

SISTEMAS OPERATIVOS E/S Y SISTEMA de FICHEROS

Víctor Rob

Pedro de Miguel

M^a de los Santos Pérez

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



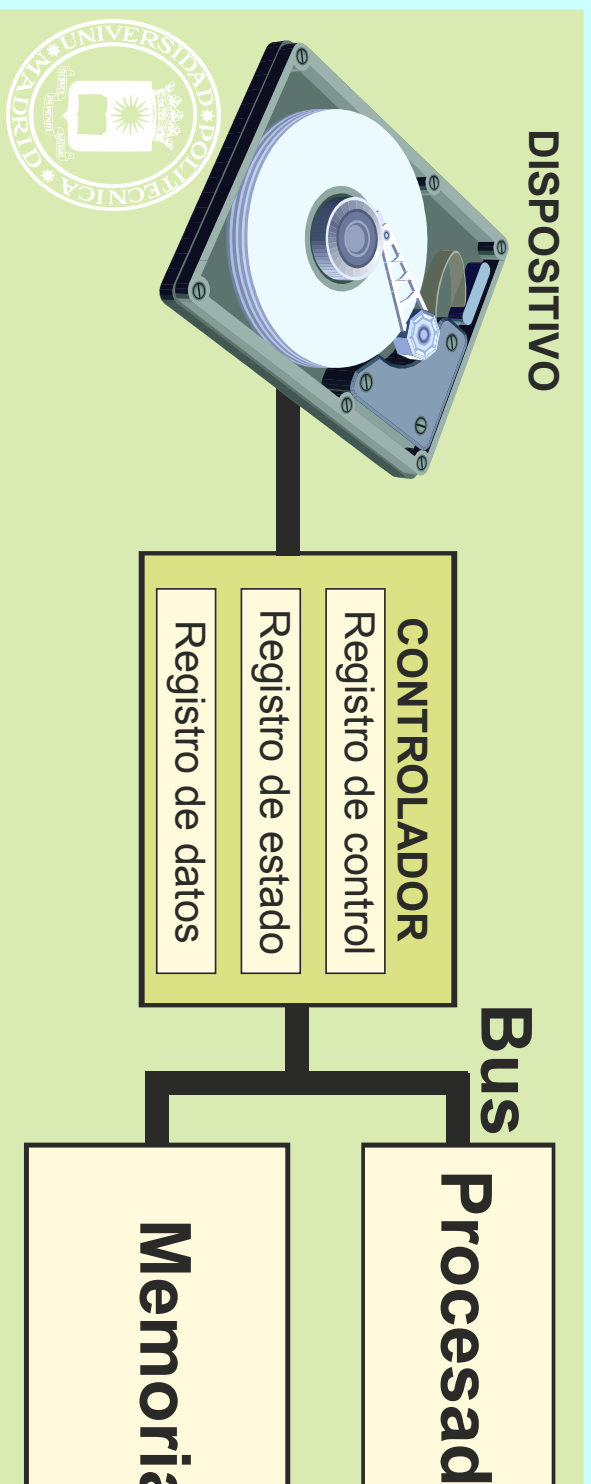
CONCEPTOS BÁSICOS DE E/S

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

CONCEPTOS BÁSICOS DE E/S



- **Arquitectura con mapa de E/S propio IN y OUT (p.e. Intel)**
- **Arquitectura con mapa de E/S en memoria LOAD y STORE (p.e. ARM)**

El acceso a los dispositivos es:

- **Complejo**
 - **Detalles físicos de los dispositivos**
 - **Dependiente de las direcciones físicas**
- **Sin protección**
 - **Si el usuario accede a nivel físico no tiene restricciones del dispositivo no limita**

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

CONCEPTOS BÁSICOS DE E/S

Hardware de E/S

- Dispositivos de E/S
 - De bloques (discos, cintas, placas red)
 - De caracteres (teclado, ratón)
 - Acceso aleatorio
 - Acceso secuencial
- Controladores de dispositivos
 - **E/S programada**: No concurrencia E/S-procesador
 - **Interrupciones**: Concurrencia E/S-procesador
 - **DMA**: Máxima concurrencia E/S-procesador

Objetivos del SO en E/S

- Controlar el funcionamiento de los dispositivos de E/S
- Facilitar el manejo de los dispositivos de E/S a través de inte
- Proporcionar mecanismos de protección
- Explotar la concurrencia E/S-procesador

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70





CONCEPTO DE SISTEMA DE FICHERO

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

SISTEMA DE FICHEROS Y PARTICIONES

- El **sistema de ficheros** permite organizar la información dentro dispositivos de almacenamiento secundario en un formato de **sistema operativo**
- Previamente a la instalación del sistema de ficheros es necesario físicamente, o lógicamente, los discos en **particiones o volúmenes**
- Una **partición** es una porción de un disco a la que se le dota de una propia y que puede ser manipulada por el sistema operativo como entidad lógica independiente
- Relación sistema de ficheros-dispositivo:
- Típico: 1 dispositivo N sistemas de ficheros (particiones)
- Típicamente cada dispositivo se divide en una o más particiones. En cada partición, su propio sistema de ficheros, que puede ser diferente
- Grandes sistemas: N dispositivos 1 sistema de ficheros
- La **tabla de particiones** indica el principio, el tamaño y el tipo de partición

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

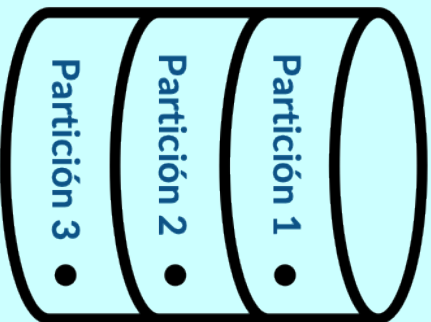
...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

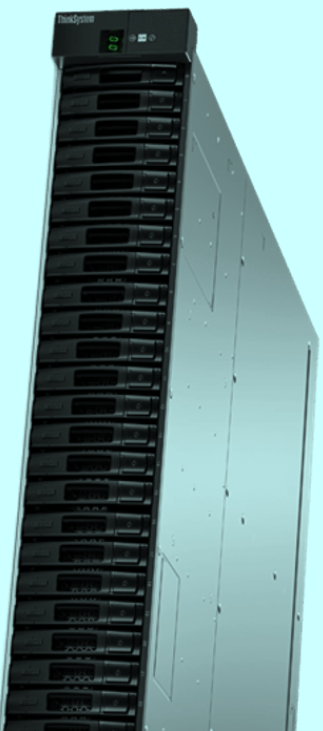
Cartagena99



SISTEMA DE FICHEROS Y PARTICIONES



Ejemplo de 1 dispositivo con
3 particiones o volúmenes



NAS (Network Area Storage) all-flash LENOVO ThinkSystem
Hasta 1,47 PB
Hasta 300.000 IOPS
Hasta 10 Gbps
Interfaces: Fibre Channel de 16/32 Gb, iSCSI de 10/20 Gb
Desde 1 hasta 512 particiones

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

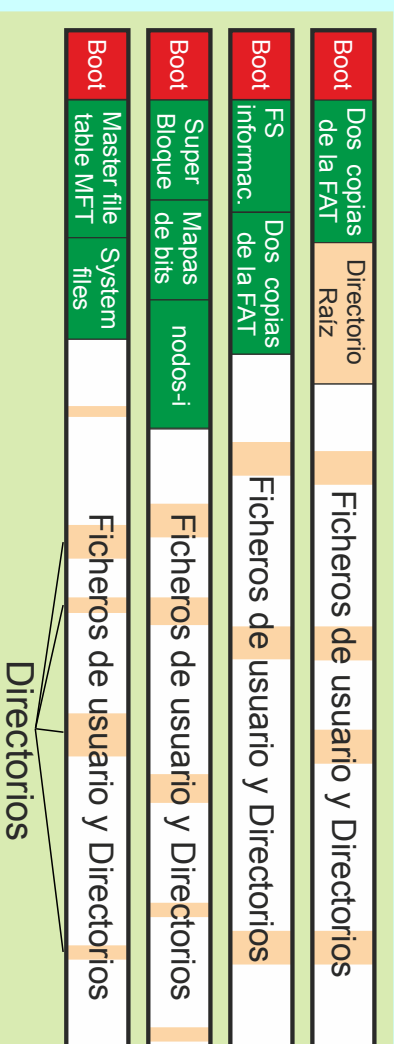


FICHERO Y SISTEMA DE FICHEROS

Fichero. Conjunto de informaciones relacionadas que se almacenan en un almacenamiento secundario y que se identifica mediante un nombre.

Sistema de ficheros. Conjunto autónomo de informaciones incluidas en una unidad de almacenamiento (partición o volumen) que permiten su explotación. Se compone de:

- **Información neta:** Ficheros de usuario (programas y datos).
- **Metainformación**
 - Estructura física de los ficheros.
 - Información asociada a los ficheros.
 - Directorios.



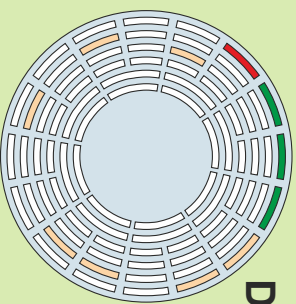
Disco

— Boot + tabla de particiones del disco

— Metainformación: FAT, nodos-i o mapas de bits

— Agrupaciones asignadas a los directorios

— Agrupaciones asignadas a ficheros



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



SISTEMA DE FICHEROS FAT



Utilizado en Microsoft Windows

Espacio dividido en

- **BOOT:** Bloque de carga: Arranque del sistema operativo
- **FAT:** Tabla de asignación de ficheros
- **Copia de FAT:** Copia de seguridad
- **Directorio Raíz:** Directorio principal del volumen
- **Zona de directorios y ficheros**

Sistemas de Ficheros actuales

FAT32: Partición máxima 4TB, fichero máximo fichero 2[^]32 byte

AexFAT: Tamaño máximo de fichero 2[^]64 bytes (16 Exbibytes)

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



SISTEMA DE FICHEROS TIPO UNIX



Estructura:

- **BOOT:** Bloque de carga: Arranque del sistema operativo
- **Superbloque:** Información descriptiva de la estructura del sistema
- **Mapa de bits de i-nodos:** Un bit para cada i-nodo de la sección de indicando libre/ocupado
- **Mapa de bits de agrupaciones:** Un bit para cada agrupación indicando libre/ocupado
- **i-nodos:** Tantos i-nodos como número de ficheros que pueda albergar
- **Agrupaciones de datos** para almacenar los ficheros y directorios

Sistemas de Ficheros actuales

UFS2: Unix File System

EXT4: Fourth Extended Filesystem. Volúmenes con tamaños de hasta 1 EiB (EiB) y archivos individuales con tamaños de hasta 16 tebibytes (TiB) tamaño de bloque estándar de 4 KiB

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



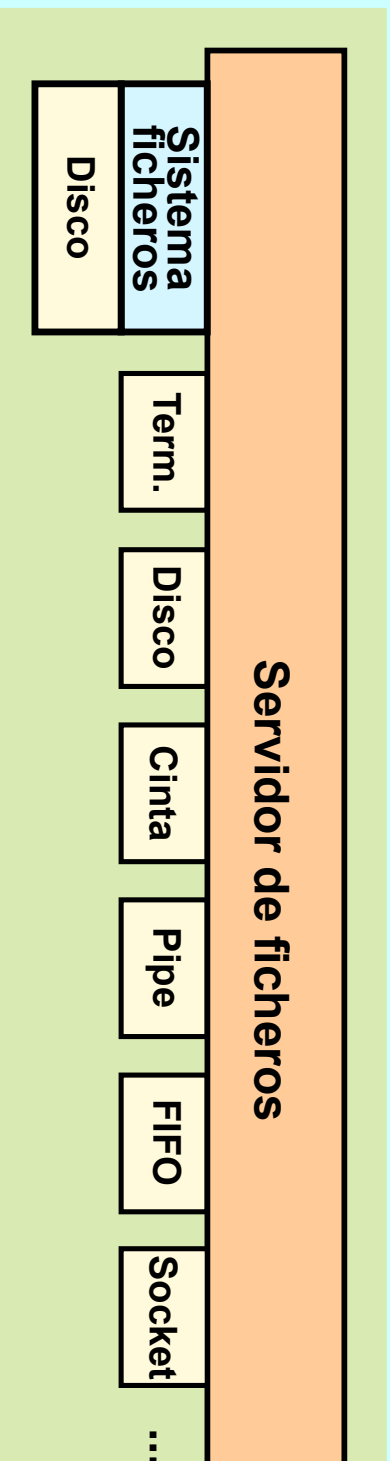
SERVIDOR DE FICHEROS

Servidor de Ficheros: capa de software entre dispositivos y usu

- Suministra una **visión lógica uniforme** de los dispositivos, p como ficheros
- Ofrece **primitivas de acceso cómodas e independientes** de l físicos
- Incorpora **mecanismos de protección**

El servidor de ficheros permite manejar los siguientes tipos de

- Ficheros de usuario (datos y programas)
- Ficheros especiales : Orientado a carácter (ej. terminal) Orientado a bloque (ej. disco)
- Mecanismos de comunicación con y sin nombre
 - Pipe, FIFO, socket UNIX



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



Unidades básicas de información en bytes

Nombre de magnitud	Símbolo	Factor en sistema decimal	Valor en sistema binario (en Bytes)	Nombre de magnitud	Símbolo	Factor en sistema binario	Valor
Byte	B	10^0	1	Byte	B	2^0	1
Kilobyte	KB	10^3	1.000	Kibibyte	KB	2^{10}	1.024
Megabyte	MB	10^6	1.000.000	Mebibyte	MB	2^{20}	1.048.576
Gigabyte	GB	10^9	1.000.000.000	Gibibyte	GB	2^{30}	1.073.741.824
Terabyte	TB	10^{12}	1.000.000.000.000	Tebibyte	TB	2^{40}	1.101.004.480.000
Petabyte	PB	10^{15}	1.000.000.000.000.000	Pebibyte	PB	2^{50}	1.125.899.904.000.000
Exabyte	EB	10^{18}	1.000.000.000.000.000.000	Exbibyte	EB	2^{60}	1.152.921.504.000.000.000
Zettabyte	ZB	10^{21}	1.000.000.000.000.000.000.000	Zebibyte	ZB	2^{70}	1.181.269.653.000.000.000.000
Yottabyte	YB	10^{24}	1.000.000.000.000.000.000.000.000	Yobibyte	YB	2^{80}	1.210.100.640.000.000.000.000.000

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

Cartagena99





FICHEROS

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

VISIÓN FÍSICA DE UN FICHERO

Sector: Unidad mínima de transferencia que puede manejar el c disco (2^m bytes, normalmente 2^9)

Bloque: Es un conjunto de sectores de disco y es la unidad de t mínima que usa el sistema de ficheros (bloque = 2^n sectores)

- Única para cada sistema de ficheros y definible por el us
- El bloque se puede direccionar de manera independiente

Agrupación: Conjunto de bloques que se utilizan como una unid gestión de almacenamiento (agrupación = 2^p bloques).

