Computadors
Sistemes Operatius (3/3)

Grau en Ciència i Enginyeria de Dades

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC)

Creative Commons License

Aquest document utilitza Creative Commons Attribution 3.0 Unported License

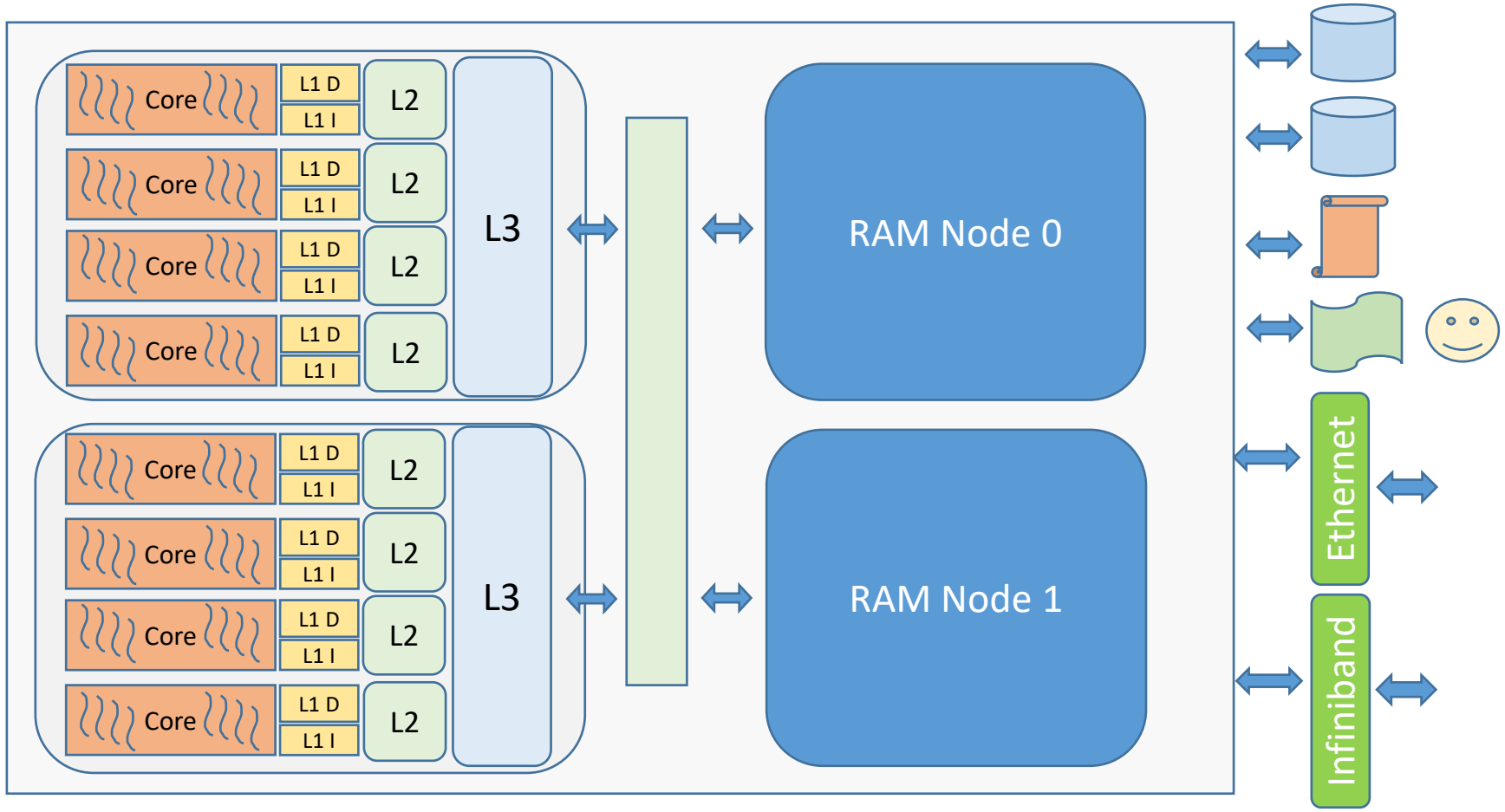
Els detalls d'aquesta licencia es poden trovar a
https://creativecommons.org/licenses/by-nc-nd/4.0

