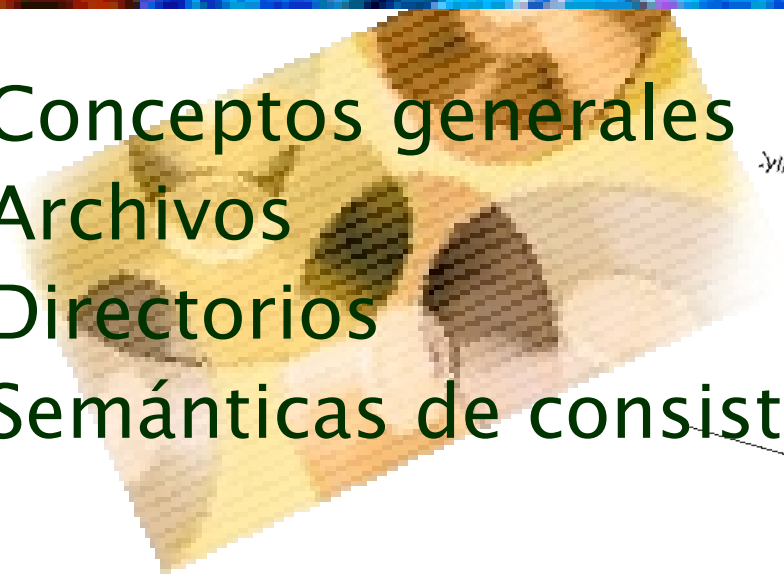
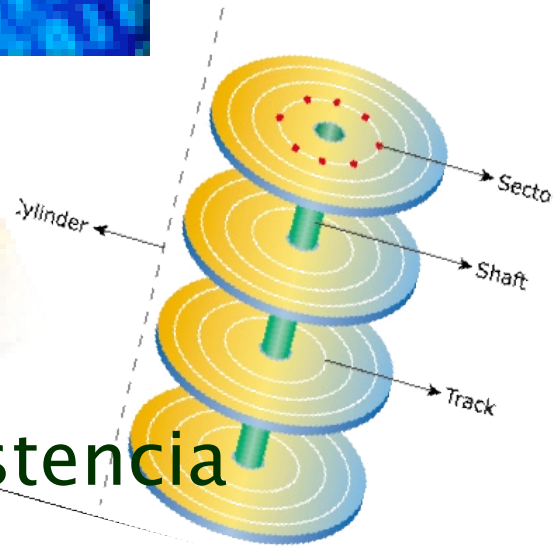
6

Sistemas de Archivos





- Conceptos generales
- Archivos
- Directorios
- Semánticas de consistencia

Persistencia de datos

- Los sistemas de archivos implementan el almacenamiento persistente – los datos creados por un programa persisten a la finalización del mismo.
- Los sistemas de archivos suministran una versión mejorada de los dispositivos del almacenamiento (discos, cintas, CD-ROM …)
- Existen sistemas de computación especiales que no tienen, o no necesitan, sistema de archivos. P. ej. un SO de un satélite, o en un microondas.

Abstracciones

- El SO suele implementar cuatro abstracciones básicas relativas al almacenamiento permanente:
  - Archivo
  - Directorio
  - Descriptor de archivo
  - Sistema de archivos
- En este tema, vamos a definir cada una de ellas, y como se implementan de forma genérica.

Concepto de archivo

- Un archivo es una colección de información relacionada con nombre que se guarda en almacenamiento secundario.
- Podemos verlos como un espacio de direcciones lógicas contiguas.
- Un archivo puede:
  - Tener cierta estructura interna, p.ej, registros, campos, etc.
  - No tener estructura. En cuyo caso, si es necesaria, se simula por el SO o la aplicación.