- Reserva espacio

Al fichero se le asignan agrupaciones, pero se accede siempre e
En algunos sistemas Bloque = Agrupación

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

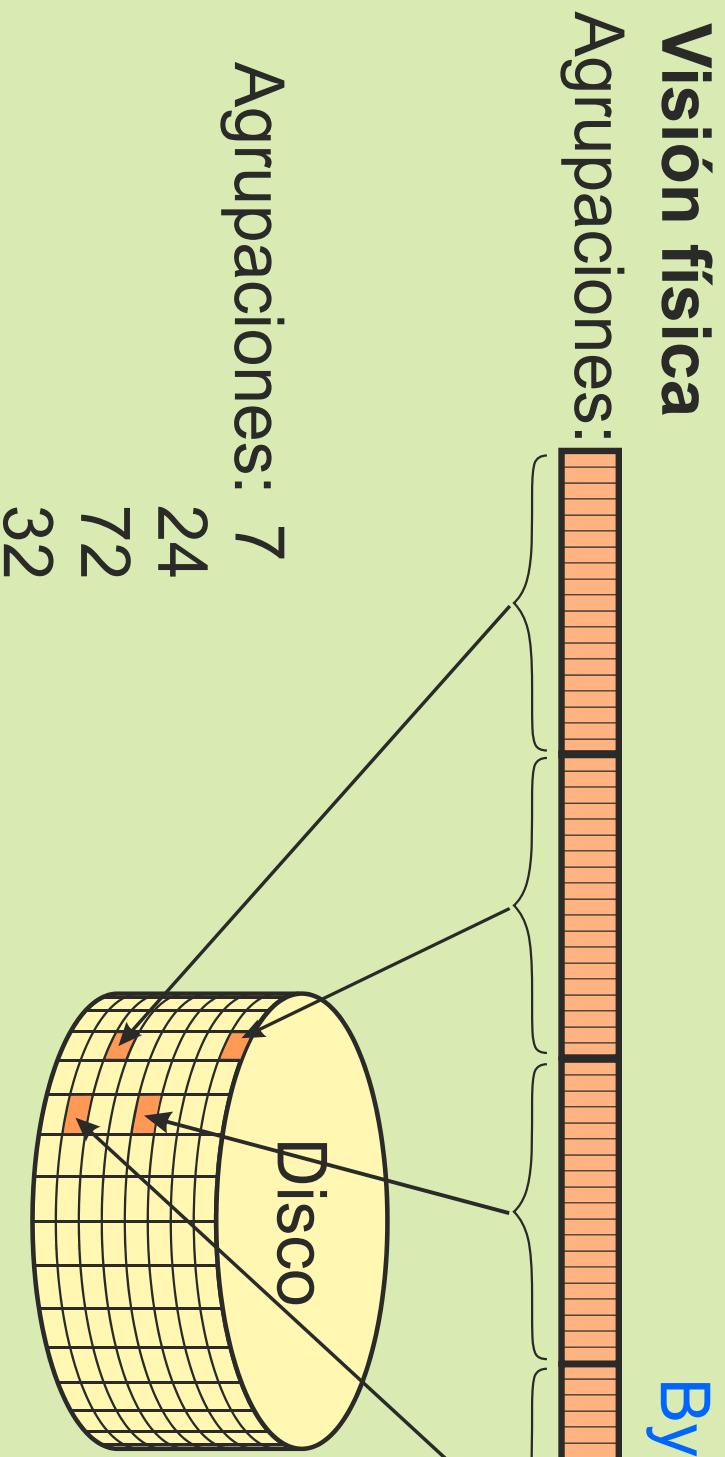
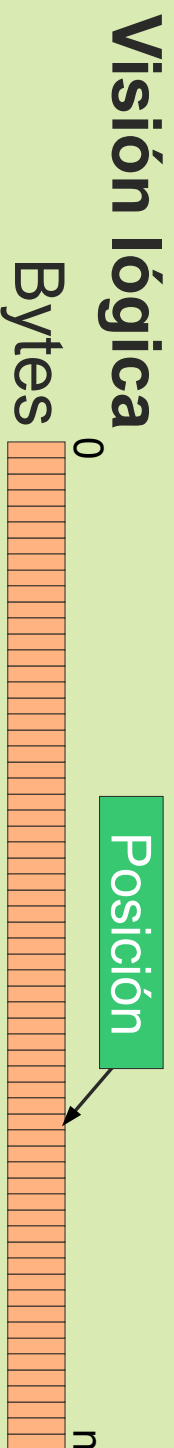
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



ESTRUCTURA LÓGICA Y FÍSICA DE UN FICHERO

- **Estructura lógica del fichero.** Secuencia ordenada de bytes o puntero que indica la posición a partir de la cual se lee o escribe.
 - Sobre esta cadena de bytes se pueden montar otras estructuras, ejemplo, conjunto de registros.
- **Estructura física del fichero.** Secuencia ordenada de agrupaciones



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

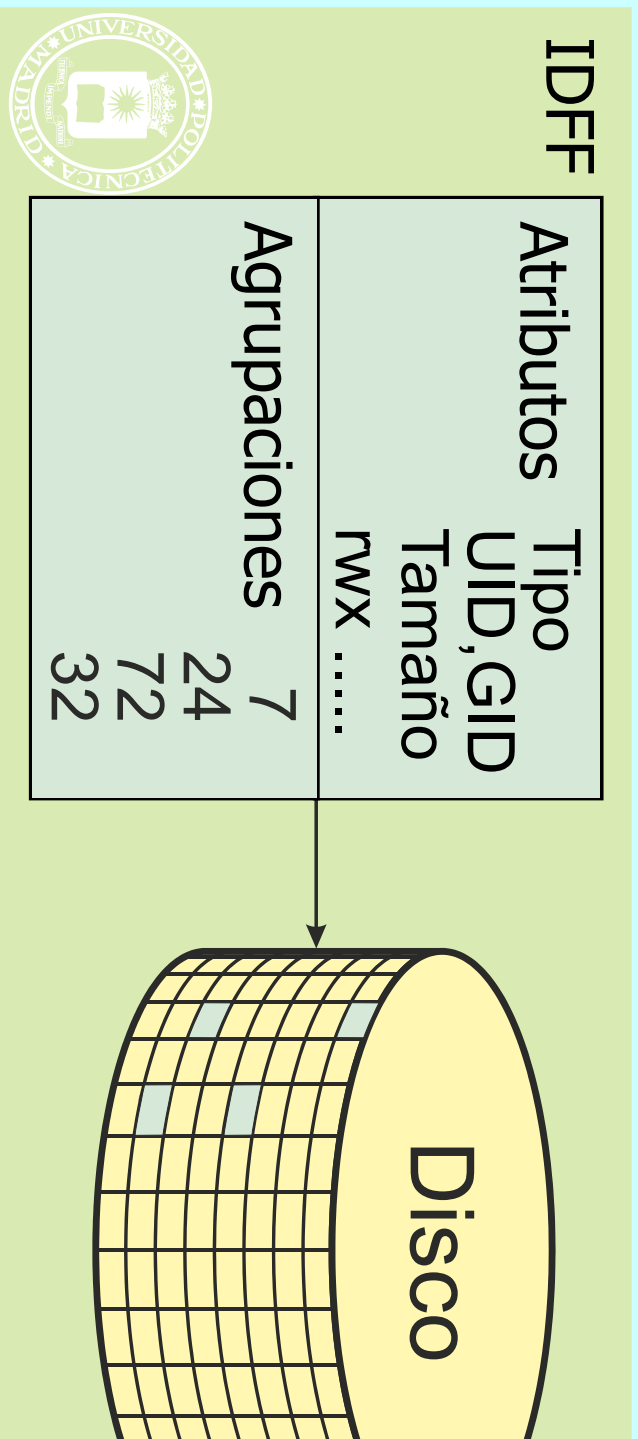


IDENTIFICADOR DE DESCRIPCIÓN FÍSICA DE FICHERO (

DDF: Estructura de información que recoge la descripción física

IDDF: Número identificador del DFF.

- Secuencia ordenada de agrupaciones + ATRIBUTOS
- Almacenada en el propio disco
- FAT en MS-DOS
- Registro MFT (Master File Table) de NTFS (New Technology Microsoft)
- Nodo-i en UNIX



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



CARACTERÍSTICAS DE LOS FICHEROS

Atributos de un fichero

- Identificador único del fichero
- Tipo de fichero
- Dueño y grupo
- Información de protección
- Tamaño real en bytes
- Hora y fecha de creación
- Hora y fecha del último acceso
- Hora y fecha de la última modificación
- Número de enlaces (número de nombres)

Tamaño máximo de un fichero

- Depende de las limitaciones establecidas por:
 - La metainformación (Atributo tamaño real, direccionamiento agrupaciones –tamaño direcciones y nº de direcciones–)
 - Las agrupaciones de datos disponibles (agrupaciones en disco o cuotas establecidas por el administrador)

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

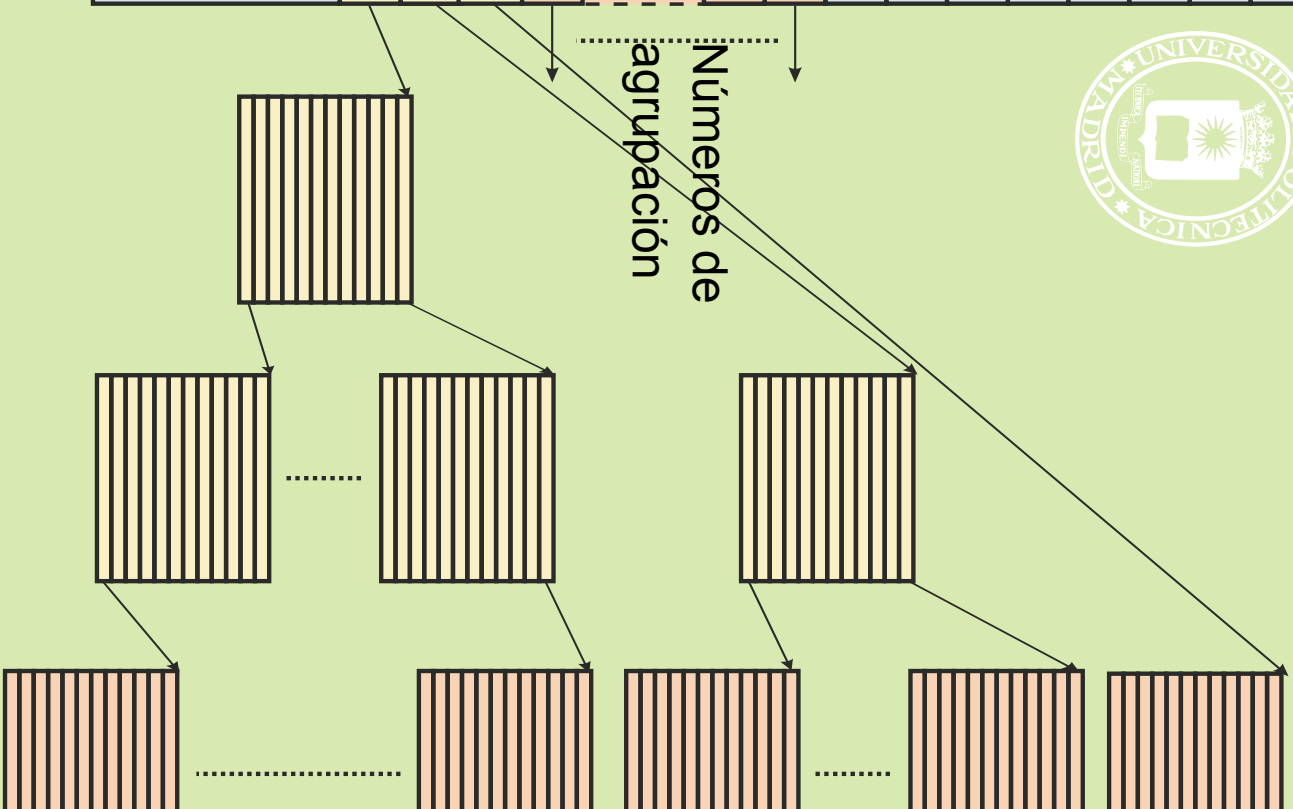


NODO I

- **Tamaño típico 128 B. Punteros de 4B.**



2 B	Tipo de archivo y Protección
2 B	Enlaces (número de Nombres)
2 B	UID propietario
2 B	GID propietario
4 B	Tamaño en bytes
4 B	Instante de creación
4 B	Instante del último acceso
4 B	Instante de la última modificación
4 B	Número de la agrupación 0
4 B	Número de la agrupación 1
4 B
4 B	Número de la agrupación 11
4 B	Puntero indirecto simple
4 B	Puntero indirecto doble
4 B	Puntero indirecto triple
Otras informaciones 28 B	



Nodo-i de ext2

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



Asignación de espacio al fichero

- Se hace por **agrupaciones**.
- El fichero puede quedar disperso o **fragmentado**. Sus agrupaciones tienen por qué ser contiguas.
 - Objetivo de la desfragmentación: mejorar el acceso secuencial.
 - SSD no se desfragmentan.

Gestión del espacio libre

- Agrupaciones libres/ocupadas.
- DFF (p.e. nodos-i) libres/ocupados.
- (mapas de bits)

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99





DIRECTORIOS

Cartagena99

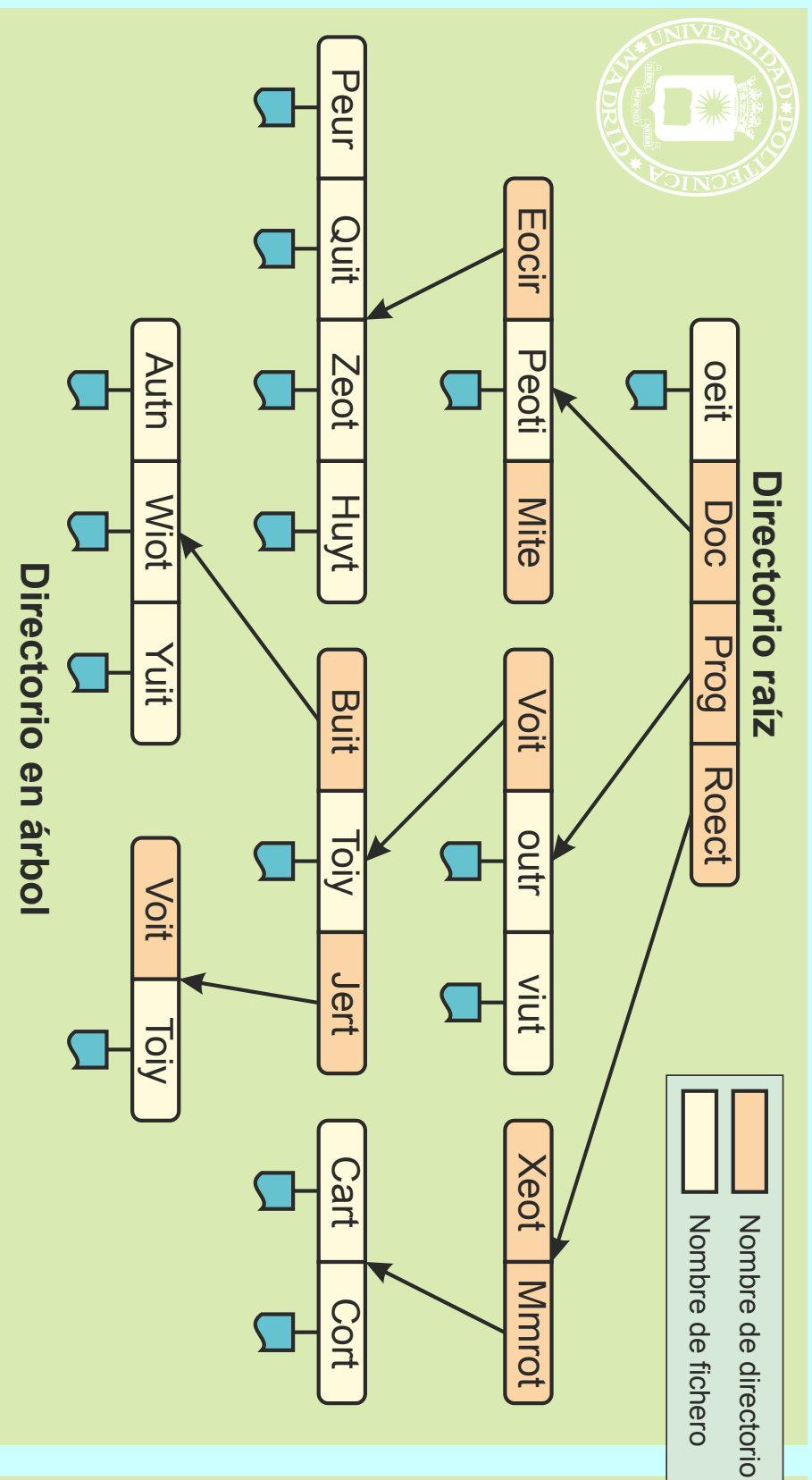
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

DIRECTORIOS

Estructura lógica: Esquema jerárquico de nombres:

- Directorio raíz.
- Subdirectorios.



- Ejemplo UNIX: /usr/include/stdio.h
 - Ejemplo Windows: C:\DocenciaS\osos2\transparencias\pipe
- Directorio único versus directorio por dispositivo. Operación de

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

ESTRUCTURA LÓGICA DE LOS DIRECTORIOS

- Un directorio es una tabla con entradas que asocian nombres con identificadores de fichero (IDFF / n° nodo_i)
- La tabla directorio suele incluir el propio directorio y el padre
 - Esquema jerárquico. La organización jerárquica de un directorio
 - Simplifica el nombrado de ficheros (nombres únicos)
 - Proporciona una gestión de la distribución => agrupar ficheros en forma lógica (mismo usuario, misma aplicación)
 - Cuando se abre un fichero el SO busca el nombre en la estructura de directorios y comprueba los privilegios. Es una operación lateral

Nombre	n° nodo_i
.	146
..	43
Sec1	239
Sec2	240

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



- El SO mantiene en el **BCP** el nombre del directorio actual o c
- El SO mantiene en memoria el nodo_*i* del directorio de traba
- Un proceso puede cambiar su directorio de trabajo
 - El SO comprueba los permisos antes del cambio
- Nombre **absoluto**
 - Es el nombre desde el directorio raíz
 - Empieza por / en UNIX o por el nombre de la unidad en W
- Nombre **relativo**
 - Es el nombre a partir del directorio de trabajo
 - Ahorro de accesos a disco
 - Se parte del nodo_*i* del directorio de trabajo
 - Se requieren menos accesos a disco para analizar el nom que el nombre absoluto

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



NOMBRADO DE FICHEROS Y DIRECTORIOS

Nombres de fichero y directorio Unix:

- **Nombre completo** (empieza por /)
/usr/include/stdio.h
 - **Nombre relativo** al directorio actual o de trabajo (no empieza stdio.h asumiendo que /usr/include es el directorio para formar rutas de acceso
../include/stdio.h
./../include/stdio.h
/usr/./include/./../include/stdio.h
- Estos tres ficheros hacen referencia a stdio.h si /usr/include directorio actual

- **Árbol único** (los dispositivos se montan en un punto del árbol

Nombres de fichero y directorio Windows:

- En lugar de utilizar /, se utiliza \
- Entradas . (propio directorio) y .. (directorio padre)
- Varios árboles (uno por sistema de ficheros). NTFS permite