Index

- Components Bàsics del SO
- Gestió de Processos
- Gestió de la Memòria
- **Gestió de l'E/S i Sistema de Fitxers**
  - **Conceptes Bàsics**
  - Suport del SO

Introducció

- Necessitat de moure dades des de/cap a el sistema des de/cap a altres dispositius

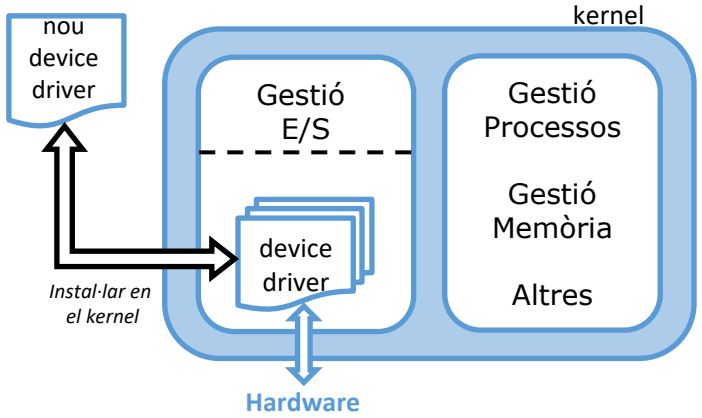


Gestió de Entrada/Sortida

- E/S és una de les activitats clau dels processos
  - L'E/S és la capacitat d'enviar/rebre dades des de/cap a un procés
    - L'acció és sempre realitzada des del punt de vista del procés
  - Això vol dir que també inclou comunicació entre processos
    - Per exemple, "sockets" per comunicar processos mitjançant la xarxa
- Accedir i gestionar dispositius són operacions "privilegiades"
  - Necessita tenir una gestió segura per garantir el correcte ús entre usuaris
  - Root / administrador és qui pot gestionar-ho a nivell d'administrador
- Per poder realitzar operacions d'E/S és necessari l'ús de **crides a sistema**
  - Són independents de les característiques del dispositiu
- **IMPORTANT:** una operació d'E/S pot bloquejar un procés fins que es completi
  - Estat "BLOCKED" i per tant no utilitza CPU
  - Hi ha diverses alternatives per minimitzar impacte
    - E.g. Operacions asíncrones (o no bloquejants), transferències amb buffer, planificació eficient, ...

Device Drivers

- Els programadors no poden desenvolupar codi per tots els diferents dispositius que existeixen
- Els fabricants han de proporcionar el conjunt de codi de baix nivell (codi compilat) que necessita el dispositiu per realitzar la funcionalitat
  - El codi + dades per accedir al dispositiu es conneix com **Device Driver**



- Per afegir un nou dispositiu
  1. Opció 1: recompilar el kernel
  2. Opció 2: **sense** recompilar el kernel
     - El SO ha d'oferir un mecanisme per afegir dinàmicament codi/dades al kernel de manera segura
       - *Dynamic Kernel Module Support*, *Plug & Play*

Device Drivers: controlador de dispositiu

- Conjunt de rutines que gestionen un dispositiu i permet a un programa interactuar amb ell
  - Segueix l'interfície definida per cada SO per implementar tasques genèriques
    - Obrir, llegir, escriure, tancar…
  - Implementa tasques específiques del dispositiu
  - Conté codi de baix nivell per gestionar interrupcions i altres elements del dispositiu
  - Són fitxers binaris

Independència de Dispositiu

- Els dispositius presenten una gran varietat de característiques i necessitats
  - Per exemple, velocitat, tipus de dispositiu, tipus de transferència, etc.
- Classificació interna
  - UNIX/Linux: tots els dispositius estan representats per un fitxer
    - **character/block**, → per exemple, teclat (char) / disc (block)
    - **major** (quin tipus de dispositiu?) → és un número enter >= 0
    - **minor** (quina instància és?) → és un número enter >= 0
- Per què l'independència del dispositiu és important?
  - **Operacions uniformes d'E/S** – open, close, read, write...
  - **Dispositius virtuals** – només utilitza un identificador (file descriptor)
  - **Redireccionament** – canviar el dispositiu que s'utilitza sense canviar el codi (vist en el Lab)

Per exemple: ./programa < disp1
./programa > disp2

UNIX/Linux: Sistema de Fitxers basat en I-Nodes

- Tot dispositiu té la seva representación en el sistema de fitxers
  - Mitjançant un I-Node

