

# Tema

# 7.2

## Sistema Operativos (SSOO)

## Gestión de ficheros II

# Índice

- Permisos extendidos UNIX
- Compartir ficheros
- Sistema FAT

# Permisos extendidos UNIX

## ¿Qué son?

- Caso particular de un esquema general de permisos de listas de acceso

## ¿Para qué sirven?

- Para especificar permisos de lectura, escritura y/o ejecución a usuarios concretos de la máquina y/o grupos de usuarios concretos:
  - Mayor flexibilidad para configurar permisos
  - Coste adicional (un fichero o directorio tiene asociado un fichero adicional).

# Permisos extendidos UNIX

## Formato de un fichero de permisos extendidos:

atributes: (SUID | SGID | SVTX)

base permissions:

owner (castano):            rwg

roup (icc):                r--

others:                    ---

Extended permissions:

enabled

deny                      rw-        u:juan

permit    -w-            u:pepe g:ii

specify r-x            u:pedro.

# Permisos extendidos UNIX

Comandos del shell relacionados:

## **aclget [-o Outfile] File**

- Muestra por la salida estándar la información del control de acceso del
- fichero File
- -o Outfile redirecciona la salida estándar del comando al fichero Outfile

## **aclput [-i Infile] File**

- Asigna al fichero File la información de control de acceso que se especifique por la entrada estándar
- -i Infile asigna al fichero File la información de control de acceso contenida en el fichero Infile

## **acledit File**

- Edita (con el editor definido por defecto para el usuario) la lista de permisos del fichero File y permite modificarlos

# Permisos extendidos UNIX

Ejemplos:

- `$ aclget fichero > perm; aclput -i perm fichero2`
- `$ aclget -o perm fichero; aclput -i perm fichero2`
- `$ aclget fichero | aclput fichero2`

¿Qué hace cada línea?

# Compartir ficheros

Necesidad de compartir ficheros entre usuarios

Aspectos implicados:

- Permisos de acceso
- Gestión de accesos simultáneos

## **Semántica de coutilización**

Especifica el efecto de varios procesos accediendo de forma simultánea al mismo archivo y cuándo se hacen efectivas las modificaciones.

# Compartir ficheros

Tipos de semántica de coutilización:

## **Semántica de archivos inmutables:**

- Una vez creado el archivo sólo puede ser compartido para lectura y no cambia nunca.

## **Semántica UNIX (POSIX):**

- Cada proceso independiente tiene una imagen del fichero y no comparten puntero → Mecanismos de sincronización necesarios.
- Si existe relación de parentesco entre procesos comparten el puntero y las escrituras son inmediatamente visibles para todos los procesos con el archivo abierto. La coutilización afecta también a los metadatos.



# Compartir ficheros

Tipos de semántica de coutilización:

## **Semántica de archivos inmutables:**

- Una vez creado el archivo sólo puede ser compartido para lectura y no cambia nunca.

## **Semántica UNIX (POSIX):**

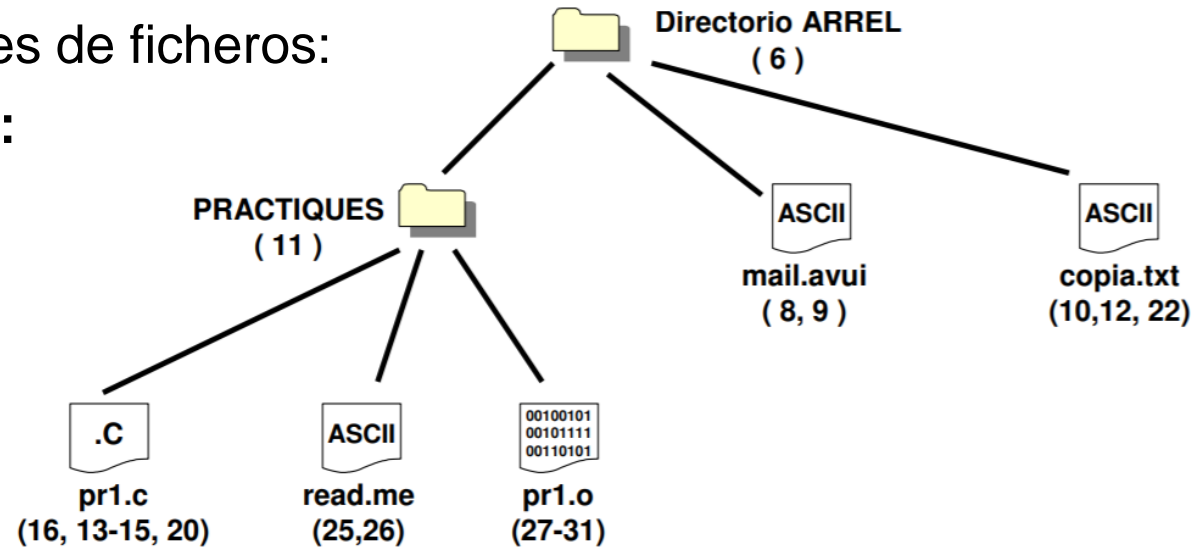
- Cada proceso independiente tiene una imagen del fichero y no comparten puntero → Mecanismos de sincronización necesarios.
- Si existe relación de parentesco entre procesos comparten el puntero y las escrituras son inmediatamente visibles para todos los procesos con el archivo abierto. La coutilización afecta también a los metadatos.