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

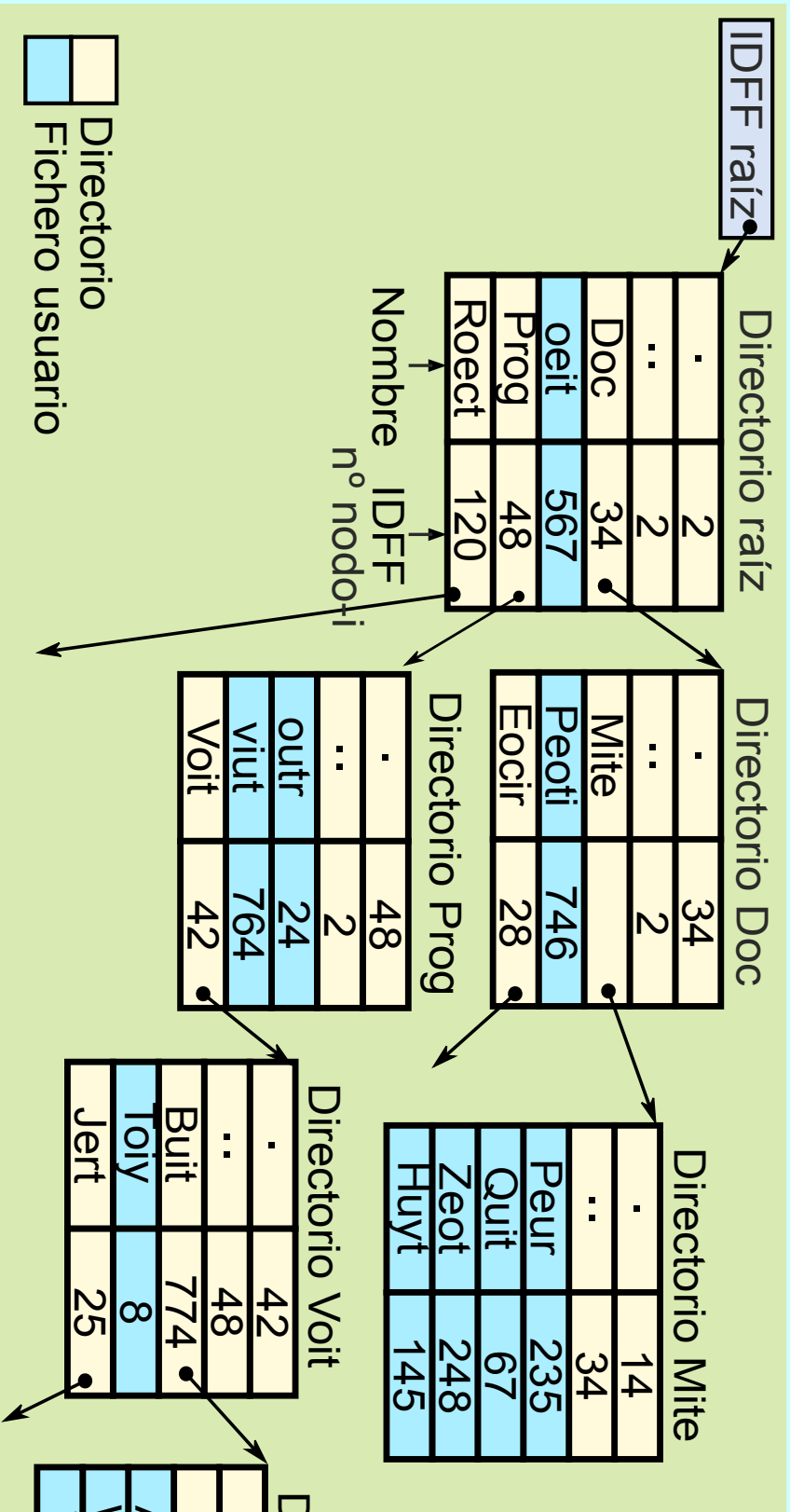
Cartagena99



ESTRUCTURA FÍSICA DE LOS DIRECTORIOS

Estructura física: Tabla Nombre-IDFF. (En UNIX es tabla Nombre-IDFF: identificador de descripción física de fichero. Permite describir el fichero (en UNIX es el nodo-i).

Para alcanzar el IDFF (n.º nodo_i) de un fichero hay que recorrer



Boot	Super Bloque	Mapas de bits	nodos-i
------	--------------	---------------	---------

Ficheros de usuario y Directorios

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

Cartagena99



ue



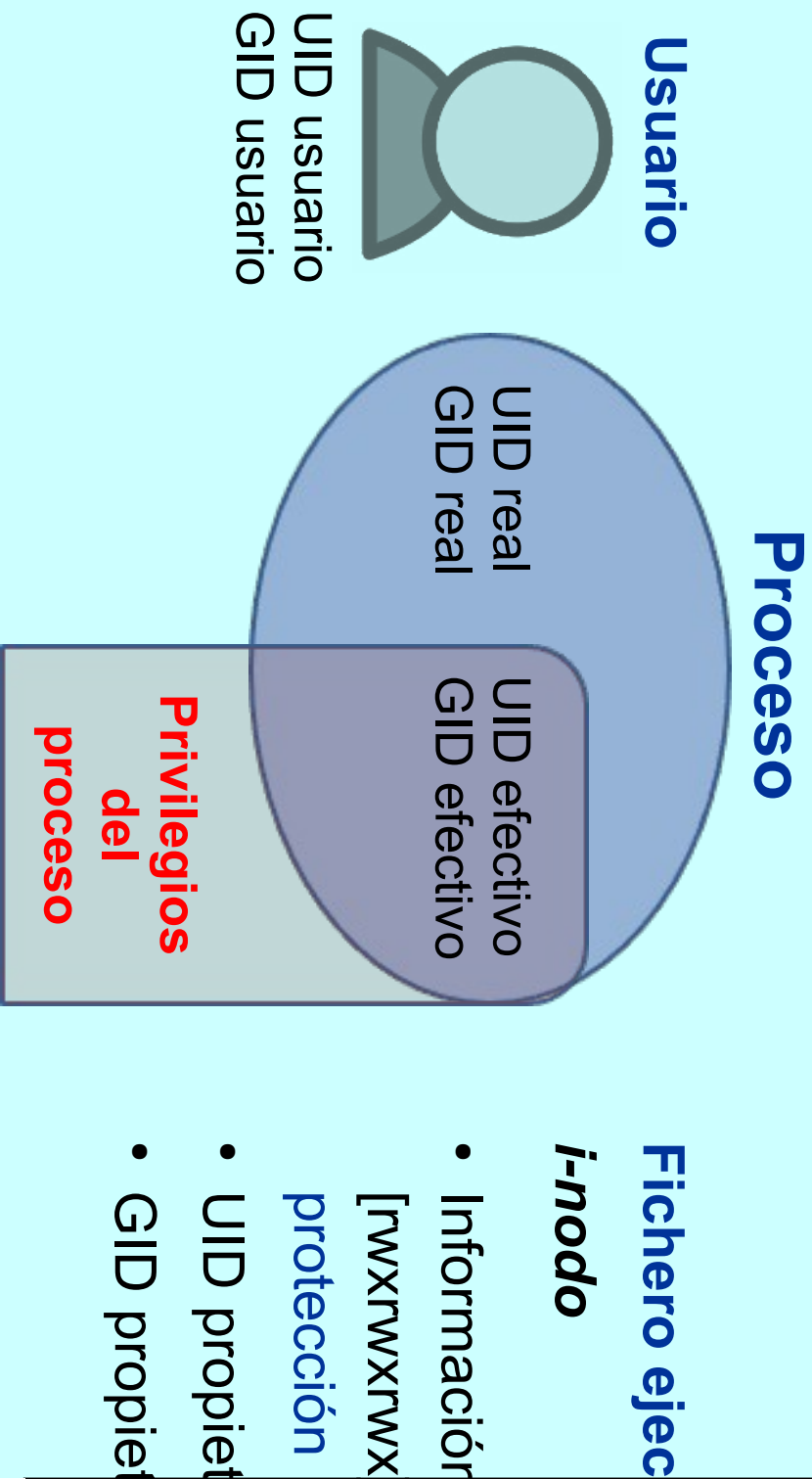
PROTECCIÓN

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

IDENTIFICADORES DE SEGURIDAD DEL PROCESO I



UID real = UID usuario	UID efectivo = UID real
GID real = GID usuario	GID efectivo = GID real

Salvo bi
SETUID o
SETGID
activos

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ón
de



Cartagena99

IDENTIFICADORES DE SEGURIDAD DEL PROCESO II

Fichero ejecutable SETUID activo

i-nodo

- Información de protección
- **UID propietario**
- GID propietario

UID real = UID usuario	UID efectivo UID propietario
GID real = GID usuario	GID efectivo

```
Añadir permisos SETUID
$ chmod u+s /home/usuario/fich
Eliminar permisos SETUID
$ chmod u-s /home/usuario/fich
```

Fichero ejecutable SETGID activo

i-nodo

- Información de protección
- UID propietario
- **GID propietario**

UID real = UID usuario	UID efectivo
GID real = GID usuario	GID efectivo GID propietario

```
Añadir permisos SETUID
$ chmod g+s /home/usuario/fich
Eliminar permisos SETUID
$ chmod g-s /home/usuario/fich
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



PROTECCIÓN UNIX SOBRE UN FICHERO O DIRECTORIO

Por tanto, los procesos se ejecutan con

- UID real
- UID efectivo (se utiliza para permisos)
- GID real
- GID efectivo (se utiliza para permisos)

Bits SETUID y SETGID

- Si un proceso ejecuta un fichero con el SETUID activo, el UID proceso pasa a ser el UID del propietario del fichero
- Si un proceso ejecuta un fichero con el SETGID activo, el GID proceso pasa a ser el GID del propietario del fichero

Reglas de protección:

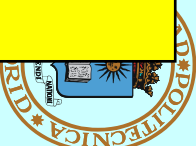
- Si UID efectivo = 0 se concede el acceso (es superusuario)
- Si UID efectivo = UID del propietario, se utiliza el primer grupo
- Si GID efectivo = GID del propietario, se utiliza el segundo grupo
- En caso contrario, se utiliza el último grupo de bits



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



EJEMPLOS DE VOLCADO DEL MANDATO IS

```
drwxr-x---- 2 pepito prof 48 Dec 26 2001 News
drwxr-xr-x  2 pepito prof 80 Sep 29 2004 bin
lrwxrwxrwx  1 root root  3 Jan 23 18:34 lvrend
lrwxrwxrwx  1 root root  3 Jan 23 18:34 lvrend
drwxrwxrwt 16 root root 1928 Apr  9 20:26 tmp
-rwxr-xr-x  1 root root 2436 Dec 26 2001 termw
-rwsr-xr-x  1 root root 22628 Jan  5 10:15 mount
```

Carácter inicial:

-	fichero normal
d	directorio
l	enlace simbólico
b	dispositivo de bloques
c	dispositivo de caracteres
p	FIFO
s	socket UNIX

Bit x:

- Si aparece una “s” significa que está activo el SETUID o SET posición.
- Si aparece una “t” en un directorio se permite crear y borrar el UID efectivo del proceso (se usa para directorios temporal podrán borrar las entradas de otros usuarios).

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



ESTABLECER PERMISOS ESPECIALES

-----	0	Predeterminado, sin permisos especiales. No ha
-----t	1	Bit de persistencia, sticky bit
-----s---	2	Bit sgid de grupo
-----s--t	3	Bit sgid y sticky
-----s-----	4	Bit suid
---s-----t	5	Bit suid y sticky
---s---s---	6	Bit suid y sgid
---s---s--t	7	Bit suid, sgid y sticky

```
#> ls -l
/usr/prog -r-x--x--x 24 root root 4096 sep 25 18:1
#> chmod 4511 /usr/prog
#> ls -l
/usr/prog -r-s--x--x 24 root root 4096 sep 25 18:1
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70





SERVICIOS DEL SERVIDOR DE FICHER

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

- Se crea.
 - Se abre → se obtiene un **fd** (descriptor de fichero).
 - Se escribe y lee a través del **fd**.
 - Se cierra.
 - Se elimina (no significa que se borre).
 - Para trabajar con un fichero hay que definir una **sesión** con **open** y **close**.
 - Las operaciones de lectura y escritura se hacen a partir del **fd** en la posición, que queda modificado.
 - Las escrituras pueden hacer crecer al fichero.

Cartagena99

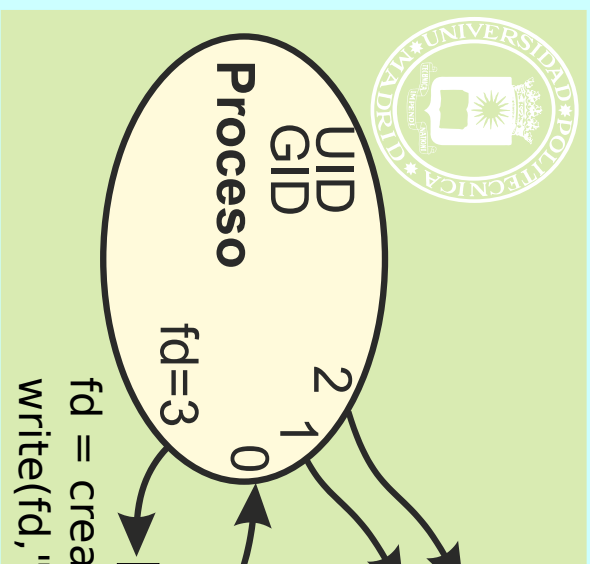
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



DESCRIPTOR DE FICHERO fd

- Entero no negativo que identifica un fichero abierto
- Se almacenan en el BCP
- Se asignan en orden 0, 1, 2, ...
- Los procesos tienen al menos:
 - 0 entrada estándar
 - 1 salida estándar
 - 2 salida de error
- El proceso hijo **hereda** los descriptors del padre (fork())
- Los ficheros abiertos **siguen abiertos** después del exec()
- **Importante:** Todas las llamadas al sistema que devuelven un fichero, devuelven el más bajo disponible (por proceso), salvo



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

SEMÁNTICA DE COUTILIZACIÓN

Semántica de coutilización: especifica el efecto de varios procesos accediendo de forma simultánea al mismo fichero

Existen diferentes tipos de semánticas

Semántica de coutilización UNIX

- Los procesos pueden compartir ficheros de forma simultánea
- Las escrituras son inmediatamente visibles para todos los procesos en el fichero abierto
- Los datos se escriben en orden de llegada al SO (si se quiere específico => usar cerrojos)
- El puntero se comparte cuando se hereda o duplica el descriptor
- El puntero se crea en el open
- La coutilización afecta a los metadatos

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

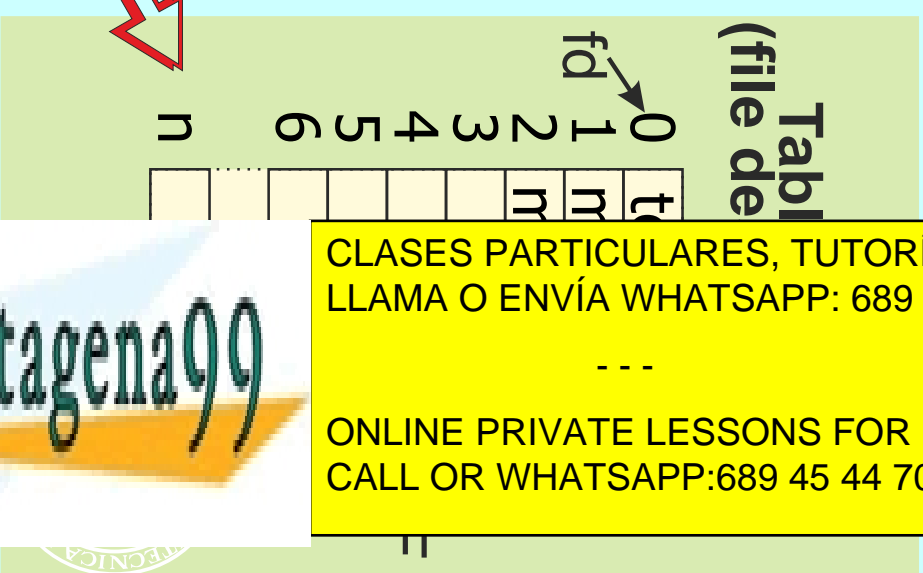
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



- Analiza el nombre completo hasta obtener el IDFF (n° Nodo_ i)
- Llena la primera entrada libre de la tabla de descriptores de BCP. De momento, consideramos que se introduce el IDFF.
- El programa utiliza el fd mientras que el SO utiliza el IDFF.