- I-Node: Estructura de dades que conté tota l'información relativa a un fitxer
  - Alguns camps són:
    - Identificador (ID) de I-Node
    - Mida
    - Tipus de fitxer (regular, directori, disp.caracter, disp.bloc, etc.)
    - Major/minor IDs
    - Protecció (Read / Write / Execute (RWX) per Owner, Group, Others)
    - …
  - Altres camps els veurem en la part de Sistema de Fitxers

Tipus de Dispositius

- Dispositius de Tipus Caràcter – Transferència de byte en byte
  - Terminals, consoles, teclat, ratolí, pantalla...
  - Impressores
  - Real time clock (rtc)
  - Dispositius especials de dades bàsiques
    - null, zero, full, random, urandom
  - SO
    - Missatges, memòria, ports d'E/S, memòria física
  - Hardware
    - Firmware, registres hardware

Dispositius de Tipus Caràcter

```
crw-r-----  1 root kmem     1,   1 Jan 16 08:14 mem
crw-r-----  1 root kmem     1,   2 Jan 16 08:14 kmem
crw-rw-rw-  1 root root     1,   3 Jan 16 08:14 null
crw-r-----  1 root kmem     1,   4 Jan 16 08:14 port
crw-rw-rw-  1 root root     1,   5 Jan 16 08:14 zero
crw-rw-rw-  1 root root     1,   7 Jan 16 08:14 full
crw-rw-rw-  1 root root     1,   8 Jan 16 08:14 random
crw-rw-rw-  1 root root     1,   9 Jan 16 08:14 urandom
crw-r--r--  1 root root     1,  11 Jan 16 08:14 kmsg
```

Tipus de Dispositius

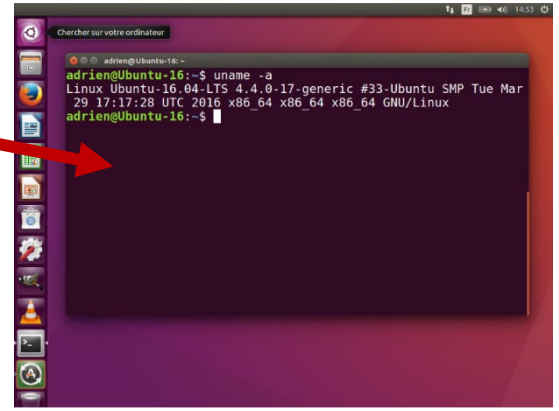
- Dispositius de Tipus Bloc – Transferència de bloc de bytes
  - Disc /dev/sda, /dev/sdb...
    - SCSI, SATA, ATA
    - IDE (/dev/hda, /dev/hdb...)
  - CD/DVD /dev/dvdrom
  - RAMDisks /dev/ram0

Dispositius de Tipus Bloc

```
brw-rw----  1 root disk      8,   0 Jan 16 08:14 sda
brw-rw----  1 root disk      8,   1 Jan 16 08:14 sda1
brw-rw----  1 root disk      8,   2 Jan 16 08:14 sda2

brw-rw----  1 root disk      7,   0 May 28 10:46 loop0
brw-rw----  1 root disk      7,   1 Jan 16 08:14 loop1
brw-rw----  1 root disk      7,   2 Jan 16 08:14 loop2

brw-rw----  1 root cdrom    18,   0 Dec  1 02:56 sr0

lrwxrwxrwx  1 root root          8   Ago 17 17:32 cdrom -> /dev/sr0
```

Suport del SO pel subsistema d'E/S

- Qualsevol dispositiu d'E/S necessita ser accessible
  - **Nom simbòlic** és el nom que el representa en el Sistema de Fitxers
    - E.g.: un terminal pot ser: "/dev/pts/0"

