



Sistemas Informáticos: Conjunto de recursos físicos, lógicos y humanos que resuelven las necesidades de obtención, proceso y almacenamiento de información de un usuario o conjunto utilizando medios informáticos.

Un solo usuario: Sistema informático monousuario

Conjunto de usuarios: Sistema informático multiusuario

Sistema operativo: Conjunto de programas que permiten a los usuarios, utilizar, compartir y competir por los recursos físicos y lógicos del sistema informático. En este caso la complejidad es mayor y entran en juego factores como el competir y compartir.

Recursos humanos:

- Usuarios finales: Únicamente usan el sistema con fines productivos, ocio etc. El único conocimiento es de cómo funciona la aplicación.
- Usuarios programadores: Escriben programas o herramientas para las necesidades del resto de usuarios del sistema.
- Usuarios administradores: Procurar que todo el sistema funcione para que los usuarios puedan usarlos correctamente.

- **Funciones:**

1. Instalar aplicaciones
2. Actualizaciones de programas
3. Instalación y configuración de componentes físicos
4. Formación al resto de usuarios
5. Dar soporte a usuarios

Tema 2: Administración y aseguramiento de la información.

2.1 Introducción a las estructuras de directorios.

Para permitir el almacenamiento se necesita un sistema de archivos. MS-DOS primer sistema operativo de Microsoft.

El árbol (invertido) de directorios se basaba en tres conceptos:

- Unidades lógicas: Representaba los discos con letras que iba de la A(unidades de disco extraíbles(disqueteras)) C(disco duro)...Z(unidades de CD etc.).

- Unidad Física: dividida en varias partes lógicas
- Archivos o ficheros: Para poder identificarlo se utilizaba nombre.ext (nombre de 1 a 8 caracteres y extensión (de 0 a 3)).
- Directorio: Permite organizar y archivar ficheros (carpetas). Están sujetos a las mismas reglas de los archivos para nombrarlos. Pero los nombres de directorios se suelen poner sin extensión.

Notación SintaxisBNF

Notación	Significado
Arg	Obligatorio
{arg1 arg2}	Obligatorio uno de los dos
[arg]	Opcional
[arg1 arg2]	Opcional uno de los dos

Unidad activa y Directorio activo: Nos indica la posición activa dentro del árbol de directorio. Para eso el sistema mantiene una serie de apuntadores: Solo habrá un único apuntador de unidad activa(- - ->), mientras que cada unidad tendrá un apuntador de directorio activo(◇).

Nombre completo de archivo o directorio: Permite que el nombre del archivo sea único dentro de una unidad lógica.

Letra unidad: ruta(8) nombre archivo(3) Letra unidad: ruta. Nombre directorio

Rutas absolutas: Es aquella que se especifica desde la raíz de la unidad lógica. Por tanto empezará por “\”

Rutas relativa: Aquella que se especifica desde el directorio a priori. Donde se encuentra “la flecha” del directorio activo.

Los sistemas operativos trabajan con el nombre completo del archivo o directorio. Para no tener que escribir el nombre entero utilizaremos un mecanismo basado en la unidad activa, directorio activo y rutas absolutas y relativas. (Tabla cuaderno)

“.” Representa a un mismo directorio

“..” Representa el directorio padre

Definición de prompt: Elemento indicador que utilizan muchos entornos para informar al usuario de la disponibilidad del sistema para recibir una orden. MS-DOS y Windows utilizan el prompt, todo por teclado. En Windows el prompt indicará la unidad activa y el directorio activo.

Órdenes:

1. ¿Cómo cambiar de unidad activa?

Letra: intro◇ Escribir letra unidad a la que quiero pasar.

Ejemplo: C:\ Windows> H:

2. Creación de archivos:

copy con nombre archivo (forma absoluta o relativa)

se pulsa F6 para terminar.

3. Visualizar contenido del archivo:

Type [Letra:] [ruta] nombre archivo

4. Creación directorio(o carpeta)

MD [Letra:][ruta] nombre directorio

5. Cambio de directorio activo

CD [Letra:][ruta] nombre directorio

6. Visualizar contenido directorio

DIR [Letra:][ruta]nombre archivo

➤ Opciones de las órdenes

Estas opciones vendrán precedidas por /

➤ Comodines:

Carácter especial que podemos usar en un nombre de archivo para identificar a más de un archivo.

- Tipos:

“*” vale para más de una posición

“?” solo vale para la posición en la que lo utilizamos

7. Borrado de archivos

DEL [unidad:][ruta] nombre archivo

8. Borrado de directorios

RD [unidad][ruta]

1. El directorio a borrar debe estar vacío.
2. Para borrarlo no podemos estar posicionado en él.

9. Copia de archivos

copy__origen__destino

10. Mover archivos

Move__origen__destino

10.1 Sobre un directorio renombra el directorio

11. Renombrar archivo

Rename(REN)__origen__destino(nuevo nombre)

12. Ver árbol gráficamente

Tree__ [unidad][ruta] *solo carpetas

Si queremos ver archivos /F | more

Tabla resumen

archivo	directorio
Nombre.ext	Nombre
Copy_con	MD
DEL	RD
TYPE	DIR
RENAME REN	MOVE

2.3 Gestión de árboles de directorios desde el entorno gráfico. Equivalencia con el entorno de comandos

2.3.1 Nombres largos y cortos de archivos y directorios.

A partir de W 95 se podía modificar la restricción de 8.3(nombre.txt), más la utilización de algunos caracteres.

- **Long file names:** un nombre será largo si:
 1. Tiene más de 8 posiciones en el nombre
 2. Tiene más de 3 posiciones en la extensión
 3. Contiene espacios en blanco o algún carácter no válido en la anotación 8.3
- **Observaciones**
 1. La longitud máxima para un nombre de archivo es de 255 caracteres.
 2. Longitud máxima de un nombre completo (ruta+nombre) 260 caracteres.
 3. Caracteres prohibidos: "<>, :\",/\\, |,?,*"
 4. Palabras reservadas: "PRN, AUX, NULL, CON1, CON2,..., LPT1, LPT2,... No pueden usarse como nombre de archivo
 5. Como ahora podemos utilizar espacios en blanco en los nombres. Ahora para separar utilizaremos ""

Ejemplo: copy "memoria_curso_2010.doc"._

Cada vez que creamos un nombre largo el sistema genera un nombre corto (normativa 8.3)

- **Para crear un nombre corto a partir de un nombre largo:**

1. Se suprimen todos los espacios en blanco y algunos caracteres (vocales acentuadas...)
2. Se suprimen todos los puntos extras, excepto el último
3. Los caracteres no permitidos en 8.3 se substituyen por ' _ '
4. La extensión se tronca a los tres(o menos) primeros caracteres después del '.'
5. Se cogen los 6 primeros y se le añade el carácter 'ASCII 126' seguido de un nº(1...9) si necesitamos el carácter número 10 solo cogería 5 caracteres.
6. Se traducen todos los caracteres a mayúsculas

7. Se pueden consultar los nombres cortos asociados a los nombres largo. "Dir /x".

- **A partir de W.XP**

Las normas 4 Y 5 cambian hasta los 5 primeros nombres permanecen igual. A partir de ahí coge los 2 primeros caracteres y colocará un código por ejemplo: __A1C1~1.___ . En el ejemplo de carta si llegará a carta~5 ésta quedará así ca----~1.doc

Si en algún momento se borra un archivo quedará un hueco en la numeración. Al crear uno nuevo con el mismo nombre.ext lo nombrará con el número bacante.

2.3.2 Acceso directos y puntos de unión (junctions)

- ❖ Un acceso directo es una extensión. LNK. En un acceso directo no hay ramas, hay que manejarlo como si fuera un archivo.
- ❖ Junctions: Los árboles de W.XP y W7 han cambiado por lo tanto ahora en W7 hay un directorio llamado "documents and settings" que re direcciona a los directorios que contiene la carpeta "users". Se utiliza para programas que utilizan directorios anteriores de W.XP.

(dir /ad te enseña junctions + carpetas; dir /al te enseña solo junctions)

1. Están ocultos
2. Tienen el mismo aspecto del acceso directo
3. No podemos acceder a ellos.(Acceso denegado) Impide que se pueda acceder al contenido de las carpetas, pero se pueden borrar. En XP los junctions si se borran, borras los originales en W7 en teoría no

2.3.3 Carpetas comprimidas

Carpetas que contienen ficheros, carpetas etc. Pero para la consola son ficheros.zip

2.3.4 Archivo desktop.ini

Archivo de configuración de directorios, que sirve para controlar el aspecto que se le muestra al usuario. En principio está oculto. Es un archivo ASCII se puede editar con un bloc de notas y las líneas que contiene. Por eso puede haber dos archivos con el mismo nombre. Él es el culpable de que archivos de programa no sea program files.

2.3.5 bibliotecas

Nos permite reunir en un mismo sitio archivos de distintas ubicaciones, mostrándolos en una única ubicación pero sin moverlas de donde están ubicados. Puede haber archivos con el mismo nombre pero pertenecen a ubicaciones distintas se configura por usuario. La configuración esta guardada en un fichero xml de extensión .library-ms y se encuentra x:\users\asixs\appdata\roaming\microsoft\windows\libraries. Es una carpeta oculta

17/10/11

2.3. Re direccionamiento de la entrada y la salida estándar, filtros y canalizaciones.

-

Cada proceso abre tres archivos estándar:

Canal 0-Entrada Standard (STDIN): Lugar por el cual el proceso prevé obtener los datos que va a procesar. Por defecto la entrada será el teclado

Canal 1-Salida Standard (STDOUT): Lugar por el cual prevé que va a devolver los resultados. La salida será la pantalla.

Canal 2-Error Standard (STDERR): Es el lugar en el cual va a escribir los mensajes de error en el caso de que produzca algún error durante el proceso. La salida será la pantalla.

Estos canales por defecto se pueden re direccionar por ejemplo si quiero que me lo tire por la impresora o en un fichero... Esto se hace mediante los operadores:

- "<" Re direcciona la entrada ó 0<.
- ">" Re direcciona la salida ó 1>.
- ">>" Re direcciona la salida con adición ó 1>>.
- "2>" Re direcciona error.

- "2>>" Re direcciona error con adición.

Ejemplo: H:\> DIR > Lista.txt por pantalla no saldrá nada enviaré los datos al archivo Lista.txt

H:\> DIR c:\ Windows >> Lista.txt Le añadirá la lista del contenido de Windows a lo que ya había en lista. Si fuera un ">" sustituiría lo que hay en Lista.txt

(No existe D1) H:\> RD_D1 si queremos que se guarde el mensaje de error utilizaremos: H:\> RD_ D1_2>_err.txt

H:\> DIR > PRN el resultado de DIR lo enviara a la impresora y lo imprimirá.

- **Filtros:**

Sort: En su forma primitiva, recibía una serie de líneas por la entrada estándar y las ordenaba según la tabla ASCII y las sacaba por la salida estándar

H:\> SORT

4

3

1

2

F6--- ^z

1

2

3

4

H:\>

H:\>Sort< Numeros.txt > Numord.txt Cambio la entrada para que me ordene el contenido que hay en Numeros .txt pero el fichero sigue estando igual y lo meterá en Numord.txt

Observaciones:

1. Se basa en la tabla ASCII

2. Si quiero meter los datos ordenados en el mismo fichero hay que ir con cuidado porque se puede perder la información. Por eso crearemos un fichero auxiliar con otro nombre y luego borraremos el original y renombraremos el auxiliar. Si ordenamos sobre el mismo fichero consideraremos que perdemos la información así que hay que hacer lo anteriormente dicho

3. La orden sort puede utilizar los siguientes modificadores

"/R" ordenará en orden decreciente

"/+nº" ordenará respecto otro carácter, columna (cuaderno)

4. Para ordenar siempre lo haremos a través de direccionamiento

Find: Muestra por salida estándar las líneas de un conjunto de ficheros en las que aparece una determinada cadena de caracteres.

Find_ "Palma de mallorca" __clientes.txt __proveed.txt > RDO.txt Aparecerán todos los clientes y proveedores en palma de Mallorca y lo re direcciona a una salida RDO.txt, no saldrán por pantalla pero los tendré en el fichero.

Observaciones:

-Puede ir acompañado de una serie de modificadores. Mirar en ayuda, se pueden utilizar todos

More: Parecido al SORT. Pero te lo muestra por diferentes pantallas. Pero lo utilizaremos para que nos lo muestre por archivos.

H:\> more < FITX.txt

H:\>

-Canalizaciones: Es posible conectar dos o más órdenes de forma que la salida estándar de una orden sea la entrada estándar de la siguiente y así sucesivamente

|comando1|comando2| (pantalla) La salida de comando 1 es la entrada de comando 2

Para hacer eso hay que utilizar el carácter ASCII 124 (|) Este carácter es el que tenemos que poner de los dos comandos.

Ejemplo: Type__ fitx.txt | sort | more. Lo enseña ordenado y por pantalla.

19/10/2011

2.4 Variable de entorno

Es una posición de memoria que contiene un valor.

El entorno es una particular área de memoria que almacena variables con información acerca del sistema operativo y que son utilizadas por este y por otros programas para guardar y leer información necesaria para su funcionamiento.

Las variables de entorno son cadenas que contienen información como por ejemplo: letras de unidad, rutas, nombre de ficheros etc. Permite definir el directorio dónde se guardará los archivos temporales. Esto es útil porque si el programa se cierra incorrectamente el archivo temporal estará en el directorio elegido por la variable. Tipos:

- **De sistema:** Las define el sistema operativo y se aplican a todos los usuarios del equipo. Las modificaciones hechas a variables del sistema se escriben en el registro de Windows y por lo general requieren que se reinicie el equipo para hacerse efectivos estos cambios. Tenemos que ser administradores para poder modificar una variable del sistema.
- **De usuario (o locales):** Son diferentes para cada usuario, algunas son creadas por el SO, otras por programas y otras por el usuario. Los cambios en estas variables pueden o no escribirse en el registro y por lo general entran en vigor inmediatamente, puede suceder que al cambiar una variable de entorno que pertenece al registro de un programa, se nos pida reiniciar el programa.

