

# SISTEMAS OPERATIVOS: SISTEMAS DE FICHEROS

Ficheros, directorios y sistema de ficheros

# Objetivos

- ❑ Conocer el concepto de fichero y directorio, así como sus principales características.
- ❑ Utilizar los servicios de gestión de ficheros y directorios ofrecidos por el sistema operativo.
- ❑ Comprender los mecanismos generales usados para trabajar con un fichero y directorio en el sistema operativo.

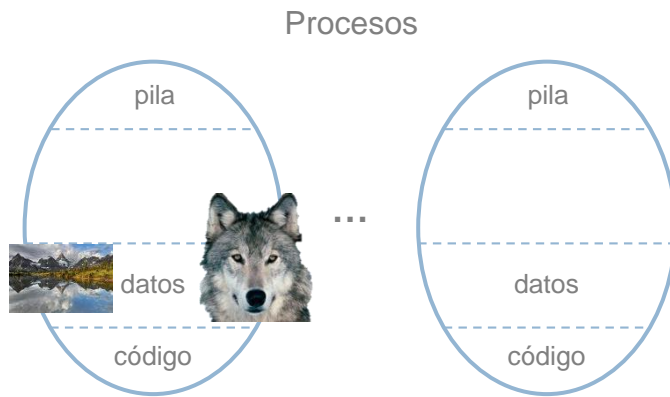
# Contenidos

- ❑ Introducción
- ❑ Fichero
- ❑ Directorio
- ❑ Sistema de ficheros
- ❑ Particiones/Volúmenes
- ❑ Dispositivos
- ❑ Software de sistema

# Contenidos

- ❑ **Introducción**
- ❑ Fichero
- ❑ Directorio
- ❑ Sistema de ficheros
- ❑ Particiones/Volúmenes
- ❑ Dispositivos
- ❑ Software de sistema

# Introducción

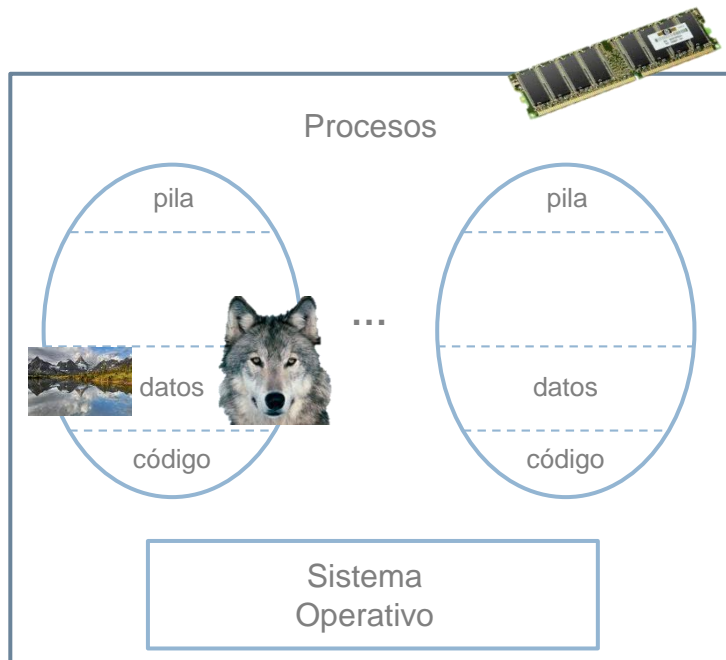


- Un proceso de edición de fotografías (por ejemplo) tiene en memoria su código y datos.
  - ▣ Cada proceso trabaja con sus datos, pudiendo generar nuevos datos.

# Introducción

6

Sistemas Operativos - ARCOS

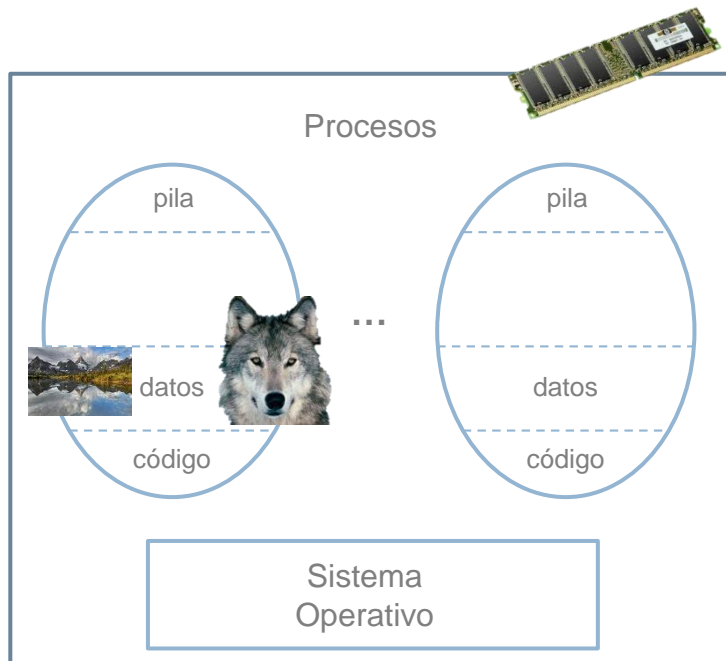


- Un proceso de edición de fotografías (por ejemplo) tiene en memoria su código y datos.
  - ▣ Cada proceso trabaja con sus datos, pudiendo generar nuevos datos.
- Puede haber varios procesos en memoria, siendo el sistema operativo el que reparte y organiza la memoria.

# Introducción

7

Sistemas Operativos - ARCOS

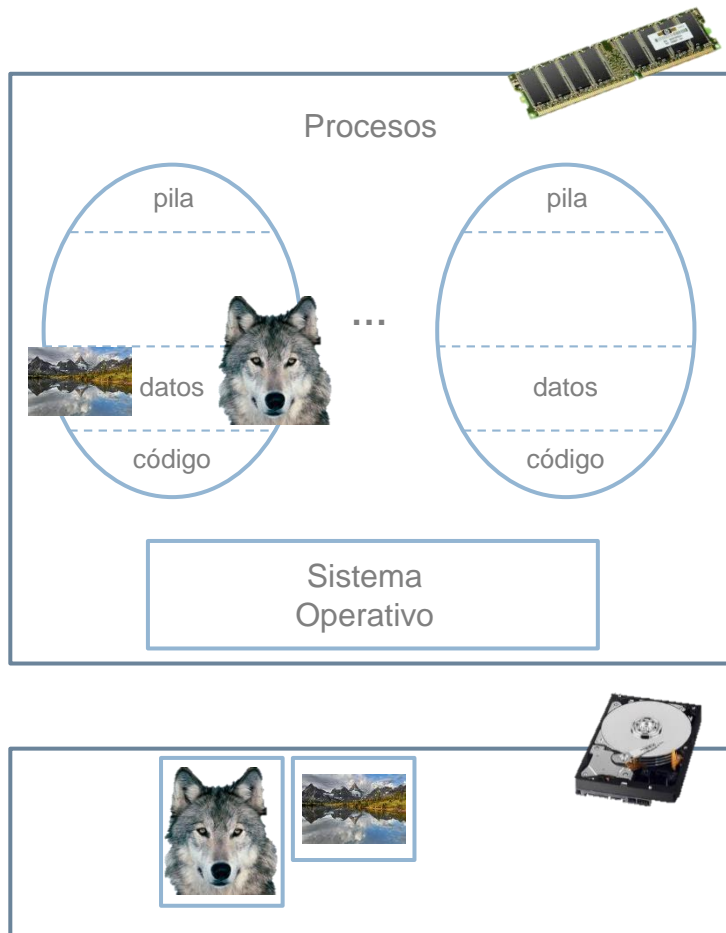


- La memoria principal en los sistemas actuales es de pequeño tamaño, acceso a palabra y volátil.
  - ▣ Los datos almacenados no son persistentes (sin electricidad).
  - ▣ Solo se usa para guardar los datos accedidos por el procesador durante un periodo.
  - ▣ Se puede acceder a cualquier palabra directamente.
- ¿Dónde guardar los datos?

# Introducción

8

Sistemas Operativos - ARCOS



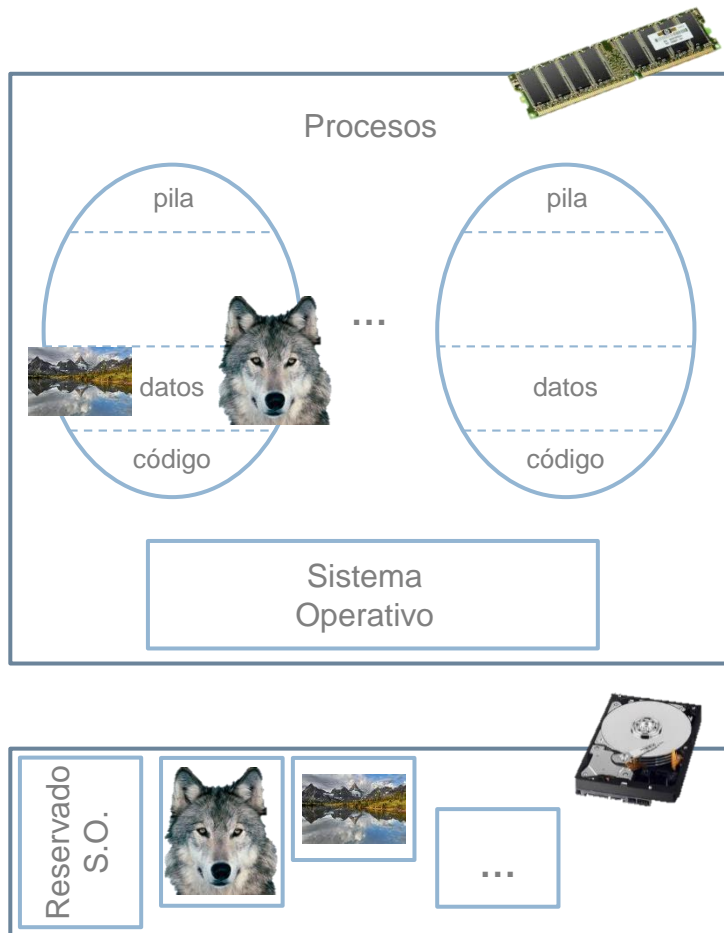
- La memoria secundaria es de mayor tamaño, acceso a bloque y no volátil.
  - ▣ Datos persistentes
    - Al proceso que lo usa, a la lectura concurrente entre procesos.
  - ▣ Permitirá guardar mayor cantidad de datos que en M.P.
  - ▣ Organizada en bloques, lo que supone tener que gestionar el uso de estos bloques.
- Los datos se guardarán en M.S.: disco duro, flash, etc..



# Introducción

9

Sistemas Operativos - ARCOS

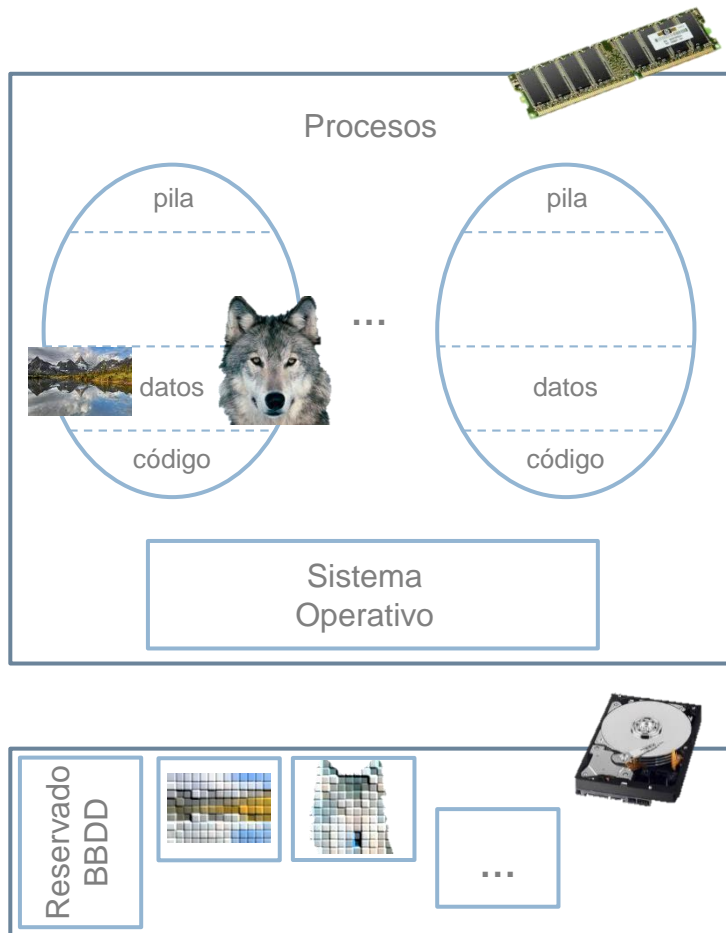


- Parte del sistema operativo se encarga de repartir y organizar la M.S.
  - ▣ Sistema de ficheros.
- El sistema de ficheros ofrece servicios para almacenar y recuperar los datos de forma simple
  - ▣ Oculta los detalles de la organización de la M.S. mediante abstracciones: ficheros, directorios, etc.