TABLA DE PROCESOS		
BCP 0	BCP 1	BCP N
pid	pid	pid
uid, gid real	uid, gid real	uid, gid real
uid, gid efect.	uid, gid efect.	uid, gid efect.
pid padre	pid padre	pid padre
Estado (registros)	Estado (registros)	Estado (registros)
Segmentos memoria	Segmentos memoria	Segmentos memoria
Tabla de fd	Tabla de fd	Tabla de fd



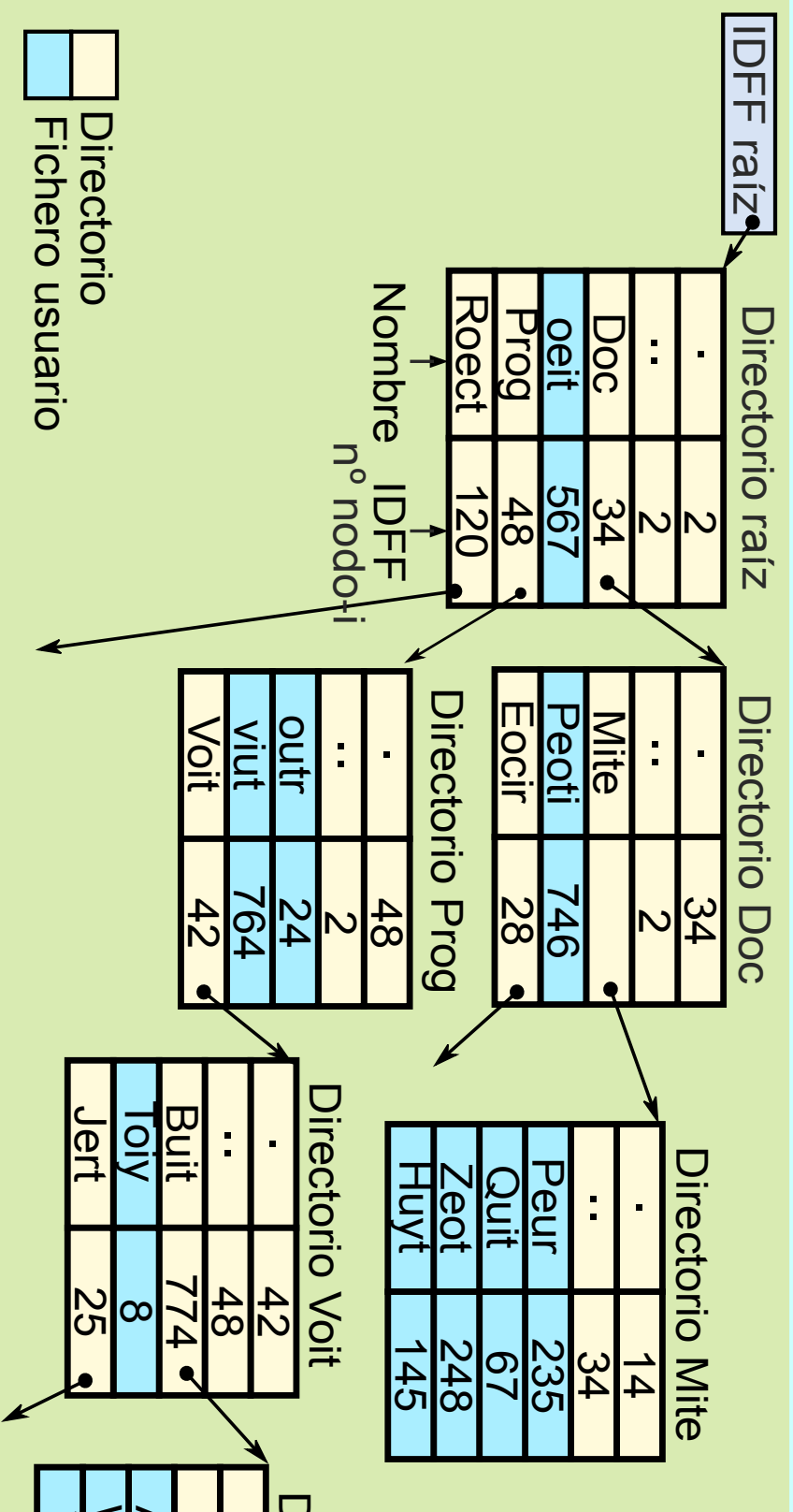
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



- Recorrer las tablas de directorio hasta encontrar el nombre k
- Se analizan los permisos que están almacenados en los DFF
- Se parte del directorio raíz o del directorio de trabajo.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



- La tabla fd existe por proceso y contiene Identificadores Internos
- En memoria hay una tabla intermedia de punteros única que II con los punteros de posición para escritura y lectura y con
- El valor II = 0 indica que ese elemento de la tabla de fd está libre (p.e. el programa usa fd = 2 → II = 4 → IDFF = 678; Pos

Tabla de procesos BCP

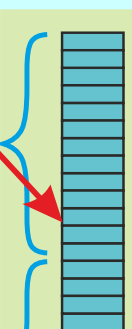
pid, uid, etc.	
Estado	
Segmentos	
Tabla fd	II
0	2
1	463
2	4
3	0
4	68
n	0

Tabla intermedia de punteros

(Única en el sistema)

nºnodo-i Puntero		
0	463	37485
1	3556	0
2	745	47635
3	678	724
4		
n		

Visión del archivo



Agrupaciones
Tipo: norma
Tamaño: 2.7
Nodo-i Vi

Tabla de memoria

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

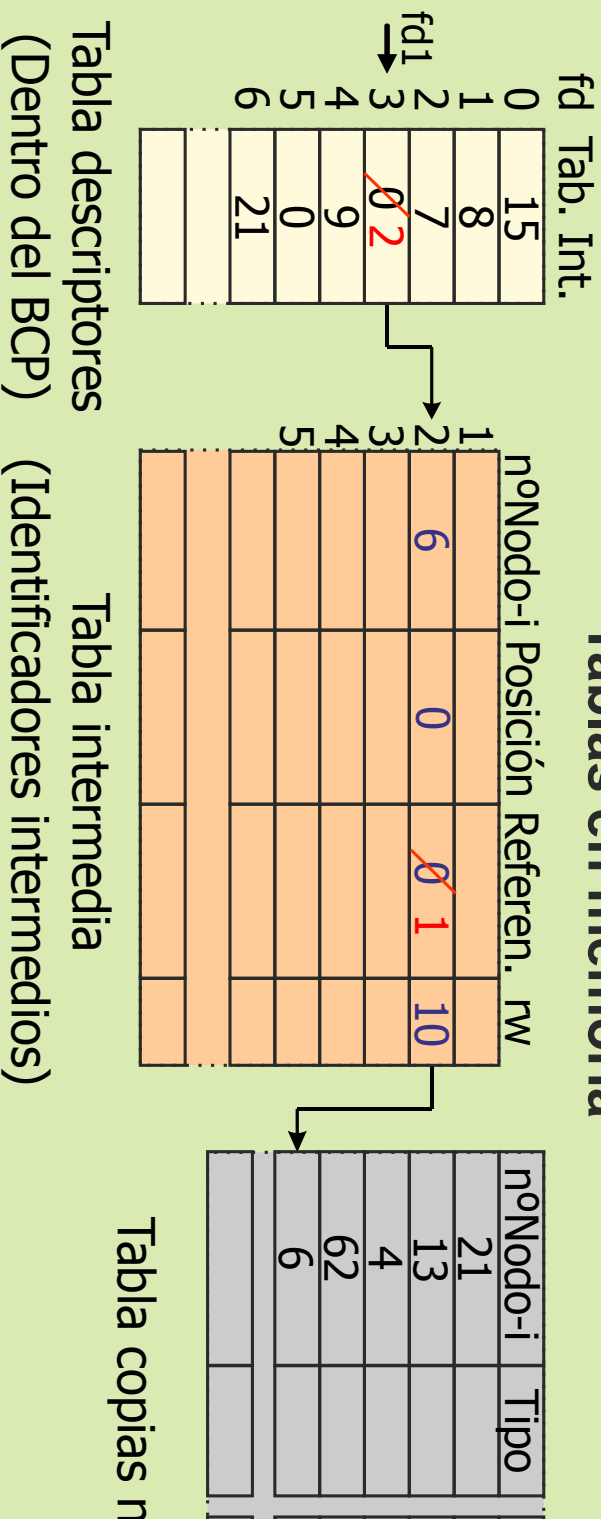
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



- Exige recorrer el árbol de directorios
- Se comprueban los derechos de acceso en cada directorio r
- Se añade una entrada en la tabla de ficheros abiertos que ex del proceso
- Se añade una entrada en la tabla intermedia de punteros, po puntero a 0 y las referencias a 1. (Al crear un hijo se increme campo de referencias)
- Si el fichero no estaba abierto se copia el nodo_i en la tabla residente en memoria y se añade el campo nopens (nº de op nopens = 1
- Si el fichero ya estaba abierto se hace nopens = nopens + 1, cuenta de cuantas veces está abierto el fichero

Tablas en memoria



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



OPEN. ABRE UN FICHERO

Servicio: (**man 2 open**)

```
int open(char *name, int flags, mode_t mode);
```

Argumentos:

- **name**: Nombre del fichero
- **flags**: Opciones de apertura:
 - **O_RDONLY**: Sólo lectura
 - **O_WRONLY**: Sólo escritura
 - **O_RDWR**: Lectura y escritura
 - **O_APPEND**: Se accede siempre al final del fichero
 - **O_CREAT**: Si existe no tiene efecto. Si no existe lo crea
 - **O_TRUNC**: Trunca a cero si se abre para escritura
- **mode**: Bits de permiso para el fichero. Valen sólo cuando se (**O_CREAT**)

Devuelve: Un descriptor de fichero o -1 si hay error o señal (error)

Descripción:

- Se comprueban los derechos de acceso de todo el camino e (absoluto o relativo)
- Se asigna descriptor de fichero en la tabla del BCP, una entrada intermedia y se copia el nodo_i a memoria o incrementa su r

Ejemplos:

```
fd = open("/home/juan/dat.txt", O_RDONLY);  
fd = open("/home/juan/dat.txt", O_WRONLY|O_CREAT|O_TRUNC,
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

CREAT. CREA Y ABRE PARA ESCRITURA UN FICHERO

Servicio:

```
int creat(char *name, mode_t mode);
```

Argumentos:

- **name:** Nombre de fichero
- **mode:** Bits de permiso para el fichero (en caso de nueva cre

Devuelve:

- Devuelve un descriptor de fichero o -1 si fracasa

Descripción:

- Se comprueban los derechos de acceso de todo el camino
- El fichero se abre sólo para escritura
- Si no existe crea un fichero vacío
 - UID_duño = UID_efectivo
 - GID_duño = GID_efectivo
 - Con los permisos indicados enmascarados (mode & ~um
- Si existe lo trunca (tamaño = 0) sin cambiar los bits de perm
- Se asigna descriptor de fichero en la tabla del BCP, una entr intermedia y se copia el nodo_i en memoria o incrementa su

Ejemplos que producen el mismo resultado:

```
fd = creat("datos.txt", 0751); ¿Tienen sentido estos permisos:  
fd = open("datos.txt", O_WRONLY | O_CREAT | O_TRUNC, 0751
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

a



READ. LECTURA DE UN FICHERO*

Servicio:

`ssize_t read(int fd, void *buf, size_t n_bytes);`

Argumentos:

- *fd*: descriptor de fichero
- *buf*: zona donde almacenar los datos
- *n_bytes*: número de bytes a leer

Devuelve:

- Número de bytes realmente leídos o -1 si fracasa

Descripción:

- Transfiere *n_bytes* como máximo
- Puede leer menos datos de los solicitados si se llega al fin de También si lee de un terminal, de un pipe o de un socket
- El servicio puede fracasar por una señal, retornando -1. NO!
- Después de la lectura se incrementa el puntero del fichero *count* de bytes realmente transferidos
- Si retorna 0, indica final de fichero
- En ficheros especiales se queda bloqueado hasta que hay datos

*: Objeto asociado a un descriptor de fichero, en g

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

WRITE. ESCRITURA DE UN FICHERO *

Servicio:

```
ssize_t write(int fd, void *buf, size_t n_bytes);
```

Argumentos:

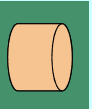
- *fd*: descriptor de fichero
- *buf*: zona de datos a escribir
- *n_bytes*: número de bytes a escribir

Devuelve:

- Número de bytes realmente escritos -1 si fracasa

Descripción:

- Transfiere *n_bytes* o menos
- Si se rebasa el fin de fichero el fichero aumenta de tamaño
- Puede escribir menos datos de los solicitados si se llega al máximo del fichero o se rebasa algún límite de implementación sistema operativo
- El servicio puede fracasar por una señal, retornando -1. NO
- Después de la escritura se incrementa el puntero del fichero número de bytes realmente transferidos



*: Objeto asociado a un descriptor de fichero, en

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



EJEMPLOS

```
int total;

n = read(d, &total, sizeof(int)); /*Se lee un entero
total = 1246; /*Si sizeof(int) = 4 total es 00 00 04 DE
n = write(d, &total, sizeof(int));

/*Se escriben los cuatro bytes 00, 00, 04 y DE, no los
caracteres ASCII 1, 2, 4 y 6 */
float m[160];
n = read(d, m, sizeof(float)*160);
/*Se leen hasta 160 float, lo que pueden ser 4*160 bytes
typedef struct registro{
    int identificador;
    float edad, altura, peso;
} registro;
registro individuo[500];
n = read(d, individuo, 500*sizeof(registro));
estadistica_ciega(individuo, n/sizeof(registro));
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



CLOSE. CIERRE DE UN DESCRIPTOR DE FICHERO

Servicio:

```
int close (int fd) ;
```

Argumentos:

- *fd*: descriptor de fichero

Devuelve:

- 0 o -1 si fracasa

Descripción:

- El proceso pierde la asociación entre el descriptor y el fichero correspondiente)
- Se decrementa el número de referencias en la tabla intermedia referencias = 0 entonces:
 - Se libera la correspondiente entrada de la tabla intermedia
 - Se decrementa *nopens* (número de duplicados) de la tabla *nodos_i*
- Si *nopens* = 0 se libera la entrada en la tabla copias nodo

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



CLOSE. CIERRE DE UN DESCRIPTOR DE FICHERO

BCP A	BCP B	BCP C
fd	fd	fd
0 11	0 15	0 15
1 34	1 48	1 48
2 34	2 48	2 48
3 12	3 2	3 2
4 3	4 9	4 9
5 0	5 0	5 0
6 0	6 5	6 5

Tablas en memoria

no	Nodo-i	Posición	Referen.	rw
1				
2	6	1827	2	11
3	6	574	1	10
4				
5	14	47	2	10
6				

Tabla intermedia

no	Nodo-i	Tipo	nopens
21			
13			
14			1
62			
6			2

Tabla copias nodos-i

Proceso B ejecuta: close (3) ;

BCP A	BCP B	BCP C
fd	fd	fd
0 11	0 15	0 15
1 34	1 48	1 48
2 34	2 48	2 48
3 12	3 2	3 2
4 3	4 9	4 9
5 0	5 0	5 0
6 0	6 5	6 5

Tablas en memoria

no	Nodo-i	Posición	Referen.	rw
1				
2	6	1827	2	11
3	6	574	1	10
4				
5	14	47	2	10
6				

Tabla intermedia

no	Nodo-i	Tipo	nopens
21			
13			
14			1
62			
6			2

Tabla copias nodos-i

Proceso C ejecuta: close (3) ;

BCP A	BCP B	BCP C
fd	fd	fd
0 11	0 15	0 15
1 34	1 48	1 48
2 34	2 48	2 48
3 12	3 0	3 2
4 3	4 9	4 9
5 0	5 0	5 0
6 0	6 5	6 5

Tablas en memoria

no	Nodo-i	Posición	Referen.	rw
1				
2	6	1827	1	11
3	6	574	1	10
4				
5	14	47	2	10
6				

Tabla intermedia

no	Nodo-i	Tipo	nopens
21			
13			
14			1
62			
6			2

Tabla copias nodos-i

Tabla de procesos

Tabla intermedia (Identificadores intermedios)

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



EJEMPLO FICHEROS. COPIA UN FICHERO EN OTRO I

```
#define BUFSIZE 4096

int main(int argc, char **argv) {
    int fd_ent, fd_sal, n_read;
    char buffer[BUFSIZE];

    //Abre fichero origen o entrada
    fd_ent = open(argv[1], O_RDONLY);
    if (fd_ent < 0) { //error apertura
        perror("open"); //transforma errno en un mensaje de
        exit(1);
    }

    //Crea fichero de destino o salida
    fd_sal = creat(argv[2], 0666);
    if (fd_sal < 0) { //error apertura
        close(fd_ent);
        perror("creat");
        exit(1);
    }
}
```

Los nombres de los
pasan como argum

Sigue página sigui

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



EJEMPLO FICHEROS. COPIA UN FICHERO EN OTRO II

```
/* bucle de lectura del fichero de entrada */
while ((n_read = read(fd_ent, buffer, BUFSIZE)) > 0)
/* escribir el buffer al fichero de salida */
    if (write(fd_sal, buffer, n_read) < n_read) { //e
        perror("write");
        close(fd_ent);
        close(fd_sal);
        exit(1);
    }
}
close(fd_ent);
close(fd_sal);
if (n_read < 0) { //se trata el error en lectura
    perror("read");
    exit(1);
}
return 0;
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



LSEEK. MODIFICA EL PUNTERO DE POSICIÓN

Servicio:

`off_t lseek (int fd, off_t offset, int whence) ;`

Argumentos:

- *fd*: Descriptor de fichero
- *offset*: desplazamiento (positivo o negativo)
- *whence*: base del desplazamiento

Devuelve:

- La nueva posición del puntero o -1 si fracasa (terminal, pipe, FIFO errno=EPIPE).