Funciones de la gestión de archivos

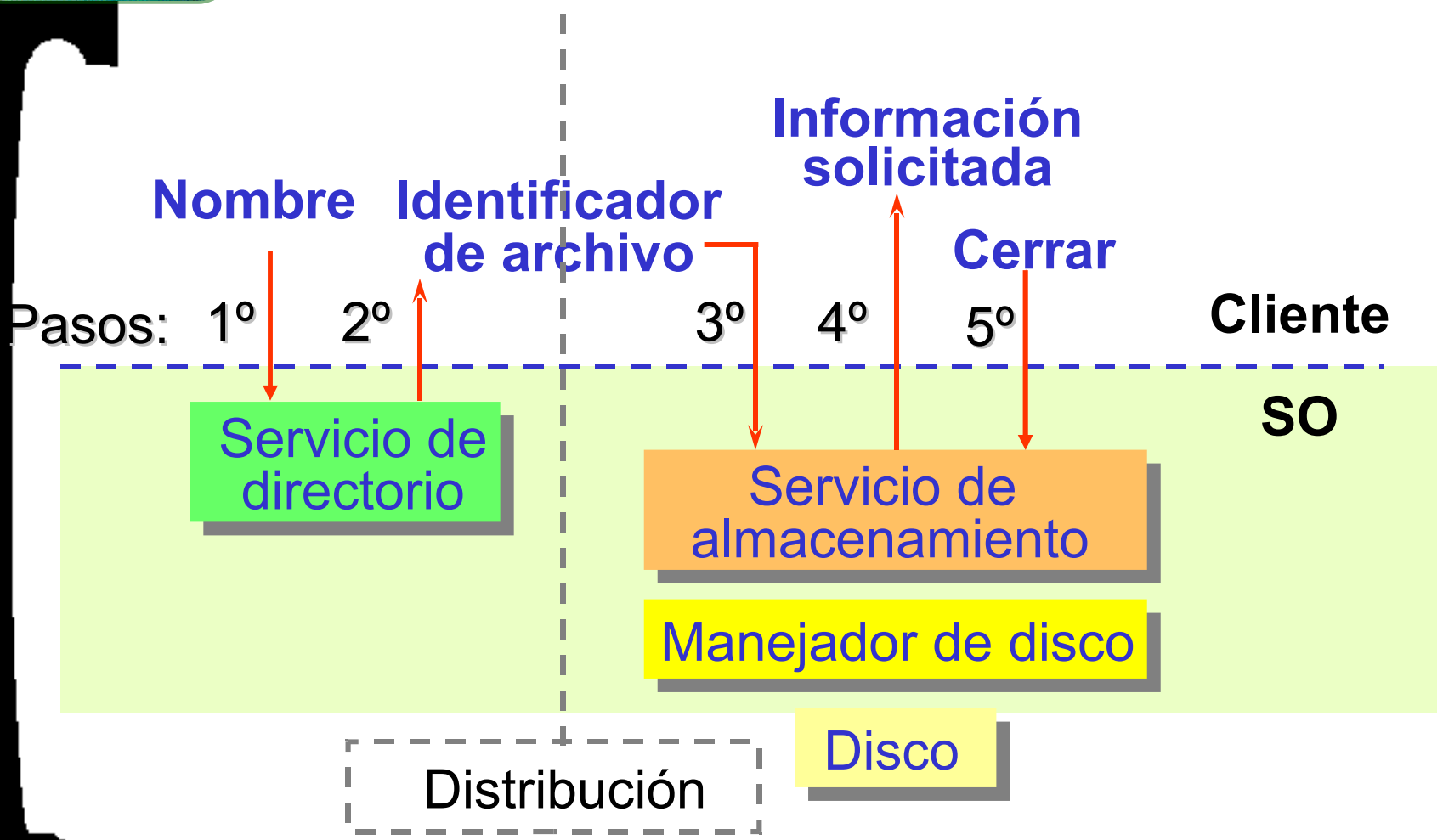
- Gestión de disco – cómo organizar bloques de disco en archivos.
- Designación (*naming*) – nombres de archivos dados por el usuario.
- Protección – mantener segura la información.
- Fiabilidad/durabilidad – cuando cae el sistema, se mantiene información en disco.
- Control de concurrencia o bloqueo de archivos– accesos concurrentes al mismo archivo.

Servicios de archivos

Podemos agrupar las funciones anteriores:

- Servicio de almacenamiento permanente – la abstracción archivo actúa como un contenedor de datos. Permite al usuario crear, borrar, …, archivos y controlar la compartición y el acceso.
- Servicio de directorio (gestión de datos) – la abstracción sistema de archivos permite organizar, manipular, y acceder a diferentes archivos. Suministra un sistema de designación lógico.

Solicitud de servicio



Pasos de la solicitud

- Solicitud de apertura dado su nombre. El SO realiza un control de acceso (el cliente esta autorizado a usarlo), y la traducción de nombre de usuario al nombre en el SO.
- El SO devuelve un identificador de archivo, descriptor, para manipulaciones posteriores.
- El usuario lo manipula con las operaciones *read*, *write*, *lseek*, *close*, etc., pasando como argumento el descriptor de archivo.
- El sistema suministra la información deseada

