

# *Introducción a los Sistemas Operativos*

Administración de  
Archivos - I



- Versión: Noviembre 2017
- Palabras Claves: Archivo, Directorio, File System,

Algunas diapositivas han sido extraídas de las ofrecidas para docentes desde el libro de Stallings (Sistemas Operativos) y el de Silberschatz (Operating Systems Concepts). También se incluyen diapositivas cedidas por Microsoft S.A.



# *Porque necesitamos archivos?*

- Almacenar grandes cantidades de datos
- Tener almacenamiento a largo plazo
- Permitir a distintos procesos acceder al mismo conjunto de información



# Archivo

- Entidad abstracta con nombre
- Espacio lógico continuo y direccionable
- Provee a los programas de datos (entrada)
- Permite a los programas guardar datos (salida)
- El programa mismo es información que debe guardarse



# *Archivos - Punto de vista del Usuario*

- Que operaciones se pueden llevar a cabo
- Como nombrar a un archivo
- Como asegurar la protección
- Como compartir archivos
- No tratar con aspectos físicos
- Etc.



# *Archivos - Punto de vista del Diseño*

- Implementar archivos
- Implementar directorios
- Manejo del espacio en disco
- Manejo del espacio libre
- Eficiencia y mantenimiento



# *Sistema de Manejo de Archivos*

- Conjunto de unidades de software que proveen los servicios necesarios para la utilización de archivos
  - ✓ Crear
  - ✓ Borrar
  - ✓ Buscar
  - ✓ Copiar
  - ✓ Leer
  - ✓ Escribir
  - ✓ Etc.



# *Sistema de Manejo de Archivos (cont.)*

- Facilita el acceso a los archivos por parte de las aplicaciones
- Permite la abstracción al programador, en cuanto al acceso de bajo nivel (el programador no desarrolla el soft de administración de archivos)



# *Objetivos del SO en cuanto a archivos*

- Cumplir con la gestión de datos
- Cumplir con las solicitudes del usuario.
- Minimizar / eliminar la posibilidad de perder o destruir datos
  - ✓ Garantizar la integridad del contenido de los archivos
- Dar soporte de E/S a distintos dispositivos
- Brindar un conjunto de interfaces de E/S para tratamiento de archivos.



# *Tipos de Archivos*

## □ Archivos Regulares

- ✓ Texto Plano
  - ◆ Source File
- ✓ Binarios
  - ◆ Object File
  - ◆ Executable File

## □ Directorios

- ✓ Archivos que mantienen la estructura en el FileSystem



# *Atributos de un Archivo*

- Nombre
- Identificador
- Tipo
- Localización
- Tamaño
- Protección, Seguridad y Monitoreo
  - ✓ Owner, Permisos, Password
  - ✓ Momento en que el usuario lo modifco, creo, accedio por ultima vez
  - ✓ ACLs



# Ej: Tipos de archivos y atributos

		Referencias	Tamaño	Fecha hora de modificación	
# ls -la					
drwxr-xr-x	5	yoko yoko	4096	May 16 18:02	.
drwxr-xr-x	44	yoko yoko	4096	May 16 18:13	..
-rw-r--r--	1	yoko grupo1	0	May 16 18:01	archivo1
-rw-r--r--	1	yoko grupo1	0	May 16 17:54	archivo2
-rw-r--r--	1	yoko grupo1	0	May 16 17:57	archivo3
drwxr-xr-x	2	yoko grupo1	4096	May 16 17:55	directorio1
drwxr-xr-x	2	yoko grupo1	4096	May 16 17:54	directorio2
drwxr-xr-x	2	yoko grupo1	4096	May 16 18:02	directorio3
lrwxrwxrwx	1	yoko yoko	23	May 16 18:02	pepe99 -> directorio1/archivo9999

Diagrama que muestra la estructura de los resultados de 'ls -la' con sus respectivos atributos:

- Permisos:** Indicados por la primera columna.
- Referencias:** Indicadas por el número en la segunda columna.
- Tamaño:** Indicado por el número en la tercera columna.
- Fecha hora de modificación:** Indicada por la cuarta columna.
- Nombre:** Indicado por la sexta columna.
- Usuario dueño y Grupo dueño:** Indicado por la quinta columna.
- d : directorio**, **I : link (soft)**, **- : archivo**: Explicación de los tipos de archivos.



# Directarios

- Contiene información acerca de archivos y directorios que están dentro de él
- El directorio es, en si mismo, un archivo
- Interviene en la resolución entre el nombre y el archivo mismo.
- Operaciones en directarios:
  - ✓ Buscar un archivo
  - ✓ Crear un archivo (entrada de directorio)
  - ✓ Borrar un archivo
  - ✓ Listar el contenido
  - ✓ Renombrar archivos
  - ✓ Etc.