# Introducción

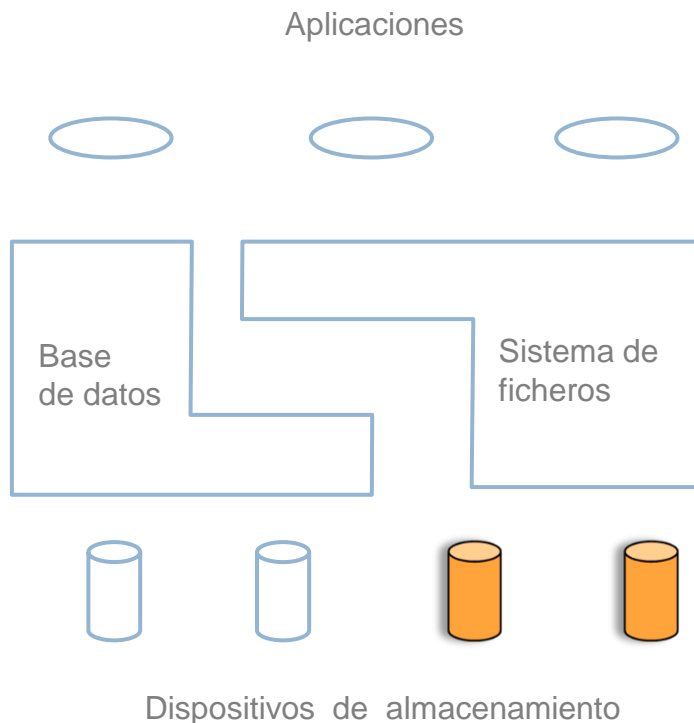
10

Sistemas Operativos - ARCOS



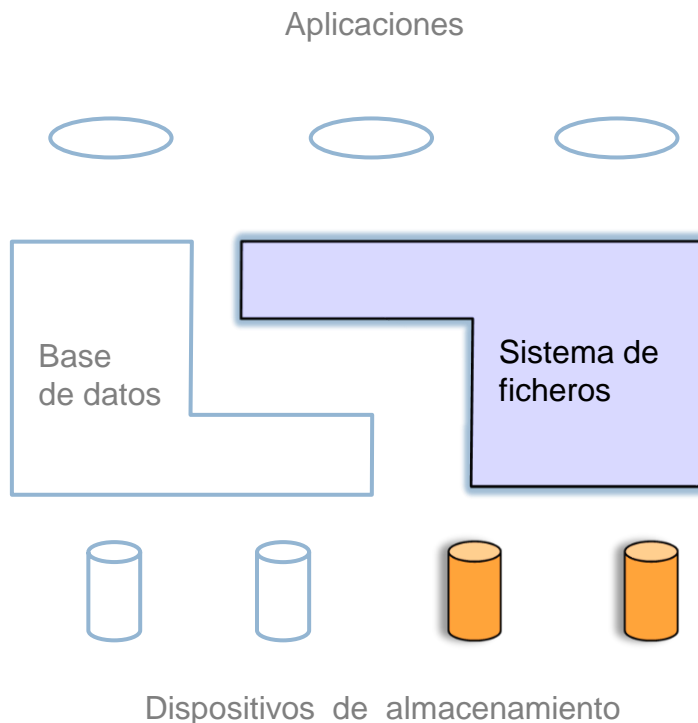
- Pero también es posible que ciertas aplicaciones organicen la M.S.:
  - ▣ Gestores de bases de datos.
- El sistema operativo ofrece acceso a todo el dispositivo.
- Es posible también una organización mixta
  - ▣ parte el sistema operativo y parte la aplicación

# Resumen: arquitectura



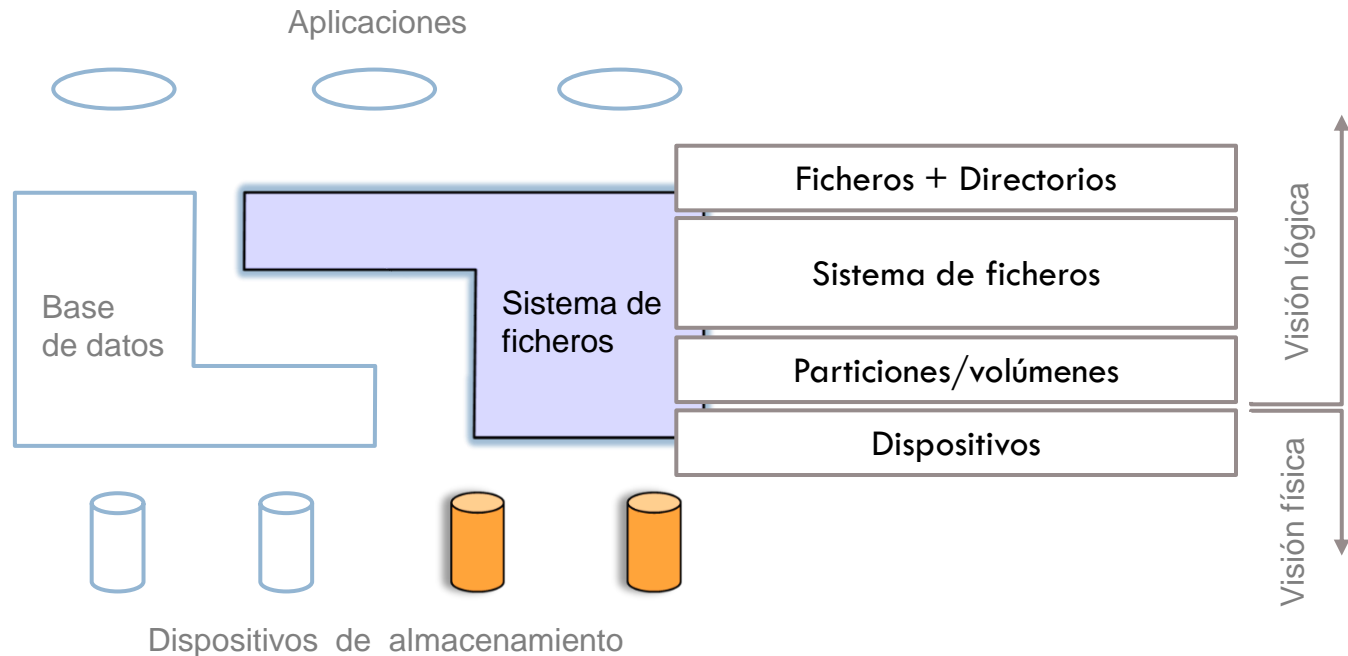
- Tenemos ambas posibilidades en ilustración que propone la SNIA:
  - ▣ *Storage Networking Industry Association*
  - ▣ <http://www.snia.org>
- Las aplicaciones acceden a los datos almacenados en los dispositivos de almacenamiento usando BBDD y/o Sist. Fich.

# Resumen: arquitectura



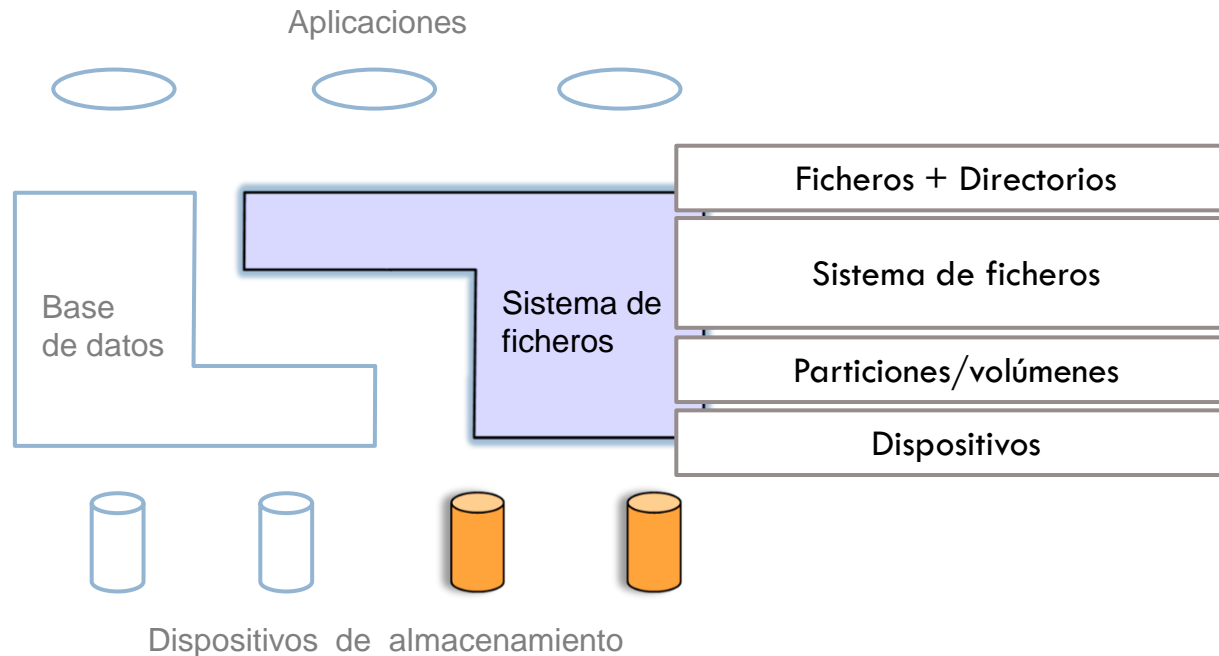
- En este tema nos centraremos en la gestión mediante el S.O. a través del sistema de ficheros:
  - ▣ Organización
  - ▣ Almacenamiento
  - ▣ Recuperación
  - ▣ Gestión de nombres
  - ▣ Implementación de la semántica de coutilización
  - ▣ Protección

# Resumen: abstracciones



- A estudiar: ficheros, directorios, sistema de ficheros, volúmenes y dispositivos  
Visión lógica                      Visión física

# Resumen: abstracciones



- ❑ Cuidado con el término “sistema de ficheros” que es usado para nombrar tanto al software gestor como para las estructuras de datos en disco

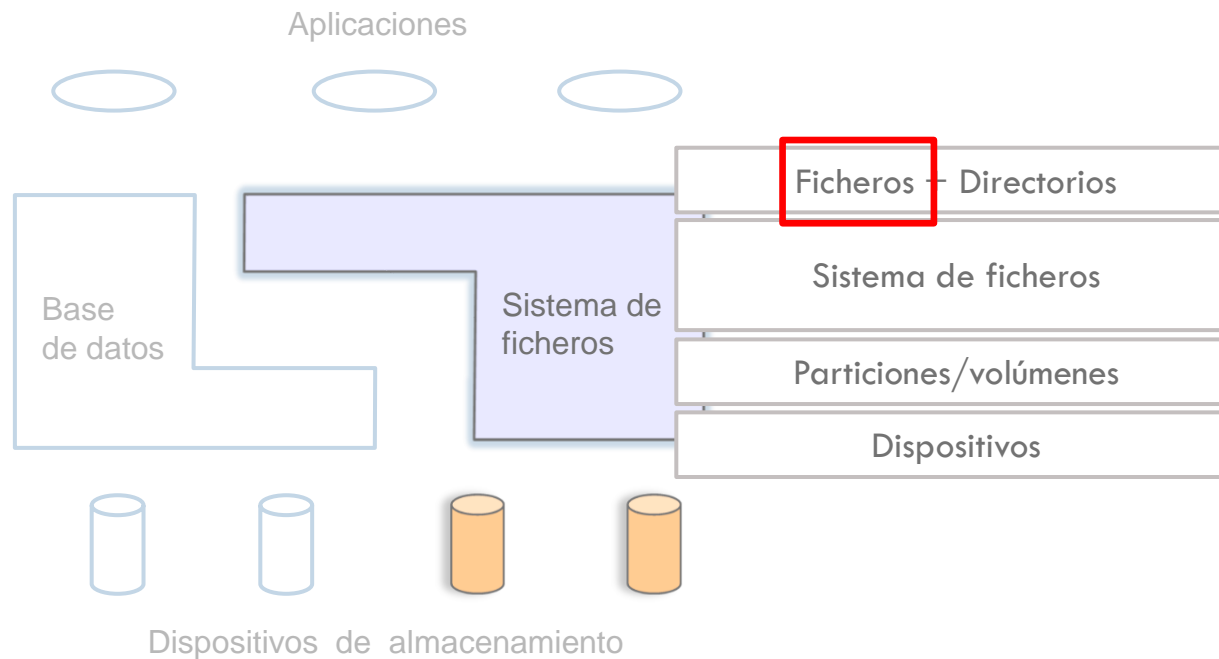
# Contenidos

- Introducción
- **Fichero**
  - **Definición**
  - Metadatos
  - Interfaz
  - Métodos de acceso
  - Semántica de compartición
- Directorio
- Sistema de ficheros
- Particiones/Volúmenes
- Dispositivos
- Software de sistema

# Fichero o archivo

16

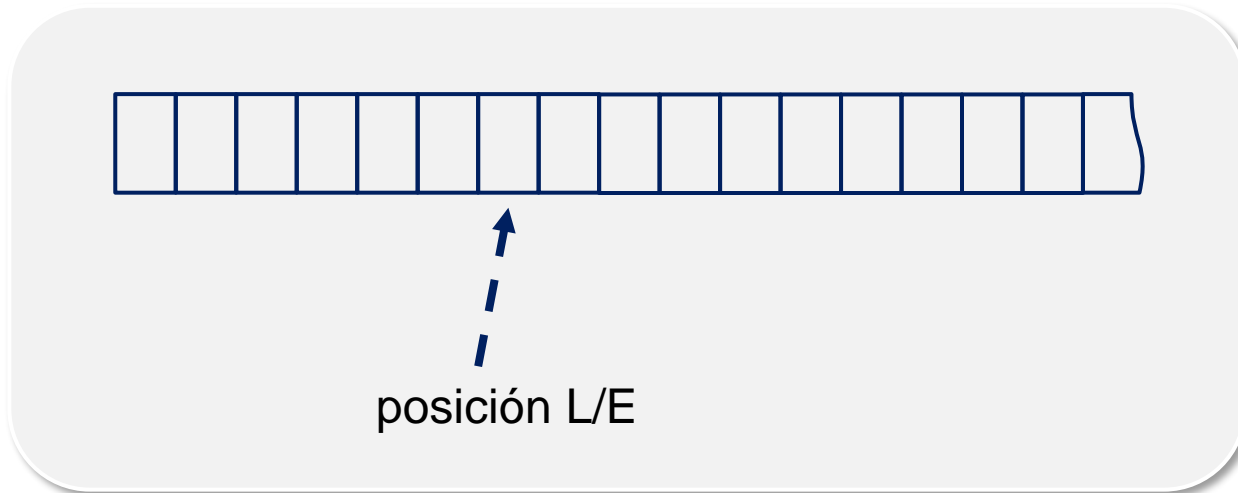
Sistemas Operativos - ARCOS





# Fichero o archivo

- Conjunto de información relacionada que ha sido definida por su creador.
- Habitualmente el contenido es representado por una secuencia o tira de bytes:



# Fichero o archivo

18

Sistemas Operativos - ARCOS

## □ Diferentes tipos de información:



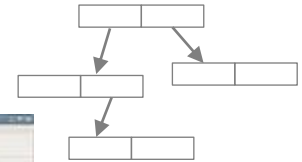
# Fichero o archivo

19

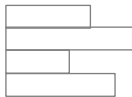
Sistemas Operativos - ARCOS

## □ Diferentes tipos de estructura de esa información:

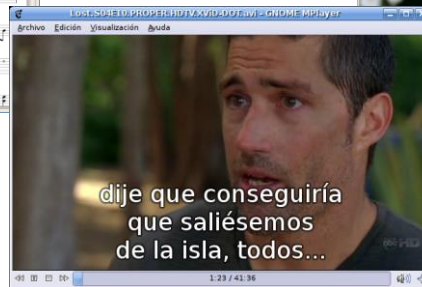
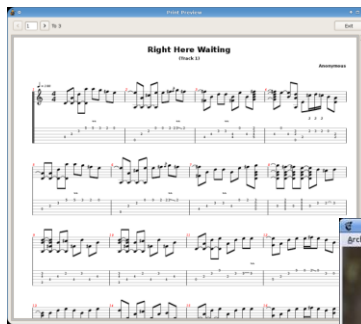
- Complejos
  - Formato (XML, etc.)
  - Reubicables