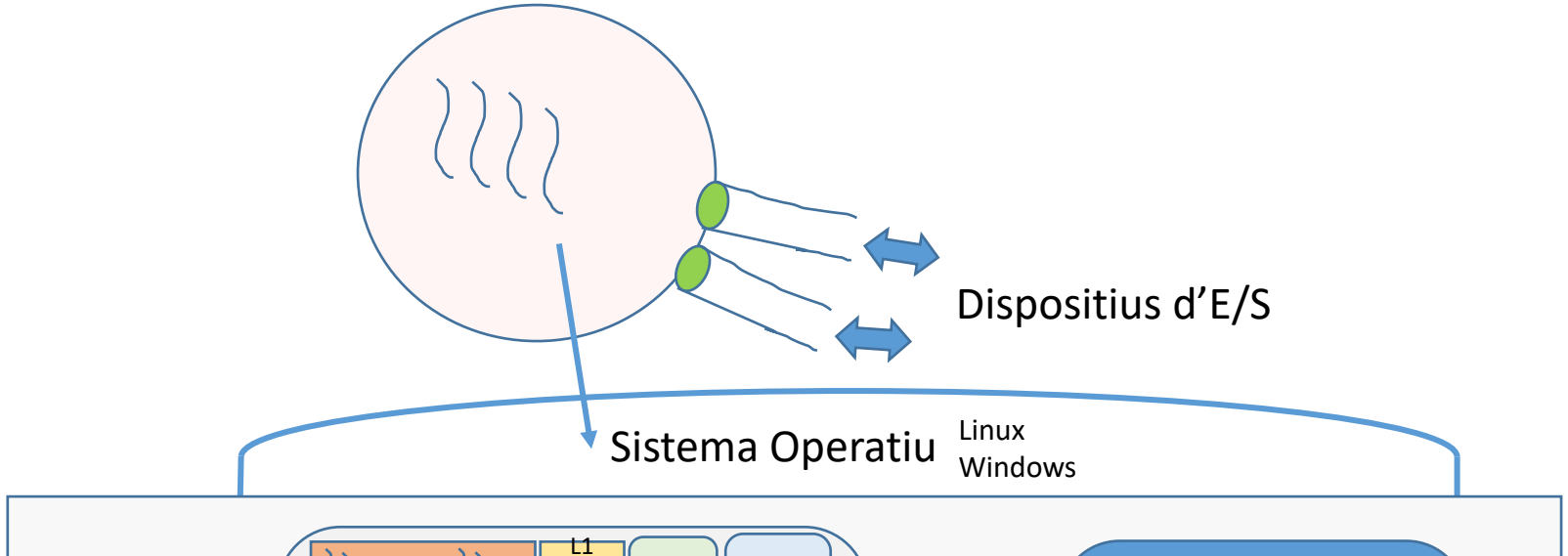


- El SO introdueix varies capes d'abstracció…
  - …per simplificar la gestió d'E/S
    - Ha d'oferir crides a sistema independents dels internals dels dispositius

- El SO gestiona, per cada procés, una taula privada de dispositius que utilitza. Pel procés són dispositius abstractes (o "virtuals")
  - **File descriptor table**: una taula de punters a una altra estructura global del SO

Taula de descriptors de fitxers

- Cada procés té una taula privada
  - És el punt d'accés per part del codi d'usuari per utilitzar dispositius
    - Totalment independent de les característiques del dispositiu
  - Cada dispositiu està representat pel número de l'entrada que se li ha assignat
    - Quan es demana començar a utilitzar un dispositiu s'assigna la primera entrada disponible
    - Els tres primers (0→STDIN; 1→STDOUT; 2→STDERR) per defecte estan assignats (per defecte a la terminal)

