



# Sistemas Operativos 2

Unidad 4: Sistemas de archivos

Interfaz del sistema de archivo

René Ornelis  
Primer semestre de 2025

## Contenido

1	Introducción .....	4
2	Tipos de archivos .....	4
3	Organización de los directorios .....	4
5	Arquitectura de sistemas de archivos.....	7
5.1	Virtual File System .....	7

## Índice de figuras

Figura 1: Directorio de un nivel.....	4
Figura 2: Directorio de dos niveles.....	5
Figura 3: Árbol de directorios.....	5
Figura 4: Directorios en grafo acíclico .....	6
Figura 5: Directorios como grafo cíclico .....	6
Figura 6: Arquitectura del Virtual File System.....	7

# Interfaz del sistema de archivos

## 1 Introducción

Ya vimos la estructura de los sistemas de archivos los cuales tienen diferentes formas de organizar la información dentro del disco duro. Igual de importante que esto es la forma en que la información se presenta a los usuarios, las opciones que el sistema de archivos en conjunto con el sistema operativo nos da, para poder trabajar nuestra información. Recordemos que la información es un activo muy valioso y entre más fácil podamos manejarlo y resguardarlo, mayor será la usabilidad de nuestro sistema operativo.

## 2 Tipos de archivos

El SO debe facilitar el manejo de archivos. El sistema debe asociar acciones o aplicaciones a cada archivo, dependiendo del tipo de archivo.

En un GUI, un doble clic, significa diferente acción, para diferentes tipos de archivos. Lo mismo para un clic derecho.

Para distinguir los tipos de archivo se tienen dos estrategias básicas:

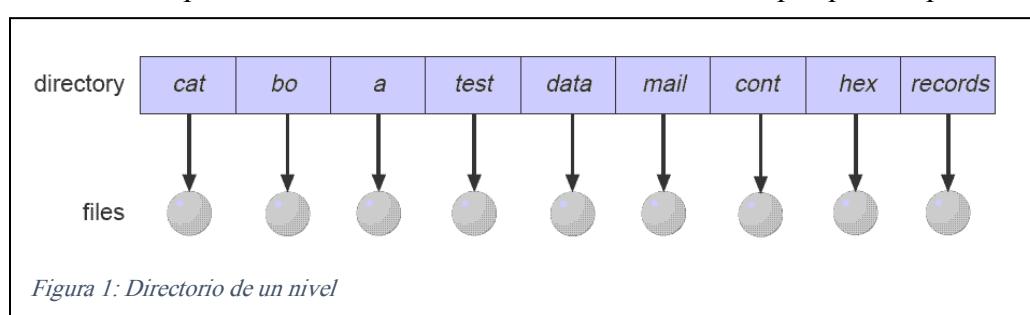
- **Extensiones:** asociación fácil para el SO y para el usuario. Es vulnerable a la decisión del usuario.
- **Firmas:** Todo tipo de archivo, tiene un inicio conocido, lo cual lo identifica, y es conocido como la firma del tipo de archivo. Da más seguridad al SO sobre el tipo de archivo y lo asociará a la aplicación adecuada, por lo que el usuario no tiene que preocuparse de las extensiones.

El uso de las extensiones se ha popularizado por la facilidad al usuario.

KDE sigue una estrategia híbrida, primero basado en la extensión, ya que es más rápido, pero en proceso de fondo, revisa las firmas de los archivos y hace las correcciones a la presentación

## 3 Organización de los directorios

Los directorios son una importante en la interfaz del sistema de archivos porque nos permite organizar la información agrupando los archivos relacionados, en vez de tener un gran conjunto de archivos.



Analicemos cómo ha evolucionado la organización de los directorios a lo largo del tiempo: En los primeros sistemas de archivos, toda la información se almacenaba en un único nivel (ver

Figura 1), es decir, existía únicamente un directorio principal que contenía todos los archivos del sistema. Este diseño era suficiente en sistemas operativos monousuarios, donde no se requería

seguridad avanzada ni un entorno más complejo. Sin embargo, con la evolución de los sistemas operativos y la introducción de funcionalidades multiusuario, este esquema de un solo nivel resultó