5. Cerramos el archivos al terminar.

Descriptor de archivo

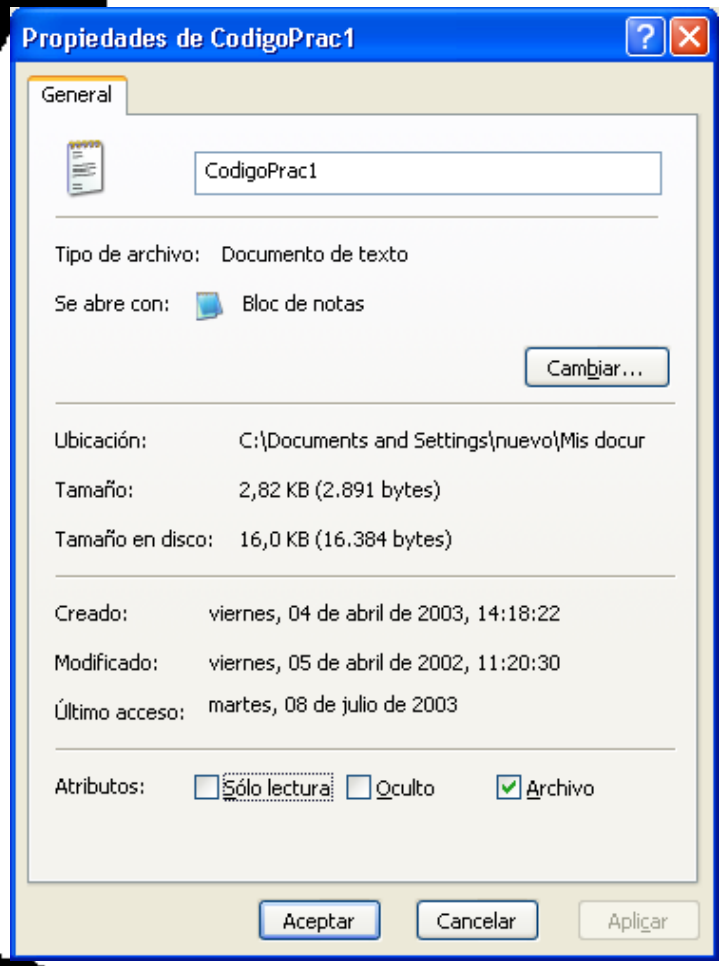
- Podemos verlo como un puntero "protegido" que nos da acceso a una sesión de trabajo sobre un archivo.
- Podemos abrir un mismo archivo de varias formas (lectura, escritura, lectura/escritura), cada sesión se aísla permitiendo el acceso a ella sólo a través del descriptor (*handle* en la terminología Microsoft).

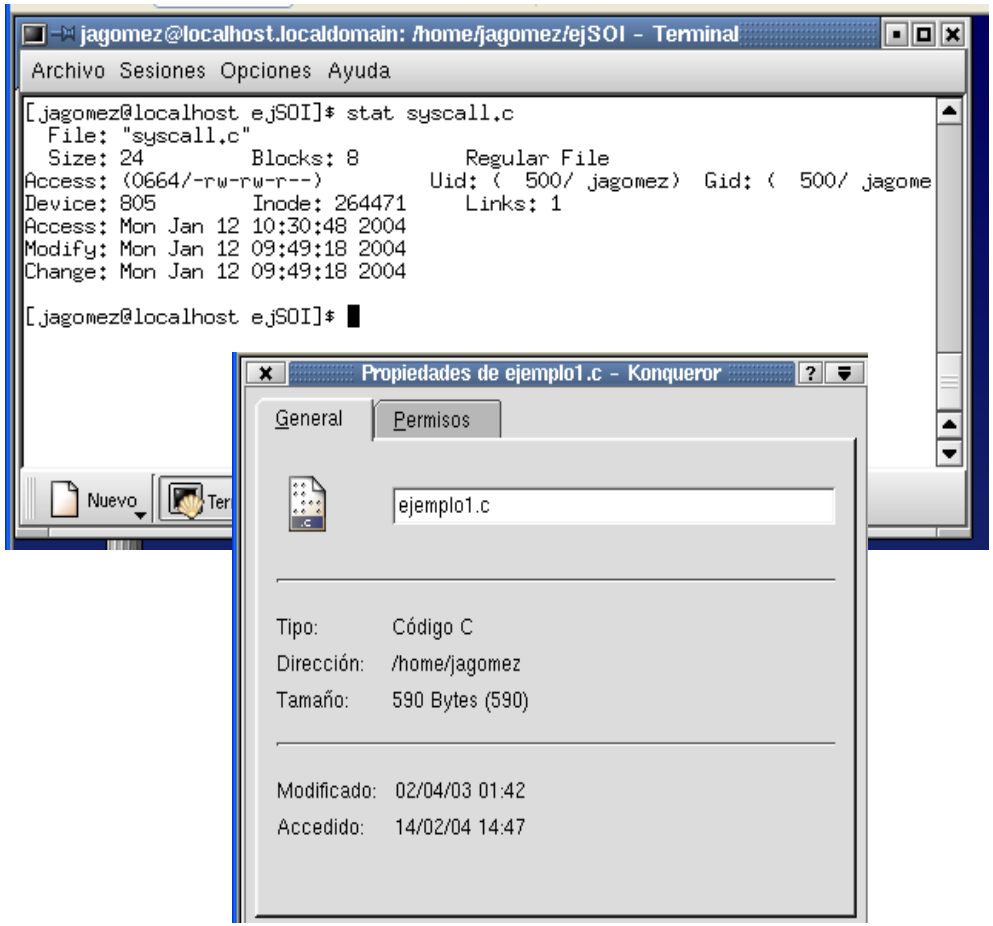
Atributos de archivo

Metadatos – información que mantiene el SO para describir el archivo. Suelen contener:

- *Nombre* – nombre de usuario
- *Tipo* – caracteriza el contenido del archivo
- *Ubicación* – su localización en el dispositivo
- *Tamaño* – tamaño actual (bytes, bloques, ...)
- *Protección* – información de control de acceso: quién y qué puede hacer con él
- *Tiempos de creación, modificación, último acceso* – por seguridad y vigilancia de uso.

Atributos: ejemplos





Archivos: operaciones y métodos de acceso

- Operaciones
  - create
  - write
  - read
  - lseek
  - delete
  - truncate
  - open
  - close
  - . . .
- Métodos de acceso
  - Secuencial
  - Aleatorio o directo
  - Indexado
  - Archivos proyectados en memoria

Los directorios

- Un directorio es un objeto que relaciona nombres de usuario de archivos con el nombre interno del archivo en el SO.
- Pueden implementarse como:
  - Archivos especiales
  - archivos normales.
- Tanto la estructura directorio como los archivos residen en disco.

Organización del directorio

La organización de directorios permite obtener

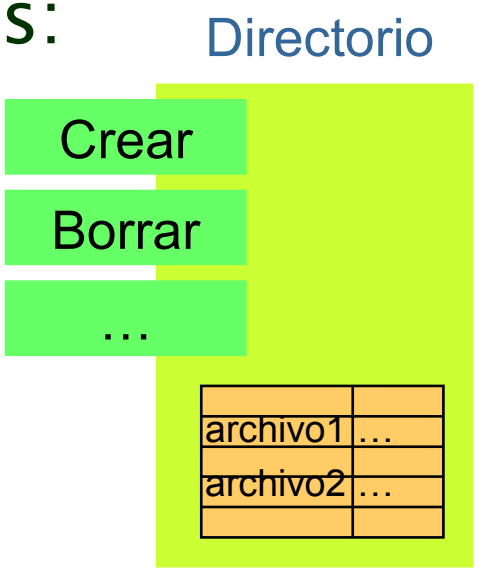
- Eficiencia – localización rápida de un archivo
- Designación – conveniente para usuarios
  - Dos usuarios pueden tener el mismo nombre para dos archivos diferentes
  - El mismo archivo puede tener varios nombres
- Agrupación – agrupar lógicamente archivos por propiedades, p. ej. programas C, juegos, etc.

Directorio de dispositivo

- Un disco se suele estructurar en particiones o volúmenes, que podemos ver como dispositivos virtuales
- Los metadatos de todos los archivos de la partición se almacena en el directorio de dispositivo o tabla de volumen.
- P. ej. en Unix esta información se almacena en el *superbloque*; en Windows NT, se almacena en el *MFT* (*Master File Table*).

Operaciones sobre directorios

- Los podemos ver como un tipo de dato abstracto con las operaciones:
  - Búsqueda de un archivo
  - Creación de un archivo
  - Borrado de un archivo
  - Lista un directorio
  - Renombrado de archivos
  - Atravesar el sistema de archivos, ...