Ordenes de las variables en consola de comandos:

- **SET:** Nos muestra una lista de todas las variables de entorno que tenemos definidas.
- **SET nombrevariable:** Nos informa únicamente del valor almacenado en esta variable. Esta posibilidad **no** está presente en todas las versiones de la consola.
- **SET nombrevariable = valor:** Si la variable no existe la creamos asignándole el valor que le hemos dado y si ya existía modificamos su valor.
- **SET nombrevariable=:** Liberamos el espacio que estábamos gastando.
- **% nombrevariable %:** Para utilizar el valor de la variable.

Ejemplo:

```
E:\> set DVD=G:
```

```
E:\> set dirtxt= E:\users\asix\desktop
```

```
E:\>%DVD% // cambia la ruta a G:\
```

```
G:\>copy con %dirtxt%\fich1.txt // crea archivo fich1.txt en escritorio
```

-

-

```
G:\>copy con %dirtxt% fich1.txt //crea fich1.txt en E:\users\asix\Desktop\fich1.txt
```

- **ECHO %variable%:** Saca por pantalla el valor de la variable.

Variable temp: Contiene todos los archivos temporales.

Variable windir: Contiene archivos .EXE del directorio en el que está instalado el sistema operativo.

Variables PATH y PROMPT:

Para utilizar un programa desde la consola de comandos pondremos el nombre del archivo ejecutable.

Comandos internos y externos:

-**Comandos internos:** son aquellos que se encuentran dentro de CMD.EXE (archivo ejecutable de la consola). DIR, TYPE, MD...

-**Comandos externos:** Aquellos que no están resueltos dentro del CMD.EXE. XCOPY... Para cada una de ellas existirá un fichero con el programa para hacer esta orden.

-La variable **PATH** guarda una lista de directorios en los que se buscará un ejecutable cuando este no se encuentre en el directorio actual. Por eso las órdenes externas se pueden ejecutar desde CMD.EXE. Hay una orden PATH

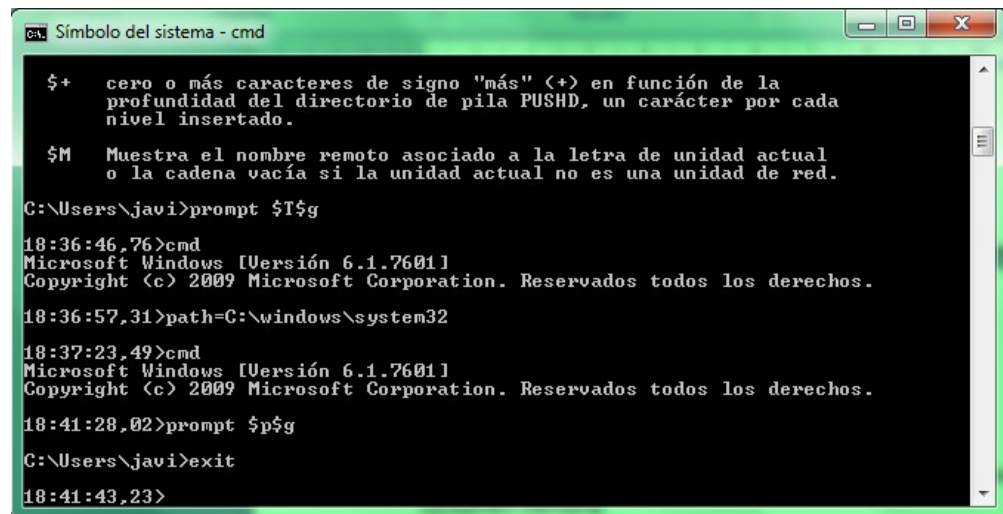
-La variable **PROMPT** permite configurar el aspecto del prompt para ello se le ha de asignar a esta variable una serie de códigos \$p\$g(\$g = >). Hay una orden PROMPT Por ejemplo:

H:\>Set prompt = hola\$g

Ó // se puede hacer igual.

H:\>prompt hola\$g

hola>.....



```

C:\Users\javi>cmd
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

18:36:46,76>path=C:\windows\system32

18:37:23,49>cmd
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

18:41:28,02>prompt $T$g

C:\Users\javi>exit

18:41:43,23>

```

Herencia de variables

Cada programa que se inicia recibe una copia del entorno (entorno hijo) de manera que si esta aplicación edita alguna variable lo hace sobre su copia. De esta manera se preserva la identidad del entorno padre.

Windows

↙ ↘

Hijo Hijo

↓

Hijo

Cada hijo hereda una copia del entorno de su padre. Las variables heredadas dejan de ser de sistema. Es decir no solo puede editarlas el administrador. Para hacer un hijo de un hijo tienes que poner **cmd** en el prompt. Para volver al padre del hijo tienes que utilizar la orden **exit**

-Pseudo variables

No almacenan un valor fijo sino que se evalúan al momento de pedir su valor, es decir, no consumen memoria.

%CD%: Se expande a la ruta del directorio actual

%DATE%: Enseña la fecha

%TIME%: Enseña la hora

%RANDOM%: Me devuelve un número aleatorio

%ERRORLEVEL%: Devuelve un código de salida de la última orden ejecutada. Si la última orden ejecutada ha tenido éxito devuelve un 0 y si no devolverá un 1

```
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\javi>type fich1.txt
El sistema no puede encontrar el archivo especificado.

C:\Users\javi>echo %errorlevel%
1

C:\Users\javi>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 18F9-77C7

Directorio de C:\Users\javi

01/10/2011  13:16    <DIR>          .
01/10/2011  13:16    <DIR>          ..
26/10/2010  22:53    <DIR>          .nbi
26/10/2010  22:53    <DIR>          .netbeans
26/10/2010  22:52    <DIR>          .netbeans-registration
19/09/2011  12:23    <DIR>          Contacts
19/10/2011  18:42    <DIR>          Desktop
17/10/2011  16:19    <DIR>          Documents
17/10/2011  11:29    <DIR>          Downloads
19/10/2011  11:09    <DIR>          Dropbox
```

2.6 Administración de discos: Estructura física de un disco.

La estructura del disco se crea cuando sale fábrica. Estructura:

- Caras (o cabezas). En los discos duros, 16 caras, es decir un disco está formado por 8 discos. En un disco con n caras se numeran de 0,1...,n-1
- Pistas: Círculos concéntricos. En los discos duros se puede hablar de cilindros en lugar de pistas. Si dividimos las caras en círculos concéntricos. La pista uno de todas las caras formará el cilindro. Numeración de 0 a n-1
- Sectores: Cada pista o cilindro se divide en sectores. La información que cabe en un sector es la misma. Lo que pasa es que la densidad será mayor o menor en relación a la distancia del centro. La técnica ZBR(zone bit recording) aprovecha las pistas exteriores para tener más sectores mientras que las pistas internas tendrán menos(**mirar dibujo internet**).

2.7 Administración de discos: Estructura lógica de un disco.

2.7.1 Particiones

Esta estructura se crea cuando el usuario prepara el disco para que pueda contener software y/o datos. Esta preparación consiste en particionamiento y formateo de unidades lógicas.

- Partición: División lógica que se realiza en un disco duro para poder segmentarlo en varias entidades independientes. La función de una partición es doble
 1. Permitir la coexistencia de varios sistemas operativos.
 2. Dividir la unidad física en varias unidades lógicas.

- Partición activa: Es la partición que toma el control cuando se “arranca” el ordenador en todo momento solo puede haber una y solo una, partición activa en el sistema.
- Partición primaria y partición extendida:
 1. Una partición primaria es una
 - partición obligatoria inicialmente solo podían tener un sistema operativo este tipo de particiones por tanto
 - solo podía definirse como activa una partición de este tipo.
 - Designa una única letra de unidad lógica
 2. La extendida es
 - opcional sirve para dividir el disco en distintas unidades lógicas que inicialmente no podían almacenar un sistema operativo.
- Tipos de disco según el estilo de tabla de particiones. Podemos distinguir dos tipos de disco:
 1. Discos MBR(Master Boot Record): El disco se estructura de la siguiente manera: El primer sector del disco recibe el nombre de registro maestro ... En el tenemos dos elementos: **(Dibujo) cuaderno**
 - La tabla de particiones: que permite registrar un máximo de 4 particiones. Nos indica donde empieza y donde acaba la partición. Podría tener como máximo 3 primarias y una extendida o 4 primarias. Con la partición extendida podremos superar el máximo porque en ella podremos almacenar más divisiones(particiones).
 - Master bootCode (IPL): Es un pequeño programa que lee en la tabla de particiones, cual es la partición que está marcada como activa para cederle a ella el control durante el proceso de arranque del sistema informático.

Cada una de estas particiones puede ser de hasta 4 TB.

2. DiscosGPT (GuidPartitionTable): Están basados en el guid un número pseudoaleatorio que permite manejar hasta 128 particiones y cada una de ellas de hasta 18 Exabytes. Estos discos fueron diseñados para ofrecer un gran rendimiento en equipos basados en el tipo “ITANIUM” y se pretende que esto llegue a desbancar el otro tipo de discos que están basados en “BIOS”.

Secuencia de arranque en PC’s basados en BIOS(dibujo cuaderno): Consiste en la ejecución en el orden correcto de unos 4 ó 5 pequeños programas.

El primer programa se encuentra en un chip de la placa base y se llama "BIOS". Realiza el test del hardware cuando se inicia la máquina y cuando termina ese test si todo ha ido bien le cede el control al segundo programa.

El segundo programa es el IPL. Le cede el control al primer sector de la partición activa. Este sector recibe el nombre de PBR (partitonboot record) este es el tercer programa.

El tercer programa PBR depende del sistema operativo que está instalado. Cada sistema operativo tiene su propia secuencia de inicio. Por lo tanto este programa será distinto dependiendo del sistema operativo serán más o menos pasos. Si es un W.XP le cede el control a un cuarto programa.

El NTLDR es el primer fichero que podemos ver haciendo un dir. Este programa es un gestor de arranque del W.XP que lo que hace es leer un fichero de iniciación llamado boot.ini e irá al directorio en el cual tenemos instalado W.XP y lo iniciará.

Si es un W.Vista o W.7 el *PBR* Buscará un elemento de inicialización llamado bootmgr que este le cederá el control a un ejecutable llamado winload.exe que iniciará el sistema (5º programa).

Herramientas para manejar esto

Hasta Windows millenium... se incluía una herramienta que se llamaba FDISK que tenia muchas limitaciones y tenias que recurrir a un programa de terceros que se llamaba partitionmagic.

A partir de W.XP ya no está esta utilidad de FDISK y se incluyen utilidades tanto de modo texto como gráfico.

Modo texto:

- DISKPART: Es una utilidad que tiene su propio prompt

Modo gráfico: En administrar llegareis a una utilidad llamada:

- Administración de discos:

2.7.2 Sistemas de archivos

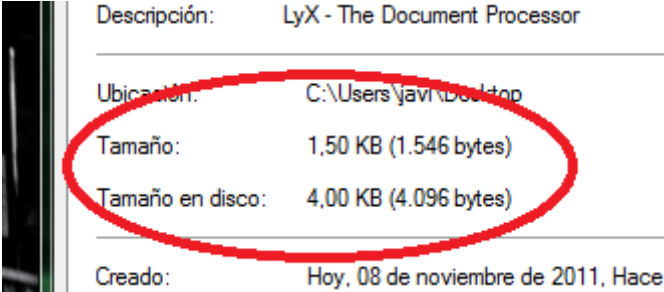
Cuando se hace las particiones tenemos que dotarlas de un sistema de archivos. Durante el proceso de formateo de la unidad lógica se crea la sistema de archivos.

(cluster paréntesis)

- **Cluster o unidad de asignación:** un cluster es un conjunto de sectores contiguos de disco. Unidad mínima desde el punto de vista lógico. Su tamaño depende del tamaño de la unidad lógica normalmente, a mayor tamaño de la unidad mayor cluster. El tamaño del cluster es siempre potencia de 2

1 cluster sectores. Un sector es ½ Kb . Un archivo ocupa como mínimo un cluster, aunque no lo llene. Si un archivo ocupa varios clusters estos no tiene porque ser contiguos, los archivos pueden estar fragmentados. El tamaño del

cluster afecta a la utilización del espacio de disco y al tiempo de acceso. Tamaño en disco en un acceso directo refleja el tamaño entero del cluster.



Descripción:	LyX - The Document Processor
Ubicación:	C:\Users\javi\Desktop
Tamaño:	1,50 KB (1.546 bytes)
Tamaño en disco:	4,00 KB (4.096 bytes)
Creado:	Hoy, 08 de noviembre de 2011, Hace

Tipos de sistemas de archivos:

- **FAT(dibujo cuaderno):** Es una tabla con tantas entradas como clusters tiene la partición o unidad lógica y que sirve para averiguar la secuencia de clusters en los que está almacenado un fichero. Cada casilla de la FAT puede tener uno de los siguientes valores:
 - 0 para indicar que es un cluster libre
 - EOF (end of file) marca para indicar que es el último cluster de un fichero
 - Una dirección de cluster que se interpretará como la dirección del siguiente cluster de un archivo

5. Estructura después de ejecutar el FAT:

- MPB(Media Parameter Bloc): Tabla con información sobre la partición: tamaño cluster, etiqueta del volumen etc.
- VBC(VolumeBootCode): Pequeño programa específico de cada sistema operativo, que será invocada por el IPL del MBR cuando esta partición esté definida como activa.
- Área de datos: área utilizada para almacenar los datos del usuario y ocupa la práctica totalidad del espacio de la partición (98%). Está dividida en clusters.
- Directorio raíz: Para cada elemento archivo o directorio, almacenado en la raíz del árbol de directorios de esta unidad tenemos una entrada en la que se registra, nombre, extensión, tamaño, fecha y hora, atributos y la dirección del cluster de inicio de dicho elemento.

(EJEMPLO cuaderno)

Tenemos que hay distintos tipos de archivos que utilizan el formateo FAT:

Fat16: Porque utiliza 16 bits para cada entrada de la FAT. Esto nos informa de la cantidad de cluster que podemos tener. Como máximo 2^{16} clusters. **(tabla fotocopia)**. Sistema originario de DOS y se sigue utilizando actualmente en disquetes

VFAT: Este sistema de archivos aparece con W.95. Mejora del Fat16 para poder contemplar el uso de nombres largos.