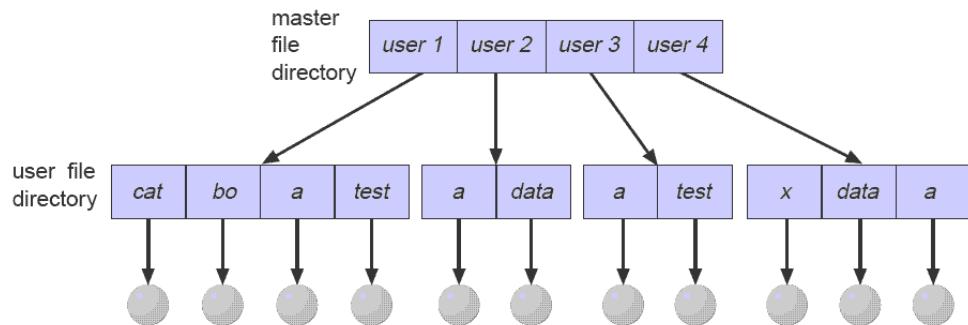


Figura 2: Directorio de dos niveles

rápidamente insuficiente. En consecuencia, se implementaron sistemas con múltiples niveles de directorios (ver Figura 2), lo que permitió una mejor organización y escalabilidad en estos entornos.

Este esquema fue más adecuado para los sistemas multiusuario en el que cada usuario veía su propio conjunto de archivos o partición, pero no existían subdirectorios para organizar la información del usuario, lo cual también rápidamente fue insuficiente dada las necesidades de los usuarios, por lo que se evolucionó a tener lo que actualmente conocemos como árbol de directorios, tal como se muestra en la Figura 3.

En un árbol de directorios, cada usuario puede organizar su información organizando sus archivos en directorios y subdirectorios. Sin embargo, se tiene la dificultad de compartir los archivos de un usuario con otro usuario, sin dar acceso a su directorio personal.

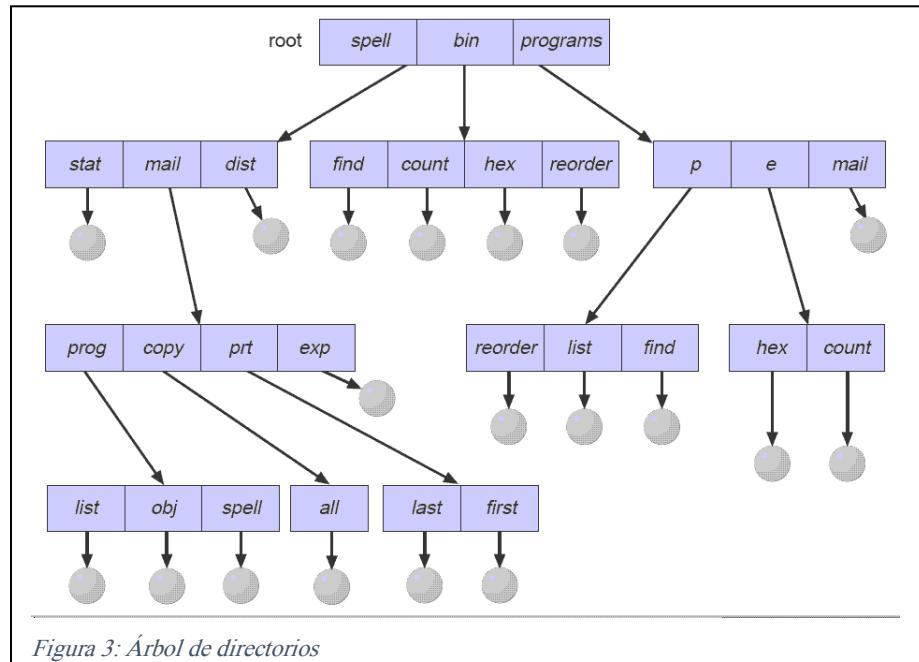
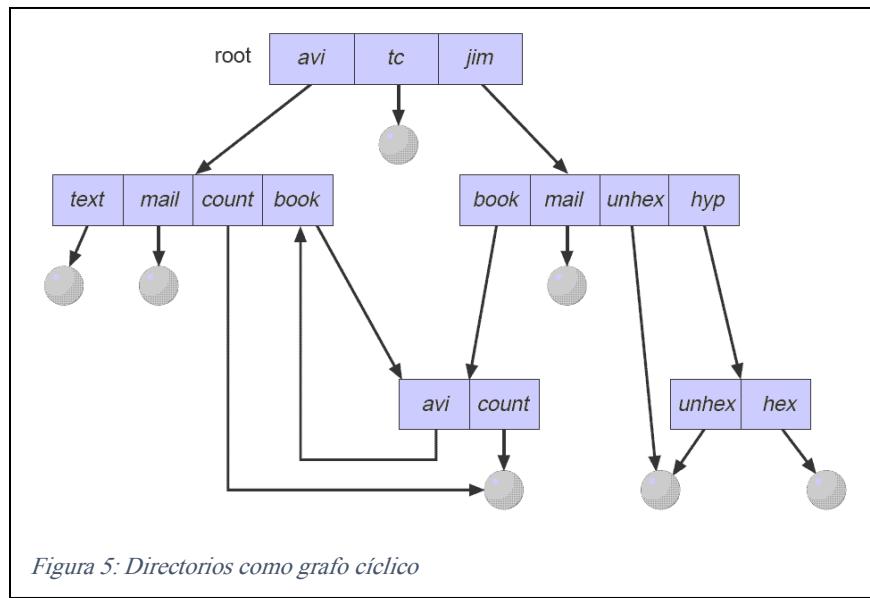
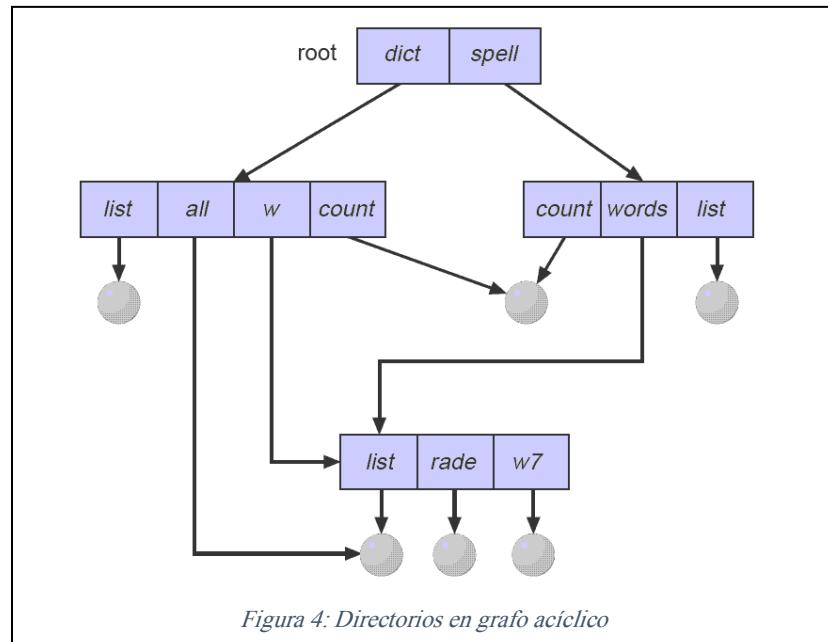