- Registros
  - Longitud fija
  - Longitud variable

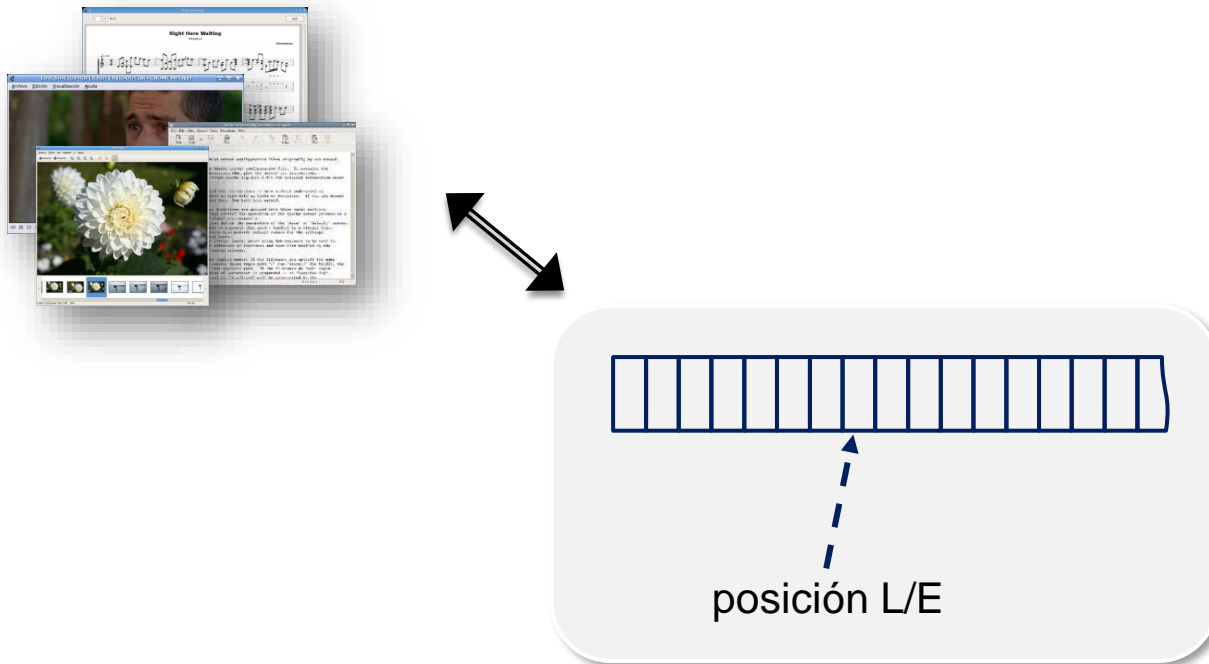


- Secuencia de palabras



# Fichero o archivo

- Las aplicaciones convierten y almacenan como una **secuencia o tira de bytes**:



# Contenidos

- Introducción
- **Fichero**
  - Definición
  - **Metadatos**
  - Interfaz
  - Métodos de acceso
  - Semántica de compartición
- Directorio
- Sistema de ficheros
- Particiones/Volúmenes
- Dispositivos
- Software de sistema

# Fichero o archivo

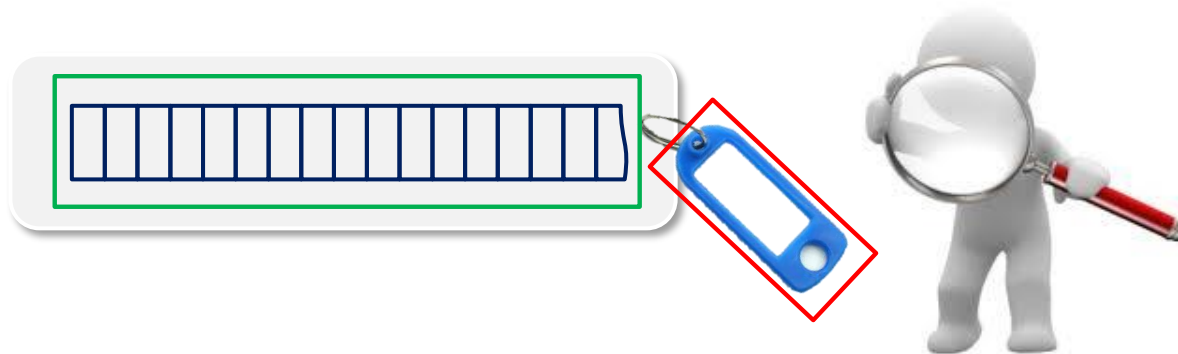
## □ Información de un archivo:

### □ Datos

- Información que almacena el archivo.

### □ Metadatos



- Información sobre el archivo: distintos **atributos** sobre el archivo.



# Fichero o archivo: atributos

- Atributos típicos de un fichero:
  - **Nombre**: identificador para los usuarios del fichero.
  - **Tipo**: tipo de archivo (para los sistemas que lo necesiten)
    - Ej.: extensión (.exe, .pdf, etc.)
  - **Localización**: identificador que ayuda a la localización de los bloques del dispositivo que pertenecen al archivo.
  - **Tamaño**: tamaño actual del fichero.
  - **Protección**: control de qué usuario puede leer, escribir, etc.
  - **Día y hora**: instante de tiempo de último acceso, de creación, etc. que permite la monitorización del uso del archivo.
  - **Identificación** de usuario: identificador del creador, dueño del archivo, etc.

# Nombre de fichero

- Se utiliza tiras de caracteres:
  - ▣ Permite a los usuarios organizarse mejor
  - ▣ Los usuarios no recuerdan nombres del tipo 001 1 2233
- Es característico de cada sistema de ficheros:
  - ▣ Longitud del nombre: fijo (MS-DOS) o variable (UNIX)
  - ▣ Sensibles a mayúsculas/minúsculas (Unix) o no (MS-DOS)
    - INMA e inma
  - ▣ Necesario extensión: si y fija (MS-DOS), no (UNIX)
    -  remain.zip    .zip -> identifica el tipo de fichero (y la aplicación a usar)
    -  /bin/ls    file nombre -> identifica por contenido (número mágico)

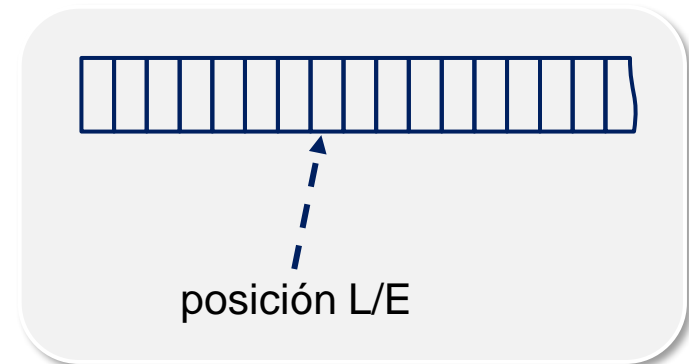


# Contenidos

- Introducción
- **Fichero**
  - ▣ Definición
  - ▣ Metadatos
  - ▣ **Interfaz**
  - ▣ Métodos de acceso
  - ▣ Semántica de compartición
- Directorio
- Sistema de ficheros
- Particiones/Volúmenes
- Dispositivos
- Software de sistema

# Fichero o archivo: interfaz

- **Interfaz genérica** para acceder a la información:
  - `descriptor` ← `open` (nombre, flags, modo)
  - `close` (descriptor)
  - `read` (descriptor, puntero, tamaño)
  - `write` (descriptor, puntero, tamaño)
  - `lseek` (descriptor, desplazamiento, origen)
  - `ioctl` (descriptor, operación, puntero\_a\_parámetros)



# Fichero o archivo: interfaz POSIX

27

Sistemas Operativos - ARCOS

## escritura

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <fcntl.h>

int main ( int argc, char *argv[] )
{
    int fd1 ;
    char str1[10] ;
    int nb ;

    fd1 = open ("/tmp/txt1",
                O_CREAT|O_RDWR, S_IRWXU);
    if (-1 == fd1) {
        perror("open:");
        exit(-1);
    }

    strcpy(str1,"hola");
    nb = write (fd1,str1,strlen(str1));
    printf("bytes escritos = %d\n",nb);

    close (fd1);
    return (0) ;
}
```

## lectura

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <fcntl.h>

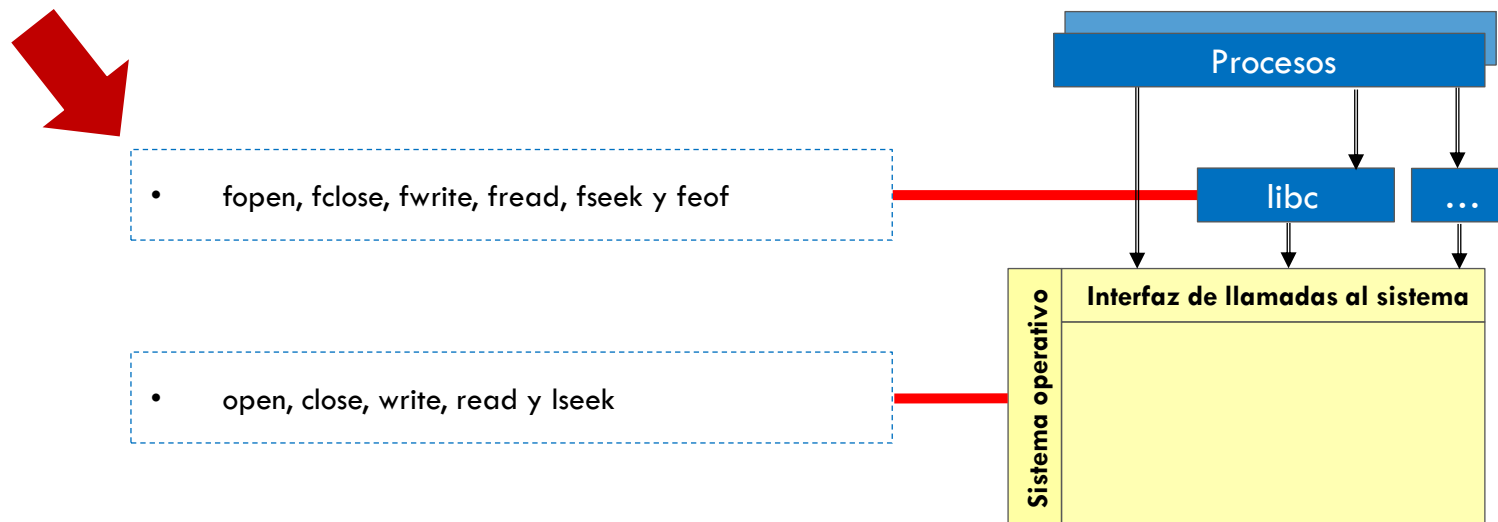
int main ( int argc, char *argv[] )
{
    int fd1 ;
    char str1[10] ;
    int nb, i ;

    fd1 = open ("/tmp/txt1",O_RDONLY);
    if (-1 == fd1) {
        perror("open:");
        exit(-1);
    }

    i=0;
    do {
        nb = read (fd1,&(str1[i]),1); i++;
    } while (nb != 0) ;
    str1[i] = '\0';
    printf("%s\n",str1);

    close (fd1);
    return (0);
}
```

# Fichero o archivo: interfaz C99



# Fichero o archivo: interfaz C99

29

Sistemas Operativos - ARCOS

## escritura

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main ( int argc, char *argv[] )
{
    FILE *fd1 ;
    char str1[10] ;
    int nb ;

    fd1 = fopen ("/tmp/txt2","w+");
    if (NULL == fd1) {
        printf("fopen: error\n");
        exit(-1) ;
    }

    strcpy(str1,"mundo");
    nb = fwrite (str1,strlen(str1),1,fd1);
    printf("items escritos = %d\n",nb);

    fclose (fd1) ;
    return (0) ;
}
```

## lectura

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main ( int argc, char *argv[] )
{
    FILE *fd1 ;
    char str1[10] ;
    int nb, i ;

    fd1 = fopen ("/tmp/txt2","r");
    if (NULL == fd1) {
        printf("fopen: error\n");
        exit(-1) ;
    }

    i=0;
    do {
        nb = fread (&(str1[i]),1,1,fd1) ;
        i++ ;
    } while (nb != 0) ; /* feof() */
    str1[i] = '\0' ;
    printf("%s\n",str1);

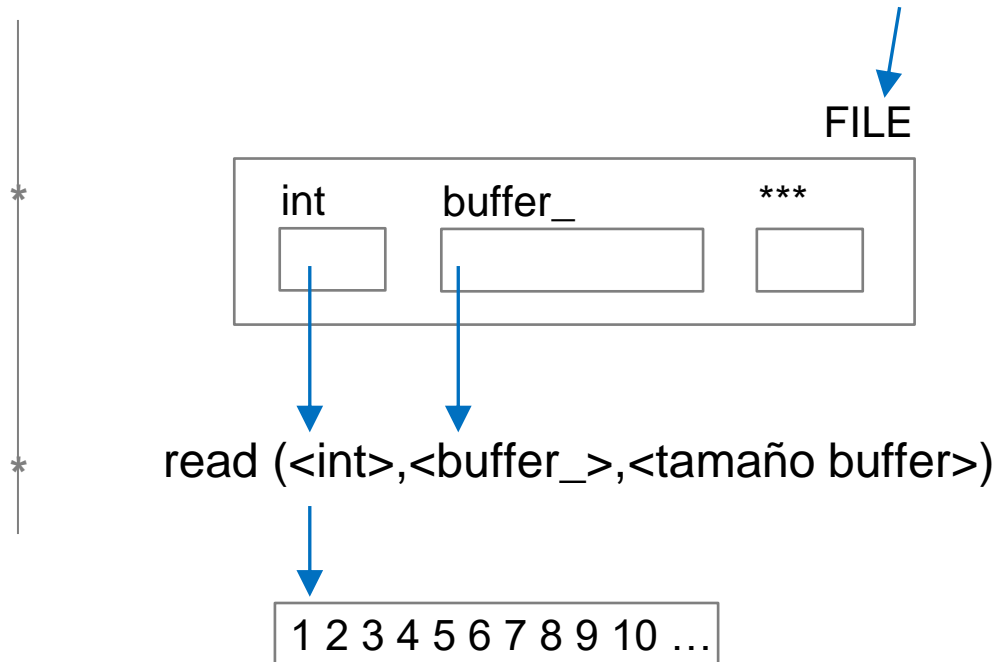
    fclose (fd1);
    return (0);
}
```

# Fichero o archivo: interfaz C99

30

Sistemas Operativos - ARCOS

fread (<buffer>,<tamaño 1 elto>,<n<sup>o</sup> eltos>,<FILE \*>)



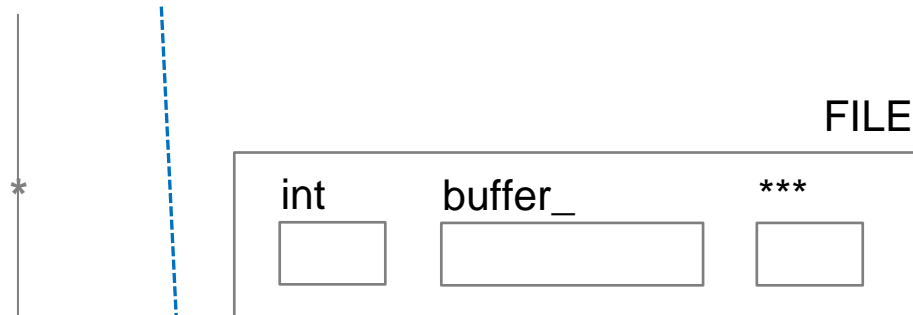
Un puntero a FILE contiene el descriptor de fichero y un buffer intermedio (principalmente)...