FAT32: Porque utiliza 32 bits para cada entrada de la FAT. El número de clusters posibles se dispara notablemente. 2^{32} clusters posibles.

- **NTFS(NT File Sistem)(dibujo cuaderno):** Aparece con Windows NT, para este sistema de archivos todos son archivos hay dos tipos de archivos:

4. Metadata Files: Son los ficheros que contienen las estructuras y la información que sustenta el sistema de archivos. El nombre de estos archivos empieza por el carácter '\$'

5. Normal Files: Ficheros de usuario.

- **MFT(master file table):** Tabla maestra de ficheros. Tabla que hay que consultar para saber la secuencia de clusters en la que está almacenado un fichero. Ella misma está registrada en esa tabla. (Dibujo cuaderno MFT).

- **Ficheros Metadata:** Los más destacables:

- \$MFT Mirr: Copia exacta de los 4 primeros registros de MFT. Permite recuperar el volumen si se daña esta tabla
- \$LogFile: Registro de las operaciones de escritura de disco pendientes
- '\$': Directorio raíz.
- \$Bitmap: Mapa de bits que contiene un Bit por cluster para indicar si un cluster está libre bit=1 u ocupado bit=0
- \$Badclus: Tabla de clusters erróneos.
- Etc.

- **Espacio libre 1 MFT zone:** La tabla MFT va creciendo por lo tanto necesitamos el espacio libre 1 que es como un buffer de crecimiento de la MFT. Para prevenir que la MFT pueda estar fragmentada, se mantiene un buffer que podemos configurar, al 12'5%, 25%, 37'5% ó 50% del total de la unidad y que no será utilizado para almacenar datos de usuario, a menos que todo el espacio restante ya este ocupado. Este es el motivo de que en el esquema anterior aparezca el espacio libre entre la MFT y el resto de ficheros metadatafile. En principio mientras podamos guardar datos en el espacio libre 2, el espacio 1 será utilizado como buffer de crecimiento de la MFT. Si llegará el momento de tener que utilizar espacio 1 para datos, éste se irá cogiendo del final para seguir manteniendo la MFT sin fragmentar

2.7.3 Formatear una unidad lógica:

Cuando compras una unidad de almacenamiento ya viene formateada- Nosotros deberemos formatear siempre que definamos nuevas particiones. Hay que distinguir dos tipos de formateo:

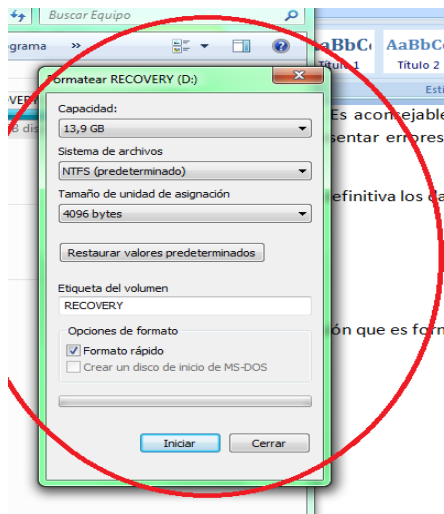
- **Rápido:** Aplicable únicamente a unidades ya formateadas en algún momento o sesión de trabajo previa. No se crea una estructura lógica nueva, sino que se aprovecha la estructura ya creada en un formateo anterior. Se marca como libres todos los clusters sin borrar físicamente ningún cluster del disco. Por este motivo con la herramienta adecuada todavía es posible recuperar la información.
- **Completo e incondicional:** Destruye todos los datos que pudiera haber en el disco creando una nueva estructura lógica completa en el disco. Por este motivo tarda más que el método anterior y ya no es posible recuperar de ninguna manera los datos. Es aconsejable utilizar este método en discos que hayan podido presentar errores de lectura o escritura

Observaciones: Formatear una unidad podemos perder de forma definitiva los datos. Desde el entorno texto:

tenemos un comando para formatear *format*, otra orden es:

Convert convierte volúmenes FAT en NTFS.

En el entorno gráfico, menú contextual de la unidad, hay una opción que es formatear.



La etiqueta en fat hasta 11 con prohibidos y en nfts hasta 32 sin caracteres prohibidos. Esta etiqueta se guarda en la NPB. Para la gestión de estas etiquetas tenemos dos órdenes.

Vol: consultas la etiqueta.

Label: Podemos consultarla y modificarla.

(2ª parte de las fotocopias)

3. Virtualización

3.1 Introducción

El hardware informático actual se ha diseñado para ejecutar un solo sistema operativo y una sola aplicación, lo que supone la infrautilización de gran parte de las máquinas. La virtualización permite ejecutar varias máquinas virtuales en una misma máquina física donde cada una de las máquinas virtuales comparte los recursos de ese ordenador físico, único, entre varios entornos.

Las distintas máquinas virtuales pueden ejecutar SO diferentes y varias aplicaciones en el mismo ordenador físico. Además de su utilización en entornos productivos es una herramienta importante para un administrador de sistemas ya que permite estudiar y analizar el comportamiento de distintos recursos y entornos antes de su implantación real en la empresa.

3.2 Máquinas virtuales

Una máquina virtual es un contenedor de software perfectamente aislado, que puede ejecutar sus propios sistemas operativos y aplicaciones como si fuera un ordenador físico. Una máquina virtual se comporta exactamente igual que lo hace un ordenador físico y contiene sus propias CPU, RAM, disco duro y tarjetas de interfaz de red virtuales, es decir, basadas en software.

Ni el sistema operativos, ni los programas pueden establecer una diferencia entre una máquina virtual y una máquina física incluso otros ordenadores de una red tampoco podrán hacer esa distinción. Incluso la propia máquina virtual considera que es un ordenador real, sin embargo una máquina virtual, se compone exclusivamente de software y no contiene ninguna clase de componente hardware.

3.3 Funcionamiento

La virtualización inserta una ligera capa de software en el hardware del ordenador o en el sistema operativo HOST. Contiene un monitor de máquina virtual (**VMM**, virtual machine monitor o **hipervisor**) asigna los recursos de hardware de manera dinámica y transparente a cada una de las máquinas virtuales. Hipervisores, dos tipos:

1. Tipo 1(nativo, unhosted, o “sobre el metal desnudo”(Baremetatl)): En este caso el hipervisor es un software que se ejecuta directamente sobre el hardware.
(Dibujo)
2. Tipo 2 o HOSTED: Un software que se ejecuta sobre un sistema operativo para ofrecer estas características de virtualización que estamos describiendo.
(Esquema cuaderno). Trabajaremos con esta estructura

3.4 Ventajas de las máquinas virtuales

En general las máquinas virtuales cuentan con 4 características que benefician al usuario:

1. Compatibilidad: Las máquinas virtuales son completamente compatibles con la totalidad de SO, aplicaciones y controladores de dispositivos estándar. De manera que se puede utilizar una máquina virtual para ejecutar el mismo software que se puede ejecutar en un ordenador físico.
2. Aislamiento: Aunque las máquinas virtuales pueden compartir los recursos físicos de un único ordenador real, permanecen completamente aisladas unas de otras, como si se trataran de máquinas independientes. De manera que si tenemos dos maquinas virtuales sobre un mismo ordenador físico y falla una de las máquinas las otras siguen estando disponibles.
3. Encapsulamiento: Una maquina virtual es un contenedor de software que agrupa o encapsula un conjunto completo de recursos hardware virtuales, así como un SO y todas sus aplicaciones dentro de un paquete de software. El encapsulamiento hace que las maquinas virtuales sean extraordinariamente portátiles y fáciles de gestionar por ejemplo podemos copiar y mover una máquina virtual de un lugar a otro como lo haríamos con cualquier otro archivo de software

4. Independencia del hardware: Las máquinas virtuales son completamente independientes del hardware físico subyacente, por ejemplo se puede configurar una máquina virtual con componentes físicos virtuales (Tarjeta de red etc.) que difieren totalmente de los componentes físicos presentes en el hardware subyacente. Esta característica combinada con el encapsulamiento hace que pueda coger la máquina virtual a otro ordenador y funcione correctamente.

3.5 Infraestructura virtual

Las máquinas virtuales son un componente básico fundamental de una solución mucho mayor que es la infraestructura virtual.

¿?¿?¿? Mientras que una máquina virtual representa los recursos de hardware de todo el ordenador una infraestructura virtual representa los recursos de hardware interconectados de la totalidad de una infraestructura de la tecnología de la información lo que incluye ordenadores, dispositivos de red y dispositivos de almacenamiento.¿?¿?¿?

4. Aseguramiento de la información

4.1. Atributos de ficheros y directorios

Los atributos son propiedades normalmente de seguridad de los ficheros y directorios comunes para todos los usuarios.

Depende del sistema de archivos en el que se encuentra almacenado un archivo o un directorio. Podemos distinguir dos tipos de atributos:

- Fat: Los atributos FAT Son 4:
 1. R (ReadOnly): Impide que se pueda modificar o eliminar un fichero. La naturaleza de algunos soportes de información, puede hacer que este atributo esté activado
 2. H (Hidden u oculto): Impide que aparezca un archivo en operaciones de listado. No se puede ni copiar, ni mover, ni suprimir a no ser que se conozca su nombre o que tengamos activa la opción de ver archivos ocultos.
 3. A (Archive)**OJO**: No se trata de un archivo, es un atributo para los programas de copia de seguridad para que estos puedan identificar los archivos que han sido creado o modificados desde la última copia. Cuando se hace la copia de seguridad se borra este atributo, si se ha modificado este archivo desde la última vez aparecerá automáticamente este atributo.

4. S(System): Indica que el archivo es para uso del SO. Es utilizado para identificar los archivos del sistema
- NTFS: 7 Atributos:
 1. Los tres anteriores(RHAS)
 2. I (NotcontentIndexed):Indica que el archivo o carpeta se indexará, para realizar una búsqueda rápida de texto en su contenido, propiedades o atributos.
 3. C (Compressed): Indica que el archivo o carpeta se comprimirá automáticamente para ahorrar espacio en disco. **No es lo mismo que una carpeta comprimida.**
 4. E(ENCRYPTED): Indica que el archivo o carpeta se cifrará para evitar que su contenido pueda ser visto por otro usuario.

Observaciones sobre 3 y 4

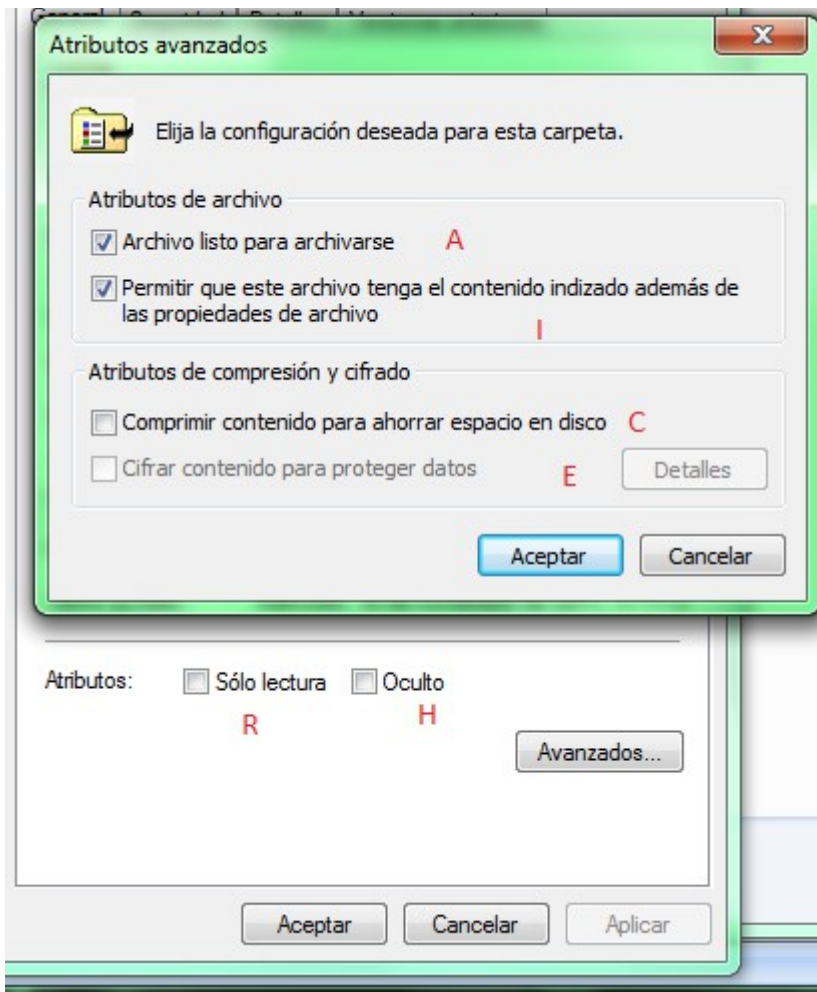
- No se pueden cifrar las carpetas ni los archivos que estén comprimidos
- Los archivos y carpetas cifrados no se pueden comprimir
- Los archivos y carpetas cifrados pueden perder este cifrado dependiendo de cómo se muevan o se copien.

Observaciones generales

- Desde el entorno gráfico no puede gestionarse el atributo **S**. Únicamente la podremos gestionar con la orden ATTRIB desde modo texto.
- Consultar atributos desde el entorno gráfico

Botón derecho propiedades. Los atributos de FAT se mantienen

Si desde una partición NTFS muevo un atributo a otra se mantienen los atributos.



- Modo texto

ATTRIB: Desde el principio de todo. Únicamente, a partir de vista, puede manejar **R, H, S, A, I**.

Compact: Gestiona **C**

Cipher: Gestiona **E**

- Atributo R:

Este atributo suele ser ignorado por Windows cuando esta aplicado a un carpeta. Le dice al SO que mire si hay un desktop.ini definido para esa carpeta.

- Estos atributos de seguridad son para todos los usuarios. Existe la posibilidad de utilizar permisos que sirve para dar diferentes atributos de seguridad a distintos usuarios. Los atributos tienen prioridad sobre los permisos.