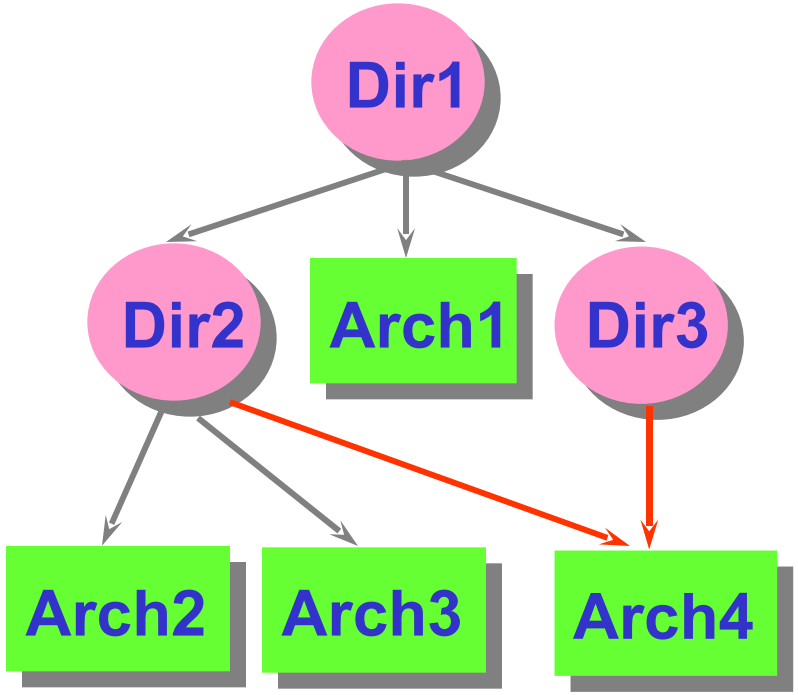


Estructuras de directorios

- Estructura de un sólo nivel – un directorio para todos los usuarios. P. ej. CP/M.
  - Problemas de designación y agrupación
- Estructura de dos niveles – un directorio separado para cada usuario
  - Búsqueda eficiente, sin capacidad de agrupar
- Estructura de árbol –
  - Búsqueda eficiente, capacidad de agrupación
  - El concepto de directorio actual permite utilizar nombres relativos.

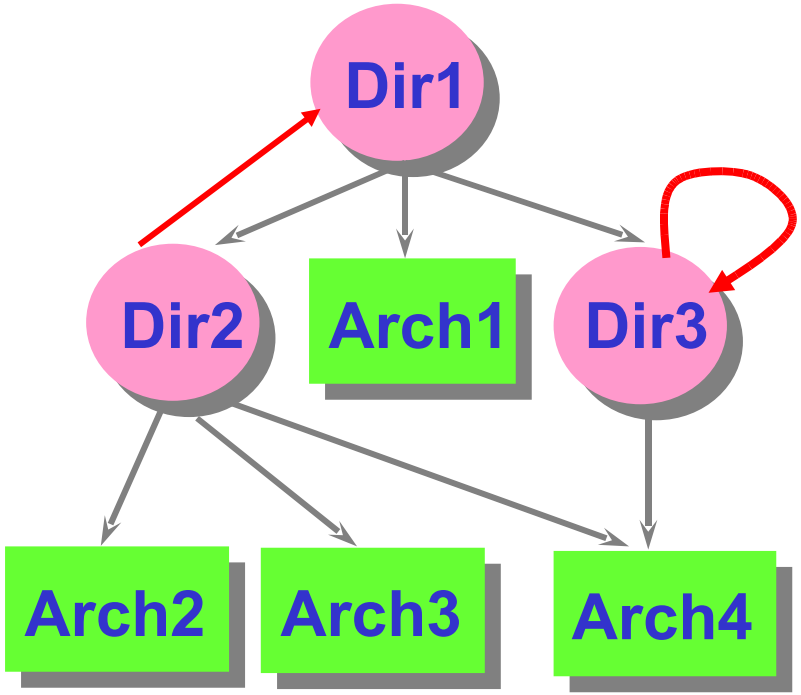
Estructura grafo acíclico

- Permite la existencia de subdirectorios y/o archivos compartidos, es decir, dos o más nombres para un mismo objeto.
- Más flexible, pero más compleja.
- La compartición se suele implementar a través de enlaces.



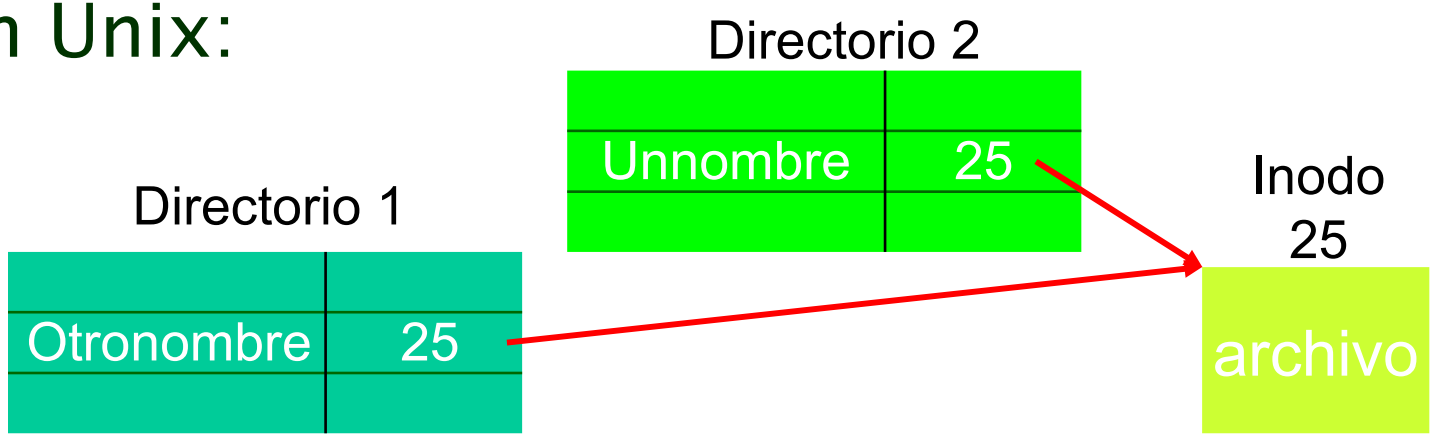
Estructura de grafo general

- Más general pero tiene problemas:
  - Podemos generar ciclos infinitos en el recorrido del mismo.
  - Debemos emplear un recolector de basura para ver cuando se puede borrar un archivo.