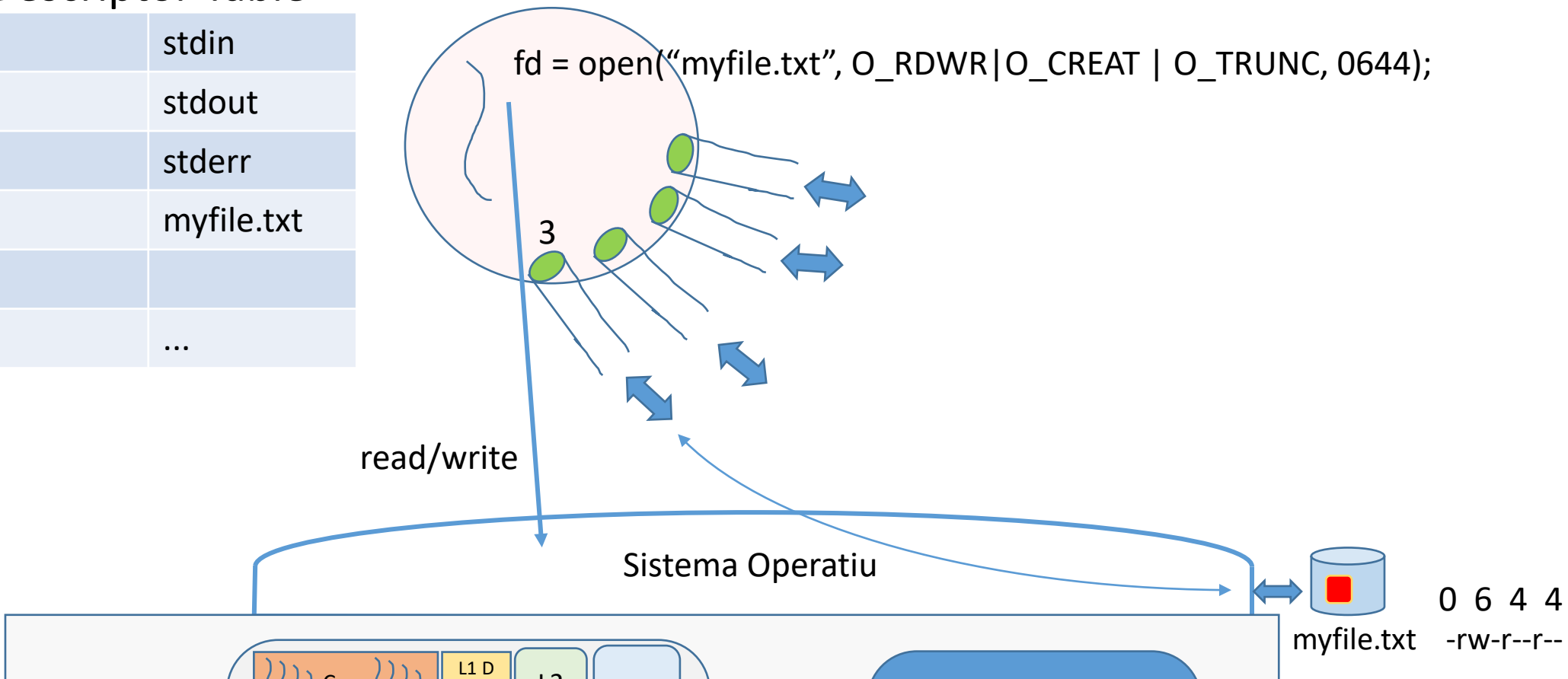


Taula de descriptors de fitxers

En Linux es pot consultar en /proc/<PID>/fd

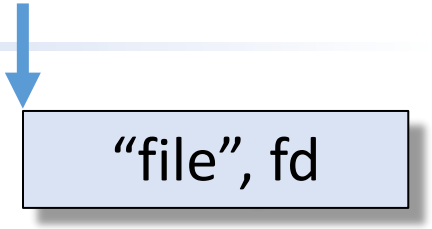
File Descriptor Table

0	stdin
1	stdout
2	stderr
3	myfile.txt
4	
...	...



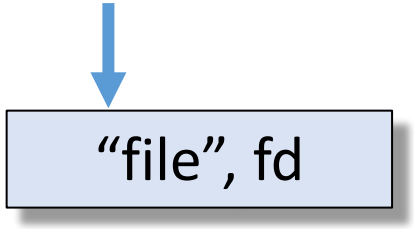
Crides a sistema bàsiques de E/S

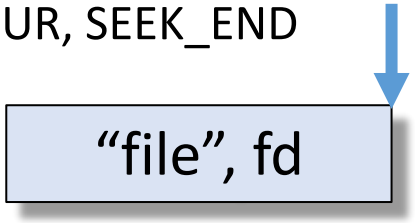
Per defecte ens posicionem al principi del fitxer quan es fa l'open



- fd = open(path, flags[, permissions]);
  - Flags bàsics:
    - O_RDONLY O_WRONLY O_RDWR
  - Combinacions de flags:
    - O_WRONLY|O_CREAT|O_TRUNC
  - Permissions:
    - Control d'accés al fitxer que es crea (0644 -> rw-r—r--)
- bytes = read(fd, @data, bytes);
- bytes = write(fd, @data, bytes);
- new_offset = lseek(fd, offset, whence);
  - Desplaçarse dins del fitxer sense accessos a disc
  - Whence: SEEK_SET, SEEK_CUR, SEEK_END
- close(fd);