# Fichero o archivo: interfaz C99

31

Sistemas Operativos - ARCOS

`fread (<buffer>,<tamaño 1 elto>,<no eltos>,<FILE *>)`



`read (<int>,<buffer_>,<tamaño buffer>)`

1 2 3 4 5 6 7 8 9 10 ...

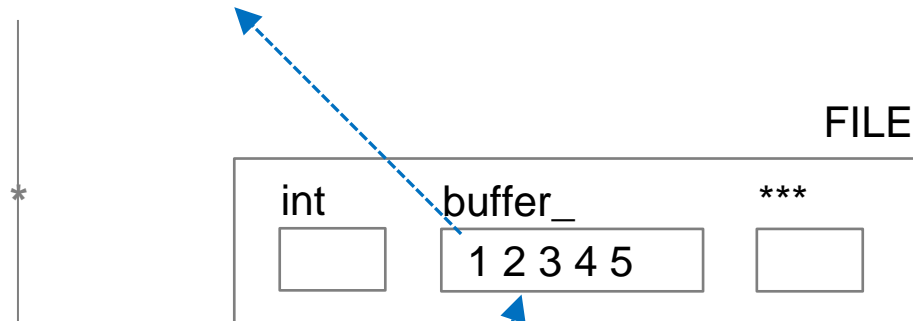
... de manera que cuando se pide la primera lectura, se realiza una lectura sobre el buffer (cuyo tamaño es mayor que el elemento pedido)...

# Fichero o archivo: interfaz C99

32

Sistemas Operativos - ARCOS

fread (<buffer>, <tamaño 1 elto>, <n<sup>o</sup> eltos>, <FILE \*>)



read (<int>, <buffer\_>, <tamaño buffer>)

1 2 3 4 5 6 7 8 9 10 ...

... los datos se cargan en el buffer y se copian la porción pedida al proceso que hace el fread...

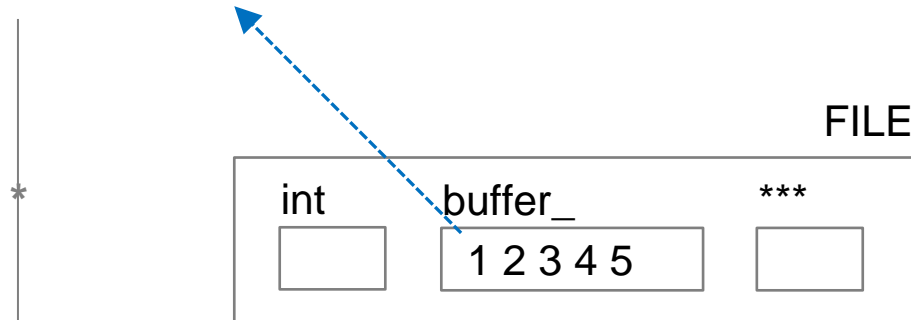


# Fichero o archivo: interfaz C99

33

Sistemas Operativos - ARCOS

`fread (<buffer>,<tamaño 1 elto>,<no eltos>,<FILE *>)`



`read (<int>,<buffer_>,<tamaño buffer>)`

1 2 3 4 5 6 7 8 9 10 ...

...y la siguiente vez que se hace una lectura, si está en el buffer (memoria) se copia directamente de él. De esta forma se reduce las llamadas al sistema, lo que acelera la ejecución.

# Fichero o archivo: interfaz C99

34

Sistemas Operativos - ARCOS

escritura

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/time.h>

#define BSIZE 1024

int main ( int argc, char *argv[] )
{
    FILE *fd1 ; int i; double tiempo ;
    char buffer1[BSIZE] ;
    struct timeval ti, tf;

    gettimeofday(&ti, NULL);
    fd1 = fopen ("/tmp/txt2","w+");
    if (NULL == fd1) {
        printf("fopen: error\n");
        exit(-1) ;
    }
    setbuffer(fd1,buffer1,BSIZE) ;
    for (i=0; i<8*1024; i++)
        fprintf(fd1,"%d",i);
    fclose (fd1) ;

    gettimeofday(&tf, NULL);
    tiempo= (tf.tv_sec - ti.tv_sec)*1000 +
            (tf.tv_usec - ti.tv_usec)/1000.0;
    printf("%g milisegundos\n", tiempo);
    return (0) ;
}
```

- Compilar (gcc -o b b.c)  
y ejecutar con
  - BSIZE=1024
  - BSIZE=0

# Fichero o archivo: interfaz C99

35

Sistemas Operativos - ARCOS

escritura

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/time.h>

#define BSIZE 1024

int main ( int argc, char *argv[] )
{
    FILE *fd1 ; int i; double tiempo ;
    char buffer1[BSIZE] ;
    struct timeval ti, tf;

    gettimeofday(&ti, NULL);
    fd1 = fopen ("/tmp/txt2", "w+");
    if (NULL == fd1) {
        printf("fopen: error\n");
        exit(-1) ;
    }
    setbuffer(fd1,buffer1,BSIZE) ;
    for (i=0; i<8*1024; i++)
        fprintf(fd1,"%d",i);
    fclose (fd1) ;

    gettimeofday(&tf, NULL);
    tiempo= (tf.tv_sec - ti.tv_sec)*1000 +
            (tf.tv_usec - ti.tv_usec)/1000.0;
    printf("%g milisegundos\n", tiempo);
    return (0) ;
}
```

□ Compilar (gcc -o b b.c)  
y ejecutar con

□ BSIZE=1024

□ BSIZE=0

□ Resultados:

□ BSIZE=1024

■ T=0.902 milisegundos

□ BSIZE=0

■ T=14.866 milisegundos

x 15

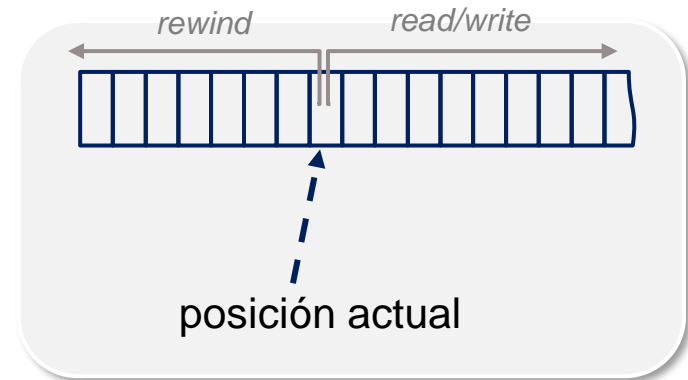
# Contenidos

- Introducción
- **Fichero**
  - ▣ Definición
  - ▣ Metadatos
  - ▣ Interfaz
  - ▣ **Métodos de acceso**
  - ▣ Semántica de compartición
- Directorio
- Sistema de ficheros
- Particiones/Volúmenes
- Dispositivos
- Software de sistema

# Fichero o archivo: método de acceso

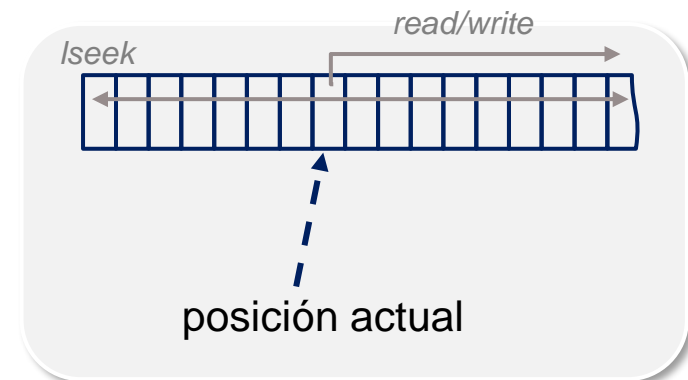
## □ Acceso secuencial:

- Dispositivos de acceso secuencial: cintas magnéticas.
- Solo es posible posicionarse (*rewind*) al principio del fichero.



## □ Acceso directo:

- Dispositivos de acceso aleatorio: discos duros.
- Es posible posicionarse (*lseek*) en cualquier posición del fichero.
  - Permite construir sobre él otros métodos de acceso (ej.: indexado)



# Contenidos

- Introducción
- **Fichero**
  - ▣ Definición
  - ▣ Metadatos
  - ▣ Interfaz
  - ▣ Métodos de acceso
  - ▣ **Semántica de compartición**
- Directorio
- Sistema de ficheros
- Particiones/Volúmenes
- Dispositivos
- Software de sistema

# Fichero o archivo: semántica de compartición

- ❑ Varios procesos pueden acceder simultáneamente a un fichero.
- ❑ Es necesario definir una semántica de coherencia:
  - ❑ ¿Cuándo son observables por otros procesos las modificaciones a un fichero?
- ❑ Opciones:
  - ❑ Semántica **UNIX**.
  - ❑ Semántica de **sesión**.
  - ❑ Semántica de **versiones**.
  - ❑ Semántica de archivos **inmutables**.

# Fichero o archivo: semántica de compartición

Semántica Unix	Semántica de sesión	Semántica de versiones	Semántica inmutable
Las <b>escrituras</b> en un archivo son <b>visibles inmediatamente</b> a todos los procesos (y el nuevo puntero de L/E)	Las <b>escrituras</b> en un archivo <b>no</b> son <b>visibles</b> por otros procesos: <b>al cerrar se hace visible</b> .	Las escrituras se hacen sobre copias con número de versión: son visibles <b>al consolidar versiones</b> .	<b>Si se declara compartido</b> un archivo, <b>no se puede modificar</b>
Una vez abierto ( <i>open</i> ), <b>la familia de procesos</b> creado ( <i>fork</i> ) <b>comparte</b> su <b>imagen</b> .	<b>Una vez cerrado</b> el fichero, <b>los siguientes</b> procesos que lo abran <b>ven las modificaciones</b> .	Usar sincronización explícita para actualizaciones inmediatas.	<b>Hasta no liberar el cerrojo</b> , ni nombre ni contenido pueden modificarse.
<b>Contención por acceso</b> exclusivo a la imagen única del fichero.	Un fichero puede estar asociado a <b>varias imágenes</b> .	Tendrá <b>varias imágenes</b> y <b>coste de consolidar</b> .	<b>No hay concurrencia</b> .
Ext3, ufs, etc.	AFS ( <i>Andrew File System</i> )	CODA	



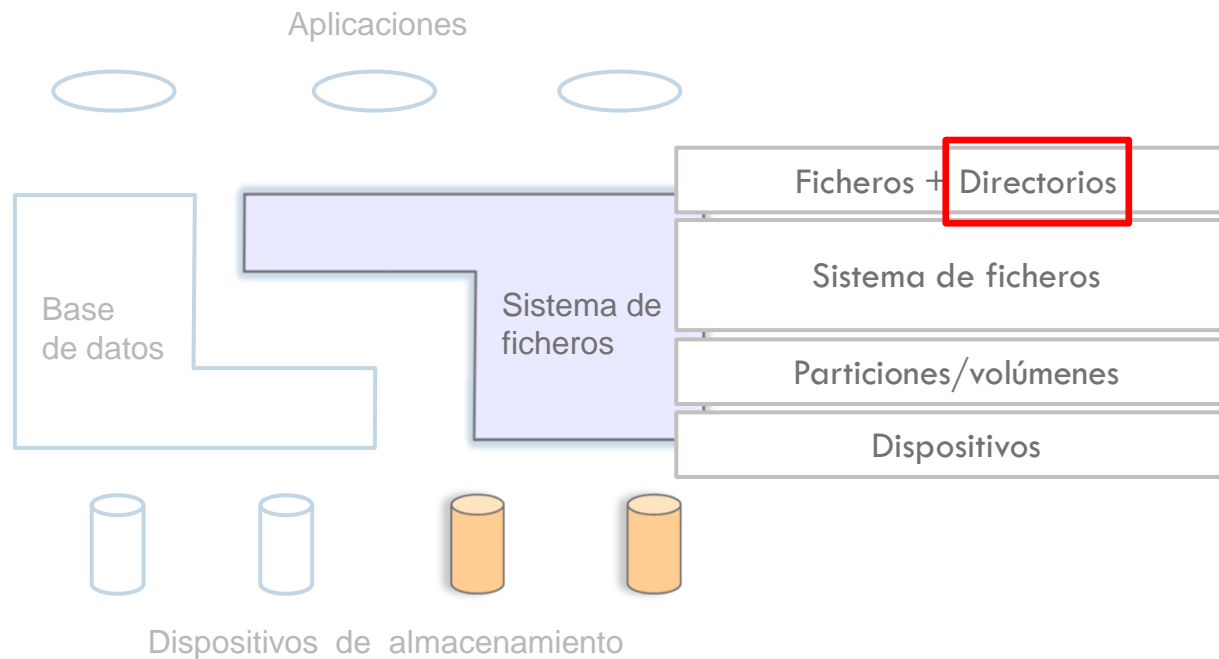
# Contenidos

- Introducción
- Fichero
- **Directorio**
  - ▣ Definición
  - ▣ Metadatos
  - ▣ Interfaz
- Sistema de ficheros
- Particiones/Volúmenes
- Dispositivos
- Software de sistema