- Se utiliza un lenguaje de colores para algunos atributos.
 - Gris ◇H
 - Azul ◇C
 - Verde ◇E

4.2 Copias de seguridad

4.2.1 Tipos de copias de seguridad

1. **Normal/completa:** Se copian todos los archivos seleccionados independientemente de cómo se encuentra el atributo A. Una vez copiado el archivo, el atributo A se desactiva. Si posteriormente este archivo fuera modificado, el SO volvería a activar el atributo.
2. **Copia:** Hace copia de todos los archivos seleccionados, independientemente de cómo este el atributo A. El atributo A no se desactiva.
3. **Diferencial:** La diferencial sirve para copiar los archivos modificados desde la última copia, es decir, copiará únicamente los que tengan el atributo A activado y no desactiva el atributo A.
4. **Incremental:** Copia únicamente aquellos archivos que tengan el A activado y desactiva el atributo A.
5. **Diaria:** No utiliza el atributo A para nada. Copia únicamente los archivos basándose en la fecha de la última modificación del archivo. De manera que si un archivo ha sido modificado en el mismo día, en el que se realiza la copia de seguridad, se incluirá en la copia.

Ejemplo de copia semanal cíclica: (dibujo)

Día	Copia 1	Copia 2
Domingo	Completa	Completa
Lunes	Diferencial	Incr
Martes	Diferencial	Incr
Miércoles	Diferencial	Incr
Jueves	Diferencial	Incr
Viernes	Diferencial	Incr
Sábado	Diferencial	Incr

4.2.2 Herramientas.

Podemos optar por usar herramientas suministradas por el propio SO para realizar estas copias de seguridad.

-W.XP/2003 server

- Entorno grafico: %windir%\System32\NTBACKUP.exe inicio, todos los programas, accesorios, herr. Del sistema, copia de seguridad
- Entorno texto: NTBACKUP

-W.7:

- Entorno grafico: %windir%\System32\SDCLT.EXE inicio, todos los programas, mantenimiento, copias de seguridad y restauración
- Entorno texto: %windir%\System32\WBADMIN.EXE (en w7 y server 2008)

4.3Recuperación de Archivos Borrados

La papelera es un concepto introducido por Apple en el año 82 y aparece por primera vez en entorno Microsoft en MSDOS-6 . En este sistema operativo teníamos una orden, **UNDELETE**. Era una orden para recuperar archivos borrados. Tenía tres modos de funcionamiento y consistía en mover a un directorio oculto llamado **Sentry** los archivos que el usuario borraba. Así funciona la papelera de reciclaje, actualmente en entornos Microsoft. Este modo no era el modo por defecto.

(Dibujo)

SO	Localización archivos borrados (oculta)
W 9X/ME	X:\Recycled
W NT/2000/XP	X:\Recycler\<suid>\
W vista/7	X:\\$Recycler.BIN\<suid>\

Observaciones a la papelera:

1. No todos los archivos borrados van a la papelera. Depende de cómo se ha eliminado y del soporte físico en el que estaba archivado. Los archivos borrados solo van a papelera desde entorno gráfico, y sólo tienen papelera también los discos duros, todos los archivos borrados desde medios extraíbles no van a papeleras, no son recuperables.
2. Cada unidad lógica ubicada en un disco duro, tiene su papelera de reciclaje. Desde modo gráfico veré una única papelera, pero realmente hay una para cada unidad lógica.(Dibujo)

3. Si dentro de la máquina tengo dos sistemas operativos para cada unidad lógica habrá dos carpetas una para cada sistema operativo, pero solo se verá, en el modo gráfico, aquella del SO que estés utilizando en ese momento.
4. Cuando tengo problemas de espacio y elimino a la papelera los archivos no libero espacio. El espacio destinado a la papelera es configurable. El porcentaje configurable se puede acceder desde las propiedades de la papelera.
5. Podéis borrar un archivo sin que vaya a la papelera pulsando mayus+supr

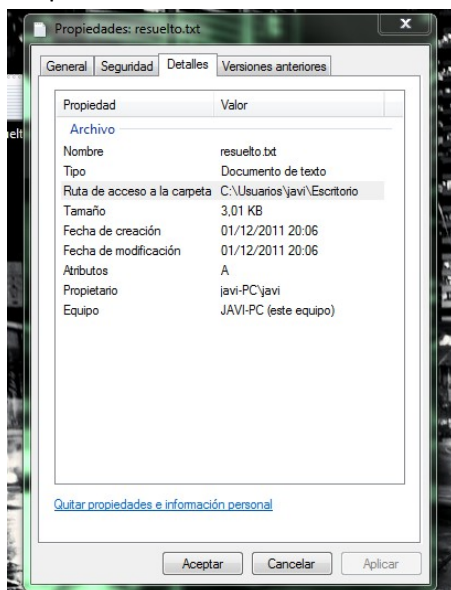
4.4 Metadatos

Estas características se almacenan en el propio sistema de archivos, pero además puede suceder que determinados formatos de archivos puedan almacenar Metadatos personalizados. Estas propiedades pueden añadirse por hardware, por software o directamente por el usuario.

Definición: Son datos que describen datos (fotos, audio, documentos)

Cuando se comparte información con otros usuarios, estos metadatos pueden revelar información que no se desea mostrar. A partir de W.7 existe una posibilidad de limpiar un archivo de metadatos no deseados.

Propiedades de archivo- Detalles-Quitar propiedades e información personal:



5. El registro de Windows

El registro de Windows es una Base de Datos jerárquica utilizada en los SO Windows para almacenar:

1. información de configuración del sistema, para uno o más usuarios.
2. información de configuración de las aplicaciones instaladas.
3. información de configuración de los dispositivos de Hardware.

Una base de datos es utilizada por el SO, al inicio de este mismo, y por todas las aplicaciones cuando necesitan información sobre el sistema informático(Hardware,Software y users) o para guardar sus propios valores.

5.1 Orígenes

Nos remontamos a los primeros Windows. Empezaron a salir aplicaciones que funcionaban sobre MS-DOS y sobre Windows(no era un SO) toda esas aplicaciones y el entorno grafico almacenaban una serie de variables de inicialización y configuración en una serie de archivos .ini estos archivos

1. Son archivos ASCII por lo tanto se puede editar con un bloc de notas...
2. Están organizados en secciones: sección= [drivers (cabecera de sección)]
wave=MMDRV.DLL, TIMER=TIMER.DRV. sección=([386 ENH(cabecera de sección)]WOAFONT=APP80850.FON

Esta organización en ficheros .ini funcionó bien en las primeras versiones de Windows pero en Windows 3.1 y 3.11 empiezan a aparecer problemas:

1. Al ser ficheros de texto su tamaño queda limitado a 64KB. Por lo tanto cada programa que instalábamos añadía un conjunto de líneas al fichero .ini y este tenía una limitación de tamaño por lo tanto me limitaba la instalación de programas. Muchos programas tenían problemas para leer entradas con más de 32K entonces era recomendable que los archivos .ini fueran de tamaño <= a 32K.
2. La información no sigue una estructura jerárquica por lo que la localización de información dentro de un archivo .ini es un proceso lento. Para evitar que esto afectara al rendimiento de sus programas, los desarrolladores de software empezaron a crear un archivo .ini para cada programa. En ese momento empiezan a aparecer publicaciones informáticas con diskets con juegos, programas etc, también los programadores cuidaban muy poco la instalación y desinstalación de los programas, porque cuando se desinstalaban quedaban restos del programa. Todo

esto desemboca en que los desarrolladores de Microsoft se plantea desarrollar una estructura diferente que fuera alternativa a estos archivos .ini.

- Lo primero que aparece son los archivos .REG en ese momento solo se utilizaban para almacenar información OLE(ObjectLinking) Permite incluir objetos de información que han sido creados por programas distintos.

Windows 95 es la primera vez que se presenta como SO y supone la llegada de tecnologías nuevas como son: el plug and play, configuraciones de múltiples usuarios etc. Entonces se diseña una base de datos dinámica y jerarquizada que cumplía con los siguientes requisitos:

1. Toda la información de configuración ha de poder almacenarse en una fuente lógica
2. La base de datos es recuperable después de una caída del sistema
3. Desaparece la limitación de los 64K
4. Los administradores pueden gestionar la información de esta base de datos mediante herramientas del panel de control u otras utilidades
5. Los desarrolladores de Software pueden acceder a las entradas de esta base de datos, de esta manera tanto los procesos de instalación, desinstalación como los programas durante la ejecución normal de estos, podrán acceder para modificar o consultar esa base de datos.
6. El registro puede examinarse tanto localmente, como por vía remota.

Entonces desde este momento hasta el actual W.7 sigue manteniéndose esta estructura centralizada que es el registro de Windows. Aunque algunas entradas y ficheros que lo sustentan han cambiado de un SO a otro.

Finalmente por razones de compatibilidad siguen existiendo, dentro del directorio Windows, System.ini y Win.ini por si hay aplicaciones tan antiguas que siguen utilizando este tipo de archivos.

5.2 Organización del registro (Ficheros y Claves)

W95-98: En ese momento el registro lo forman dos archivos: u:\windows\System.dat y v:\Windows\Users.dat esto en el caso de que solo lo utilice un usuario. Si son varios el u:\windows\System.dat y u:\windows\profiles\userx\user.dat En este caso hay un subdirectorío profiles que contiene los datos de cada usuario y dentro contendrá el User.dat

W.Millennium: Lo anterior se mantiene. Aparece un tercer archivo u:\windows\classes.dat si hay más de un usuario u:\windows\classes.dat + los dos anteriores

W.XP: (DIBUJO)

W.Vista y W.7: Son los mismos pero han cambiado los directorios. La parte de Windows sigue igual pero Documents and Settings ahora es Users(dibujo)

(Dibujo) Las claves del primer nivel se llaman claves principales y aunque al usuario se le muestran en principio 5 claves principales, para algunos autores solo hay 2.

Las 5 claves principales son las siguientes:

- HKEY_USERS: Tiene una subclave para cada usuario del equipo, en la que se guarda información específica de ese usuario. P.e información de identificación, configuración de escritorio.
- HKEY_CURRENT_USER: Contiene la información del usuario que en ese momento tiene la sesión iniciada en el equipo.
- HKEY_LOCAL_MACHINE: Contiene toda la información relativa al hardware del ordenador y a las aplicaciones instaladas.
- HKEY_CURRENT_CONFIG: Almacena la configuración actual del perfil de Hardware con la que se ha iniciado el sistema.
- HKEY_CLASSES_ROOT: Almacena información sobre la asociación, programas, extensiones e iconos, junto a información de vista rápida.

De estas 5 para algunos autores se consideran principales la 1 y 3 porque la clave de Current_User se considera una subclave que se puede acceder desde la Users. Igual que config y root se puede acceder a ellas bajando por el árbol de local_machine.

Estas 5 claves han estado en todas las versiones del registro y hay una clave adicional que únicamente estaba presente en W.98: HKEY_DYN_DATA, es una clave que almacenaba información dinámica de estado de dispositivos, es una información que era utilizada por programas de vigilancia para detectar problemas de hardware y estados de dispositivos.

5.3 Edición del Registro

Se puede acceder y modificar la base de datos del registro de distintas maneras:

1. De forma implícita y transparente al usuario: P.e: desde el panel de control, desde el explorador de Window, menú ver opciones de carpeta(u otras opciones de configuración), cuando cambiamos configuración de archivos, desde el office cuando

definiis las propiedades del programa, la configuración del escritorio. Modificaciones que el usuario no tiene porqué saber que por detrás significa cambios en el registro.

2. Si somos programadores podemos incluir dentro de nuestro código llamadas a funciones que consultan o modifican valores en el registro. P.e: programas de instalación y desinstalación de aplicaciones.
3. Utilizando una herramienta incluida en Windows REGEDIT o herramienta de edición de Registro.
4. Utilizando herramientas de terceros para la edición del registro. Si se opta por esta opción hay que estar seguros de que la herramienta que utilizamos es segura.

5.4 Copia de seguridad y recuperación del registro

Un registro estropeado impide que el SO se inicie de forma correcta por lo tanto hay que hacer copia de seguridad de los registro

En lo que respecta a las posibilidades ofrecidas del SO para la realización de estas copias han ido cambiando continuamente. Actualmente en W.XP dentro de la utilidad de copias de seguridad tenéis unas casillas de verificación que permite incluir los archivos del registro en la copia de seguridad

En W.7 esta opción ya no está. En w.7 la única posibilidad por el SO. Es desde consola de comandos mediante la utilidad *REG* en esta utilidad REG no podemos hacer la copia de seguridad completa del registro sino de partes de este registro (copia parcial).

Importar/exportar dentro de REGEDIT

Exportar te permite seleccionar un punto del árbol de registro y hacer una copia de ese árbol en un fichero .reg

Importar recuperar ese fichero .reg

No es una buena técnica porque cuando nosotros importamos un archivo reg las claves que están en el fichero reg que no estaban en el registro se añadirán, las claves que están en el fichero reg y en el registro las que llegan pisarán a las que están, pero aquellas que no están en el fichero reg y si en el registro se mantendrán, por lo tanto si alguna de ellas es corrupta se mantendrá.

6. Secuencias de Arranque Duales y Múltiples

Como se explicó anteriormente, la secuencia de arranque consiste en la ejecución de varios programas.(Gráfica)

6.1 Secuencia de Arranque Dual(w.XP + w.XP)

El proceso de instalación de W.XP reescribirá el IPL en el MBR, detecta cual es la partición activa y escribe en ella un PBR de W.XP, este dice que lo que habrá que hacer es buscar un ntldr que cargue el S.O. Entonces almacena en la raíz de la partición activa un ntldr(Sustituye el que había) y también crea un boot.ini que contemplara la existencia de dos sistemas operativos(el que teníamos y el nuevo).

En la nueva partición donde vamos a instalar el nuevo W.XP, creara el PBR de XP y ya la estructura de directorios donde estará el nuevo S.O que estamos instalando. Entonces ahora cuando se inicie la máquina y el proceso llegue al boot.ini nos saldrá un menú por pantalla en el que nos dará a elegir entre el w.XP anterior o el nuevo. Eso si la activa es la primera, en el caso de que la segunda partición sea la activa, buscaría el ntldr y como no está no funcionaría(mirar gráfica anterior).