Internament el SO actualitza l'offset (posició) en la que està després de r/w





Es pot anar directament a punts remots del fitxer sense accedir a disc

Suport de la biblioteca C

- Ofereix serveis d'E/S d'alt nivell
  - **FILE** és una estructura que engloba més que el "file descriptor"
  - Afegeix altres característiques per facilitar l'ús
    - Especialment **BUFFERING**
- Cada funció invoca a la crida a sistema corresponent
  - E.g. "fopen" crida a "open"
- Llegir i escriure...
  - ...està preparat per transferir un nombre d'elements d'una mida (*elemsize* bytes)
- Lseek funciona de la mateixa manera

```
FILE * f;
size_t size;  int res;

f = fopen(path, mode);
     // mode = "r"    read-only
     // mode = "w"  truncate, write-only
     // mode = "a"   append
     // mode = "r+"  read-write
size = fread (data, elemsize, numelems, f);
if (size < 0) perror ("fread");

size = fwrite (data, elemsize, numelems, f);

res = fseek(f, newoffset, whence);
     // whence = SEEK_SET, SEEK_CUR, SEEK_END

res = fclose(f);
```

Suport de la biblioteca C

- Hi ha altres funcions addicionals per fer E/S pensades per text
  - printf, escriure per la sortida estàndard **stdout**
  - fprintf, utilitzar una sortida diferent
  - sprintf, escriure en un array de chars

```
FILE *f;
f = fopen("myfile", "r+");
if (f == NULL) {
    fprintf (stderr, "fopen: %s\n",strerror(errno));
    exit(1);
}
fprintf (f, "my message");
```

  - scanf, llegir una string de format concret des de l'entrada estàndard **stdin**
  - fscanf, llegir una string de format concret des d'una entrada diferent
  - sscanf, llegir una string de format concret des d'una string

```
char line[4096];
int i; double d; char str[11];

n = sscanf(line, "%d %lf %10s", &i, &d, str);
     // will understand lines like   10   4.5   hola
     // and assign i=10, d=4.5, str="hola"
```

Impacte en el Rendiment

- Una operació d'E/S pot **bloquejar** un procés (operació síncrona)
  - Si les dades es poden disposar al moment, torna immediatament
  - Si les dades no estan disponibles, el procés passa d'estat "Running" a "BLOCKED" i per tant no utilitza CPU fins que es completi
- Una operació d'E/S pot ser **no bloquejant**
  - Gestió d'identificadors de l'operació asíncrona
  - Retorna immediatament tant si l'operació s'ha completat com si no
- Les rutines d'E/S són crides a sistema
  - Tenen l'overhead implícit per passar a mode privilegiat (executar codi del kernel)
  - Utilitzar buffers per reduir el nombre de crides a sistema