Descripción:

- Coloca el puntero de acceso asociado a *fd*
- La nueva posición, que no puede ser negativa, se calcula se:
 - SEEK_SET: posición = offset
 - SEEK_CUR: posición = posición actual + offset
 - SEEK_END: posición = final del fichero + offset

- Nos podemos salir del tamaño del fichero (no se aumenta el que se haga una escritura posterior)

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

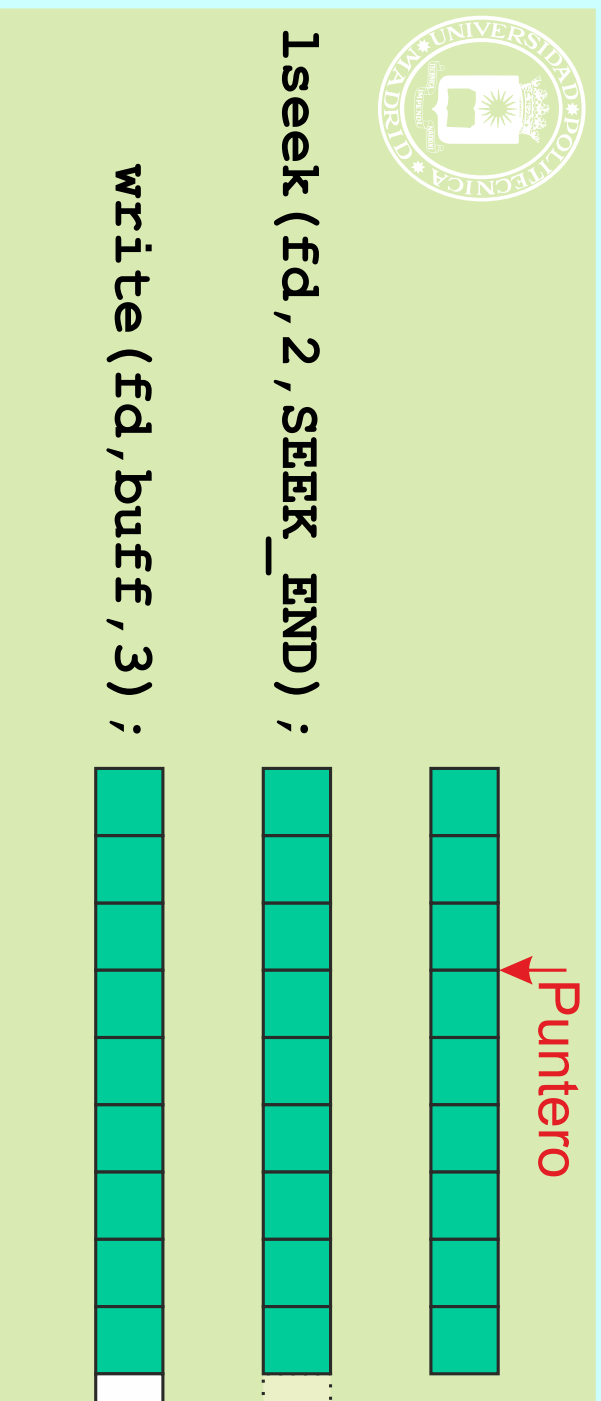
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



LSEEK. EJEMPLOS

- Tamaño de fichero:
`tam = lseek (fd, 0, SEEK_END) ;` ← ¿qué pasa si hay huecos?
- Posición actual:
`pos = lseek (fd, 0, SEEK_CUR) ;`
- Creación de huecos:



- Al escribir es cuando se aumenta el tamaño real
- El tamaño ocupado puede ser < que el real (agrupaciones de asignadas)
- Si se lee del hueco no escrito se obtienen nulos (0x00)

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

DUP. DUPLICA UN DESCRIPTOR DE FICHERO

Servicio:

`int dup (int fd) ;`

Argumentos:

- *fd*: descriptor de fichero

Devuelve:

- Un descriptor de fichero que comparte todas las propiedades del *fd* o -1 si fracasa

Descripción:

- Crea un nuevo descriptor de fichero que tiene en común con
 - Accede al mismo fichero
 - Comparte el mismo puntero de posición
 - El modo de acceso es idéntico
- El nuevo descriptor tendrá el menor valor numérico posible
- Se incrementa en 1 el número de referencias (tabla intermed

fd
0
1
2
3
4
5
6
n

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



DUP2. DUPLICA UN DESCRIPTOR DE FICHERO

Servicio:

`int dup2 (int oldfd, int newfd) ;`

Argumentos:

- *oldfd*: descriptor de fichero existente
- *newfd*: nuevo descriptor de fichero

Devuelve:

- El nuevo descriptor de fichero o -1 si fracasa

Descripción:

- Crea un nuevo descriptor de fichero, cuyo número es *newfd*, común con *oldfd*:
 - Accede al mismo fichero
 - Comparte el mismo puntero de posición
 - El modo de acceso es idéntico
- Si *newfd* estaba abierto, lo cierra antes de realizar el duplica
- Se incrementa en 1 el número de referencias (tabla intermed

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

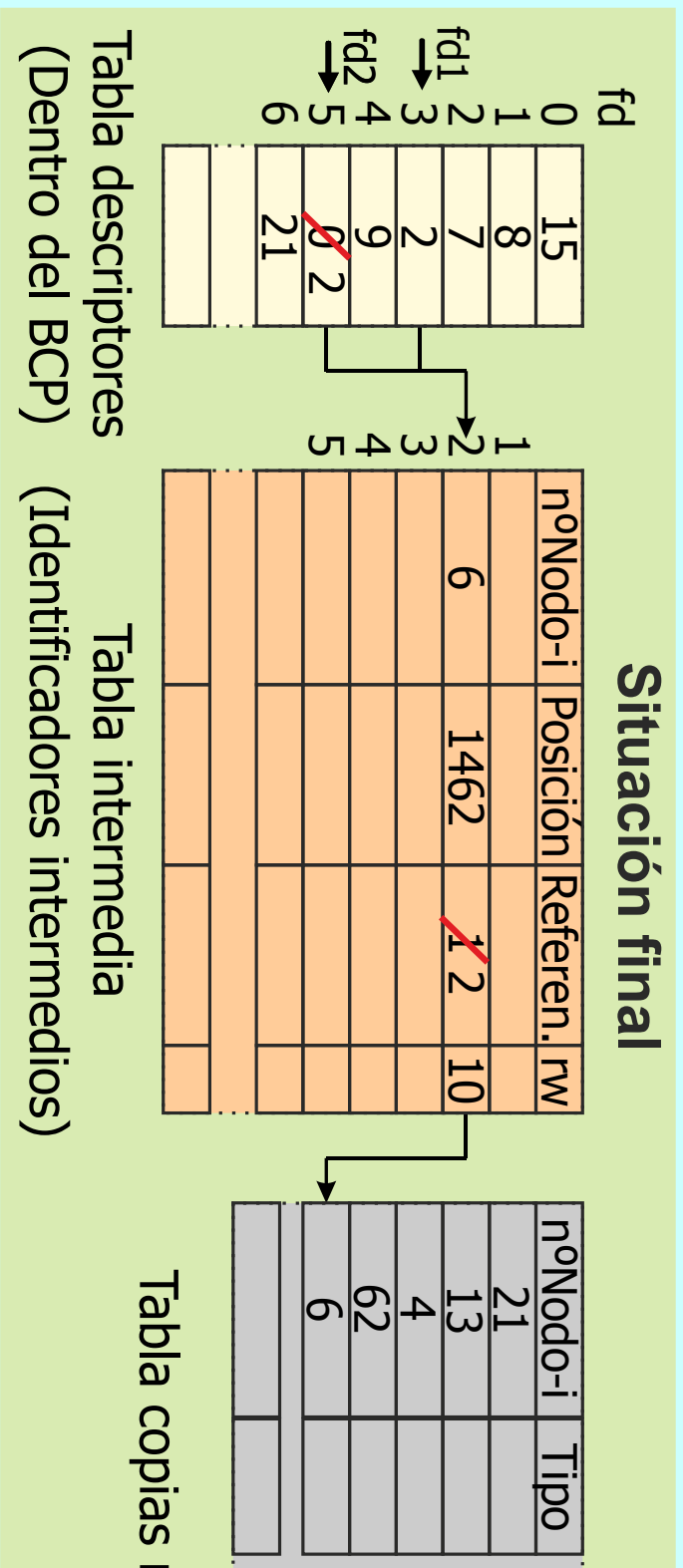
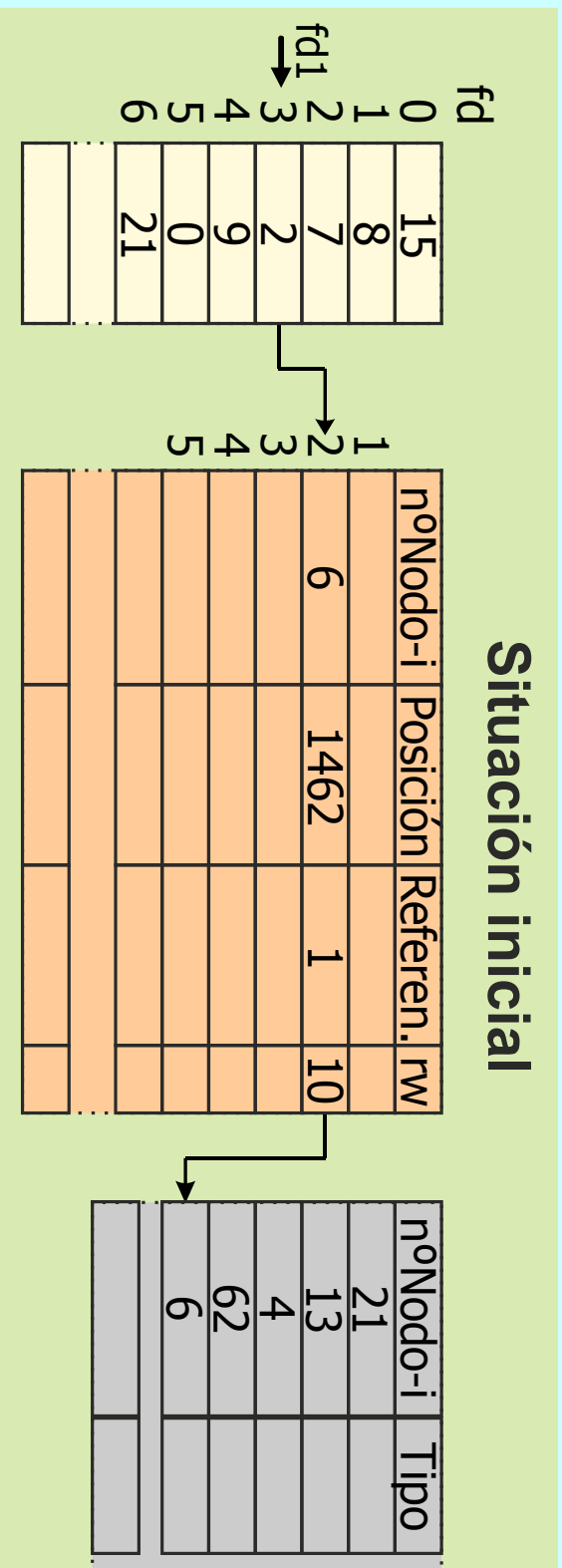
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



GESTIÓN DE PUNTEROS. DUP

fd2 = **dup** (fd1) ;



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70

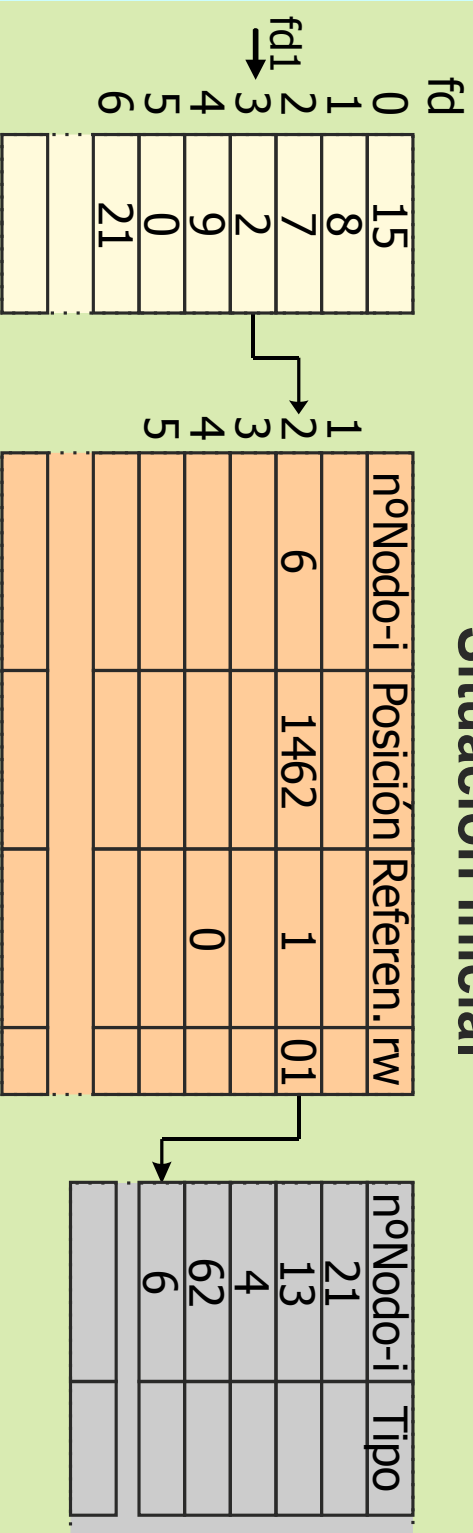


GESTIÓN DE PUNTEROS. OPEN

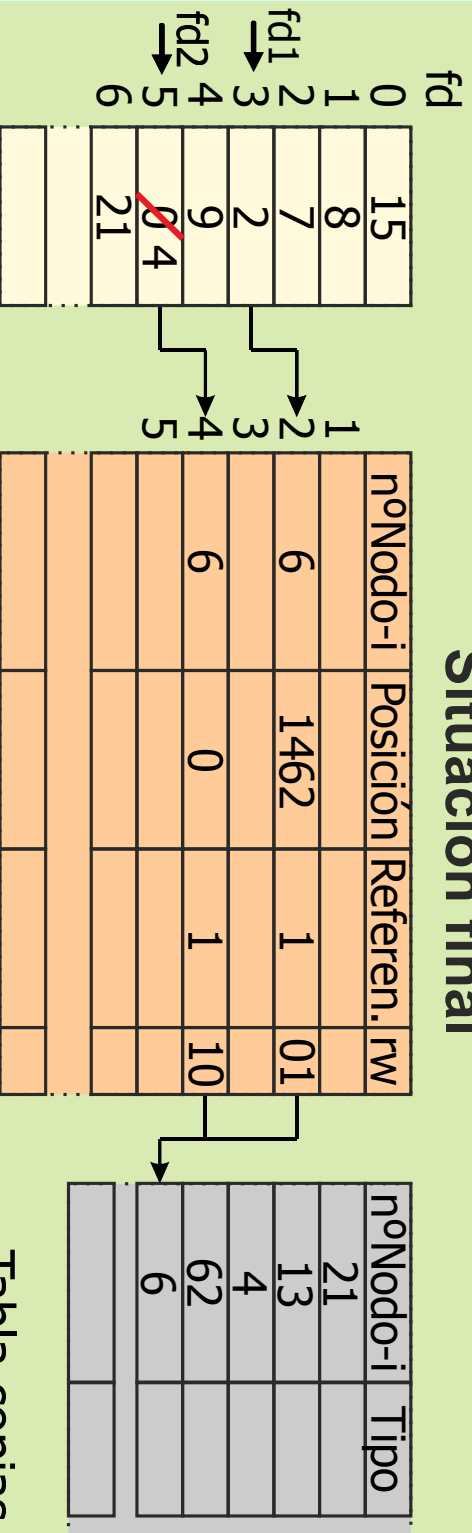
El mismo proceso vuelve a abrir el mismo fichero datos.txt de n

fd2 = **open** ("datos.txt", O_RDONLY);

Situación inicial



Situación final



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

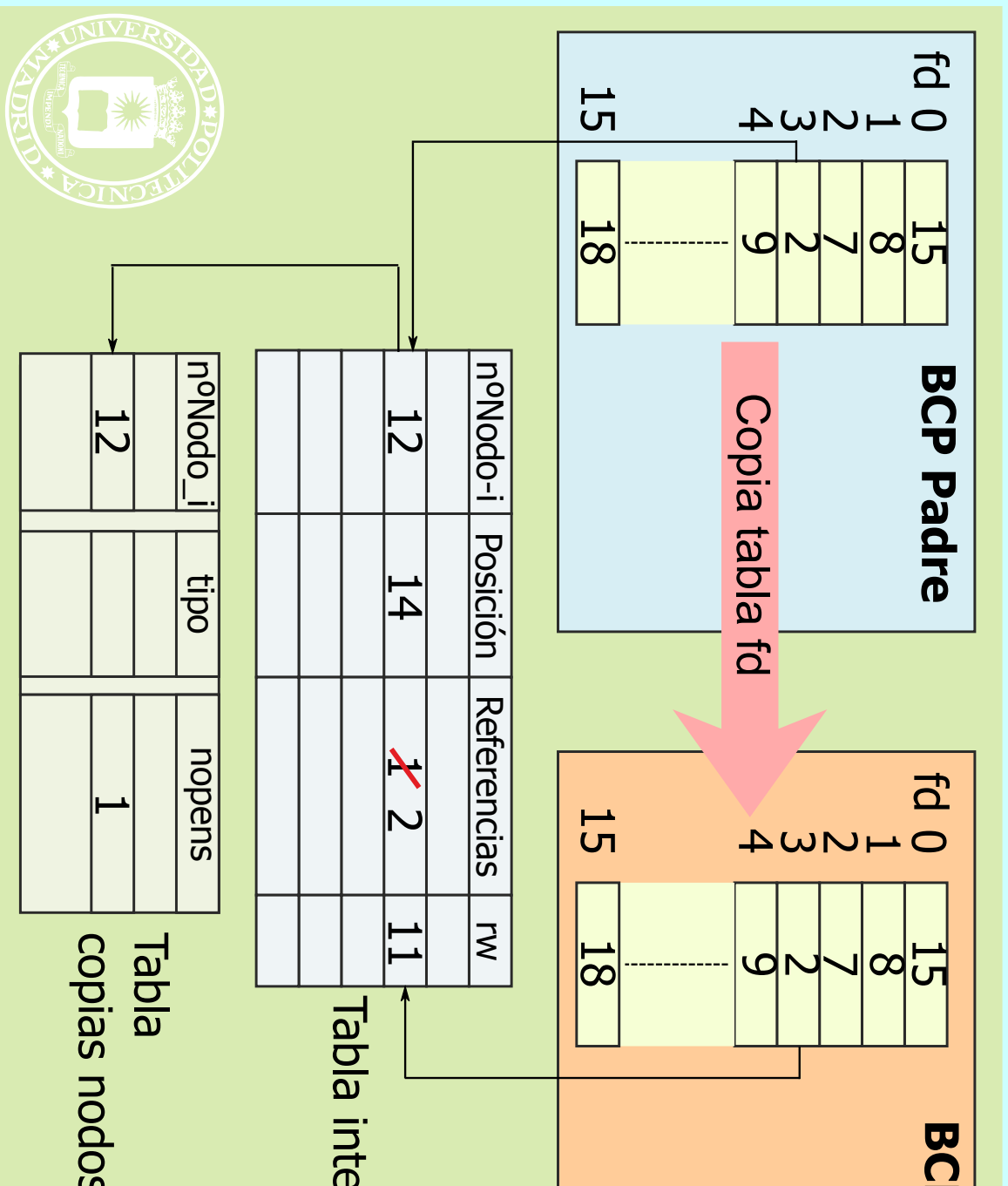
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