# Directorio (carpetas)

42

Sistemas Operativos - ARCOS

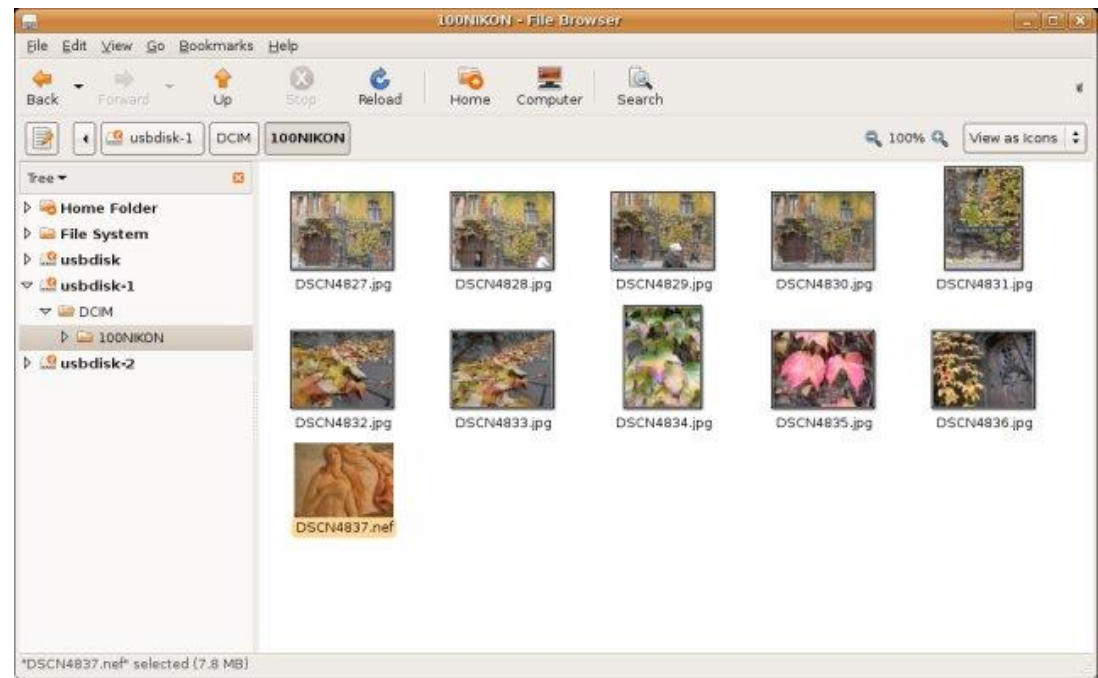
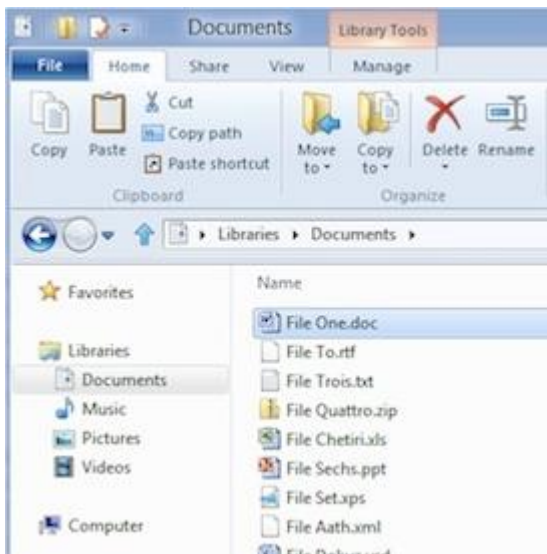


# Directorio (carpetas)

43

Sistemas Operativos - ARCOS

- Estructura de datos que permite agrupar un conjunto de ficheros según el criterio del usuario.

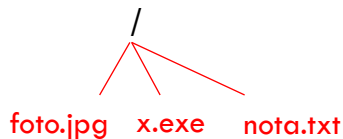


# Directorios: organización

44

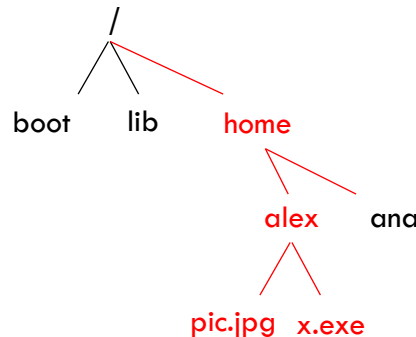
Sistemas Operativos - ARCOS

- Organizan y **proporcionan** información sobre la **estructuración** de los **sistemas de archivos**:



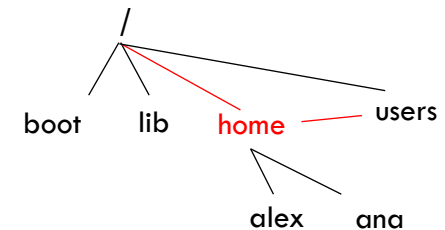
## ► De un nivel

- 1 dir con **n** **ficheros**
- 1 **fichero** con 1 dir.



## ► Jerárquico (árbol)

- 1 dir con **n** entradas
- 1 entrada con **1** dir.



## ► Árbol a-cíclico

- 1 dir. con **n** entradas
- 1 entrada con **n** dir.

# Directorios: nombres jerárquicos

45

Sistemas Operativos - ARCOS

## □ Nombres jerárquicos para la identificación.

### □ Tipo de nombrado de directorio:

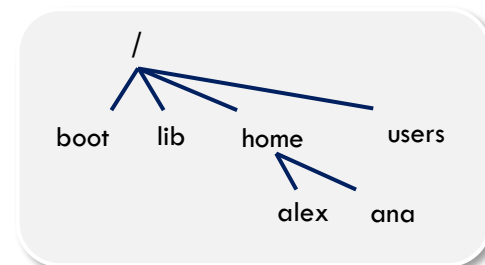
- **Nombre absoluto:** especificación del nombre respecto al directorio raíz (/ en LINUX, \ en Windows)

- **Nombre relativo:** especificación del nombre respecto a un directorio distinto del raíz.

  - Ejemplo: (estando en /users/) alex/correo.txt

  - Relativos al directorio de trabajo o actual:

basado en el directorio en el que se encuentre el usuario (directorio de trabajo)



### □ Directorios especiales:

- Directorio actual o directorio de trabajo: `.` (Ej.: `cp /alex/correo.txt .`)

- Directorio padre: `..` (Ej.: `ls ..`)

- Directorio base del usuario: `$HOME` (Ej.: `ls -las $HOME`)

# Contenidos

- Introducción
- Fichero
- **Directorio**
  - ▣ Definición
  - ▣ **Metadatos**
  - ▣ Interfaz
- Sistema de ficheros
- Particiones/Volúmenes
- Dispositivos
- Software de sistema

# Directorio (carpetas)

47

Sistemas Operativos - ARCOS

## □ Información de un directorio:

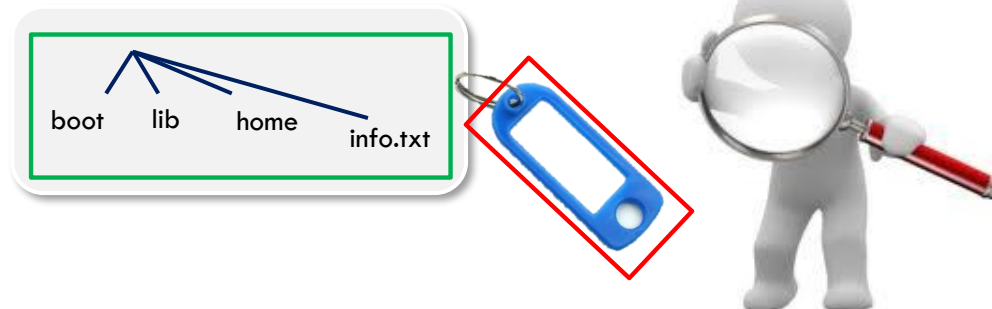
### □ Datos

fichero | directorio

- “fichero especial” cuyo contenido es un listado con los **entradas** que contiene.

### □ Metadatos

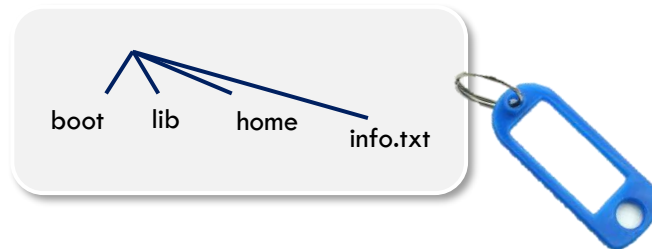
- Información sobre el directorio en sí: distintos **atributos** sobre el directorio.



# Directorios: atributos

## □ Atributos típicos de un directorio:

- **Nombre:** identificador para los usuarios del directorio.
- **Tamaño:** número de ficheros en el directorio.
- **Protección:** control de qué usuario puede leer, acceder, etc.
- **Día y hora:** instante de tiempo de último acceso, de creación, etc. que permite la monitorización del uso del directorio.
- **Identificación** de usuario: identificador del creador, etc.





# Contenidos

- Introducción
- Fichero
- **Directorio**
  - ▣ Definición
  - ▣ Metadatos
  - ▣ **Interfaz**
- Sistema de ficheros
- Particiones/Volúmenes
- Dispositivos
- Software de sistema

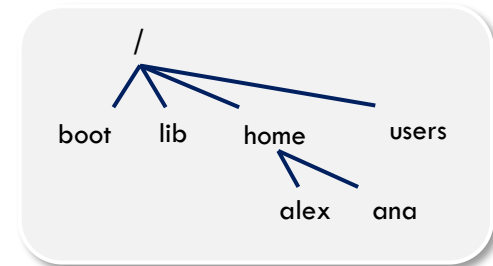
# Directorios: interfaz

50

Sistemas Operativos - ARCOS

## □ Interfaz genérica para gestión de directorios:

- `mkdir` (nombre, modo)
- `rmdir` (nombre)
- `chdir` (nombre)
- `getcwd` (nombre, tamaño\_nombre)
- `descriptor` ← `opendir` (nombre)
- `closedir` (descriptor)
- `estructura` ← `readdir` (descriptor)
- `rewindir` (descriptor)
- `unlink` (nombre)
- `rename` (antiguo\_nombre, nuevo\_nombre)



# Directorios: interfaz POSIX

51

Sistemas Operativos - ARCOS

lectura de /tmp

```
#include <unistd.h>
#include <sys/types.h>
#include <dirent.h>
#include <stdio.h>

int main ( int argc, char *argv[] )
{
    DIR *dir1 ;
    struct dirent *dp ;
    char nombre[256] ;
    int ret ;

    ret = chdir ("/tmp/") ;
    if (ret < 0) exit(-1) ;

    getcwd (nombre, 256);
    printf("%s\n", nombre);

    dir1 = opendir (nombre);
    if (NULL == dir1) exit(-1) ;
    while ( (dp = readdir(dir1)) != NULL) {
        printf("%s\n", nombre, dp->d_name);
    }
    closedir (dir1);

    return (0) ;
}
```

Cambiar de directorio de trabajo

Imprimir el directorio actual de trabajo

Abrir un directorio para trabajar con él

Leer entradas del directorio e imprimir el nombre de cada entrada

Cerrar el directorio de trabajo

# Directorios: interfaz POSIX

52

Sistemas Operativos - ARCOS

lectura de argv[1]

```
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
#include <stdio.h>

int main ( int argc, char *argv[] )
{
    DIR *dir1 ;
    struct dirent *dp ;
    struct stat s ;

    dir1 = opendir (argv[1]);
    if (NULL == dir1) {
        perror("opendir:");
        return (-1);
    }

    while ( (dp = readdir (dir1)) != NULL) {
        stat(dp->d_name,&s);
        if (S_ISDIR(s.st_mode))
            printf("dir: %s\n", dp->d_name);
        else printf("fch: %s\n", dp->d_name);
    }

    closedir (dir1);
    return (0) ;
}
```

Abrir un directorio para trabajar con él

Leer entradas del directorio...

...para cada entrada obtener los metadatos de la misma e imprimir si es fichero o directorio junto con el nombre de la entrada

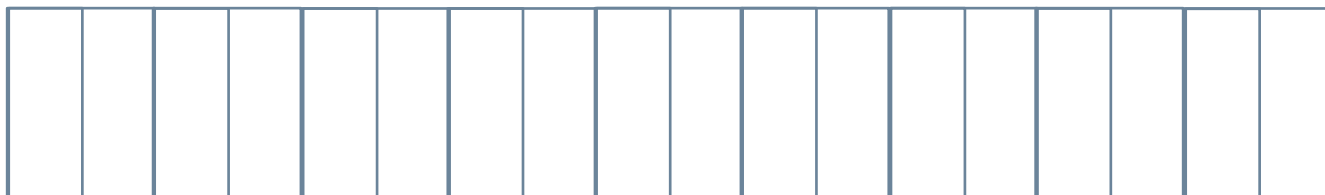
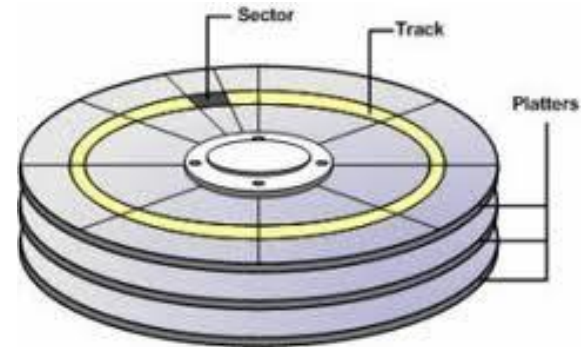
Cerrar el directorio de trabajo

# Contenidos

- Introducción
- Fichero
- Directorio
- **Sistema de ficheros**
- Particiones/Volúmenes
- Dispositivos
- Software de sistema

# Sectores

- El dispositivo de almacenamiento se divide en **sectores**, pistas y cilindros.

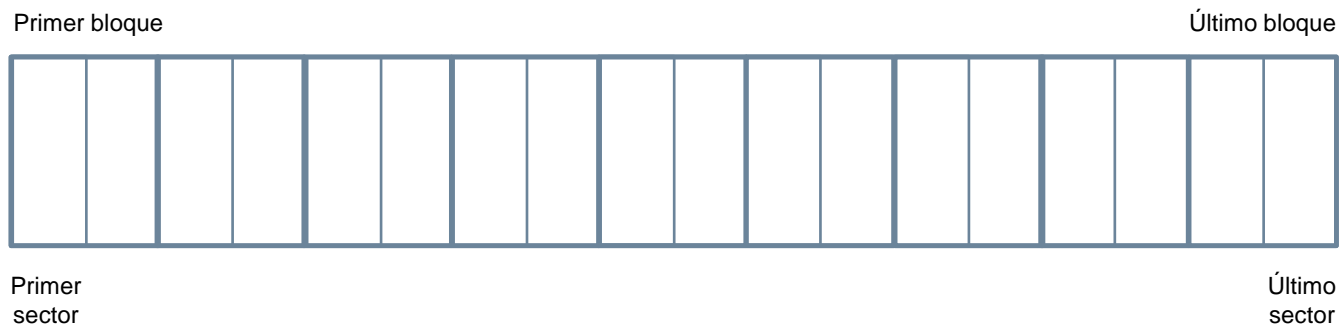


Primer  
sector

Último  
sector

# Bloques

- **Bloque:** *agrupación lógica de sectores de disco ( $2^n$  sectores)*
  - ▣ Es la unidad de transferencia mínima usado por el S.O.
  - ▣ Optimizar la eficiencia de la entrada/salida de los dispositivos.
  - ▣ Los usuarios pueden definir el tamaño de bloque al crear el sistema de ficheros, o usar el ofrecido por defecto en el S.O.