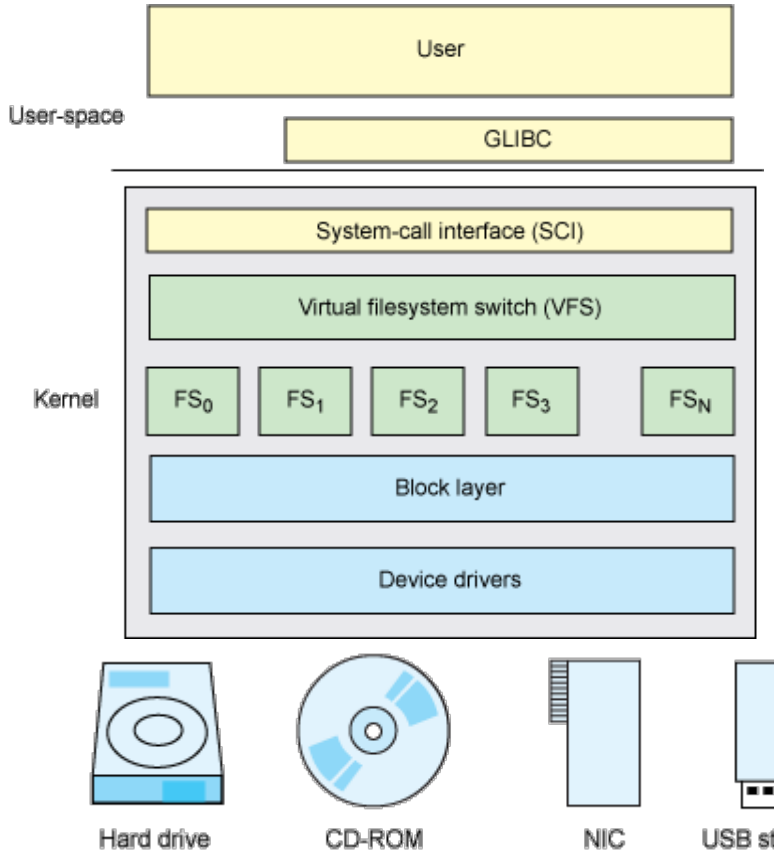
Index

- Components Bàsics del SO
- Gestió de Processos
- Gestió de la Memòria
- **Gestió de l'E/S i Sistema de Fitxers**
  - Conceptes Bàsics
  - **Suport del SO**

Virtual File System (VFS)

- Capa d'abstracció per gestionar particions de diferents tipus de sistema de fitxers
  - Proporcionar un única interfície de crides a sistema per qualsevol tipus de sistema de fitxers
  - VFS és per UNIX/Linux. Altres SOs utilitzen mètodes similars

- Tipus de Sistemes de Fitxers
  - FAT (File Allocation Table)
    - Dispositius extraïbles
  - exFAT (Extended FAT)
    - Dispositius extraíbles més grans de 4GB
  - NTFS (New Technology Transfer)
    - Windows
  - **Sistemes de Fitxers basats en I-node (UNIX/Linux)**
    - Ext3, ext4, Reiser4, XFS, F2FS
  - Sistemes de Fitxers en el Núvol
    - GlusterFS, Ceph, HadoopFS, ElasticFileSystem (Amazon)



https://en.wikipedia.org/wiki/Comparison_of_file_systems

Altres Sistemes de Fitxers

- Espai Swap: en SOs UNIX/Linux és un Sistema de Fitxers que estén la memòria principal
  - Windows implementa l'espai swap en un únic fitxer redimensionable

- FUSE: Filesystem in Userspace
  - Permet definir un sistema de fitxers en espai d'usuari. És a dir, sense modificar o afegir mòduls en el kernel (les rutines s'executen en espai d'usuari)
    - E.g.: GDFS (Google Drive), WikipediaFS, propietary File Systems

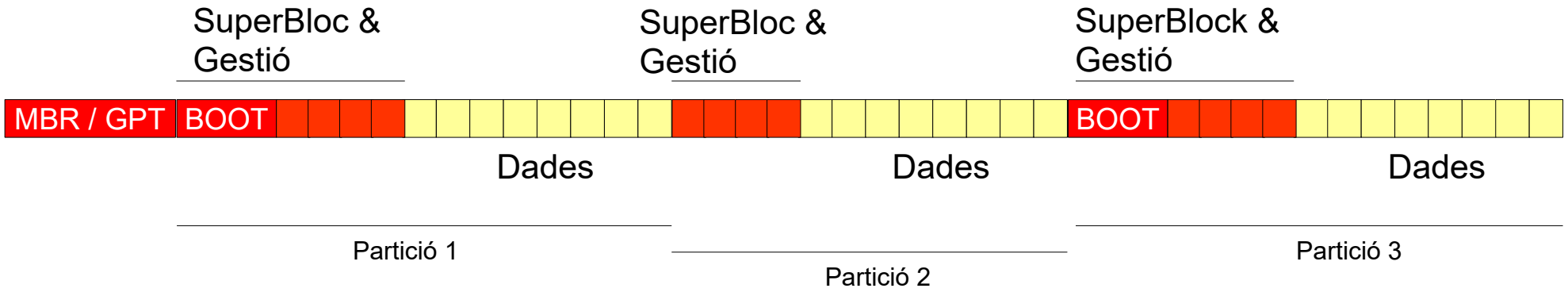
- Les característiques dels Sistemes de Fitxers tenen impacte en el rendiment, fiabilitat, resiliencia, seguretat, etc

Partició/Unitats Lògiques

- Partició: Un grup de sectors consecutius. És identificat amb una etiqueta única
  - Pot ser gestionat com una **unitat lògica** independent
    - Windows: C: D: Linux: /dev/sda1 /dev/sda2
  - Un dispositiu d'emmagatzematge físic pot ser dividit en varies particions
- Sectors de Gestió
  - MBR: Master Boot Record (fins a unitats físiques de 2TB)
  - GPT: GUID Partition Table (substitueix MBR)