GESTIÓN DE PUNTEROS (FORK)

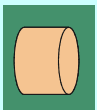
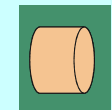
pid = **fork()** ; /*crea proceso hijo */



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



EJEMPLO: REDIRECCIONES

Programa que ejecuta: `ls > fichero`

```
int main(void) {
    pid_t pid;
    int status, fd;
    fd = open("fichero", O_WRONLY|O_CREAT|O_TRUNC, 0666);
    if (fd < 0) {perror("open"); exit(1);}
    pid = fork();
    switch(pid) {
        Case -1: // error
            perror("fork"); exit(1);
        Case 0: //proceso hijo ejecuta "ls"
            close(1);
            dup(fd); //descriptor se copia al 1
            close(fd);
            execlp("ls", "ls", NULL);
            perror("execlp"); exit(1);
        Default: //proceso padre
            close(fd);
            while (pid != wait(&status));
    }
    return 0;
}
```

fd	Proc P
0	teclado
1	monitor
2	monitor
3	2
4	9
5	0
6	21

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



EJEMPLO: REDIRECCIONES

Programa que ejecuta: `ls > fichero`

```
int main(void) {
    pid_t pid;
    int_status, fd;
    fd = open("fichero", O_WRONLY|O_CREAT|O_TRUNC,
    if (fd < 0) {
        perror("open"); exit(1);
    }
    pid = fork();
    switch(pid) {
        case -1: /* error */
            perror("fork"); exit(1);
        case 0: /* proceso hijo ejecuta "ls" */
            dup2(fd, 1); /* el descriptor se copia en el
            close(fd);
            execlp("ls", "ls", NULL);
            perror("execlp");
            exit(1);
        default: /* proceso padre */
            close(fd);
            while (pid != wait(&status));
    }
    return 0;
}
```

The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized, outlined font with a blue and orange gradient background.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70





SERVICIOS UNIX SOBRE DIRECTORIO

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

OPENDIR. ABRE UN DIRECTORIO

El fichero directorio es un fichero de registros de tipo `struct dirent` pero incluye:

- `ino_t d_ino`: número de nodo
- `ino_t d_ino`: nombre del fichero (de tamaño variable)

Servicio:

```
DIR *opendir (char *dirname) ;
```

Argumentos:

- `dirname`: Nombre del directorio

Devuelve:

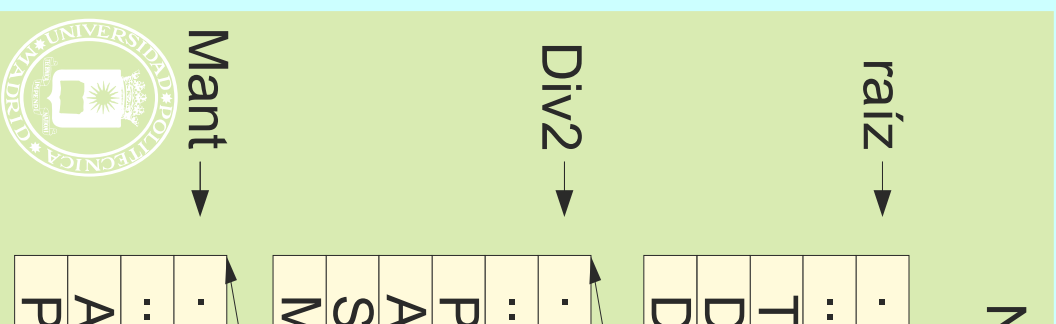
- Un puntero para utilizarse en `readdir()`, `rewindir()` o `closedir()` y `NULL` si fracasa

Descripción:

- Abre un directorio y se coloca en el primer registro o entrada del mismo

Permisos:

- Permiso de lectura de `dirname`



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

READDIR. LECTURA DE UN REGISTRO DE DIRECTORIO

Servicio:

`struct dirent *readdir(DIR *dirp);`

Argumentos:

- `dirp`: puntero devuelto por `opendir()`

Devuelve:

- Un puntero a un objeto del tipo `struct dirent` que representa de directorio o `NULL`, si hubo error o se ha llegado al final del directorio

Descripción:

- Devuelve el siguiente registro del directorio `dirp`
- Avanza el puntero al siguiente registro

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



CLOSEDIR. CIERRA UN DIRECTORIO

Servicio:

`int closedir (DIR *dirp) ;`

Argumentos:

- *dirp*: puntero devuelto por *opendir* ()

Devuelve:

- Cero o -1 si fracasa

Descripción:

- Cierra la asociación entre *dirp* y la secuencia de entradas d

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, green, serif font. The '99' is significantly larger and more prominent than the 'Cartagena' part. The text is set against a light blue background with a white arrow pointing to the right, which is partially obscured by the text.

REWINDIR. INICIALIZA EL PUNTERO DEL DIRECTORIO

Servicio:

void rewindir (DIR *dirp) ;

Argumentos:

- *dirp* puntero devuelto por *opendir ()*

Descripción:

- Sitúa el puntero de posición del directorio en el primer registro.

The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized, green, serif font with a shadow effect, set against a light blue and orange background.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



MKDIR. CREA UN DIRECTORIO

Servicio:

```
int mkdir (char *name, mode_t mode);
```

Argumentos:

- **name:** nombre del directorio
- **mode:** bits de protección

Devuelve:

- Cero o -1 si fracasa

Descripción:

- Crea un directorio de nombre name y permisos mode & ~umk
- UID_ dueño directorio = UID_efectivo proceso
- GID_ dueño directorio = GID_efectivo proceso

Permisos

- Escritura en directorio donde se añade name

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



RMDIR. BORRA UN DIRECTORIO

Servicio:

```
int rmdir (char *name) ;
```

Argumentos:

- **name:** nombre del directorio

Devuelve:

- Cero o -1 si fracasa

Descripción:

- Borra el directorio si está vacío
- Si el directorio no está vacío no se borra

Permisos

- Escritura en directorio de donde se elimina name

The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized, green, serif font with a slight shadow effect, set against a light blue and white background with a subtle arrow shape pointing right.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



PROGRAMA QUE LISTA UN DIRECTORIO

```
#define MAX_BUF 256
int main(int *argc, char **argv) { /* directorio se pasa
argumento */
    DIR *dirp;
    struct dirent *dp;
    char buf[MAX_BUF];
    /* imprime el directorio actual */
    getcwd(buf, MAX_BUF);
    printf("Directorio actual: %s\n", buf);
    /* abre el directorio pasado como argumento */
    dirp = opendir(argv[1]);
    if (dirp == NULL) {
        fprintf(stderr, "No puedo abrir %s\n", argv[1]);
    } else {
        /* lee entrada a entrada */
        while ((dp = readdir(dirp)) != NULL)
            printf("%s\n", dp->d_name);
        closedir(dirp);
    }
    return 0;
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

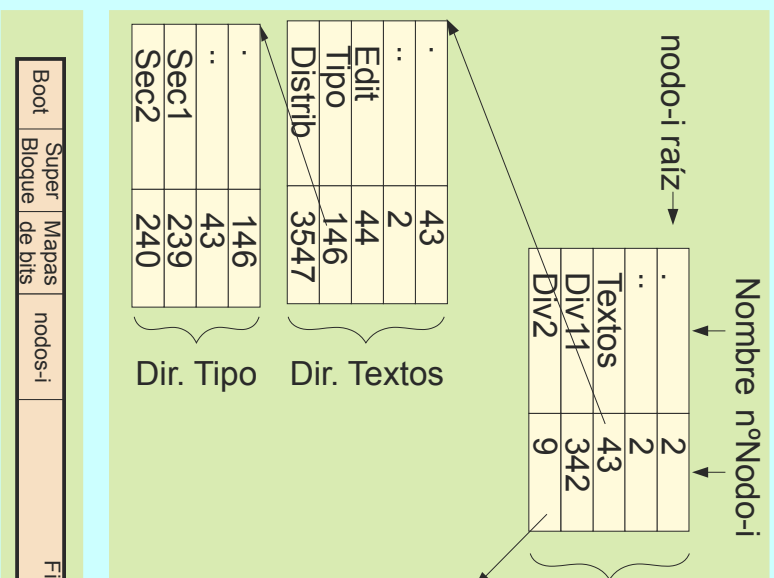
Cartagena99



ENLACES

Permite que dos o más nombres hagan referencia al mismo fichero

- Enlace **físico**
 - El fichero sólo se elimina cuando se borran todos sus nombres
 - Contador en el nodo-i del fichero. Necesario para saber cuándo puede eliminarse un fichero. Con contador = 0 se elimina el fichero
 - Sólo se permiten enlazar ficheros y subdirectorios del mismo nivel
 - No perder de vista que los nombres **.** y **..** también cuentan.
- Enlace **simbólico**
 - El fichero se elimina cuando el contador de enlaces físicos = 0. Si permanece el enlace simbólico se provocará un error al tratar de abrirlo
 - Se puede hacer con ficheros y directorios
 - Se puede hacer entre ficheros de diferentes volúmenes



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

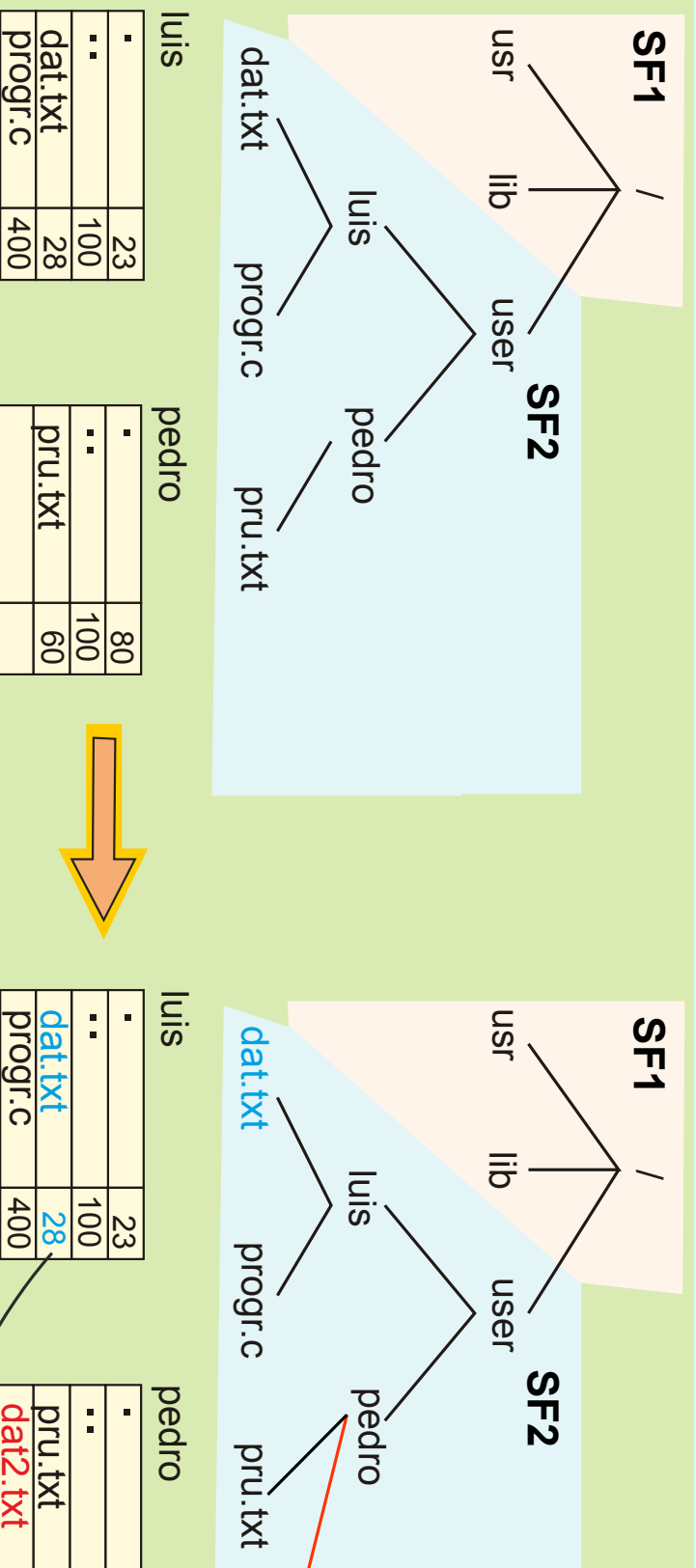
Cartagena99

Dir. Mant Dir. Div2



ENLACE FÍSICO

- Dos entradas de directorio al mismo nodo-i
- Solamente para ficheros de un mismo sistema de ficheros
- Hay que incrementar el contador de enlaces en el nodo-i



Mandato: ln /user/luis/dat.txt /user/pedro/dat2.txt
Servicio: link ("/user/luis/dat.txt", "/user/pedro/da

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



LINK. CREA UNA ENTRADA (ENLACE) DE DIRECTORIO

Servicio:

`int link (char *existing, char *new) ;`

Argumentos:

- *existing*: nombre del fichero existente
- *new*: nombre de la nueva entrada que será un enlace al fichero. No puede existir previamente

Devuelve: Cero o -1 si fracasa

Descripción:

- Crea un nuevo **enlace físico** para un fichero existente
- Requiere permisos de búsqueda para *existing* y de escritura para *new* en el directorio donde esté *new*
- El sistema no registra cuál es el enlace original
- *existing* no debe ser el nombre de un directorio salvo que se le otorga privilegio suficiente y la implementación soporte el enlace de directorio. El enlace de directorios puede dar lugar a ciclos en el árbol de directorios.
- Las entradas `.` y `..` son enlaces físicos de directorio, pero corren por el sistema operativo

El `open` y `creat` también pueden crear una entrada de directorio.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

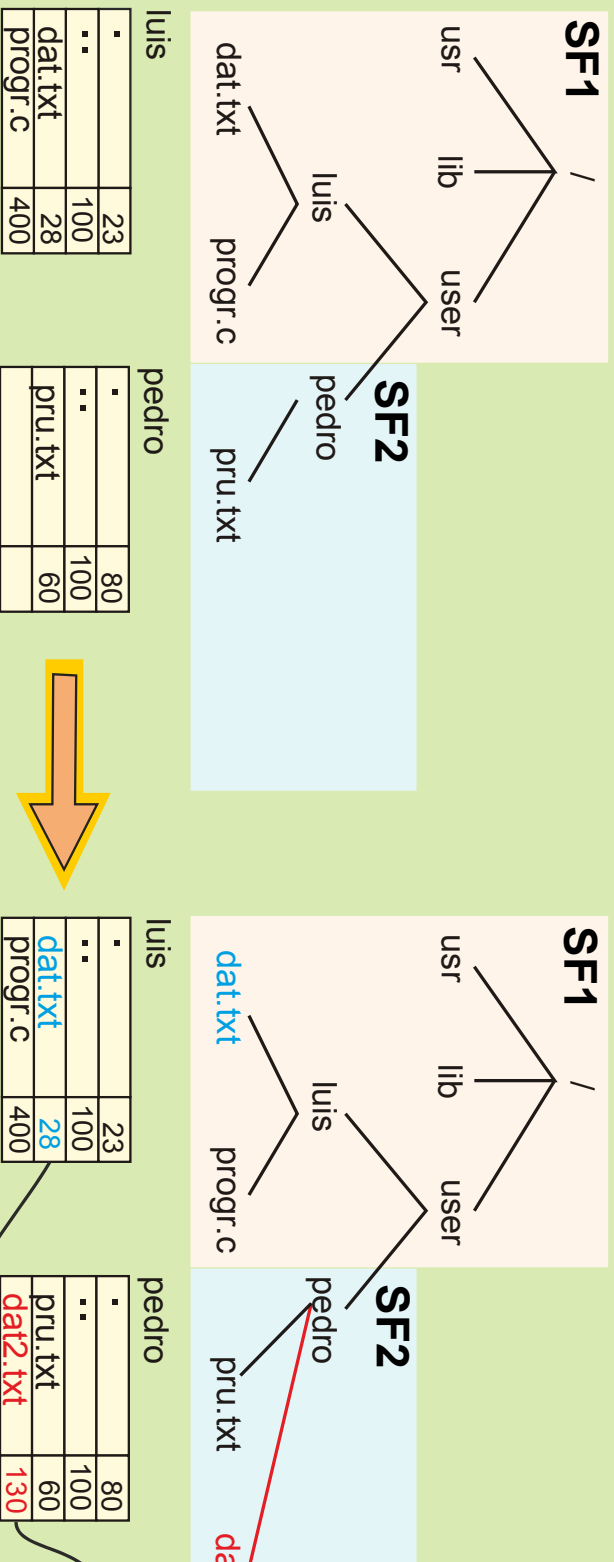
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



ENLACE SIMBÓLICO

- Para ficheros y directorios de un mismo o diferentes SF
- Se utiliza un nodo-i para almacenar la referencia
- No se incrementa el contador de enlaces del fichero
- Para abrir un fichero se comprueban los derechos del camino del enlace simbólico



Mandato: `ln -s /user/luis/dat.txt /user/pedro/dat2.txt`
Servicio: `symlink ("/user/luis/dat.txt", "/user/pedro/dat2")`

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



SYMLINK. CREA UNA ENTRADA DE DIRECTORIO

Servicio:

`int symlink (char *existing, char *new) ;`

Argumentos:

- *existing*: nombre del fichero existente (no se comprueba que exista)
- *new*: nombre de la nueva entrada que será un enlace al fichero *existing*. Es necesario tener permisos de escritura en el directorio

Devuelve:

- Cero o -1 si fracasa

Descripción:

- Crea un nuevo enlace simbólico físico para un fichero o directorio existente
- Requiere permiso de escritura en el directorio donde esté el enlace
- El fichero o directorio puede desaparecer dejando el enlace
- Se pueden crear ciclos en el árbol de directorio, lo que da lugar a problemas de acceso
- Para abrir un fichero se comprueban también los derechos de acceso
- Una vez creado el enlace, la gran mayoría de los servicios que se ejecutan como argumento se realizarán realmente sobre *existing*. Por ejemplo, `chmod` se hará sobre *existing*

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



UNLINK. ELIMINA UNA ENTRADA DEL DIRECTORIO

Servicio:

`int unlink (char *name) ;`

Argumentos:

- `name`: nombre de fichero

Devuelve:

- Cero o -1 si fracasa

Descripción:

- Elimina la entrada del directorio y decrementa el número de fichero correspondiente
- Cuando el número de enlaces es igual a cero:
 - Si ningún proceso lo mantiene abierto, se libera el espacio ocupado por el fichero
 - Si algún proceso lo mantiene abierto, se conserva el fichero lo cierren todos
- Si se hace sobre un enlace simbólico se decrementa el n° de nodo_*i* simbólico. Cuando llega a 0 se libera dicho nodo_*i*. Nada sobre el fichero enlazado
- Permiso de escritura en directorio de donde se elimina `name`



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



Servicio:

`int chdir (char *name) ;`

Argumentos:

- *name*: nombre de un directorio

Devuelve:

- Cero o -1 si fracasa

Descripción:

- Modifica el directorio actual o de trabajo, aquel a partir de forman los nombres relativos

Permisos:

- Requiere permisos de búsqueda para *name*

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



RENAME. CAMBIA EL NOMBRE DE UN FICHERO

Servicio:

`int rename (char *old, char *new) ;`

Argumentos:

- **old:** nombre de un fichero existente
- **new:** nuevo nombre del fichero

Devuelve:

- Cero o -1 si fracasa

Descripción:

- Cambia el nombre del fichero `old`. El nuevo nombre es `new`

Permisos:

- Permiso de escritura en los directorios que contienen a `old`

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



GETCWD. OBTIENE EL NOMBRE DEL DIRECTORIO ACTU

Servicio:

```
char *getcwd(char *buf, size_t size);
```

Argumentos:

- *buf*: dirección del espacio donde se va a almacenar el nombre del directorio actual
- *size*: longitud en bytes de dicho espacio

Devuelve:

- Puntero a *buf* o **NULL** si fracasa (p.e. si el nombre ocupa r

Descripción:

- Obtiene el nombre del directorio actual o de trabajo

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

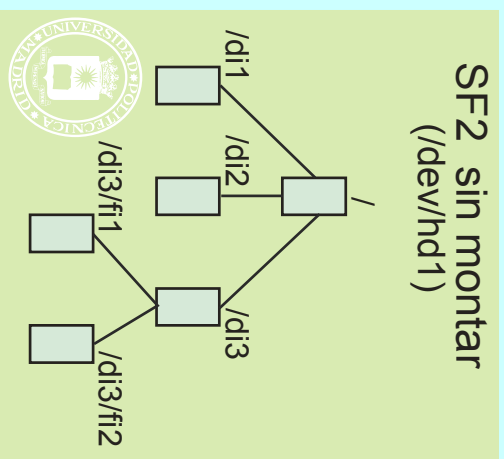
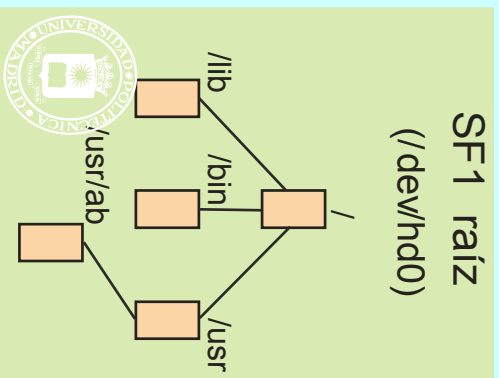
...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

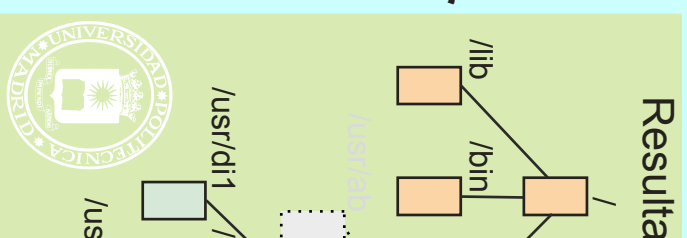
The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized, green, serif font with a slight shadow effect, set against a light blue and white background with a subtle arrow shape pointing to the right.

MONTADO DE VOLÚMENES O PARTICIONES

- Añade a un árbol de directorios los directorios de un dispositivo de ficheros virtual conjunto
- El montado es una operación realizada en memoria. No se modifica el sistema de ficheros en los discos
- Oculta el nombre del dispositivo físico o partición que se monta



mount /dev/hd1 /usr



¿Qué pasa con los números de los nodos-i?

- Si en el directorio de montaje existen ficheros o directorios que son inaccesibles. (el directorio /usr/ab ya no está disponible). Los directorios desmontados vuelven a ser accesibles.
- En UNIX /etc/fstab contiene los sistemas de ficheros disponibles y las opciones de montaje para el mandato mount

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

/fi2

MOUNT. MONTA UN SISTEMA DE FICHEROS

Servicio:

```
int mount (const char *source, const char *target,
           const char *filesystemtype,
           unsigned long mountflags, const void *
```

Argumentos:

- **source** : sistema de ficheros que se monta, generalmente un como /dev/cdrom, /dev/hdb0, /dev/sda1 o /home/image
- **target** : directorio sobre el que se monta
- **filesystemtype** : tipo de sistema de ficheros, como "minix", "ext4", "msdos", "vfat", "proc" o "nfs"
- **mountflags** : Opciones como:
 - MS_NOEXEC enmascara los bits de ejecución del sistema
 - MS_NOSUID enmascara los bits SUID y SGID del sistema
 - MS_RDONLY montado para lectura solamente
- **data** : opciones que dependen del tipo de sistema de fichero

Devuelve:

Cero o -1 si fracasas

Descripción:

- Monta un sistema de ficheros
- Requiere privilegios de superusuario

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



No confundir el servicio mount con el mandato mount

- Cada sistema de ficheros numera los nodos_i empezando en 0
- Al montar sistemas de ficheros, los n° de nodo_i no son únicos
- Para identificar un nodo_i es necesario especificar el sistema de ficheros y en n° de nodo_i dentro de él. Esto es necesario, por ejemplo en la tabla de copias de nodos_i
- Si en el directorio de montaje existen ficheros o directorios con el mismo nombre que los de los sistemas de ficheros, estos son inaccesibles

Tablas en memoria

fd Tab. Int.

0	15
1	8
2	7
3	2
4	9
5	0
6	21

SF-nºNodo-i Posición Referen. rw

1	hda-6	28373	1	11
2	hdb-37	3847	1	10
3	hda-43	7635	0	01
4	hda-6	0	1	10
5	hdb-6	56	1	11
	hdb-234	0	1	10
	hda-238	0	0	10

SF-nºNodo-i

hda-21
hdb-13
hda-4
hdb-6
hda-6

Tabla descriptores
(Dentro del BCP)

Tabla intermedia
(Identificadores intermedios)

Tabla co

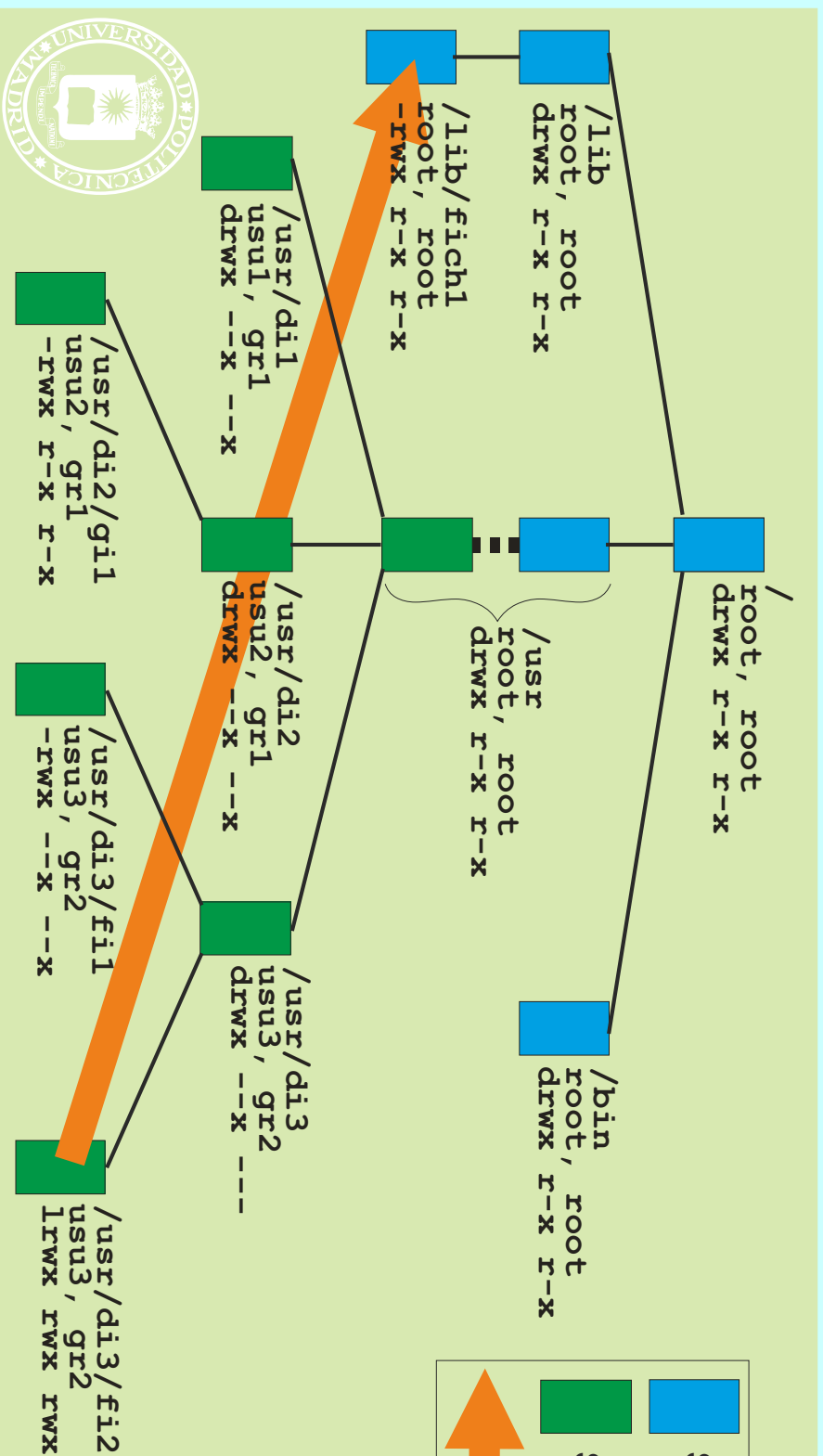
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



UNIX: PERMISOS EN VOLÚMENES MONTADOS



```
$ ls -l "/usr/di3/fi2"
```

```
lrwxr-xr-x root root 10 Mar 11 2004 /usr/di3/fi2 ->
```

Para abrir /usr/di3/fi2 se comprueban los permisos de:

“/” (raíz SF A), “usr/” (raíz SF B), “di3/”, “/” (raíz SF A), “lib/”, “fic/” (no se comprueban los permisos del nodo_i de “fi2” ni del nodo_“usr/”)

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

UMOUNT. DESMONTA UN SISTEMA DE FICHEROS

Servicio:

`int umount (const char *target) ;`

Argumentos:

- *target* : directorio que se desmonta

Devuelve:

- Cero o -1 si fracasa

Descripción:

- Desmonta un sistema de ficheros
- Si hay ficheros abiertos no desmonta y devuelve error
- Requiere privilegios de superusuario

The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized, green, serif font with a blue shadow effect, set against a white background with a blue and orange arrow-like shape behind it.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70





SERVICIOS UNIX PARA PROTECCIÓN

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

ACCESS. ACCESIBILIDAD SOBRE UN FICHERO

Servicio:

```
int access (char *name, int amode);
```

Argumentos:

- **name**: nombre del fichero o directorio
 - **amode**: modo de acceso que se quiere comprobar. amode es de R_OK, W_OK, X_OK o F_OK (comprobar existencia)
- Devuelve:
- 0 si el proceso tiene acceso al fichero (para lectura, escritura -1 en caso contrario)

Descripción:

- Utiliza el UID real y el GID real (en lugar de los efectivos) para derechos de acceso sobre un fichero

Ejemplos:

- **access** ("fichero", F_OK); devuelve 0 si el fichero existe
- **access** ("fichero", R_OK|W_OK); devuelve 0 si el proceso llamada tiene permisos de acceso de lectura y escritura sobre (utilizando el UID real y el GID real)

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



ACCESS. Ejemplo

Ejemplo de uso `access` para que un programa que tenga activo `SETUID` o el de `SETGID`.

Antes de abrir un fichero se comprueba si el usuario real tiene el acceso. Solamente se abrirá en dicho caso.