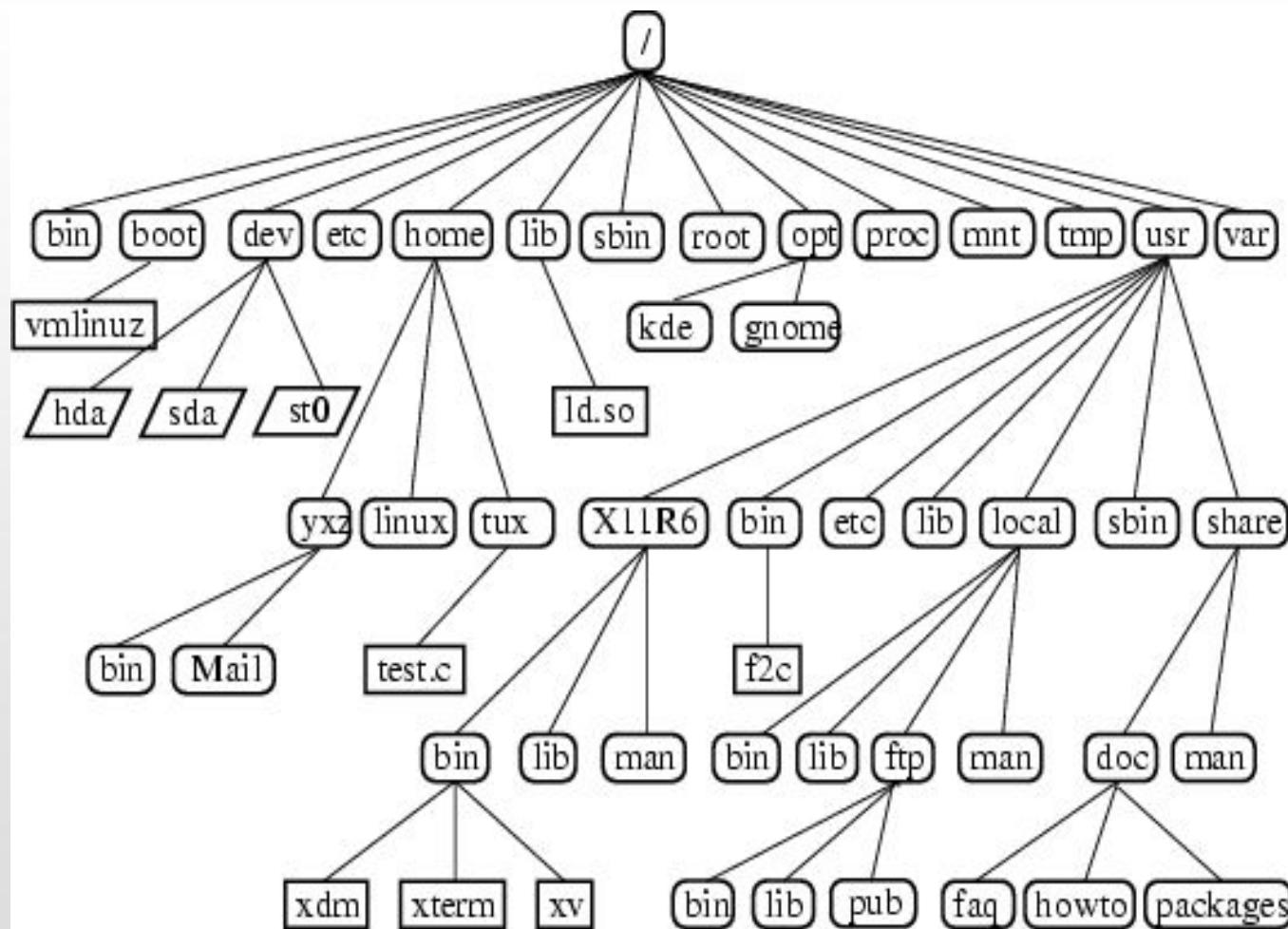


# Directarios de Archivos (cont.)

- El uso de los directorios ayuda con:
  - ✓ La eficiencia: Localización rápida de archivos
  - ✓ Uso del mismo Nombre de archivo:
    - ◆ Diferentes usuarios pueden tener el mismo nombre de archivo
  - ✓ Agrupación: Agrupación lógica de archivos por propiedades/funciones:
    - Ejemplo: Programas Java, Juegos, Librerias, etc.



# Estructura de Dir. Jerárquica o Arbol



# *Estructura de Directorios*

- Los archivos pueden ubicarse siguiendo un path desde el directorio raíz y sus sucesivas referencias (**full pathname** del archivo o **PATH absoluto**)
- Distintos archivos pueden tener el mismo nombre pero el fullpathname es único



# *Estructura de Directorios*

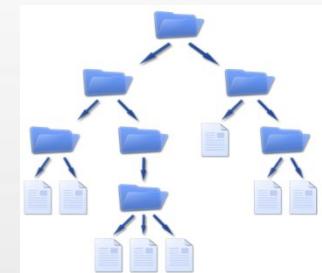
- El directorio actual se lo llama “directorio de trabajo (working directory)
- Dentro del directorio de trabajo, se pueden referenciar los archivos tanto por su **PATH absoluto** como por su **PATH relativo** indicando solamente la ruta al archivo desde el directorio de trabajo.