Boot manager y bootloader

boot manager o gestor de arranque: nos permite elegir un sistema operativo con el que iniciar un ordenador

bootloader:es el programa que inicia la carga del S.O elegido

En el caso de W.XP las dos funciones están en el ntldr mientras que en W.7 esas dos funciones están separadas. La de boot manager la tiene bootmgr y la de bootloader la tiene winload.exe

System partition y Boot Partition

System Partition: Partición en la que reside el gestor de arranque.

Boot partition: Partición en la que reside el S.O que hemos iniciado.

En nuestro esquema W.XP System partition la primera. Boot partition depende del sistema operativo que elijamos

En W.7 System partition es la partición reservada para sistema (100MB) y la boot partition es la partición de W.7

6.2 Secuencia de arranque dual (W.XP + W7)

Iniciamos la instalación se escribe un nuevo IPL y se detecta cual es la partición activa. Entonces se sobrescribe el PBR que es de W.XP y se lo pondrá el PBR de W.7. Como es de W.7 tendrá unas líneas que dirán que tiene que buscar el bootmgr, por lo tanto se crearan el bootmgr y el BCD de esta manera podremos elegir con que S.O iniciamos. Y tendremos otra partición con un PBR de W.7 con la estructura de directorios de W.7 y el winload.exe. Con el BCD detectará que hay otro Sistema operativo que no es W.7, se nos ofrecerá un menú para elegir que S.O iniciar. Si es W.XP el BCD dará permiso al ntldr para iniciar el W.XP.

6.3 Secuencia de arranque dual (W.7+W.XP)

Microsoft recomienda que lo ideal es instalar el sistema más viejo y luego el más nuevo.

Pero vamos a instalar un W.XP en una máquina que ya tiene W.7 (gráfica W.7). Metemos el W.XP sustituye el W.XP. En la partición donde el PBR era de W.7 ahora será un PBR de W.XP. Entonces crea el ntldr y boot.ini para iniciar el S.O y en una nueva partición crea un PBR de XP y crea una estructura de directorios de XP. Cuando iniciamos en W.XP el ntldr no detectará que existe otro S.O y siempre iniciará el W.XP. Para solucionar el problema metemos el cd de W.7 y recuperaremos el PBR de W.7. Entonces cuando inicies la máquina se ejecutara el PBR de W.7 que irá a bootmgr que no detectará el W.XP pero W.7 sabe que existe el W.XP. Por lo tanto para poder ejecutar los dos S.O deberemos editar el BCD para que nos muestre el menú de elección. Deberemos utilizar la consola de recuperación.

6.4 Secuencia Multiarranque

W.XP+W.XP+W7

Queremos instalar un W.7 en una máquina que ya existen dos W.XP. Se reescribe el IPL y en la partición activa se sustituye el PBR de W.XP por el de W.7, crea bootmgr y BCD entonces BCD detectará que hay W.XP instalado. Detecta que hay S.O anteriores pero solo mostrará dos líneas el de W.7 y otra que contiene los S.O anteriores. Al elegir la opción de los S.O anteriores y luego nos saldrá un segundo menú que nos dará a elegir entre uno de los dos XP. Y esto es así porque las líneas del BCD únicamente son redireccionamientos al cargador de XP(ntldr) que luego nos enseñará el menú de boot.ini

(Esquema ideal)Caso de que todas los sistemas tengas la estructura completa:

-MBR bootmgr: Esta opción consiste en reemplazar el IPL de Microsoft del MBR por otro que me permite elegir al principio del arranque cual es la partición que quiero activa.

Inconvenientes:

- Cada vez que instalo un nuevo S.O reescribe el IPL

- Hay algunas características de W.7 que requieren que esté instalado el IPL de Microsoft

-BootmanagerPartition: Se trata de tener una partición de gestión de arranque que tenga un boot manager de otro fabricante y que esta sea siempre la partición activa por lo tanto el IPL de Microsoft apuntará a esta partición donde se encontrará el IPL del otro fabricante que nos llevara a iniciar el S.O deseado.

Cuentas de usuario y cuentas de grupos

Las cuentas de usuario: Representan a cada una de las personas que utilizan un equipo y se utilizan para lo siguiente

1. Autenticar la identidad
2. Autorizar/denegar acceso a Recursos
3. Realizar Auditorias de eventos del sistema(inicio de sesión o utilización de archivos u otros recursos)

Las cuentas de grupo: se utilizan para para conceder permisos a usuarios de características similares y simplificar la administración de cuentas:

Una cuenta de usuario puede pertenecer a un grupo, a más de uno o ninguno. Ejemplo:

Tenemos a un departamento que tienen acceso a archivos de contabilidad, tenemos la posibilidad de que todos puedan acceder a esos archivos desde una única cuenta¿?.

Tipo de cuenta de usuario

Representa una forma sencilla de describir la pertenencia a un grupo de seguridad, así Windows clasifica estas cuentas de usuario en uno de estos tres tipos:

- Administrador
- Usuario estándar
- Invitado

Administrador: Se incluye la primera cuenta que se crea al configurar el equipo y también una cuenta llamada administrador, que en principio se encuentra deshabilitada y oculta de manera predeterminada.

- Usuario Administrador: Esta cuenta está oculta porque este usuario tiene una capacidad exclusiva y es que no está sujeto al UAC(UserAccount Control), en principio la 1ª cuenta se pide permiso para hacer esas modificaciones. El administrador no está sometido al UAC para evitar estas peticiones. Todas las demás cuentas administrativas se ejecutan en principio con privilegios de cuenta estándar a menos de que el usuario de su consentimiento para su elevación. La cuenta administrador se ejecuta siempre con todos los privilegios administrativos y nunca necesita consentimiento para la elevación.

Usuario estándar(cuenta limitada en XP): muchos de los privilegios que en XP no estaban añadidos ahora en W7 sí que se permite, como por ejemplo (cambiar IP, actualizaciones de sistema(en xp no podían realizarse si eras de esta cuenta...))

Invitados: los privilegios de estas cuentas son los mismos que las estándar pero con algunas limitaciones: No se puede crear una contraseña para estas cuentas

En grandes redes basada en dominios en Directorios Activos los grupos son una valiosísima herramienta administrativa ya que simplifican la tarea de garantizar que todos los miembros con necesidades de acceso comunes tengan privilegios idénticos, sin embargo no es aconsejable crear o usar grupos distintos de los predefinidos (administrados, usuario estándar, invitado) en equipos independientes o basados en grupos de trabajo.

Para trabajar con cuentas de trabajo en W7 se ofrecen las siguientes posibilidades:

1. Panel de control-cuentas de usuario
2. Opciones avanzadas de cuentas de usuario(mediante NETPLWIZ.exe)
3. Menú contextual equipo-Administrar-usuarios y grupos locales.
4. Línea de comandos-(mediante NET USER o NET LOCALGROUP)

Propiedad (propietario) de un objeto

Todos los recursos del sistema (archivos, impresoras...) cuentan con un propietario que decide quién accede o no a dicho recurso, normalmente el creador del recurso es el propietario, no obstante los administradores pueden tomar posesión de un recurso que ellos no han creado.

Control de acceso de Windows

Para determinar que usuarios tienen acceso a un recurso determinado, Windows asigna un SID(identificador de seguridad) a cada cuenta de usuario, de manera que cuando un usuario inicia sesión el sistema operativo:

1. Valida nombre usuario y contraseña.
2. Crea un Token de seguridad. Esto incluye el nombre de usuario, SID e información sobre los grupos a los cuales pertenece este usuario.

Cuando un usuario inicia un programa este programa recibe una copia del token de manera que cuando se intenta acceder a un recurso, controlado por el sistema operativo, bien directamente o bien a través de este programa que hemos iniciado, el sistema operativo examina el token y decide si permite o no el acceso. Para tomar esta decisión(de si tiene acceso o no) el S.O consulta la ACL o lista de control de acceso del recurso, esta lista tiene una entrada para cada usuario o grupo que tiene acceso al recurso. Estas entrada se llaman ACE(Access Control Entry) en cada una de estas entrada figura lo que puede hacer el grupo o usuario correspondiente con este recurso. El S.O compara el token de seguridad con las entradas que el corresponde dentro de esta ACE y a partir de ahí decide si se le concede o no acceso al recurso que se le ha solicitado. Estas ACE pueden permitir o denegar distintos comportamientos según el tipo de objeto. Así por ejemplo en el caso de un archivo pueden ser leer, escribir o ejecutar, mientras que en una impresora podría ser imprimir, administrar impresora, administrar documentos etc. En el caso de archivos y directorios solo funcionará esto cuando sean NTFS.

Privilegios: Permisos y derechos

Un permiso es la capacidad de acceder a un objeto de una forma definida por ejemplo, escribir en un archivo NTFS o modificar una cola de impresión los permisos los asigna el propietario de un recurso o un administrador a través del cuadro de diálogo propiedades del recurso, se asignan a objetos(todo esto de los permisos).

Los derechos es la capacidad de ejecutar una serie de acciones en el sistema por ejemplo iniciar sesión configurar el reloj instalar un DRIVER. Los derechos los configuran los administradores a través de la consola de *directiva de seguridad local* esta consola no esta disponible en todas las versiones de un mismo S.O en estos casos los derechos de algunos grupos están predefinidos y no se pueden modificar. Los derechos se asignan a cuentas de usuario o grupos.

En muchos manuales y cuadros de diálogos de Windows se utiliza el término privilegio para referirse tanto a los permisos como a los derechos. Los derechos y permisos de los miembros de un grupo son acumulables, es decir, si una cuenta de usuario pertenece a más de un grupo, el usuario disfruta de todos los privilegios de los grupos a los que pertenece.

Herencia de permisos

Esta es otra característica que te permite administrar permisos de manera sencilla, esta característica hace que los objetos de un contenedor hereden automáticamente todos los permisos heredables de ese contenedor por ejemplo cuando se crean archivos en una carpeta los archivos heredaran los permisos de la carpeta y solo se heredan aquellos que han sido marcados como heredables. Podemos distinguir dos tipos de permisos:

- **Explícitos:** Son aquellos que se establecen de forma predeterminada en objetos que no son secundarios cuando se crea el objeto o los que crea el usuario en objetos secundarios, primarios o que no son secundarios.
- **Heredados:** Son los que se propagan a un objeto desde un objeto primario

Los permisos explícitos tienen prioridad sobre los heredados, incluidos los permisos denegar heredado. Si en la carpeta D1 tengo activado el permiso denegar por lo tanto el archivo tendrá por herencia el permiso denegar pero si yo explícitamente le cambio el permiso al archivo como es explícito ganará al heredado.

Estructura de red de trabajo en grupo

Grupo de trabajo o workgroup

Es un concepto Windows, no es más que un grupo de ordenadores en red que comparten recursos (ficheros e impresoras), en este modelo no existe un servidor central, ni ordenadores clientes, sino que son redes de igual a igual (peer to peer). Donde cualquier ordenador puede jugar ambos roles. Cualquier ordenador puede dar a compartir un recurso propio y cualquiera de los ordenadores haría de cliente al ordenador. Este concepto se introduce a partir de Windows 3.11 (nombre del S.O: Windows 3.11/trabajo en grupo). Este modelo de redes hay una diferencia notable entre antes de XP y después. Antes de XP la autenticación se producía a nivel de recursos, el recurso compartido (carpeta, impresora...) podía ser protegida por contraseña, entonces para acceder al recurso compartido, bastaba con estar en la red, conocer la ubicación en la cual está el recurso y su contraseña. A partir de Windows XP se introduce el concepto de usuario, entonces cada equipo conserva una lista de usuarios autorizados y de recursos disponibles, de esta manera para poder acceder a un recurso compartido el usuario debe de estar dado de alta en esta lista, que mantiene el equipo que comparte el recurso.

Este modelo de redes tienen un inconveniente, el número de accesos a un recurso compartido es limitado en W.XP un máximo de 10 accesos simultáneos al **recurso** compartido. Si hay diez que comparten el recurso y otro quiere acceder no podrá hasta que alguno de los anteriores suelte el recurso compartido.

En W7 se ha visto incrementado a 20 accesos.

Este modelo de trabajo contrasta con el modelo de cliente-servidor, esta máquina que hace de servidor tiene un SO especial y el resto de máquinas pueden tener cualquier sistema operativo.

En un modelo cliente-servidor solo hay una base de datos que es la que contiene el servidor en cambio en el workgroup hay una base de datos para cada ordenador ya que cada una tiene que tener dado de alta al usuario para que se pueda acceder (¿?).

Observaciones a este modelo

1. En W.XP, en opciones de carpeta-ver-utilizar uso simple de archivos
2. En W7, en opciones de carpeta-ver-utilizar el asistente para compartir

Importante conocerlas porque permiten trabajar y compartir recursos de una manera simple sin que el usuario tenga que saber permisos etc... Pero limita bastante las posibilidades de configuración. En propiedades de una carpeta o impresora, pueden aparecer otras fichas de seguridad o compartir se ve afectado según las opciones anteriores. Estas opciones condicionan a la hora de compartir recursos, por lo tanto recomienda tenerlas aplicadas.

3. Diferencia Permisos de recursos compartidos/permisos NTFS. La manera de gestionarlos es similar, por lo que lleva a confusiones, pero son dos niveles de

seguridad distintos. Únicamente aquellas conexiones de red que atraviesen con éxito estas dos barreras podrán acceder al recurso en cuestión.

- a. Permisos de recursos compartidos: controlan el acceso de red sobre un recurso y no afectan a aquellos usuarios que inician sesión de forma local. Se configuran en propiedades/compartir.
- b. Permisos NTFS: se aplican a las carpetas y archivos de una unidad formateada NTFS para cada usuario al que se concede acceso (local o red), afectan tanto en local, como en red. Se configuran en propiedades/seguridad.

Estos dos tipos se combinan de la forma más estricta posible, entonces primero se pasa el control de recursos compartidos y si se consigue pasarlo a continuación hay que enfrentarse con el de permisos NTFS, este segundo control podrá eliminar permisos del nivel anterior pero nunca podrá añadirle permisos.

Por ejemplo: Si el primero dice lectura y el segundo control total solo se podría leer ya que no puede añadirle controles. Si fuera al revés eliminaría permisos. Si accedes de manera local solo se tendrán en cuenta los NTFS si accedes en red se aplicarán los primeros y luego los segundos eliminando si hace falta los primeros pero nunca añadiendo.