## **Semántica de sesión:**

- Las escrituras que hace un proceso no son inmediatamente visibles para los demás procesos con el archivo abierto.
- Cuando se cierra el archivo los cambios se hacen visibles para las futuras sesiones.
- Un archivo puede tener temporalmente varias imágenes (versiones).
- Sincronización explícita necesaria si se requiere actualización inmediata.

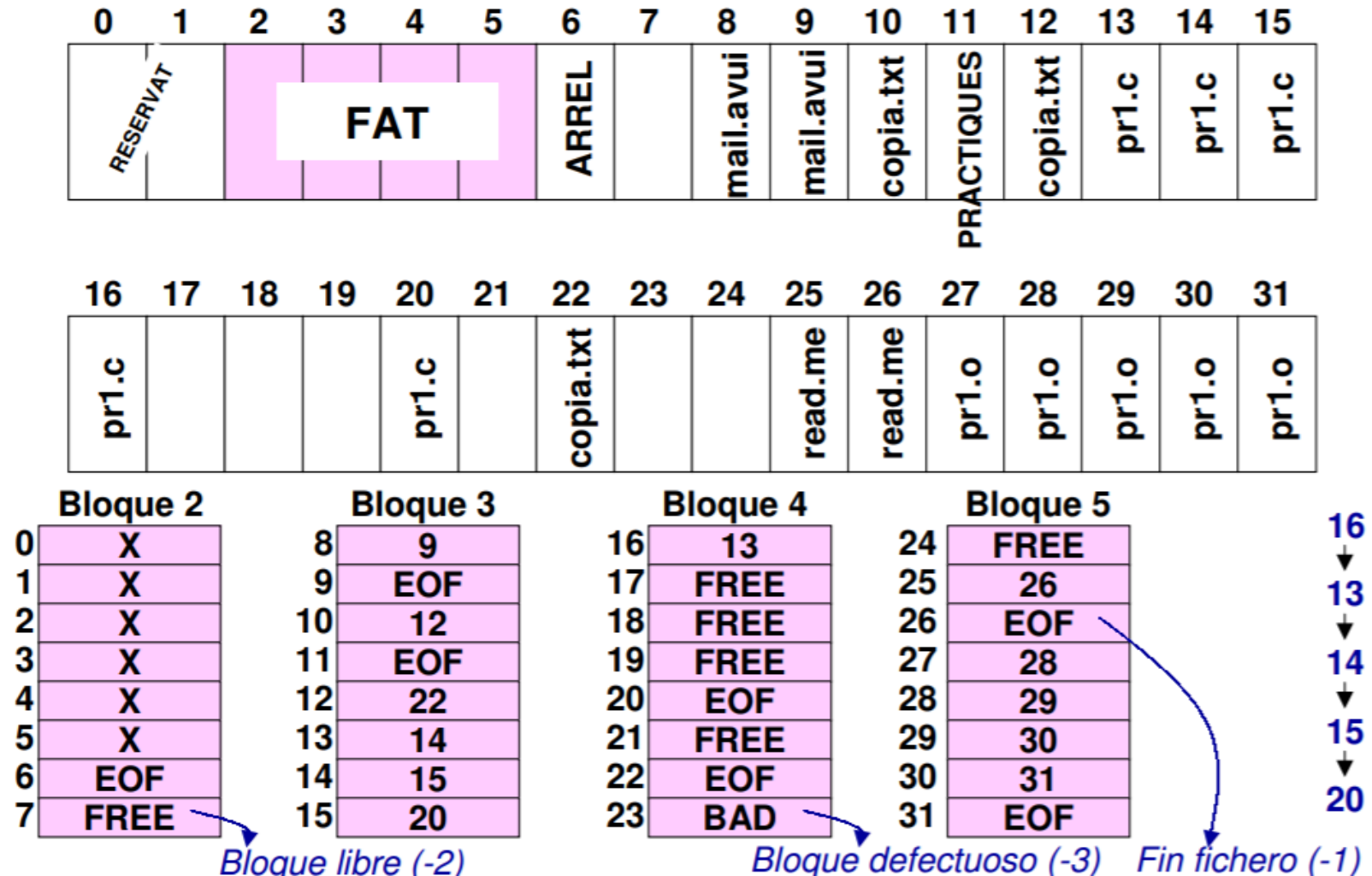
# Sistema de fichero: FAT

Asignación enlazada de bloques de ficheros:  
**FAT (“File Allocation Table”):**



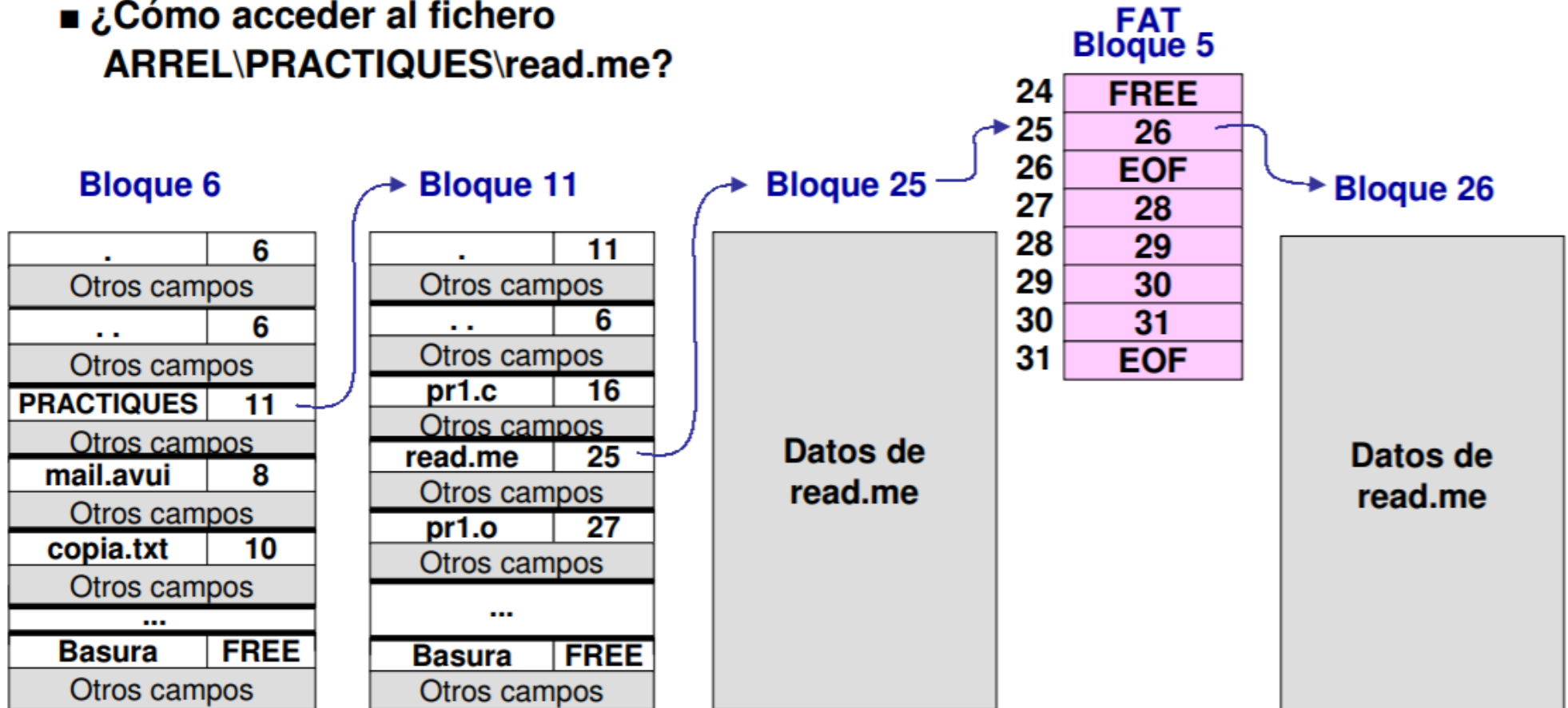
- Una entrada en la tabla por cada bloque del disco
- Cada entrada apunta al siguiente bloque del fichero
- Inconveniente:
  - Hace falta toda la FAT en memoria aunque sólo haya un fichero abierto.
  - Si no está toda la FAT en memoria hay que acceder a disco para seguir la lista de bloques (localización lenta).
- Viable para PC's pero no para ordenadores con grandes SF.

# Sistema de fichero: FAT



# Sistema de fichero: FAT

## ■ ¿Cómo acceder al fichero ARREL\PRACTIQUES\read.me?



# Bibliografía

- [Germán Fabregat Lluca](#), Arquitectura y Tecnología de Computadores (ATC). Departamento de Ingeniería y Ciencia de los Computadores. Universitat Jaume I (UJI).
- **FCO. MANUEL MARQUEZ**, UNIX Programación avanzada, 2da edición.
- **CARRETERO**, Jesús, **GARCÍA**, Félix, **DE MIGUEL**, Pedro, **PÉREZ**, Fernando. Sistemas Operativos: una visión aplicada. McGraw-Hill, 2001.
- **STALLINGS, William. Sistemas operativos: aspectos internos y principios de diseño. 5ª Edición. Editorial Pearson Educación. 2005. ISBN: 978-84-205-4462-5.**
- **TANENBAUM**, Andrew S. Sistemas operativos modernos. 3ª Edición. Editorial Prentice Hall. 2009. ISBN: 978-607- 442-046-3.