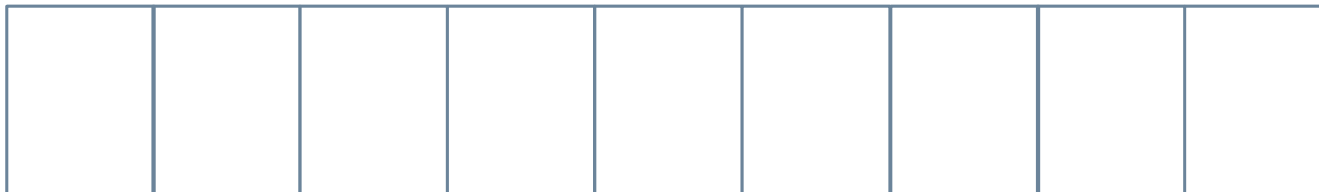


# Bloques

- **Bloque:** *agrupación lógica de sectores de disco ( $2^n$  sectores)*
  - ▣ Es la unidad de transferencia mínima usado por el S.O.
  - ▣ Optimizar la eficiencia de la entrada/salida de los dispositivos.
  - ▣ Los usuarios pueden definir el tamaño de bloque al crear el sistema de ficheros, o usar el ofrecido por defecto en el S.O.

Primer bloque

Último bloque



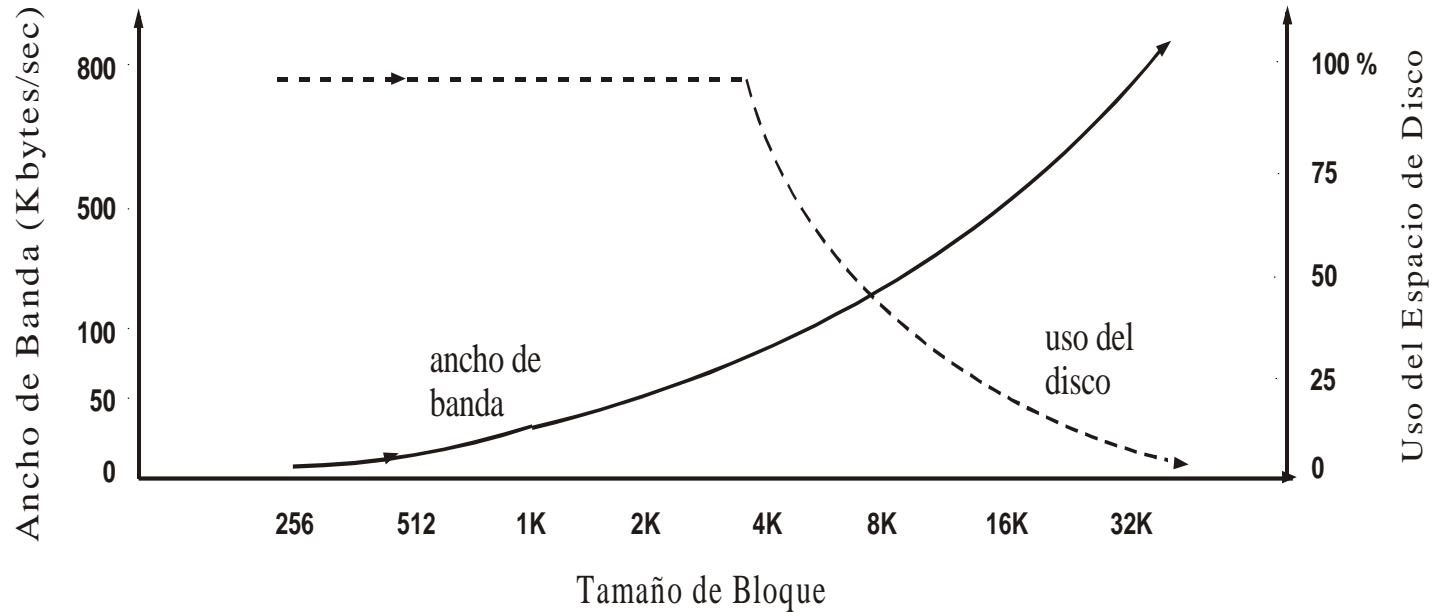


# Tamaño de bloque

Sistemas operativos: una visión aplicada

57

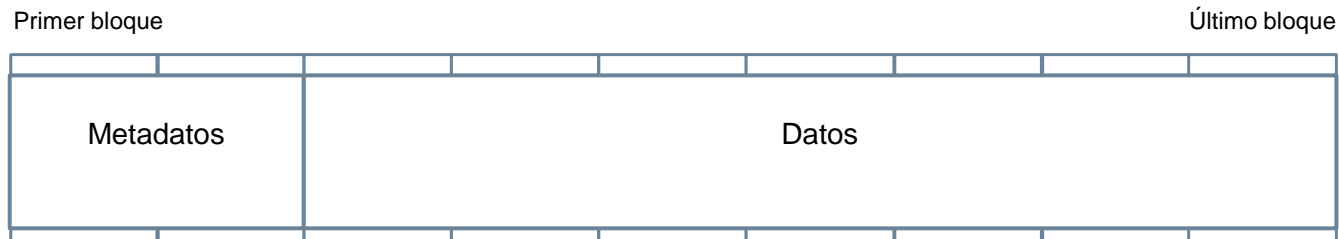
Sistemas Operativos - ARCOS



- La elección del tamaño del bloque es importante para balancear:
  - Ancho de banda: mayor número de sectores inicialmente, mejor ancho de banda
  - Uso del disco: menor número de sectores, menos fragmentación interna

# Sistema de ficheros

- El sistema de archivos permite organizar la información dentro de los dispositivos de almacenamiento en un formato inteligible para el sistema operativo:
  - ▣ Es un *conjunto coherente de metainformación y datos*.



# Sistema de ficheros: atributos

- Atributos típicos de un sistema de fichero:
  - **Tamaños usados:**
    - **Número de bloques:** cantidad de bloques gestionados (datos + metadatos)
    - **Tamaño de bloque:** tamaño del bloque (en bytes o en sectores).
    - **Número de entradas:** número de entradas (ficheros y directorios) gestionados.
    - **Tamaño de la zona de metadatos:** número de bloques dedicados.
  - **Gestión de espacio libre:** identificación de qué bloque está libre.
  - **Gestión de entradas:** para cada entrada (fichero o directorio) se reserva un espacio para los metadatos que la describe:
    - Atributos generales: fechas, permisos, identificación de usuario, etc.
    - Atributos para la gestión de ocupado: bloques usados por esta entrada.
  - **Referencia a la entrada del directorio raíz:** identificación de la entrada que contiene la información del directorio raíz.

# Sistema de ficheros: operaciones

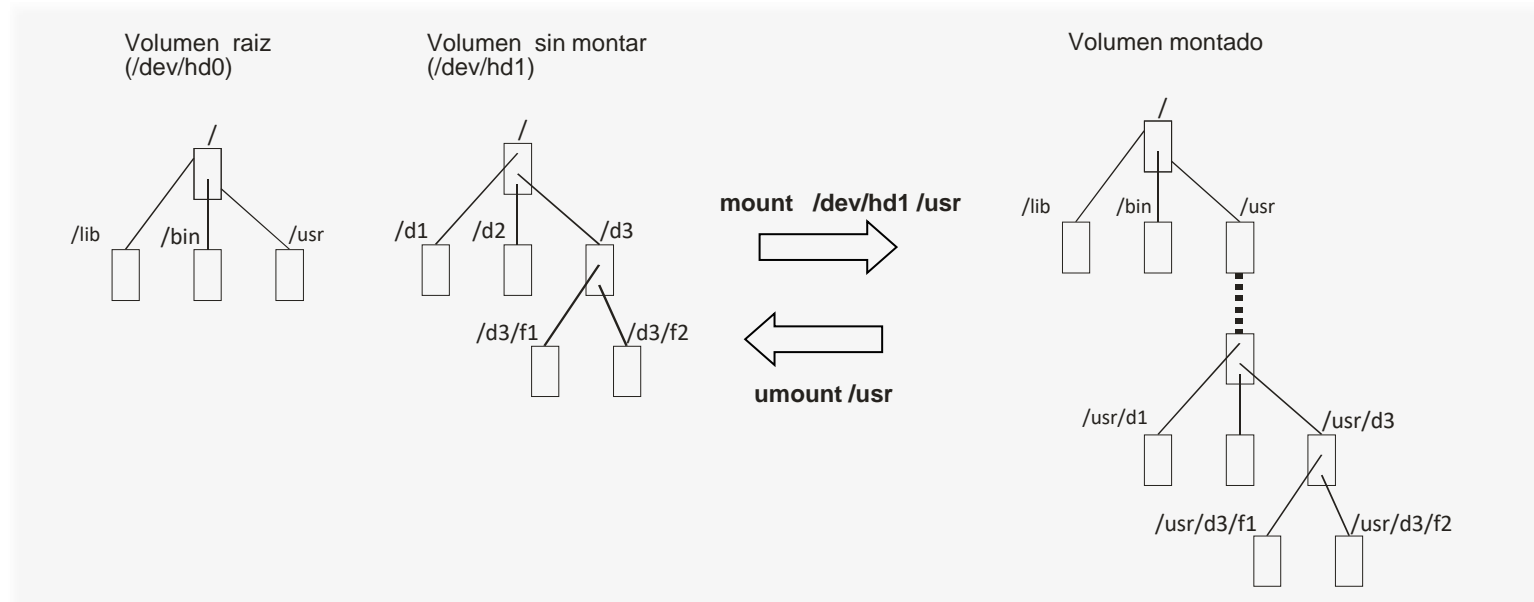
Sistemas operativos: una visión aplicada

60

Sistemas Operativos - ARCOS

## □ Operaciones con sistemas de ficheros:

- ▣ Crear
- ▣ Montar
- ▣ Desmontar



# Sistema de ficheros

[http://en.wikipedia.org/wiki/List\\_of\\_file\\_systems](http://en.wikipedia.org/wiki/List_of_file_systems)

61

Sistemas Operativos - ARCOS

## □ Gran cantidad de sistemas de ficheros.

### □ Para dispositivos de almacenamiento:

- minix (Minix)
- ext2 (Linux)
- ext3 (Linux)
- ufs (BSD)
- fat (DOS)
- vfat (win 95)
- hpfs (OS/2)
- hfs (Mac OS)
- ntfs (win NT/2K/XP)
- ...

### □ Especiales:

- procfs (/proc)
- devFS (/dev)
- umsdos (Unix sobre DOS)
- ...

### □ En red:

- NFS
- CODA
- SMBFS
- NCPFS (Novell)
- ...

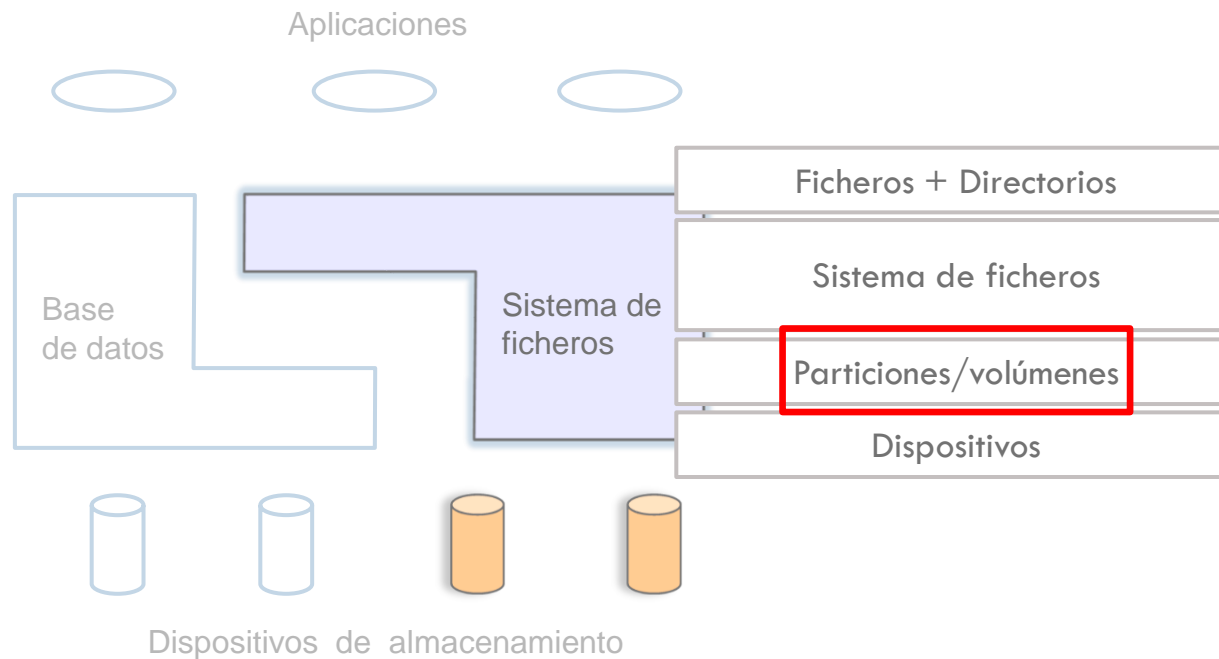
# Contenidos

- ❑ Introducción
- ❑ Fichero
- ❑ Directorio
- ❑ Sistema de ficheros
- ❑ **Particiones/Volúmenes**
- ❑ Dispositivos
- ❑ Software de sistema

# Particiones/Volúmenes

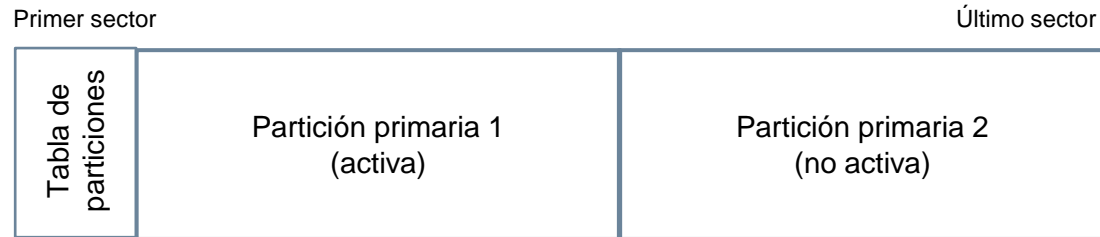
63

Sistemas Operativos - ARCOS



# Particiones

- Contenedor de un sistema de ficheros.