Estos permisos hay que tener en cuenta el concepto de grupo. Si el usuario que estamos validando pertenece algún grupo de usuarios hay que tenerlo en cuenta

Linux

Para conectarnos a la máquina debemos hacerlo a través de la consola utilizando el comando

TELNET 192.168.1.152

Login: user17

Passwd:17

El prompt será: \$ no sabemos dónde estamos.

Para salir: \$logout, \$ exit, Ctrl-D

Características:

1. Entorno basado en prompt
2. Lista llamada histórico de las ordenes previamente introducidas este histórico lo podemos recorrer con las teclas de movimiento. Tiene memoria...
3. Podéis interrumpir la ejecución de la orden mediante Ctrl-C
4. En una misma línea podéis introducir más de una orden separándolas por “;”
“orden;orden;”
5. Podéis dividir una orden en varias líneas, mediante

\$inicio-orden\ \$sigue-orden\ Final-orden

Estructura de las órdenes de Linux

Una orden tendrá tres partes

\$ orden opciones argumentos

Opciones: son códigos normalmente formados por una única letra precedida de un guión que modifican el tipo de acción que realiza la orden.

Argumentos: Objetos sobre los que actúa la orden.

1. No todas las ordenes tiene porque tener diferenciada estas partes, podemos utilizar con opciones y sin argumentos, etc.

Ejemplo: \$cp -i arch1 d1/arch2 ◊copia(cp) si el destino existe pedirá confirmación de si sobrescribe(-i) destino (argumentos)

2. En Unix es obligatorio seguir este orden: orden opciones argumentos. En Linux se puede mezclar todo p.e: \$ cp arch1 -i d1/arch2. Hay que ajustarse a la sintaxis de Unix.

3. Cuando se indican dentro de una orden más de una opción podemos reunir todas las opciones en un mismo guión p.e: `ls -l -a` – equivaldría al `DIR` en Windows `-l` detallado `-a` muestra ocultos. Esto es equivalente a: `ls-la`

Órdenes:

- `Passwd`- Para el administrador del sistema tiene una utilidad muy alta pero para un usuario limitado está bastante “capada” solo podremos cambiar la password.
- `Whoami`- Esta orden muestra el nombre del usuario asociado con la cuenta actual.
- `Who`- Consultar que usuarios están conectados al sistema. También podemos saber que usuarios están dispuestos a recibir mensajes.
- `Uname`-se utiliza para obtener información sobre nuestro sistema Linux
- `Date`- permite consultar fecha y hora del sistema
- `Cal`-permite visualizar el calendario, correspondiente a un determinado mes de un determinado año.
- `Clear`-limpiar pantalla
- `Write`-permite enviar mensajes a otro usuarios en tiempo real, comunicación unidireccional. Para que sea bidireccional el otro usuario debe escribir `write`. Pero se verá mezclado por lo tanto cambio y corto. La comunicación se interrumpe con `ctrl-d`
- `Talk`-Bidireccional. El otro tiene que dar el OK para poder hablar entre los dos. En este momento la pantalla se divide en dos para diferenciar las conversaciones.`Ctrl-c`
- `Mesg`-
 - Consultar nuestro estado si estamos permitiendo la recepción.
 - Activar la recepción de mensajes.
 - Desactivar la recepción.
- Para tener ayuda Linux cuenta con una serie de utilidades
 - `Man`+orden de la que queremos pedir ayuda: utilidad (es un programa que tiene su propia interface, el cual tiene sus propios comandos)
 - Salir de la utilidad: “q”
 - Buscar una determinada cadena dentro de las páginas de ayuda
 - “/cadena” buscará hacia delante

- “?”cadena”búsqueda hacia atrás.
- N anterior búsqueda
- F siguiente página
- b

(D)Sistemas de Archivos de Linux

Está basado en una estructura jerárquica de árbol, en la que a diferencia de DOS que tenemos una estructura de árbol independiente para cada unidad, aquí tendremos un único árbol desde el cual podremos tener acceso a cualquier subárbol procedente de cualquier unidad de almacenamiento. (dibujo Linux)

Directorio home: para cada usuario que se haya creado una cuenta, se le genera un directorio con el nombre del usuario. En este directorio se crea una serie de directorios y archivos para que este usuario pueda trabajar sobre él. A cada uno de estos directorio se le llama por home y el nombre de usuario /homeUser01

En DOS teníamos \ para indicar raíz y secciones. En el caso de Linux se utiliza “/”significa raíz y separador para las rutas. Ruta absoluta: /home/user01. La ruta relativa a user02 con el directorio activo user01 seria: ../user02.

“.” Y “..” igual que en DOS.

Existe un 4º carácter especial que es “~” que depende el usuario que lo está utilizando. Representa la ruta absoluta del directorio home del usuario que hace uso del carácter.

Si el directorio activo es User01 y el usuario que ha iniciado es User02

../user02/texto

/home/user02/texto

~/texto

Es lo mismo

Observaciones del nombre de archivos:

1. En Unix max 14 posiciones
2. En Linux max 256 posiciones
3. En las primeras versiones de Linux este nombre no podía empezar por dígito
4. Aquellos que empiecen por un punto “.” serán archivos ocultos.
5. El punto “.” en medio de cualquier nombre se comporta como cualquier otro carácter. Una consecuencia de esto es que en Linux no hay extensiones. Esto no quiere decir que los usuarios no hayan tomado una serie de acuerdos para simular extensiones,

perfectamente se pueden encontrar nombre de archivos que acaben en .c(archivo que contiene lenguaje C), pero no son extensiones es como un nombre más.

6. Mayúsculas y minúsculas son distintas

Ordenes:

1. Orden **pwd**: muestra el directorio activo actual, en ese momento.
2. Orden **mkdir**: crea directorios
3. Orden **cd**: cambio de directorio activo
 - a. `cd ..` *ojo al espacio!!!sino no funciona en el caso de los puntos
 - b. `cd *` a pelo, no muestra el directorio, en este caso te lleva al directorio home del usuario.
4. Orden **rmdir**: borrar directorios.
5. Orden **rm**: borrar archivos.
6. Orden **ls**: mostrar contenido de un directorio.
7. Orden **cat**: Concatenación de archivos. Otras funcionalidades:
 - a. Mostrar por pantalla el contenido de uno o varios archivos. Ejemplo
 - a.i. `$Cat fich1 ../fich2` mostraria por pantalla
 - b. Redireccionamiento
 - b.i. `$Cat fich1 ../fich2>fich3` crearia un archivo fich3 con el contenido de fich1 y 2
 - c. Entrada estándar
 - c.i. `$cat>fich1` metará el contenido que le venga a continuación en fich1

-

-

-

Ctrl D

\$
8. Orden **script**: permite almacenar en un archivo todo lo que el usuario va tecleando y el sistema responde. Podemos guardar en un archivo el trabajo que hemos realizado.

9. Orden **head**: Permite visualizar las n primeras líneas de uno o varios ficheros.
10. Orden **Tail**: Permite visualizar las n últimas líneas de uno o varios ficheros.

Sistema de Archivos (II)

1. Orden **more**. Nos permite ver el contenido de uno o varios archivos por páginas. Es una utilidad (tiene su propio interface y su conjunto de órdenes, mismas ordenes que man), se puede utilizar directamente \$more seguido de una serie de ficheros o mediante tuberías \$cat ---|more
2. Orden **file**. Esta orden deduce el tipo de fichero a partir de un análisis que realiza sobre las primeras líneas del fichero. En Linux los directorios también son ficheros...
3. Orden **od**. Esta orden permite examinar el contenido de un archivo byte a byte. P.e podremos saber si son espacios en blanco o tabuladores a simple vista no se ve la diferencia.
4. Orden **cp**. Equivalente a la orden copy de DOS. Permite copiar archivos de un origen a un destino. Observaciones:
 - a. cp origen origen... destino. Posibilidad de más de un origen y pasarlos a un destino.
 - b. -b (backup) cuando en el destino existe un archivo con el nombre del origen. El archivo se renombra con el mismo nombre seguido del carácter ~ y luego se hace la copia del archivo P.e: cp ~/d0/arch1 ~/arch1 ◇arch1 sería arch1~
 - c. -S nos permite elegir un sufijo distinto al anterior.
 - d. -f es la opción por defecto borra los destino sin avisar, cuando hay coincidencia
 - e. -i pide confirmación antes de sobrescribir
 - f. -u solo copia cuando el que llega es más nuevo que el que ya está, en el caso de que haya coincidencia.

Sin coincidencia:

 - g. -R copia recursiva directorios
5. Orden **mv**. para mover archivos y directorios similar a cp las opciones que lleva tienen el mismo significado que en la cp pero hay menos.
6. Orden **ln**. Permite hacer referencia a un archivo usando diferentes nombres para acceder a él desde distintos directorios. Estos nombres adicionales reciben el nombre de vínculos o links. Concepto equivalente a accesos directos.
 - a. Sintaxi: ln nombre_original nombre_vinculo

b. Observaciones:

- b.i. Se puede crear más de un vínculo sobre el mismo archivo. (dibujoln)
\$ln ~/d0/arch1 ~/arch2
\$ln ~/d0/arch1 ~/d1/arch3
- b.ii. Los vínculos no pesan.
- b.iii. El contenido del archivo no se borra hasta que son borrados todos sus vínculos si borro el archivo puedo seguir accediendo apartir del vínculo.
- b.iv. Este tipo de vínculos reciben el nombre de “hard links” o vínculos duros
- b.v. Únicamente pueden utilizarse entre elementos del mismo sistemas de archivos(dentro de la misma partición).
- b.vi. No es posibles hacer vínculos duros a directorios, solo a archivos.
- b.vii. Hay otro tipos de vínculos: symbolic links, estos se crean con esta orden pero con la opción -s, el vínculo simbólico contiene el nombre de la ruta de acceso del archivo con el que está vinculado.
 - b.vii.1. Si se elimina el original los vínculos simbólicos dejan de funcionar, pero existen.
 - b.vii.2. Se pueden crear vínculos simbólicos a directorios.
 - b.vii.3. Se pueden crear vínculos a archivos de distintos sistemas de archivos.

c. Diferenciar entre hard links y symbolic links

- c.i. ls -l saca información detallada. la primera posición de la primera columna dice el tipo de archivo (d=directorio -=archivo l=symbolic link)

el número de la segunda columna indica los vínculos. En los directorios hay dos porque se puede acceder a través del padre y directamente sobre ellos.

ls -li otra opción que saca el número de inodo(número único que identifica) esto significa que los hard links tienen el mismo número de inodo que el archivo al que se refieren...

ln arch1 arch2

ls -li

cat arch2

hola

rm arch1 borro

ls -li

cat arch2 el contenido sigue siendo accesible

hola

ln -s arch2 arch3

ls -li segunda columna habrá una l porque es un vinculosimbolico

rm arch2

cat arch3 no existe el fichero porque es un vínculo simbólico.

7. Orden **find**. Permite localizar un archivo o varios en una lista de directorios

- a. \$find lista-directorios -opcion criterio [-opcion criterio ...]
 - a.i. -name patrón p.e: \$find ~ -name arch1 (dibujo find) Buscaria aquellos archivos que tengan ese nombre
 - a.ii. -ls muestra la lista detallada de lo que ha encontrado
 - a.iii. -type tipoarchivo (d,f o l) \$find ~ -name arch1 -type l buscaria los que tengan ese nombre pero que sean links simbólicos
 - a.iv. -size + ó - n^oc buscarlos de un determinado tamaño, además puede acompañarse de un más o un menos para buscar los de mayor o menor tamaño. \$find ~ -name arch1 -type l size +10c
 - a.v. -mtime n^o archivos modificados n días atrás
 - a.vi. -newer patrón aquellos archivos con una fecha por encima del patrón es decir aquellos archivos más nuevos que patrón
 - a.vii. Estas opciones las podemos utilizar con operadores lógicos NOT=! y OR= -o además puedo utilizar paréntesis para dar prioridad a determinadas operaciones \(\). \$find ~ \(-name arch1 -o -type l \) size +10c

Editor: Es un editor que está presente en todas las versiones de Linux.

vi: Para entrar al editor

\$vi invocando al editor, empieza archivo desde 0

\$vi archivo entrar abriendo un archivo ya existente o creando uno nuevo con ese nombre,

\$vi +45 archivo Posicionaríamos el cursor en la línea 45

\$vi +\$ archivo Tendríamos como línea activa la última línea

\$vi +/cadena archivo Entraríamos y posicionaríamos el cursor en la cadena especificada del archivo

Es importante el posicionamiento porque habrá que hacerlo mediante teclado.

Una vez dentro hay tres modos de funcionamiento:

1. Modo ORDEN: En este modo todas las teclas funcionan como órdenes del editor si pulsamos una A ejecutaremos la orden A no se pintará una A. Algunos ejemplos de ordenes:
 - a. Órdenes para mover el cursor dentro del documento. Usaremos las flechas.
 - b. Órdenes para modificar el texto. P.e:
 - b.i. **x** que borra el carácter dónde tenemos el cursor.
 - b.ii. **dd** borra la línea entera donde tenemos el cursor
 - b.iii. **D** Borra desde la posición del cursor hasta el final de línea
 - c. Ordenes que nos llevan al modo Entrada:
 - c.i. **i** nos llevará al modo entrada que nos permitirá escribir texto en la posición del cursor.
 - c.ii. **a** añadir texto después del cursor.
 - c.iii. **I** podemos empezar a escribir al principio de línea.
 - c.iv. **A** podemos empezar a escribir al final de línea.
 - d. Se llega al modo ULTIMA LÍNEA con “:”
 - e. Para regresar al modo Shell (prompt de Linux) hay que escribir ZZ(= a modo última línea wq).
 - f. Carácter /cadena. Se acompaña de una cadena. Busca la aparición de la cadena en el documento
 - g. ?cadena. Lo mismo que / ¿?¿?
 - h. Orden n para buscar la siguiente aparición de esa cadena

2. Modo ENTRADA: El teclado funciona para la introducción de texto. Si pulsamos A se pintará en pantalla

a. Para regresar al modo orden pulsamos esc, una vez allí posicionaremos el cursor y volveremos al modo ENTRADA.

