales de interrupción

4. ARCHIVOS ESPECIALES

- Cuando ejecutamos el comando `ls -l`, el primer carácter indica el tipo de archivo:
 - '-' denota un archivo regular
 - 'd' denota un directorio
 - 'b' denota un archivo especial de bloques
 - Un archivo especial de bloques o dispositivo de bloques corresponde a dispositivos a través de los cuales el sistema mueve datos en forma de bloques
 - A menudo representan dispositivos direccionables tales como un disco duro, unidades de CD-ROM, o regiones de memoria

4. ARCHIVOS ESPECIALES

- Cuando ejecutamos el comando `ls -l`, el primer carácter indica el tipo de archivo:
 - 'c' denota un archivo especial de caracteres
 - Un archivo especial de bloques o dispositivo de bloques corresponde a dispositivos a través de los cuales el sistema mueve datos en forma de caracteres
 - Representan dispositivos como la impresora ó la pantalla
 - 'l' denota un enlace simbólico
 - 'p' denota una tubería con nombre
 - Indica que el fichero es una tubería con un nombre (creada con el comando `mkfifo`), que permite realizar comunicaciones entre procesos

4. ARCHIVOS ESPECIALES

- Cuando ejecutamos el comando `ls -l`, el primer carácter indica el tipo de archivo:
 - 's' denota un socket de dominio
 - Los sockets proporcionan una comunicación de dos vías, punto a punto entre dos procesos. Los sockets son muy versátiles y son un componente básico de comunicación entre interprocesos e intersistemas. Un socket es un punto final de comunicación al cual se puede asociar un nombre. Este tiene un tipo y uno o más procesos asociados
 - Los sockets existen en los dominios de comunicación. Un socket de dominio es una representación que da una estructura de direccionamiento y un conjunto de protocolos. Los sockets se conectan solamente con sockets en el mismo dominio. Los sockets pueden también ser usados para comunicar procesos entre diferentes sistemas

4. ARCHIVOS ESPECIALES

- /dev/tty1 ... /dev/tty6
- /dev/tty7
- /dev/pts
- /dev/sda1 ... /dev/sda2
- /dev/sdb1
- /dev/sr0 ... /dev/sr1
- /dev/null

5. EJERCICIOS

Los ejercicios se realizarán en Word (excepto el ejercicio 2), indicando los pasos seguidos en cada uno de ellos y realizando capturas de pantalla representativas. Su entrega se realizará a través del aula virtual

1. Busca información y define los diferentes entornos gráficos que podemos utilizar en Linux, indicando en aquellos casos que sea posible los paquetes de instalación necesarios para Ubuntu y Max
2. Realiza un esquema del sistema de ficheros que se utiliza en Ubuntu 10.10 de, al menos, 3 niveles, indicando que contiene cada carpeta del segundo nivel. En el tercer nivel incluye únicamente un par de los ficheros que aparecen.
3. Tenemos un fichero denominado semana.doc que almacena un día de la semana en cada fila, con la primera letra en mayúsculas. ¿Qué nos devuelven los patrones ^M y es\$?
4. Creamos un fichero nombres que contiene: López, Gómez, Montes. ¿Qué nos devuelve el patrón .ó.ez?

5. EJERCICIOS

5. ¿Qué veremos con los patrones [P], [Pp], [PF], [n], [o-s], ^[^P] si los aplicamos al fichero países cuyo contenido es el siguiente?
- Portugal
 - Francia
 - Panamá
 - Lisboa
 - España