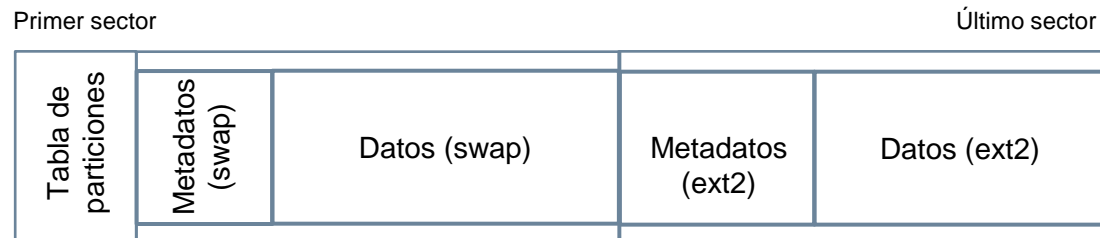


- Una **partición** es una porción de un disco a la que se la dota de una identidad propia y que puede ser manipulada por el sistema operativo como una entidad lógica independiente.



# Particiones

## ► Contenedor de un sistema de ficheros.



- Una vez creadas las particiones, el sistema operativo debe crear las estructuras de los sistemas de archivos dentro de esas particiones.

Para ello se proporcionan mandatos como `format` o `mkfs` al usuario:

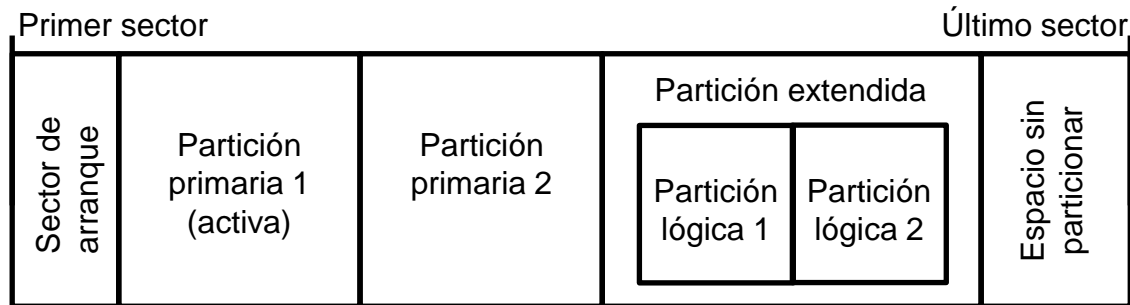
- `# mkswap -c /dev/hda1 20800`
- `# mkfs -c /dev/hda2 -b 8196 123100`

# Particiones

- Atributos típicos de una partición:
  - ▣ **Tipo**: primaria, secundaria, unidad lógica, con arranque, etc..
  - ▣ **Tamaño**: inicio y fin de partición.
  - ▣ **Sistema albergado**: linux, linux swap, vfat, etc.
  - ▣ **Identificación**: número de partición (orden o UUID).

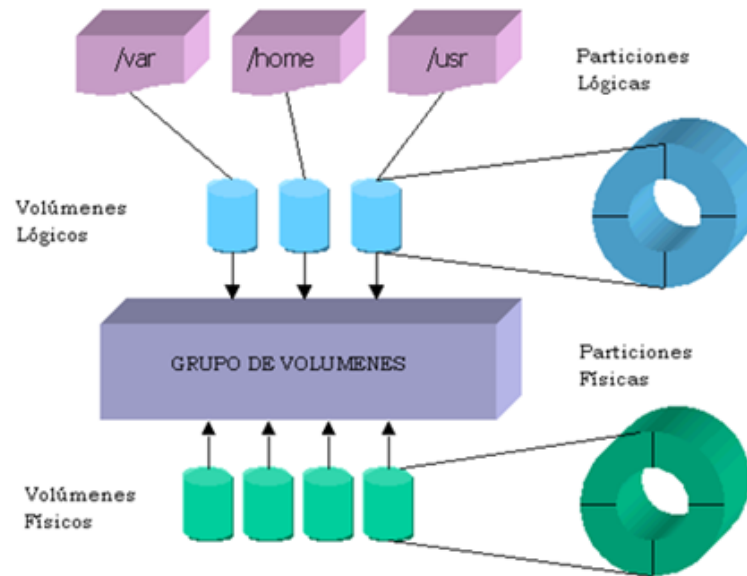


# Particiones: particionado tradicional en PC



- ❑ Sector de arranque contiene la tabla de particiones
- ❑ Partición primaria o secundaria (con unidades lógicas)
- ❑ Antiguo y limitado:
  - 4 particiones en total (primarias + secundarias)
  - No es posible cambiar el tamaño sin perder los datos

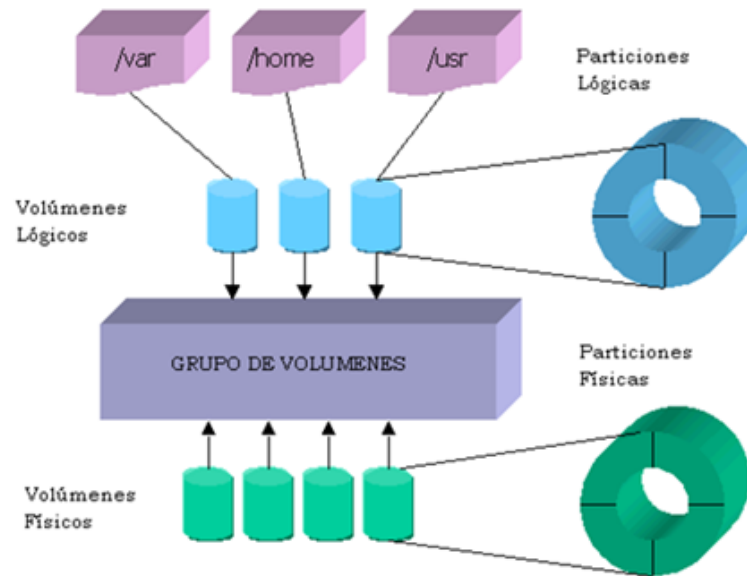
# Volúmenes



[http://www.howtoforge.com/linux\\_lvm](http://www.howtoforge.com/linux_lvm)

- ▣ Volúmenes lógicos (~antiguas particiones) sobre grupo de volúmenes compuestos de volúmenes físicos.
- ▣ Moderno y flexible:
  - Sin límite, cambio dinámico, uso de múltiples discos, etc.

# Volúmenes



## ■ Crear un volumen físico, un grupo de volúmenes y uno lógico:

1. `# pvcreate /dev/sdb1`
2. `# vgcreate vol_infoso /dev/sdb1`
3. `# lvcreate -L100M -nweb vol_infoso`
4. `# mkfs -t ext3 /dev/vol_infoso/web`
5. `# mount /dev/vol_infoso/web /mnt`

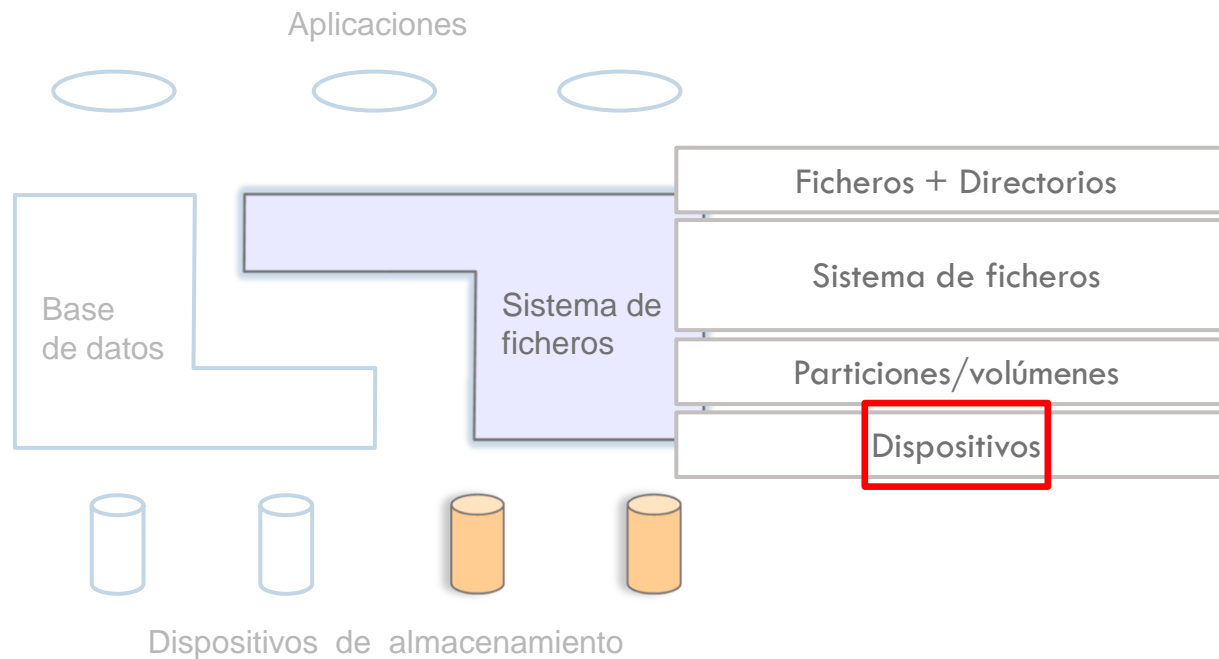
# Contenidos

- Introducción
- Fichero
- Directorio
- Sistema de ficheros
- Particiones/Volúmenes
- **Dispositivos**
- Software de sistema

# Dispositivos

71

Sistemas Operativos - ARCOS



# Dispositivos

reales

72

Sistemas Operativos - ARCOS

- Disco duro



- SSD (estado sólido)



- Sistemas ópticos



- Etc.



# Dispositivos

reales



73

Sistemas Operativos - ARCOS

## □ Listar los dispositivos PCI:

```
acaldero@phoenix:~/infodso/$ lspci
```

```
00:00.0 Host bridge: Intel Corporation 82Q35 Express DRAM Controller (rev 02)
00:01.0 PCI bridge: Intel Corporation 82Q35 Express PCI Express Root Port (rev 02)
00:03.0 Communication controller: Intel Corporation 82Q35 Express MEI Controller (rev 02)
00:03.2 IDE interface: Intel Corporation 82Q35 Express PT IDER Controller (rev 02)
00:03.3 Serial controller: Intel Corporation 82Q35 Express Serial KT Controller (rev 02)
...
```

## □ Listar los dispositivos USB:

```
acaldero@phoenix:~/infodso/$ lsusb
```

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
...
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 002: ID 1241:1166 Belkin MI-2150 Trust Mouse
Bus 005 Device 002: ID 0c45:600d Microdia TwinkleCam USB camera
```

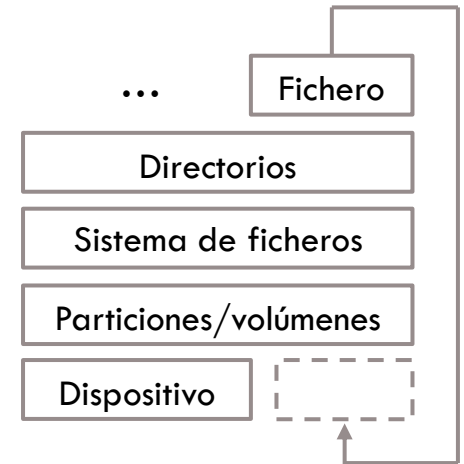
# Dispositivos

especiales



74

Sistemas Operativos - ARCOS



- Dispositivo *loopback*
  - ▣ Fichero como dispositivo de bloques

- Ejemplo de sesión de trabajo:

1. [1] Usar una imagen de CD-ROM/DVD:  
`wget ftp://ftp.rediris.es/sites/releases.ubuntu.com/releases/11.10/ubuntu-11.10-desktop-i386.iso`
2. Asociar el fichero al dispositivo de *loopback*:  
`sudo losetup /dev/loop1 /tmp/ubuntu-11.10-desktop-i386.iso`
3. Montar como dispositivo de bloques (disco):  
`mount /dev/loop1 /mnt`
4. Usar el sistema de ficheros de /mnt
5. Desmontar el dispositivo:  
`umount /dev/loop1`
6. Desasociar el dispositivo:  
`losetup -d /dev/loop1`

# Dispositivos

especiales



75

Sistemas Operativos - ARCOS

## □ Dispositivo *md*

### □ Dispositivo de dispositivos

## □ Ejemplo de sesión de trabajo:

1. [1] Crear el dispositivo md espejo:

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/loop1 /dev/loop2
```