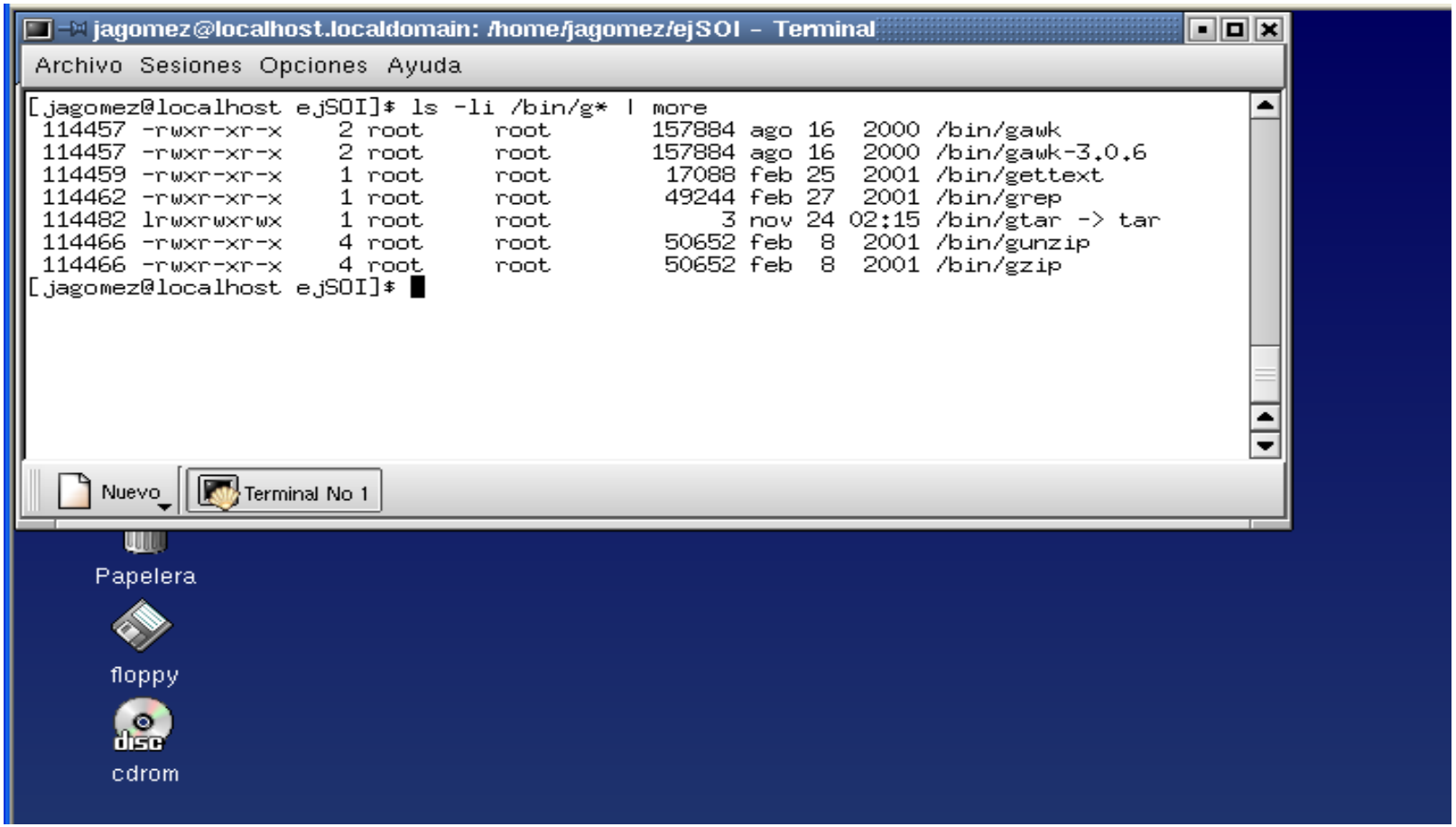


Enlaces duros

- Un enlace duro (*hard link*) es un puntero (nombre) hacia un archivo.
- Limitados a un único sistema de archivos – el puntero es único en el mismo sistema de archivos.
- En Unix:

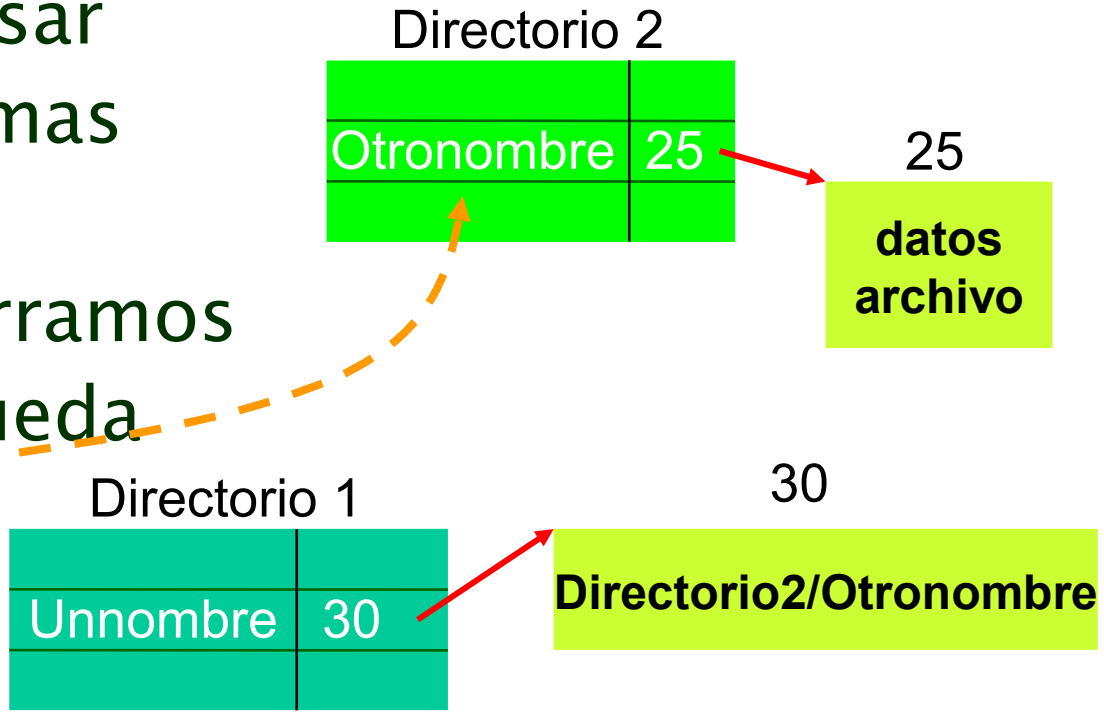


Enlace duro: ejemplo

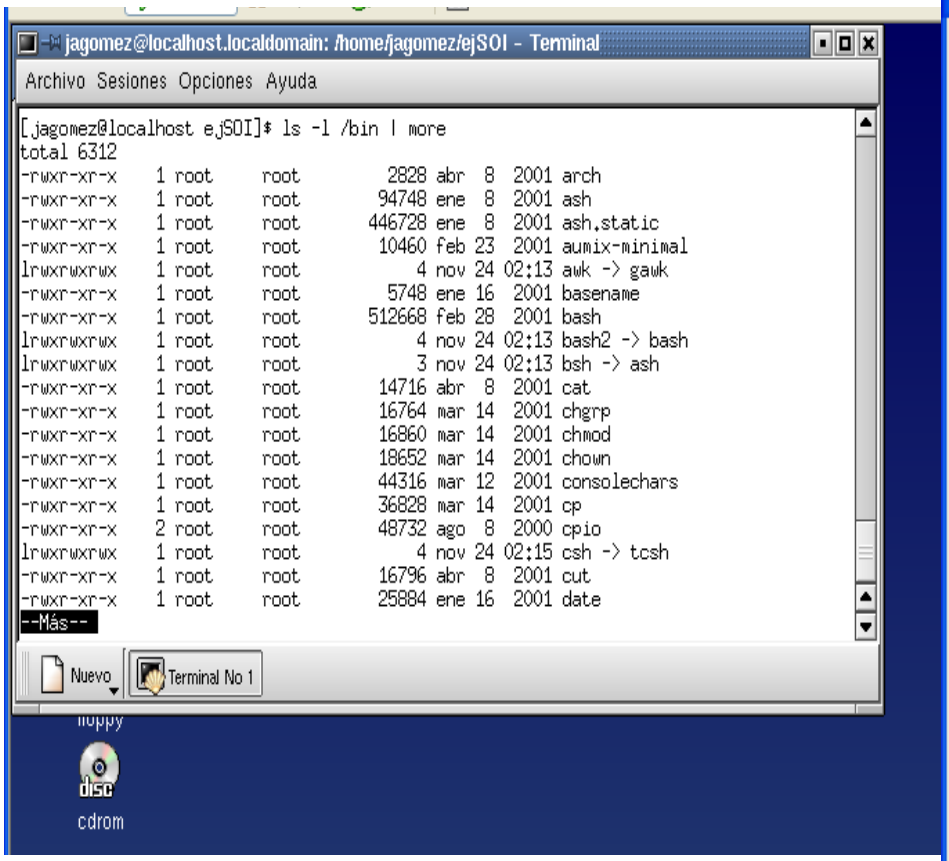


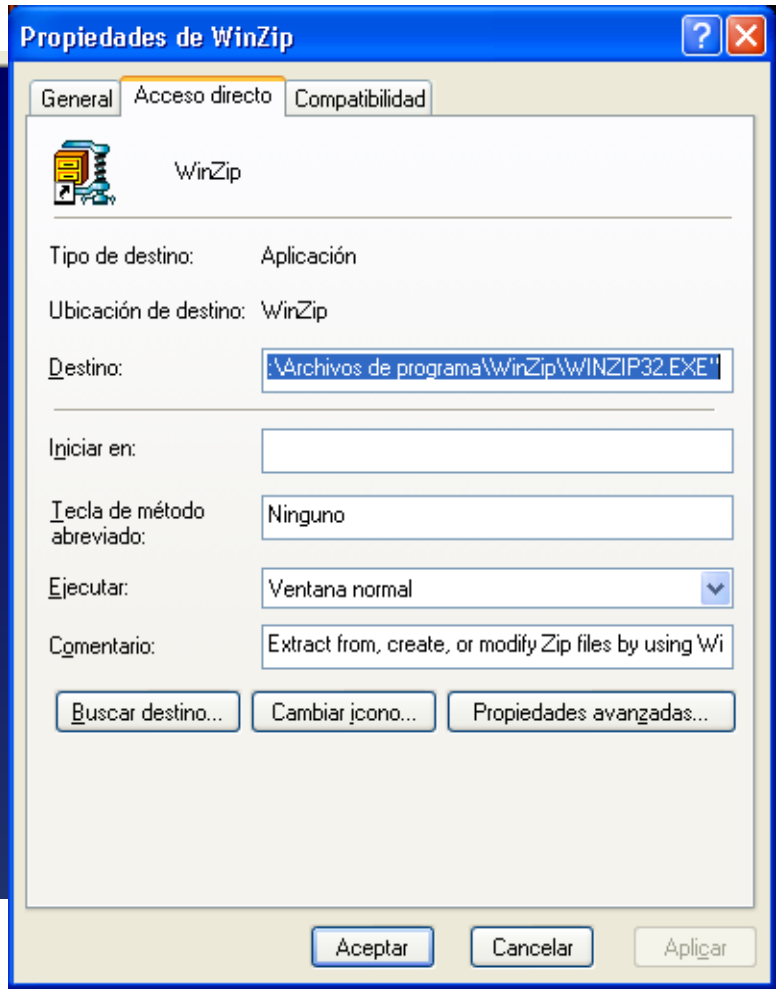
Enlaces simbólicos

- Es un archivo especial cuyo contenido es el nombre absoluto del archivo al que apunta.
- Podemos atravesar diferentes sistemas de archivos
- Problema: si borramos el puntero se queda "colgado".



Enlaces simbólicos: ejemplo





Borrado de archivos compartidos

¿ Cuando podemos desasignar el espacio asignado a estos archivos cuando se borran ?

- Para enlaces simbólicos, simplemente borrar el archivo ⇒ punteros colgados (Cualquier acceso posterior se trata como acceso a nombre ilegal)
- Asociar un contador de referencias. La creación/borrado, de un enlace incrementa/decrementa, el contador. El archivo se borra cuando contador=0. P. ej. contador enlaces duros en UNIX.

Estructura general de directorios

- La principal ventaja de un grafo acíclico es la relativa simplicidad de los algoritmos para atravesarlo y ver cuando no hay más referencias a un archivo. Por ello debemos evitar los ciclos.
- Para garantizar la no existencia de ciclos:
  - Permitir sólo enlaces a archivos, no a directorios
  - Al añadir un enlace, activar un algoritmo de detección de ciclos para determinar si todo es correcto.

Protección

- El propietario / creador de un archivo debe ser capaz de controlar:
  - qué puede hacerse con él
  - quién puede hacerlo
- Tipos de accesos:
  - lectura
  - escritura
  - ejecución
  - añadir
  - borrar
  - listar …
- La veremos en Sistemas Operativos II.

Semánticas de consistencia

Si un usuario modifica un archivo, ¿cuando es visible esta modificación por otro usuario?

- Semántica UNIX: atomicidad a nivel de operaciones *write/read*.
- Semántica de sesión: atomicidad a nivel de sesión (accesos realizados entre las operaciones *open* y *close)*. P. ej. AFS.
- Semántica de archivos compartidos inmutables: un archivo compartido, no puede ser modificado. Utilizada en sistemas distribuidos.