# *Identificación absoluta y relativa*

Tanto archivos como directorios se pueden identificar de manera:

- **Absoluta.** El nombre incluye todo el camino del archivo.
  - </var/www/index.html>
  - C:\windows\winhelp.exe
- **Relativa.** El nombre se calcula relativamente al directorio en el que se esté
  - si estoy en el directorio /var/spool/mail/
  - Entonces es: ../../www/index.html



# *Compartir archivos*

- En un ambiente multiusuario se necesita que varios usuarios puedan compartir archivos
- Debe ser realizado bajo un esquema de protección:
  - ✓ Derechos de acceso
  - ✓ Manejo de accesos simultáneos



# Protección

- El propietario/administrador debe ser capaz de controlar:
  - ✓ Que se puede hacer
    - ◆ Derechos de acceso
  - ✓ Quien lo puede hacer



# *Derechos de acceso*

- Los directorios también tienen permisos, los cuales pueden permitir el acceso al mismo para que el usuario pueda usar el archivo siempre y cuando tenga permisos.



# Derechos de acceso (cont.)

## □ Execution

✓ El usuario puede ejecutar

## □ Reading

✓ El usuario puede leer el archivo,

## □ Appending

✓ El usuario puede agregar datos pero no modificar o borrar el contenido del archivo



# Derechos de acceso (cont.)

## □ Updating

✓ El usuario puede modificar, borrar y agregar datos. Incluye la creación de archivos, sobreescribirlo y remover datos

## □ Changing protection

✓ El usuario puede modificar los derechos de acceso

## □ Deletion

✓ El usuario puede borrar el archivo



# *Derechos de acceso*

## □ Owners (propietarios)

- ✓ Tiene todos los derechos
- ✓ Pueder dar derechos a otros usuarios. Se determinan clases:
  - ◆ Usuario específico
  - ◆ Grupos de usuarios
  - ◆ Todos (archivos públicos)



# Ejemplo - Protección en UNIX

□ Derechos de acceso son definidos independientemente para:

- ✓ (u) user - Owner (creator) of a file
- ✓ (g) group - Group
- ✓ (o) other - all other users of the UNIX system

□ Derechos de Acceso:

- ✓ (r) Read access right; List right for directory
- ✓ (w) Write access right; Includes delete/append rights
- ✓ (x) Execute access right; Traverse right for directories

□ Binary representation:

- ✓ (x): Bit 0 (+1)
- ✓ (w): Bit 1 (+2)
- ✓ (r): Bit 2 (+4)

□ Rights can be combined

- ✓ Read+Write access right: 6
- ✓ Read+Execute access right: 3
- ✓ Read-only: 2

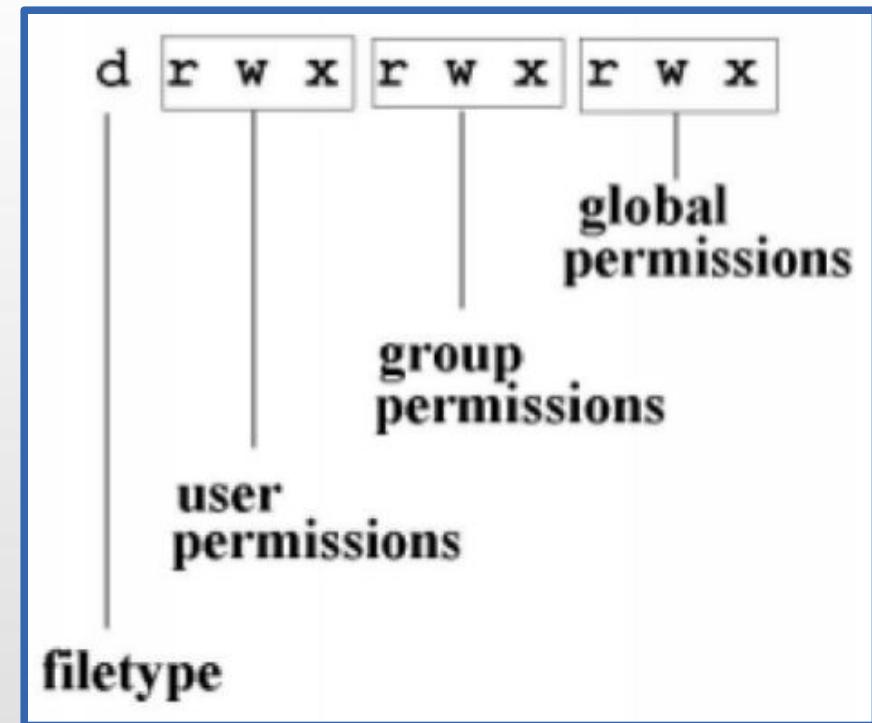


# Ejemplo - Protección en UNIX

Los permisos que se pueden dar o quitar son:

- r - de lectura
- w - de escritura
- x - de ejecución

```
$ ls -l
drwxrwxr-x 4 www      www      ..
-rw-rw-r-- 1 www      www      x_windows.tex
lrwxrwxrwx 1 lee      lee      img -> ../linux/img/
-rw-rw-r-- 1 lee      lee      test.log
```



# Ejemplo - Protección en Windows