3. Modo ULTIMA LÍNEA: Se llega desde modo orden mediante “:”. Se llama así porque cuando pasamos a este modo el cursor se pone al final de la línea.

a. :help

b. q para salir sin guardar.

c. wq salir grabando

d. w grabar y no salir

e. etc

f. Para regresar a modo orden mediante la tecla esc.

g. Carácter !(bang).

g.i. Si nosotros queremos forzar a salir sin guardar ejecutaremos q!, otro ejemplo: :w archivo(archivo ya existe y no lo guardará) podemos forzarle a que guarde: : w! archivo

g.ii. Permitir ejecutar órdenes del Shell sin cerrar el editor. Ejemplo:
: ! ls~/d1

g.iii. Si queremos seguir ejecutando comandos del shell podemos cerrarlo temporalmente con la orden :sh y nos aparecerá el prompt del entorno de comandos \$. Mediante \$ exit regresaré al modo que tenía abierto

g.iv. Me gustaría que el resultado del ls se escribiera en el documento que estoy editando. Se puede hacer con

g.iv.1. : r incluye en la línea activa el contenido del archivo que ahí se especifica

g.iv.2. : r !ls me inserta el resultado de la ejecución de una orden.

Buffers: cantidad de memoria destinada a guardar datos temporalmente en una posición de memoria: 2 tipo:

- **Numerados:** 9 buffers que van numerados del 1-9
- **Con nombre:** El nombre es una letra que va de la a-z (abecedario inglés)

Podemos conservar 35 copias o cortes para luego recuperar los que consideremos oportunos.

Por defecto el editor va guardando en los numerados las operaciones de borrado que los usuarios van haciendo. El 1 la última operación de borrado, el dos la penúltima operación de borrado... Ejemplo de copiar-pegar

7"ayy◊nºlineas|buffer|orden. cogería 7 líneas desde donde tenemos el cursor y las copiaría en el buffer a .Para indicar el buffer hay que poner "

"ap◊pega el contenido del buffer.

Para cortar solo hace falta pegar porque cuando borramos se guarda en el buffer.

"1p◊pega el último borrado que se ha guardado en el buffer.

Configuración del editor:

Para ello en modo última línea se dispone de un conjunto de ordenes ": set" que tiene una forma de utilización muy simple

:set opción ◊Activa opción

:setnoopción◊Desactiva opción

Ejemplo:

:setshowmode◊ Configura el editor para que nos indique que estamos el modo insert(entrada)

:setnoshowmode◊Desactiva.

:setnumber◊Las líneas del contenido del archivo me aparecen numeradas

:setautoindent◊ Tabulará automáticamente

:settabstop=n ◊ caracteres nº que saltara tabulador

:setignorecase◊ desactiva case sensitive

Todo esto dura hasta que lo cerramos, si queremos que se quede estas configuraciones se ha implementado el siguiente mecanismo:

En el home de cada usuario puedo utilizar un archivo oculto que se llama **.exrc** en el cual voy poniendo todas esas líneas set que me interesan.

Podría configurar un directorio que se llamará program. Y otro que se llamará manuales y tener un archivo **.exrc** para cada directorio, cada archivo con sus propias características. De manera

que cada archivo que yo cree en uno de esos directorios tendrán las características del archivo correspondiente a ese directorio.

Todo esto funciona siempre que yo tenga en el archivo .exrc de home la orden **set exrc** sino no funciona.

En algunas distribuciones esto no funciona. En las de clase no funciona porque hay un archivo que le hace la puñeta .vimrc si finaliza en rc es un archivo de configuración. Este archivo es un archivo de mejora del vi. Se soluciona renombrando o borrándolo de manera que cuando se inicie el editor no lo encuentre.

Operaciones propias del Shell

Núcleo: Podemos definirlo como un programa modular que ejecuta programas y gestiona dispositivos hardware.

Shell o interprete de comandos: es un programa que está continuamente en ejecución desde que se inicia una sesión de usuarios hasta que finaliza y que actúa de interface entre el núcleo del sistema y el usuario.

Dos grandes grupos de Shell:

1. Bourne Shell: Tienen su origen en el primer Shell de unix, que se llamaba y que había sido diseñado por Steve Bourne,
 - a. Shell BourneAgain.(Bash)
 - b. Shell ksh
2. Cshell: Escrito por Bill Joy, intentan hacer un Shell más familiar para el usuario.
 - a. Shell Tcsh

La mayoría de distribuciones de Linux y Ubuntu establecen como modelo predeterminado "bash".

Características comunes a todos los Shell

- Metacaracteres: Carácter especial que utilizado dentro de un nombre de archivo, permite especificar más de un archivo como argumento de una orden. (comodín en DOS) Son:
 - "*", carácter comodín que vale para la posición que tengo y las que vienen.
 - "?", para la posición que tengo. Una única posición
 - "[]", Sirven para especificar todos los posibles caracteres(ASCII) válidos en una determinada posición, intervalo de caracteres.
 - \$lsdoc[12] ◊orden listado por archivos de cuatro letras que contengan doc+ 1 ó 2.

- `$lsdoc[1-3]` ♦ solicitando aquellos archivos cuyo nombre tenga 4 letras que contengan 1 ó 2 ó 3.
- `$lsdoc1[12]` ♦ los que tengan 11 ó 12.
- Para negar un intervalo se utiliza signo de admiración
`lsarch[!a-d]` ♦ aquellos que no contengan estos caracteres.

○ Estos metacaracteres se pueden utilizar de manera combinada.

- `$ls *[0-9]` ♦ todos aquellos que terminan en un dígito

○ **** Los comodines en DOS eran caracteres prohibidos en nombre de archivos, en el caso de Linux no, como diferencio un comodín de un carácter normal. Cuando quiera que estos metacaracteres se comporten como un carácter normal tengo que utilizarlos precedidos de `\`. Igual para `?` y `[]`

- `Lsdoc*` ♦ aquellos que empiezen por doc
- `Lsdoc*` ♦ únicamente listará doc*(doc asterisco)

○ `~` es traducido por la ruta absoluta.

- **Redireccionamientos**

Posibilidad de direccionar la salida y entrada estándar

- `0` ó `<` entrada
- `1` ó `>` salida
- `1` ó `>>` Adición
- `2>` error
- `2>>` error

Podemos utilizar:

`&n` permite utilizar cualquiera de los números anteriores.

`$ cat fich1 fich2 >fich3 2>&1` ♦ Concatenaria fich1, fich2 meterá el contenido en fich3 y si alguno de los archivos no existe se generaría un mensaje de error que iría a `&1` que es fich3.

- **Canalizaciones**

"|" permite encadenar una serie de órdenes de manera que las salidas de una orden se convertía en la entrada de la siguiente orden y así sucesivamente.

- **Canalizaciones+redireccionamiento**

- **Carácter** – puede ser utilizado como sinónimo de entrada estándar p.e:

\$cat fich1 – fich2 >fich3 ◇concatena fich1 entrada estándar(teclado) fich2 a fich3. Meterá el contenido de fich1 en fich3 esperara a que le introduzca el texto, ctrl+dfinalizar, lo meterá en fich3, y luego fich2 lo meterá en fich3.

\$pwd | cat – fich2 >fich3 ◇la entrada estándar del cat es la salida del pwd, la entrada estándar (salida de pwd) se concatena con el fich2 y la salida a fich3. Crearía un fich3 con la línea del directorio en el que estoy + el contenido de fich2

- **Orden tee.** Hace una copia de la salida estándar en un fichero. p.e:

\$ sort fich | tee fichord◇ordena el fich y se le pasa ordenado y guardará una copia a fichord y lo mostrará ordenado por pantalla ya que es la salida estándar.

- **Ordenes Filtro**

- **mort=DOS.**
- **more=DOS.**
- **pg en unix, no en linux (que es lo mismo que el more).**
- **grep= FIND(DOS)**
- **tr:** sirve para traducir unos caracteres en otros. Permite cambiar caracteres por otros
- **cut:** Nos permite hacer una abstracción de las columnas.
- **wc:** Contador de palabras, permite sacar unas estadísticas del número de caracteres, bits, palabras...

- **Alias**

Mediante la orden alias podemos crear nombres adicionales con los que invocar determinadas órdenes. Para crearlos alias nombre= orden ó nombre='orden' con " podemos escribirlos con opciones de la orden.

Un alias para borrar. Solo duran hasta que se cierra la sesión de trabajo actual.

Para guardarlos para siempre hay que utilizar "variables de entorno"

- **Definición y utilización de variables**

- `$ nombrevariable=valor` ◇ crea variable
p.e: `destino=/home/user01/d1/` ◇ crea variable destino con la ruta
- En los Cshells sí que se utiliza set. Para exámenes **SIN** set.
- `$cp~/arch1 $destino` ◇ Para utilizar el valor almacenado en la variable utilizamos `$nombrevariable`.
- Si yo tengo `destino=/home/user01/d1/` y quiero añadirle una carpeta a2 hay que hacer `$cp~/arch1 ${destino}a2` ◇ Te crea a2 en la ruta de destino.
- `unsetnombrevariable` ◇ Libera espacio de la variable.
`$ unset destino` ◇ libera el contenido de destino.
- `$ set*(volveremos...)` ◇ saca listado de las variable que tenemos definidas.
- Entorno padre-entorno hijo:

Cuando un usuario inicia sesión se crea un Shell de entrada o de usuario, dentro de este nosotros podemos ejecutar ordenes, crear variables y podemos crear y ejecutar scripts o guión:

Fichero que contiene una secuencia de órdenes que podemos ejecutarlas invocando desde el prompt este fichero que contiene las órdenes. Todas estas órdenes que vemos de manera individual, podemos reunir todas las ordenes en un fichero ascii de manera que desde el prompt pongamos el nombre del fichero y se ejecuten simultáneamente.

Cada vez que ejecutamos un script se crea un Shell hijo del Shell de entrada.

(dibujo). En dos cualquier hijo heredaba una copia de su padre, aquí no, solo se hereda lo que de manera explícita se indica que debe heredarse. Si yo quiero que un hijo mío herede algo tengo que decir explícitamente. Mediante la orden:

export variable ◇ esta variable contiene la copia del padre que heredará el hijo

Cualquier Shell mantiene dos áreas de memoria:

1. Área local de datos:
2. Entorno

Cuando el Shell crea una variable, la variable se crea en el área local de datos. Cuando exporto una variable se quita del área y se pasa al entorno.

Cuando creo un Shell subordinado tendrá sus dos áreas de variables:

1. Área local de datos
2. Entorno

Del padre únicamente heredará el entorno. Si este subordinado hace otro hijo este heredará también el entorno... Por lo tanto el entorno del padre lo heredará el hijo, el nieto...

La orden **set** me muestra las variables, tanto del área como del entorno de manera mezclada.

La orden **env** únicamente me lista las variables del entorno.

Podemos hacer un hijo escribiendo **bash** y escribiendo **exit** saldremos del hijo y volveremos al padre.

Variables de entorno del sistema:

Mediante **manbash** llegaremos a un punto que nos mostrará las variables del sistema con su descripción.

- HOME: ruta del directorio home del usuario que consulta la variable
- LOGNAME: nombre de usuario con el que se ha hecho login
- PATH: rutas alternativas donde encontrar ejecutables.
- PS1 /PS2: Definir el prompt para que indique cual es la ruta activa.
- HISTSIZE: Número de líneas de las que se acordará el histórico
- HISTFILE: Fichero en el que se guardan las líneas de órdenes que yo voy tecleando. Por defecto apunta a un fichero oculto que tenemos en home ~/.bash-history

Permiso de archivos y directorios

Usuarios y grupos. Conceptos básicos

En un sistema Linux podemos considerar dos categorías de usuarios:

- Usuario ROOT: Usuario Administrador del sistema. (Administrador. Microsoft)
- Usuarios Normales

Los usuarios pueden unirse en grupos para facilitar la gestión de estos. Un usuario puede pertenecer a más de un grupo y esto hace que en un entorno Linux tengamos que distinguir entre:

- Grupo primario o activo:
- Grupos secundarios:

De todos los grupos a los que pertenece el usuario hay uno que está definido como primario(...), es importante porque determinadas acciones sobre un usuario utilizarán, sino se dice lo contrario, el grupo primario o activo por defecto. Tanto los usuarios como los grupos pueden identificarse mediante:

- Nombre de usuario o de grupo: es el que asigna el usuario ROOT cuando crea el usuario o el grupo.
- 6º Identificador (Nº ID): Si el nombre lo asignaba el usuario ROOT el número de identificador lo asigna el sistema de manera automática. Es un número único para cada usuario. En determinados contextos vendrán representados por las letras UID (user Identificador, número que le corresponde al usuario) ó GID (Group Identificador, número que le corresponde al grupo)

Ordenes:

- **ID:** Puede ejecutar cualquier usuario que lo que hace es informar sobre datos identificativos del usuario.
\$ id
uid=503(user05) gid=100(users) groups=100(users),101(programadores),102(mantenimiento). // como grupo activo es el de 100(users), este grupo es // un grupo por defecto, que es al grupo al que se le asigna al usuario por defecto
- **groups:** consulta los grupos a los que pertenece el usuario. Lo puede ejecutar cualquier usuario.

Propiedad de un fichero/directorio. Usuario Propietario/grupo propietario

Cuando se crea un fichero o directorio, el sistema asigna al usuario que crea el archivo como usuario propietario y al grupo activo, al que pertenece ese usuario, como grupo propietario.