Figura 3: Árbol de directorios

Para solucionar esto, se implementó la referencia entre archivos (*links*) de forma que un archivo pueda ser utilizado desde diferentes directorios, lo cual convierte la estructura de los directorios y archivos en un grafo acíclico, tal como se muestra en la Figura 4.

Aunque cada sistema de archivos tiene su propia implementación y nomenclatura, en general, los enlaces se clasifican en dos tipos:

- Enlaces duros (*Hard links*): la entrada del directorio hace referencia al mismo archivo (i-node en los POSIX). Se guardan contadores en el control de archivos para determinar cuántos enlaces se refieren al mismo archivo. En general se tiene la restricción que deben ser archivos en la misma partición.
- Enlaces suaves (*Soft links*): La entrada en el directorio es un archivo nuevo (otro i-node) cuyo contenido es la ruta completa al archivo referenciado. Esto permite que el archivo referenciado esté en otra partición o, incluso, en otro sistema de archivos. La desventaja es que puede provocar inconsistencias si el archivo referenciado es borrado.



recorridos a los directorios deben tener cuidado de no provocar un ciclo infinito.

El abuso o descontrol en el uso de enlaces puede provocar ciclos en la estructura de los directorios (ver Figura 5) cuando un enlace se refiere a un directorio ancestro del directorio referente. En este caso, los procesos que hacen

## 5 Arquitectura de sistemas de archivos

### 5.1 Virtual File System

El **Virtual File System (VFS)** es una abstracción de alto nivel que proporciona una interfaz uniforme para interactuar con diferentes sistemas de archivos dentro de un sistema operativo. Diseñado para la portabilidad y flexibilidad, el VFS permite que el sistema operativo soporte múltiples sistemas de archivos al mismo tiempo, como ext4, NTFS, FAT, y más, sin que las aplicaciones tengan que preocuparse por las diferencias subyacentes.