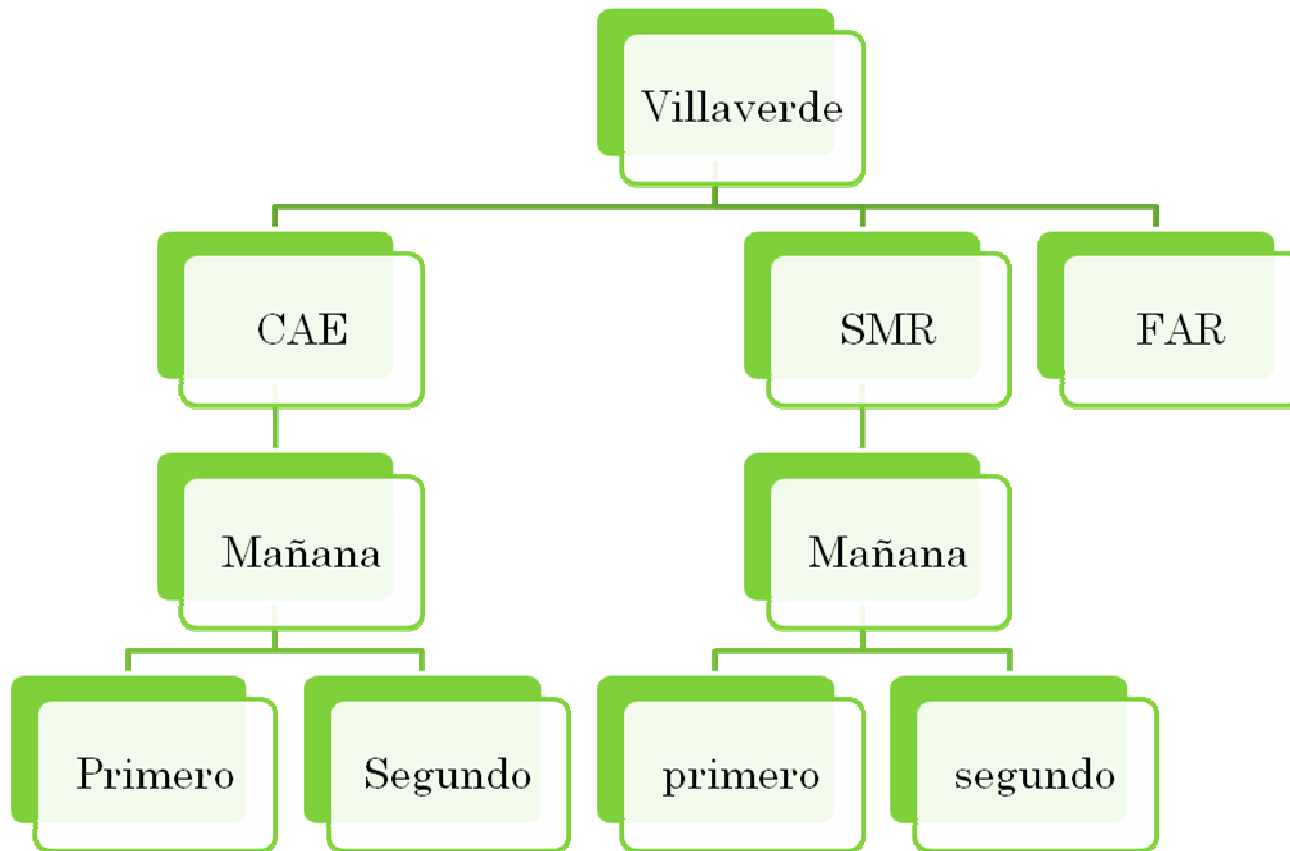
5. EJERCICIOS

6. Dado los siguientes ficheros ordena su contenido varias veces, haciendo uso de todas las opciones del comando sort:

\$ cat ciudades	\$ cat numeros	<i>\$ cat ciudades2</i>
barcelona	101	<i>España:Alicante</i>
oviedo	112	<i>España:Barcelona</i>
Paris	10	<i>Alemania:Berlín</i>
alicante	373	<i>Grecia:Atenas</i>
alicante	64	
OVIEDO		

5. Ejercicios

7. Crear el siguiente árbol con el menor número de comandos posibles:



5. Ejercicios

8. Realiza pruebas con los enlaces duros y blandos. Debes sacar conclusiones del efecto de tiene crear, cambiar o borrar tanto el archivo original como el enlace, teniendo en cuenta todas las opciones posibles
9. En el entorno gráfico realiza las siguientes acciones:
 - En tu directorio personal crea un fichero de texto (texto.txt) que contenga tu nombre y tu apellido, y crea un enlace a ese fichero con el nombre enlace.txt.
 - Crea una carpeta con el nombre nuevo, entra dentro y comprueba que está vacía.
 - Vuelve a tu carpeta personal, edita enlace.txt y añádele tu dirección.
 - Abre tu fichero texto.txt y comprueba su contenido.
 - Asocia el tipo de fichero texto.txt, para que cada vez que se abra, se abra por defecto con OpenOffice Org Procesador de textos, en vez de con editor de textos

5. Ejercicios

10. Añade un panel en la zona izquierda de la pantalla. Amplía el tamaño del panel a 61 píxeles y añade los siguientes elementos: Salir (cerrar sesión del usuario), notas adhesivas, frecuencia de la CPU, buscar archivos por su nombre o contenido unos ojos que sigan el cursor del ratón. Añade una nota adhesiva que te recuerde que debes realizar las actividades al UT8

Trabajando con comandos realiza los siguientes ejercicios:

11. Ejercicio de aplicación 4.1
12. Ejercicio de aplicación 4.2
13. Ejercicio de aplicación 4.3
14. Ejercicio de aplicación 4.4

5. Ejercicios

- 15. Ejercicio de aplicación 4.5
- 16. Ejercicio de aplicación 4.6
- 17. Ejercicio de aplicación 4.7
- 18. Ejercicio de aplicación 4.8
- 19. Ejercicio de aplicación 4.9
- 20. Ejercicio de aplicación 4.10