Con user05

\$ cat > xx

```
$ls -l
```

Aparecerá:

```
-rw-r--r-- 1 user05 users 07 may 15'25 xx ◇ Usuario propietario(user05), Grupo propietario (users)
```

Orden **newgrp**: Permite cambiar de grupo activo

```
$newgrp programadores
```

```
Passwd: ..
```

Cuando haces el cambio de grupo lo hace en otro shell y tendremos que salir dos veces...

```
$id
```

```
Uid=503(user05) gid=101(programadores) ...
```

```
$cat>zz
```

```
$ls -l
```

```
-rw-r--r-- 1 user05 users 07 may 15'25 xx
```

```
-rw-r--r-- 1 user05 programadores 07 may 15'30 zz
```

```
$exit //cierro el Shell subordinado
```

```
$id //el grupo activo volvería a ser USERS◇
```

```
$ id
```

```
uid=503(user05) gid=100(users) groups=100(users),101(programadores),
```

Permisos

Cada archivo o directorio está dotado de una palabra de permisos que indican lo que los usuarios pueden hacer con ese archivo o con ese directorio. Se trata de una palabra de permisos de 9 ó 10 posiciones (-rw-r--r--)ese 1 de diferencia es por la primera posición porque solo indica el tipo de fichero(archivo o directorios), **nosotros incluimos 10**. Las otras 9 posiciones se divide en tres grupos de tres posiciones (-|rw-|r--|r--) cada uno de estos grupos. El primer grupo indica lo que el usuario propietario puede hacer con ese archivo. El segundo grupo indica los permisos del grupo propietario. El tercer grupo son los permisos para el resto de usuarios(ni usuario activo, ni pertenecen al grupo user) cada uno de estos tres grupos tienen una estructura similar, cada una de ella puede tener los siguientes valores.

- R ó – ◇ lectura
- W ó – ◇escritura
- X ó – ◇ejecución

Si se tiene la letra se tiene el permiso si se tiene el guión no. El significado de la R, W, X no es el mismo si hablamos de directorios o archivos

	R	W	X
Fichero	Leer(cat,vi...)	Modificar(vi,redireccionar, sobre él,...)	Ejecutar
Directorio	Examinar(ls)	Creación/supresión en el directorio	Pasar por él(cd)

Para borrar un archivo no se necesitan permiso sobre el archivo, sino permisos sobre el directorio. Si borras un archivo modificas el contenido del directorio por lo tanto no te va a dejar.

Borrar: miramos la W del directorio en que está el archivo

Copiar: R de archiu i W de directoridesti

Moure: W de directori i w des directoridesti.

Ordenes que permiten manejar propiedades y permisos

1. **chown:** Permite ceder la propiedad tanto de usuario como de grupo de un fichero(tanto fichero como directorio), esta orden solo puede utilizarla usuarios ROOT.
2. **chgrp:** Permite ceder únicamente la propiedad de grupo y puede utilizarla cualquier usuario.
3. **chmod:** Tiene una primera forma de utilizar parecida a attrib que con el + añades atributo – quitas atributos.
 - a. Chmodgrupodepermisos±permisos ficheros
\$chmodgo+wx xx ◇añado wx a g(grupo propietario) y a o(grupo others) del fichero xx

Para poder hacer esto hay que ser ROOT.

Permisos Absolutos: Patrones Binarios

Esta forma de modificar los archivos(mediante chmod) recibe el nombre de método simbólico. Hay otra manera de modificar con los Permisos Absolutos.

En este modo las letras pasan a ser 1 y los guiones pasan a ser 0. 110|111|111 ◇6|7|7.(octal)
Puedo utilizar esta forma para representar los archivos.

\$chmod 677 xx

No son exactamente equivalentes el método simbolico hace referencia a los permisos que cambian mientras que los permisos absolutos ¿?te lo deja de esa manera?

Permisos por defectos

Los permisos por defecto están definidos mediante una máscara llamada **umask** que se utiliza de la siguiente manera, dependiendo de si se trata de un archivo o directorio que queremos crear.

666 AND NOT UMASK ◇Archivos 666=110 110 110

777 AND NOT UMASK ◇Directorios

Normalmente la máscara está definida como “022” 022=000 010 010. La negación del máscara es 111 101 101.

Entonces ahora hay que hacer un and entre 666 y UMASK (and es verdadero cuando las dos premisas son verdaderas)

$(110\ 110\ 110) \text{ and } (111\ 101\ 101) = 110(rw-) \ 100(r--) \ 100(r--)$

Y con 777 igual.

Esta máscara si el usuario quiere la puede cambiar con la orden umask

\$umask 025

OJO!!!-->En los libros dicen que es una resta pero es and not

Instalación y configuración

Proceso de instalación:

1. Fase de preparación: básicamente consiste en procurar una partición en la cual reside el sistema operativo.
 - a. Procurar 1 particion libre
 - a.i. Una posibilidad hacer la instalación sobre todo el disco duro,
 - a.ii. Varias particiones y elegir una de esas particiones como una en la que se va a instalar el sistema operativo, en esta, puede suceder que no hayamos tenido en cuenta el espacio del disco duro y no tenemos partición para instalar. Debemos procurarnos una herramienta de reparticionamiento, para conseguir más espacio y poder instalar el SO, esto será posible siempre que seamos capaz de conseguir un espacio grande y contiguo en el disco, libre.
 - a.iii. PartitionMagic
 - a.iv. Habrá otras distribuciones de Linux que dentro del CD de instalación podremos encontrar herramientas de particionamiento, estas serán capaces de conseguir el espacio contiguo y otras no serán capaces de hacerlo todo y se tendrán que unir con un proceso de desfragmentación. Con el desfragmentador me unirá todos los datos para poder hacer una partición contigua a las otras. Ya que si los datos estuviesen desperdigado sería muy difícil encontrar un espacio contiguo.
 - a.v. En la distribución de clase dosutils\lfs♦Te hacía una división para crear la nueva partición.
 - b. Partición de swap: La finalidad de esta partición es la de ampliar de forma virtual la memoria de nuestro ordenador, almacenando los datos que no caben la memoria RAM física, homólogo en Windows, pagefile.sys♦archivo de memoria virtual. En Windows archivo en Linux partición(**importante**). Convendría darle el doble del tamaño que tiene la RAM, para otros autores

consideran que la memoria RAM ya tiene bastante tamaño no tiene porque asignarle tanto espacio a la partición swap.

b.i. Si la RAM es $\leq 1\text{GB}$ el swap debería ser igual que la RAM

b.ii. Entre $2\text{GB} \dots 4\text{GB}$ el swap debería tener la mitad de la RAM

b.iii. $\text{RAM} \geq 4\text{GB}$ el swap no debería ser mayor de 2GB

2. Inicio de un S.O básico. Cuando se carga el sistema se hace a través de un SO

3. (SUSE) depende de la distribución

- a. Ejecución linuxrc, se configuran aspectos como el idioma, tipo de pantalla que queremos utilizar durante la instalación, carga de drivers para el disco duro etc. Se configura un poco como va a ser el proceso de instalación, Tipo de interface. No se instala nada... solo se configura.
- b. Ejecución YAST es la que hace la instalación del SO. YAST es una herramienta que va a utilizar continuamente el administrador del sistema. Para gestión de usuarios, de la red... La distribución que elijamos podrá tener un equivalente a YAST o puede que no.

Una vez instalado debería ser posible utilizar el sistema como usuario ROOT(administrador). En el caso de SUSE se hace la instalación y se tiene acceso mediante contraseña al usuario ROOT. Pero en UBUNTU el acceso a ROOT no está tan claro, se debe realizar algo para poder utilizarlo.

Otro aspecto importante como usuarios que acabamos de instalar un SO. Como detener el sistema? No podemos apagar la máquina cuando nos dé la gana porque estamos en un SO multiusuario y debemos avisar a los demás usuarios

Configuración del arranque

Cuando el SO, convive con otros SO. Existen distintas posibilidades:

- 1. ¿?(W9x-WME)◇ Posibilidad de iniciar Linux habiendo iniciado previamente un sistema Microsoft. Para ello se incluía una utilidad loadlin. En Windows le metías loadlin en el prompt y se iniciaba. Ventajas no modifica el MBR de Microsoft
- 2. Gesto de arranque suministrado por Linux. De manera que al iniciar aparezca un menú para elegir el SO.
 - a. LILO
 - b. GRUB: más utilizado
- 3. Gestor de arranque suministrado por Microsoft.

Entorno gráfico

Podemos trabajar de dos formas (comandos o gráfico), no hay un único entorno gráfico para linux. Paquete de herramientas de administración que incluye el entorno gráfico

Proceso de arranque

Normalmente el proceso de arranque, empezará con la aparición de un mensaje similar a uncompressing Linux... Indica que el núcleo esta tomando el control del hardware

En SUSE: consiste en un fichero vmlinuz que está en la raíz de la estructura de directorios, activa la lectura de los parámetros de la BIOS y la inicialización de los interfaces con el hardware, a continuación los controladores comprueban e inicializan el hardware existente, se sigue con la comprobación de particiones y montaje del root file sistema(sistema de archivos raíz (árbol de directorios que cuelga de raíz)) y a continuación se ejecuta /sbin/init.

INIT: directamente iniciado por el Kernel y a partir de ahí es el responsable de la correcta inicialización de todos los procesos del sistema. Es el padre de todos los procesos, de forma que todos los demás programas son arrancados por él o por alguno de sus procesos hijo

etc/**inittab** contiene una serie de opciones que configuran el funcionamiento de init. Entre otras cosas el **inittab** contiene el runlevel con el que se iniciará el sistema y la definición de lo que ha de suceder(qué procesos deber arrancarse) en cada runlevel. Que procesos han de iniciarse y detenerse cuando paso de un runlevel a otro. Todos estos procesos que deben ejecutarse están dentro de un directorio(init.d) etc/init.d/.

Un **runlevel**(nivel de ejecución) define como se inicia el sistema y que servicios están disponibles cuando este está en funcionamiento. Hay 8 posibles niveles de ejecución que pueden tener un significado distinto dependiendo de la distribución de Linux. Los que si son estandarizados:

1. Runlevel 0: Indica Halt (parada del sistema)
2. Runlevel 6: Reboot(reinicio)

EN SUSE:

3. runlevel 1: modo monousuario. Para evitar que algun otro usuario pueda utilizar el usuario
4. runlevel 2: modo multiusuario sin red.
5. runleve 3: modo multiusuario con red

(dibujo cuaderno)Tenemos scripts. Para cada nivel de ejecución tengo un directorio. Dentro de estos directorios, únicamente tengo vínculos simbólicos(para ahorrar espacio porque hay algunos que se utilizan más de una vez) a los scripts que tengo en *scripts* cuyo nombre empieza o bien por S o bien por K.

Los que empiezan por S(start scripts):se ejecutan los scripts del runlevel al que quiero llevar

Los que empiezan por K(stop scripts): Cuando yo decidido hacer un cambio de un runlevel a otro han de ejecutarse una serie de scripts que detienen el runlevel en el que estoy.

Varia para cada distribución pero sí que hay un nivel para cada runlevel.

Órdenes:

Init 0 ⇨ equivalente a un shutdown, empezará a ejecutar los K del nivel en el que te encuentres.

Runlevel ⇨ orden de consulta para saber en el nivel que estamos y también el anterior.

Gestión de usuarios

Para ello dispondremos de distintas posibilidades:

1. Órdenes del Shell:

- a. **useradd**: crea usuario
- b. **usermod**: modifica usuario
- c. **userdel**: elimina usuario
- d. **chfn**: Permite modificar la descripción del usuario
- e. **chsh**: cambiar el Shell del usuario.
- f. **groupadd**: añade grupo.
- g. **groupmod**: modifica grupo.
- h. **groupdel**: borra grupo.

Cualquier operación de creación de usuarios o grupos queda reflejado en los siguientes archivos:

/etc/passwd ⇨ base de datos de los usuarios que tenemos creados en el sistema

/etc/group ⇨ igual pero con grupos.

Tanto uno como otro son ficheros de texto(ASCII) con la siguiente estructura:

Passwd: para cada usuario registrado tenemos una línea con la siguiente información:

Username:

passwd:XuserId:numerodeusuariogroupId:numerodegrupoComment:descripci
ón del usuario Home_directory:del usuario login_shell shell entrada

la passwd codificada está asociada a un archivo **/etc/shadow/**

group: para cada grupo registrado tenemos una línea con la siguiente información:

groupname:nombrepasswd:contraseña group_ID:numeromembers(miembros)

- i. **pwck**: comprobar si hay inconsistencias dentro del archivo de usuarios(si hay dos usuarios con el mismo nombre...etc)
- j. **grpck**: comprobar si hay inconsistencias dentro del archivo de grupos

- k. passwd: permite hacer más cosas.
- 2. Editando directamente los archivos /etc/passwd y /etc/group
 - a. Con cat, vi, etc.
- 3. Utilidades
 - a. YAST: en su principio solo en modo consola, pero ahora también existe en modo gráfico
 - b. ...

Directorio /etc/skel: contiene los archivos y directorios que se crearan dentro del directorio home de los usuarios que se den de alta.

Ordenes su y sudo

Estas órdenes permiten no tener que cerrar nuestra sesión de trabajo actual y abrir la cuenta root. Para no tener que cerrar la cuenta y abrir la otra.

Comunicación entre administrador y usuarios

Orden Wall: escribir a todos(writeall)◊mediante esta orden podemos enviar un mensaje a todos los usuarios conectados en ese momento

Archivo /etc/motd:message of de day◊envía un mensaje cuando los usuarios inician sesión

Administración del sistema de archivos

- Archivos de dispositivos: Son los archivos que se encuentran en el directorio /dev(device) en Linux el hardware también está representado con archivos(drivers en Microsoft)
- Sistema de archivos:
 - EXT: Sistema de archivos característico de Linux
- Montaje/Desmontaje de sistema de archivos:

Los directorios van montados en la raíz principal. Necesitamos una carpeta donde montar el árbol de directorios que nos interesa poder acceder. Igual que Microsoft... Montar en carpetas vacías.

Para facilitar esta labor muchas distribuciones incluyen ya directorios por si necesitas acceder a ese volumen...

Para hacer el montaje se utiliza la orden mount como mínimo dos parámetros carpeta de destino y el otro archivo de dispositivo asociado al árbol que quiero montar(que esta en /dev), cuando no queremos acceder más a ese directorio hay que desmontar, orden umount. El proceso de arranque es capaz de montar una serie de directorios, que se encuentran en /etc/fstab. Contiene la lista de sistemas de archivos que deseamos que se monten de manera automática al iniciar el sistema.

- Otras ordenes relacionadas

mkfs◇para formatear el disco.

df(displayfilesystem)◇orden de consulta para saber el sistema de ficheros que tenemos montado y permite averiguar un elemento a que sistema de archivos pertenece

fsck(file Systemcheck)◇chekear el sistema de archivos

E2fsck◇chekear el sistema de archivos

Dumpe2fs◇chekear el sistema de archivos