#### 1. API Uniforme para Acceso a Archivos

- Ofrece funciones estándar como abrir, leer, escribir y cerrar archivos, independientemente del sistema de archivos real que se utilice.
- Esto simplifica el desarrollo de aplicaciones, ya que los programadores no necesitan adaptar su código a sistemas de archivos específicos.

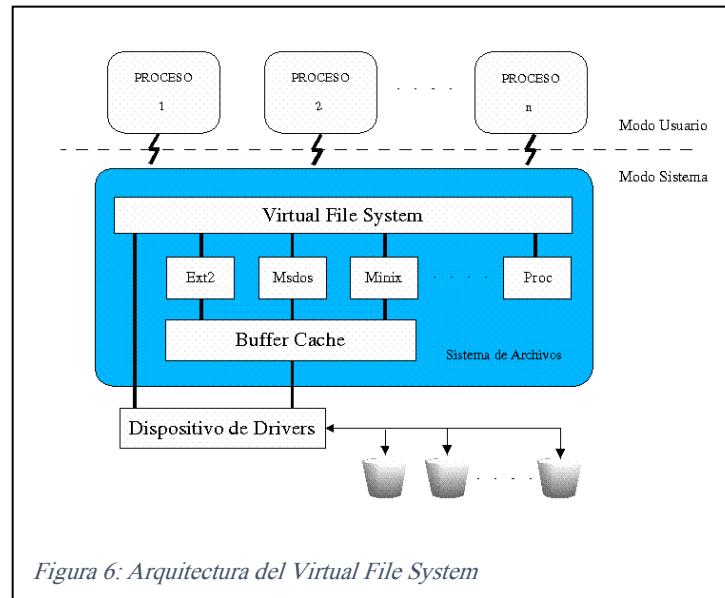
#### 2. Estructuras de Datos Abstraídas

- **Superbloque (Superblock):** Contiene metadatos sobre el sistema de archivos, como el tamaño del sistema de archivos, tipo, y estado.
- **Inodo (Inode):** Representa archivos individuales y contiene información como permisos, propietarios y ubicación de los datos.
- **Dentry (Directory Entry):** Representa una entrada en un directorio, vinculando nombres de archivos con sus inodos.
- **File:** Representa una instancia abierta de un archivo, conectando a las operaciones del sistema de archivos correspondiente.

#### 3. Módulos de Sistemas de Archivos

- Cada sistema de archivos implementa una serie de operaciones estándar definidas por el VFS, como lectura, escritura y manejo de directorios.
- Estos módulos se cargan dinámicamente, permitiendo al kernel soportar múltiples tipos de sistemas de archivos.

#### 4. Abstracción de Dispositivos



- El VFS actúa como un puente entre las operaciones de archivos a nivel de usuario y los controladores de hardware que gestionan los dispositivos de almacenamiento.

## 5. Flujo de operación

- **Solicitud de usuario:** Cuando una aplicación realiza una operación, como abrir un archivo, se envía una llamada al sistema (syscall) que pasa por el VFS.
- **Resolución de directorios:** El VFS utiliza su estructura de datos para traducir el nombre del archivo en una referencia al sistema de archivos correspondiente y el inodo del archivo.
- **Interacción con el sistema de archivos real:** El VFS delega la operación a la implementación específica del sistema de archivos que maneja el almacenamiento físico.
- **Respuesta al usuario:** El VFS toma la respuesta del sistema de archivos y la devuelve a la aplicación.

## 6. Ventajas del VFS

- **Portabilidad:** Permite que los sistemas operativos soporten múltiples sistemas de archivos sin modificar las aplicaciones.
- **Flexibilidad:** Los sistemas de archivos pueden ser añadidos o eliminados dinámicamente sin afectar el núcleo del sistema operativo.
- **Abstracción:** Oculta las diferencias entre sistemas de archivos, proporcionando una interfaz uniforme.

## 7. Ejemplo de uso

Un ejemplo común del VFS es en sistemas Linux, donde permite el acceso simultáneo a sistemas como ext4, NTFS y FAT desde una única interfaz. Esto es esencial en entornos donde se utilizan diferentes dispositivos con diferentes sistemas de archivos.

En resumen, el VFS es un componente crítico que facilita la interoperabilidad, escalabilidad y simplicidad en la gestión de sistemas de archivos en sistemas operativos modernos.