2. [1] Crear el sistema de ficheros:

```
mkfs -t ext3 /dev/md0
```

3. Montar y desmontar el dispositivo:

```
mount /dev/md0 /mnt
```

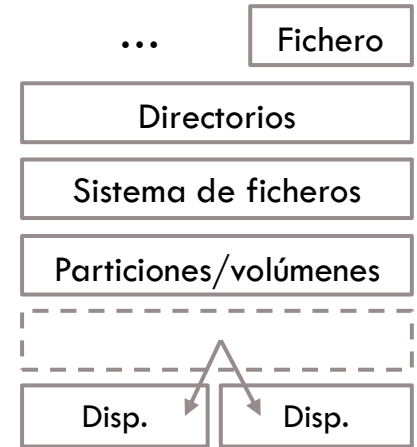
```
umount /dev/md0
```

4. Parar el dispositivo md:

```
mdadm --stop /dev/md0
```

5. Arrancar el dispositivo md:

```
mdadm --assemble /dev/md0 /dev/loop1 /dev/loop2
```



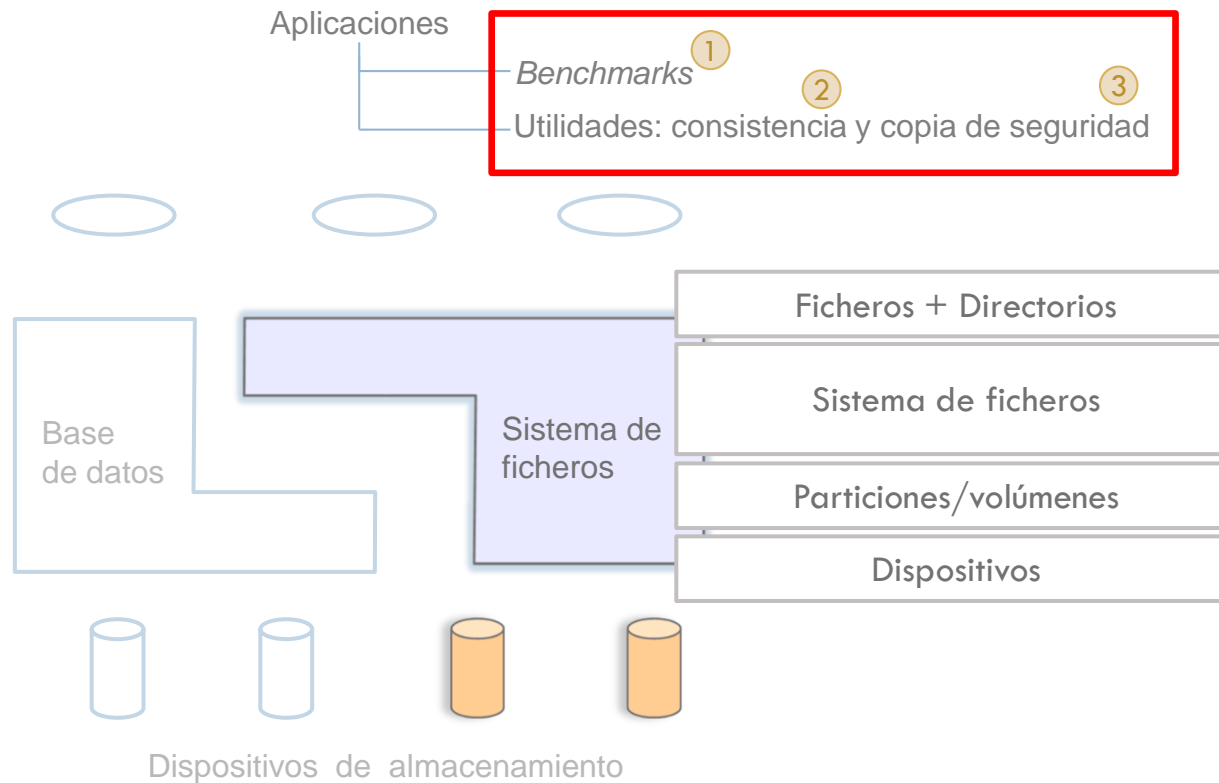
# Contenidos

- ❑ Introducción
- ❑ Fichero
- ❑ Directorio
- ❑ Sistema de ficheros
- ❑ Particiones/Volúmenes
- ❑ Dispositivos
- ❑ **Software de sistema**

# Software de sistema

77

Sistemas Operativos - ARCOS



# Benchmarks<sup>1</sup>

<http://linuxgazette.net/122/piszc.html>

78

Sistemas Operativos - ARCOS

## □ *Benchmarks:*

- Permiten medir las **prestaciones del sistema de ficheros** (y toda dependencia del mismo)
- Diseñados para medir **diferentes aspectos:** latencia, ancho de banda, número de ficheros procesados por unidad de tiempo, etc.
- Ejemplos trabajando con metadatos: fdtree, mdtest, etc.
- Ejemplos trabajando con datos: iozone, postmark, IOR, etc.

# Consistencia<sup>2</sup> del sistema de archivos

- Fallos en software pueden que la información (y metadatos) quede inconsistente.
- Solución:
  - ▣ Disponer de herramientas que revisen el sistema de archivos y permita reparar los errores encontrados.
- Dos aspectos importantes a revisar:
  - ▣ Comprobar que la **estructura física** del sistema de archivos es coherente.
  - ▣ Verificar que la **estructura lógica** del sistema de archivos es correcta.

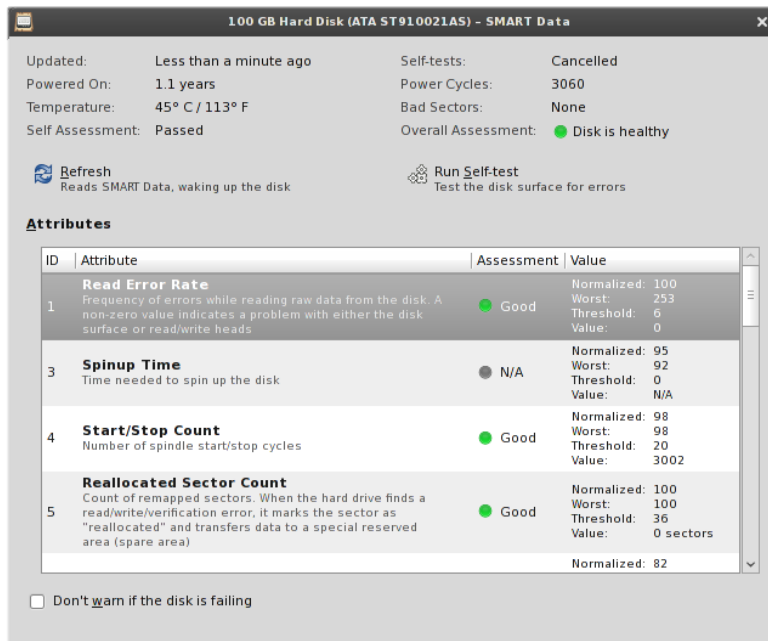
# Consistencia<sup>2</sup> del sistema de archivos

## estructura física

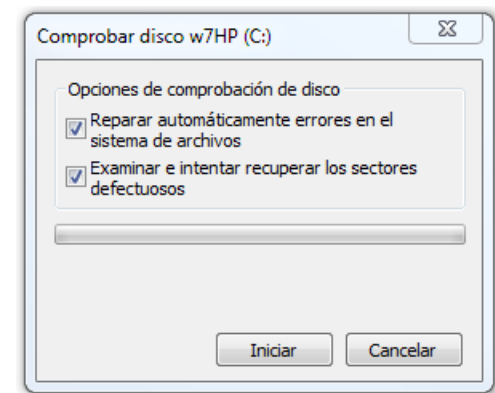
80

Sistemas Operativos - ARCOS

- ❑ Lógica del controlador:
  - ❑ Se realizan pruebas del estado del disco-controlador
  - ❑ Ej.: S.M.A.R.T.



- ❑ Superficie del disco:
  - ❑ Se lee/escribe los bloques de disco uno a uno para comprobar problemas en la superficie de parte del disco.
  - ❑ Ej.: si lo leído es diferente a lo escrito





# Consistencia<sup>2</sup> del sistema de archivos

## estructura lógica

81

Sistemas Operativos - ARCOS

- Estructuras en disco:
  - ▣ Comprobar que los metadatos en disco son coherentes para la partición, directorios y archivos almacenados
  - ▣ Ej.: fsck en Linux o scandisk en Windows

```
acaldero@phoenix:/tmp$ sudo fsck -f /dev/loop1
fsck desde util-linux-ng 2.17.2
e2fsck 1.41.12 (17-May-2010)
Paso 1: Verificando nodos-i, bloques y tamaños
Paso 2: Verificando la estructura de directorios
Paso 3: Revisando la conectividad de directorios
Paso 4: Revisando las cuentas de referencia
Paso 5: Revisando el resumen de información de grupos
/dev/loop1: 11/28560 ficheros (0.0% no contiguos), 5161/114180 bloques
acaldero@phoenix:/tmp$
```

# Backup<sup>3</sup> (copia de seguridad)

## ¿Dónde?

82

Sistemas Operativos - ARCOS

### □ Lugar:

- Distante del sistema principal
- Protegido del agua, fuego, etc.
  - Armarios ignífugos



### □ Medio:

- Disco duro
  - V: capacidad y precio, l: delicado
- Cinta
  - V: capacidad y precio, l: lentitud



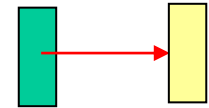
# Backup<sup>3</sup> (copia de seguridad)

## ¿Cómo?

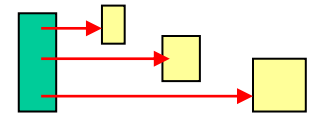
83

Sistemas Operativos - ARCOS

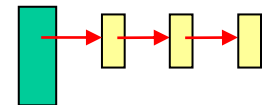
- **Completo** (*full backup*):  
copiar todo el contenido del sistema de ficheros.



- **Diferencial** (*differential backup*):  
contiene todos los ficheros que han sido modificados desde la última copia de seguridad **completa**.



- **Incremental** (*incremental backup*):  
contiene todos los ficheros que han sido modificados desde la última copia de seguridad, ya sea **completa** o **diferencial**



# Backup<sup>3</sup> (copia de seguridad)

## ¿Cuándo?

84

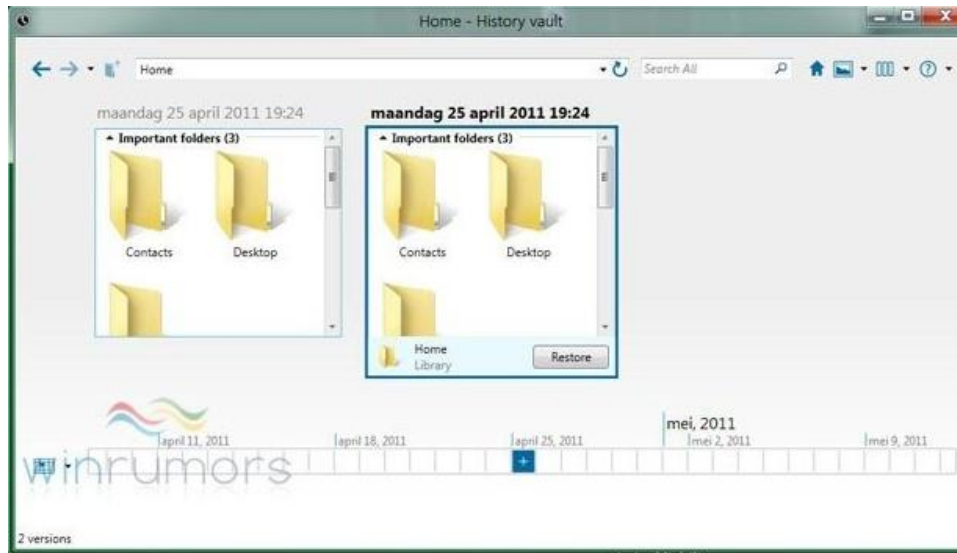
Sistemas Operativos - ARCOS

- **En parada (Off-line)**
  - ▣ La copia de seguridad se realiza en los periodos de tiempo en los que no se utilizan los datos del sistema.
- **En línea (On-line):**
  - ▣ La copia de seguridad se realiza mientras se utiliza el sistema.
  - ▣ Uso de técnicas que eviten problemas de consistencia:
    - *Snapshots*  
copia solo lectura del estado del sistema de ficheros.
    - *Copy-on-write*  
escrituras después de *snapshot* se realizan en copias.

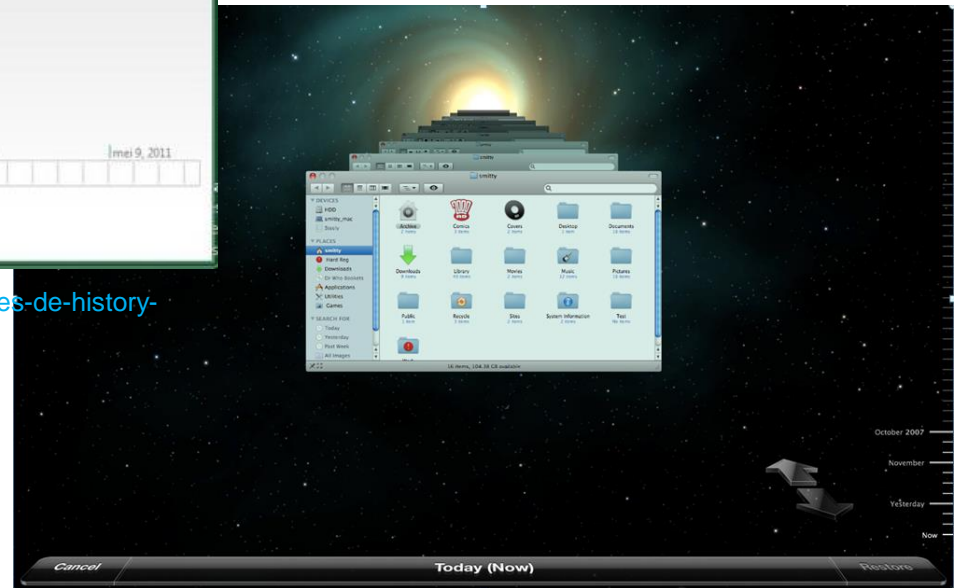
# Backup<sup>3</sup> (copia de seguridad)

85

Sistemas Operativos - ARCOS



<http://www.genbeta.com/sistemas-operativos/primeras-imagenes-de-history-vault-el-time-machine-de-windows-8>



[http://www.reghardware.com/2007/11/08/review\\_leopard\\_pt2/page2.html](http://www.reghardware.com/2007/11/08/review_leopard_pt2/page2.html)

# Lectura recomendada

## Básica

- Carretero 2007:
  - ▣ 9.1. Visión de usuario del sistema de ficheros.
  - ▣ 9.2. Ficheros.
  - ▣ 9.5. Ficheros compartidos.
  - ▣ 9.8. Estructura y almacenamiento del fichero.

## Complementaria

- Stallings 2005:
  - ▣ 12.1. Descripción básica.
  - ▣ 12.2. Organización y acceso a los ficheros.
  - ▣ 12.4. Compartición de ficheros.
  - ▣ 12.6. Gestión de almacenamiento secundario.
- Silberschatz 2006:
  - ▣ 13. Sistemas de entrada/salida.

# Lectura recomendada

## Básica

- Carretero 2007:
  - ▣ 9.3. Directorios
  - ▣ 9.4. Nombre jerárquicos.
  - ▣ 9.9. Estructura y almacenamiento del directorio.
  - ▣ 9.10. El servidor de directorios.

## Complementaria

- Stallings 2005:
  - ▣ 12.3 Directorios.
- Silberschatz 2006:
  - ▣ 10.3 Estructura de directorios.
  - ▣ 10.4 Montaje de sistemas de archivos.
  - ▣ 11.3. Implementación de directorios.

# Lectura recomendada

## Básica

- Carretero 2007:
  - 9.9. Estructura y almacenamiento del fichero.
  - 9.10. Sistemas de ficheros.
  - 9.12. El servidor de ficheros.

## Complementaria

- Stallings 2005:
  - 12.6. Gestión de almacenamiento secundario.
  - 12.7. Gestión de ficheros en UNIX.
  - 12.8. Gestión de ficheros en Linux.
- Silberschatz 2006:
  - 11.1 Estructura de un sistema de archivos.
  - 11.2 Implementación de un sistema de archivos.
  - 11.3. Implementación de directorios.
  - 11.4. Métodos de asignación.
  - 11.5. Gestión del espacio libre.
  - 11.6. Eficiencia y prestaciones.