```
if (0 == access ("archivo", R_OK|W_OK)) {
    /* Se realiza la apertura del fichero */
    fd7 = open ("archivo", O_RDWR);
    .....
} else {
    /* El acceso es negativo, por lo que no hace el
    .....
}
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



UMASK. ASIGNA LA MASCARA DE CREACION

Servicio:

`mode_t umask (mode_t cmask) ;`

Argumentos:

- `cmask`: bits de permiso a desasignar en la creación de fichero

Devuelve:

- Devuelve la máscara previa

Descripción:

- Asigna la máscara de creación de ficheros del proceso que llama
- Los bits activos en la máscara son desactivados en la palabra de protección del fichero (`permisos = mode & ~umask`)
 - Si máscara = 022, y se crea un fichero con permisos 077, los permisos con los que se crea realmente el fichero son 075
- La máscara está almacenada en el BCP del proceso

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



CHMOD. CAMBIA LOS BITS DE PERMISO

Servicio:

```
int chmod (char *name, mode_t mode);
```

Argumentos:

- **name:** nombre del fichero objetivo
- **mode:** Nuevos bits de protección

Devuelve:

- Cero o -1 si fracasa

Descripción:

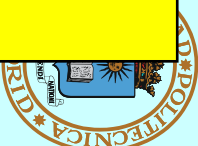
- Modifica los bits de permiso y los bits SETUID y SETGID del
 - S_ISUID = 04000
 - S_ISGID = 02000
- Sólo el propietario del fichero o el administrador pueden cambiar bits
- **NO se utiliza la máscara**

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



int chmod(char *name, mode_t mode);

Servicio:

int chown (char *name, uid_t owner, gid_t group);

Argumentos:

- **name:** nombre del fichero
- **owner:** nuevo propietario del fichero
- **group:** nuevo identificador de grupo del fichero

Devuelve:

- Cero o -1 si fracasa

Descripción:

- Modifica el identificador de usuario y de grupo del fichero
- Los bits SETUID y SETGID son borrados
- Sólo el propietario del fichero o el administrador pueden cambiar atributos

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99





SERVICIOS UNIX SOBRE ATRIBUTO

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

STAT, FSTAT. INFORMACIÓN SOBRE UN FICHERO

Servicios:

```
int stat(char *name, struct stat *buf);  
int fstat(int fd, struct stat *buf);  
int lstat(char *name, struct stat *buf);
```

Argumentos:

- **name**: nombre del fichero
- **fd**: descriptor de fichero
- **buf**: puntero a un objeto de tipo `struct stat` donde se almacena información del fichero
- **lstat** se diferencia del **stat** en que devuelve el estado del **link** simbólico y no del fichero al que apunta dicho enlace, como **stat**

Devuelve:

- Cero o -1 si fracasa

Descripción:

- Obtiene información sobre un fichero y la almacena en una estructura de tipo `struct stat`

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



STAT, FSTAT. INFORMACIÓN SOBRE UN FICHERO

```
struct stat {
    mode_t      st_mode;      /* tipo de fichero + per
    ino_t       st_ino;      /* número del fichero
    dev_t       st_dev;      /* dispositivo */
    nlink_t     st_nlink;    /* número de enlaces */
    uid_t       st_uid;      /* UID del propietario
    gid_t       st_gid;      /* GID del propietario
    off_t       st_size;     /* número de bytes */
    blksize_t   st_blksize;  /* tamaño bloque para I/O
    blkcnt_t    st_blocks;   /* n° de agrupaciones as
    time_t      st_atime;    /* último acceso */
    time_t      st_mtime;    /* última modificación d
    time_t      st_ctime;    /* última modificación d
};
```

Comprobación del tipo de fichero aplicado a st_mode:

S_ISDIR(s.st_mode)	Cierto si directorio
S_ISREG(s.st_mode)	Cierto si fichero normal
S_ISLNK(s.st_mode)	Cierto si enlace simbólico
S_ISCHR(s.st_mode)	Cierto si especial de caracteres
S_ISBLK(s.st_mode)	Cierto si especial de bloques
S_ISFIFO(s.st_mode)	Cierto si pipe o FIFO
S_ISSOCK(s.st_mode)	Cierto si socket

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

STAT, FSTAT. INFORMACIÓN SOBRE UN FICHERO

```
int main(int argc, char *argv[]) {
    struct stat infoFich;
    stat(argv[1], &infoFich)
    printf("Número de i-nodo: %ld\n", (long) infoFich.st_ino);
    printf("Número de enlaces: %ld\n", (long) infoFich.st_nlink);
    printf("Propietario: UID=%ld GID=%ld\n", (long) infoFich.st_uid
        (long) infoFich.st_gid);
    printf("Tamaño del fichero: %lld bytes\n", (long long) infoFich
    printf("Número de bloques: %lld\n", (long long) infoFich.st_bl
    printf(" Última modificación i-nodo : %s", ctime(&sb.st_ctime)
    printf("Último acceso: %s", ctime(&infoFich.st_atime));
    printf("Última modificación de datos: %s", ctime(&infoFich.st_
    exit(0);
}
```

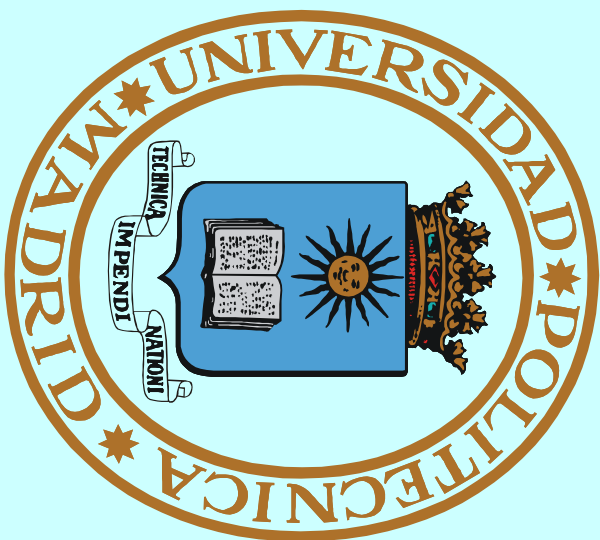
Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70





SERVIDOR DE FICHEROS ASPECTOS DE DISEÑO

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

- **Mapas de bits**, o vectores de bits: un bit por recurso existente de fichero, agrupación). Si el recurso está libre, el valor del bit mismo es 1, si está ocupado es 0.
 - Ejemplo, sea un disco en el que las agrupaciones 2, 3, 4, ocupadas y el resto libres, y en el que los descriptores de 4 están ocupados. Sus mapas de bits de serían:
 - MB de agrupaciones: 1100011100011.....
 - MB de descriptores: 1100011...
 - Fácil de implementar y sencillo de usar. Eficiente si el disco está muy lleno o muy fragmentado.
- **Listas de recursos libres**: mantener enlazados en una lista todos los recursos disponibles (agrupaciones o descriptores de fichero) manteniendo un apuntador al primer elemento de la lista.
 - Este método no es eficiente, excepto para dispositivos muy fragmentados
 - La lista puede construirse dentro de las propias agrupaciones

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

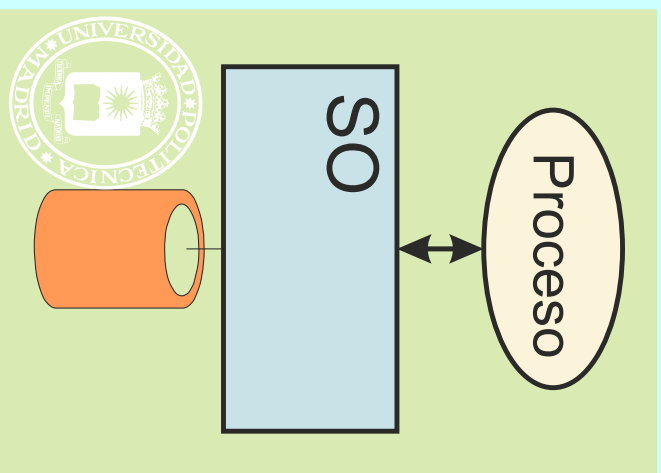
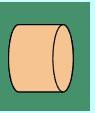
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



CACHE DE BLOQUES

- Estructura de datos en memoria con los bloques más frecuentes utilizados
 - Lecturas adelantadas
 - Limpieza de la cache (**sync**) significa escritura retardada
- Mejora las prestaciones si hay reutilización de la información referencial)
- El tiempo medio de lectura puede ser distinto del de escritura
- Principal problema: fiabilidad del sistema de ficheros por ser retardada (**delayed-write**) (30seg en Unix) o diferida (**write-back** técnica de **write-through**, pero no es eficiente.
- Asociado a la gestión de memoria virtual (bloque cache = página)



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99





TUBERÍAS

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Servicio:

```
int pipe (int fildes [2] );
```

Argumentos:

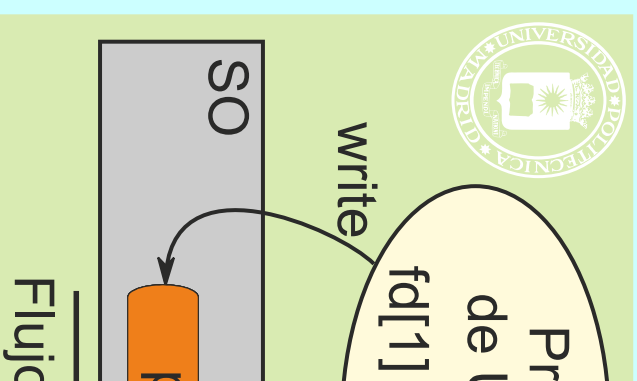
- *fildes*: Vector de dos descriptores de fichero
 - *fildes[0]*: descriptor de lectura
 - *fildes[1]*: descriptor de escritura

Devuelve:

- 0 o -1 si fracasó

Ejemplo utilización:

```
int miPipe [2] , ret;  
ret = pipe (miPipe);
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

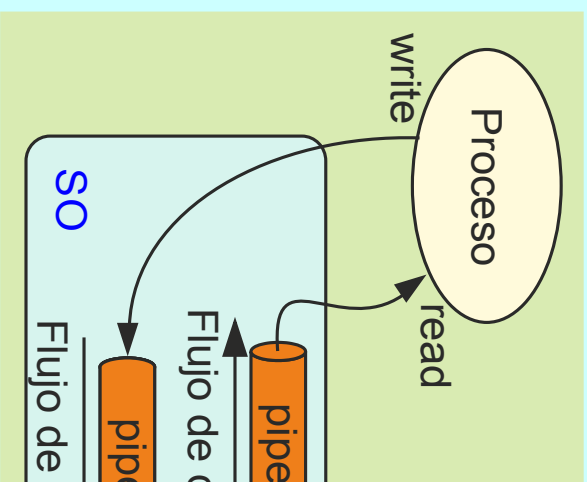
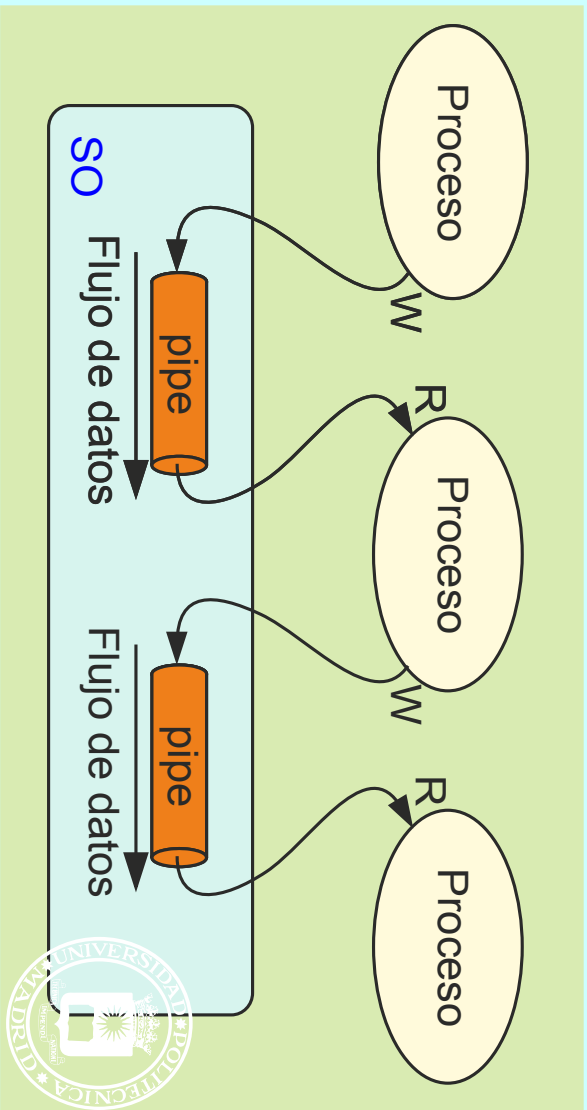
...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

TUBERÍA O PIPE

- Mecanismo de comunicación y sincronización sin nombre
- Identificación: dos descriptors de fichero
- Sólo pueden utilizarse entre procesos que crean y heredan
- En cada proceso deben cerrarse los extremos no utilizados
- Con buffering (Tamaño típico: 4 KB)



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

SEMÁNTICA DEL READ EN UN PIPE

`ssize_t read(int fd[0], void *buf, ssize_t n_bytes)`

- Pipe vacío → se bloquea el lector
- Pipe con p bytes →
 - Si $p \geq n$ devuelve n
 - Si $p < n$ devuelve p
- Si pipe vacío y no hay escritores (no hay usuarios utilizando) devuelve 0
- Lectura atómica (cuidado con tamaños grandes, debido al b

!No dejar abiertos los descriptors NO usados!

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

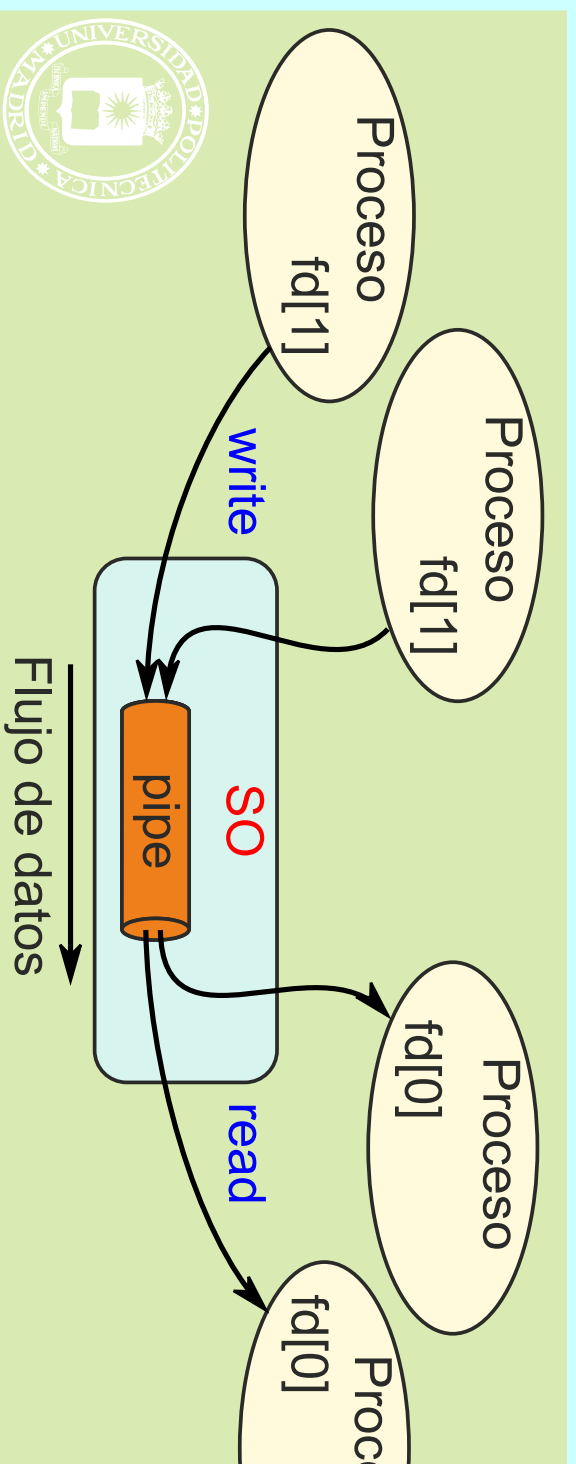


SEMÁNTICA DEL WRITE EN UN PIPE

- `ssize_t write(int fd[1], void *buf, size_t n_bytes)`
- Pipe lleno → se bloquea el escritor
- Si no hay lectores (no hay usuarios utilizando `fd[0]`) se recibe `SIGPIPE`
- Escritura **atómica** (cuidado con tamaños grandes, debido al

!No dejar abiertos los descriptors NO usados!

Varios procesos pueden leer o escribir de un mismo pipe.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



ESQUELETO DE EJEMPLO CON PIPES

```
int main(void) {
    int fd[2];          /* descriptors del pipe */
    int dato_p[4];     /* datos a producir */
    int dato_c;        /* dato a consumir */
    pipe (fd);
    if (fork() == 0) { /* productor (proceso hijo) */
        close (fd[0]);
        for (;;) {
            /* producir dato_p */
            .....
            write (fd[1], dato_p, 4*sizeof(int));
        }
    } else { /* consumidor (proceso padre) */
        close (fd[1]);
        for (;;) {
            read (fd[0], &dato_c, sizeof(int));
            /* consumir dato */
            .....
        }
    }
    return 0;
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

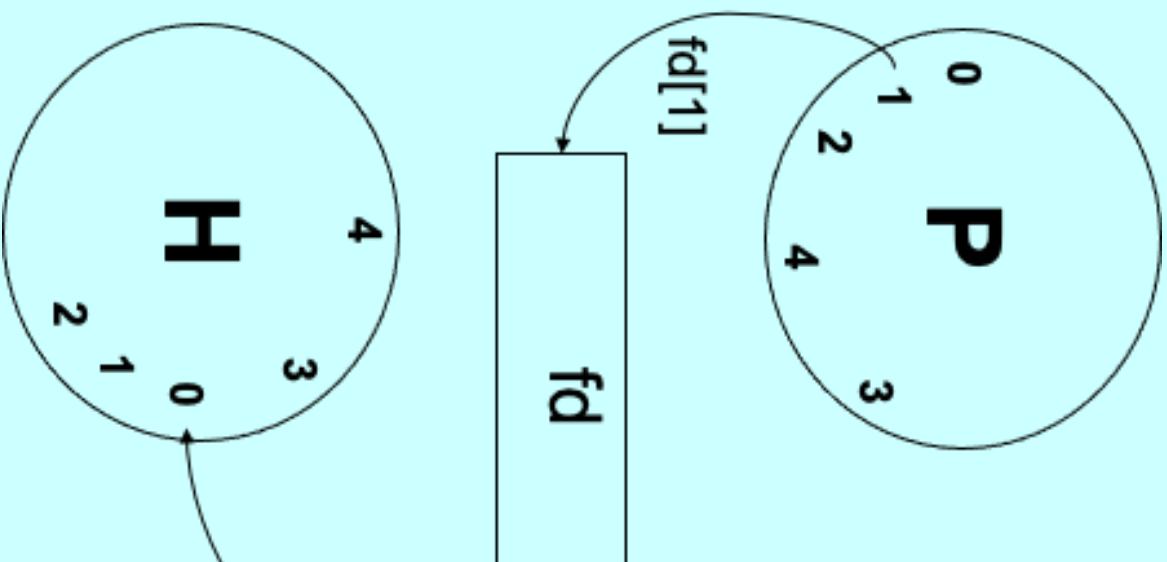
Cartagena99

TUBERÍA O PIPE

```
int fd[2], ret;

ret = pipe (fd);
pid = fork ();

if (pid > 0) { /* padre */
    close (1);
    dup (fd[1]);
    close (fd[0]);
    close (fd[1]);
}
else { /* hijo */
    close (0);
    dup (fd[0]);
    close (fd[0]);
    close (fd[1]);
}
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

...

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