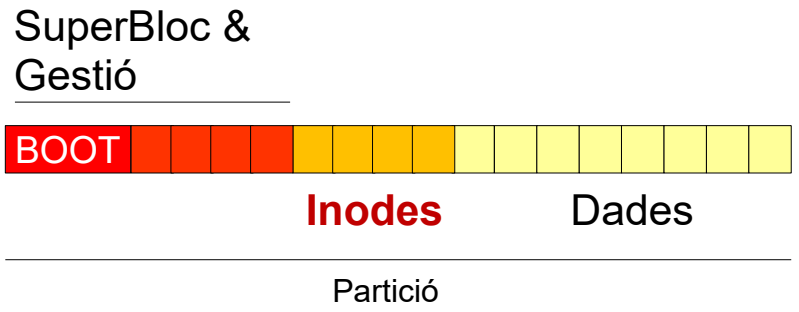
  } Per trovar particions i el SO a arrencar

  - Superbloc & gestió (una per partició): Informació crítica per gestionar la partició
  - Boot Sector: Pocs Bytes per arrencar el SO instal·lat en aquesta partició



Sistema de Fitxers basat en I-Nodes

- I-Node: Estructura de dades que conté tota l'información relativa a un fitxer
  - Alguns camps són:
    - Identificador (ID) de I-Node
    - Mida
    - Tipus de fitxer (regular, directori, disp.caracter, disp.bloc, etc.)
    - Major/minor IDs
    - Protecció (Read / Write / Execute (RWX) per Owner, Group, Others)
    - Propietari
    - Marques de temps
    - Nombre de Links (# relacions directes entre nom simbòlic i ID de I-Node)
    - Punters a blocs de dades (index multinivell). Acostuma tenir 12-13 punters.



Gestió de blocs

- Cada sistema de fitxers té el seu mecanisme per gestionar…
  - …blocs ocupats/lliures
  - …els blocs d'un fitxer donat (i.e. com acceder als continguts d'un fitxer)
- Depenent del sistema de fitxers, l'implementación canvia. Per exemple:
  - FAT: té una taula global amb tantes entrades com blocs té el dispositiu
    - Accés al fitxer basat en llista enllaçada
  - Basat en I-nodes: té una estructura, I-node, que guarda tota l'información necesaria per gestionar un fitxer
    - Accés al fitxer basat en índex (índex multinivell), guardat en el I-node
- Buffer Cache
  - Zona de memoria que guarda temporalment qualsevol transferencia des de/cap el sistema de fitxers
  - Si el I-node o bloc està en la cache, no cal fer l'accés al disc→**augment de rendiment**

Directoris

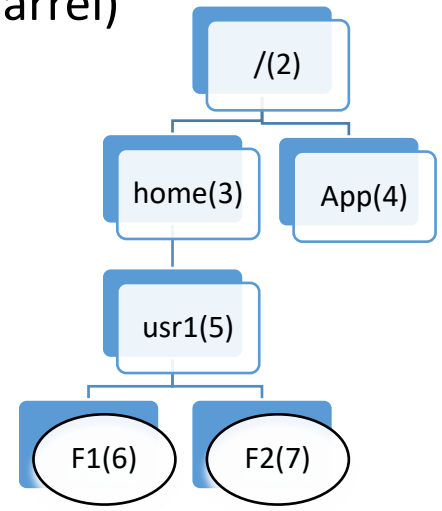
- **Directori:** Estructura lógica per organizar fitxers
  - És un tipus de fitxer particular gestionat pel SO
    - "/" és el directori arrel (el directori més superior de tot el sistema de fitxers)
      - Cada partició té el seu directori arrel **(ID de I-node 2)**
    - "." & ".." són entrades obligatories en qualsevol directori
      - Encara que el directori no tingui fitxers
- Ruta: conjunt de caràcters que identifiquen de manera única la localització d'un fitxer
  - Relativa (des de qualsevol directori) vs Absoluta (des de la carpeta arrel)
- **Hardlink:** Relació directe entre nom simbòlic i ID de I-node

Nom	I-node
.	2
..	2
home	3
App	4

Directori: /

Nom	I-node
.	5
..	3
F1	6
F2	7

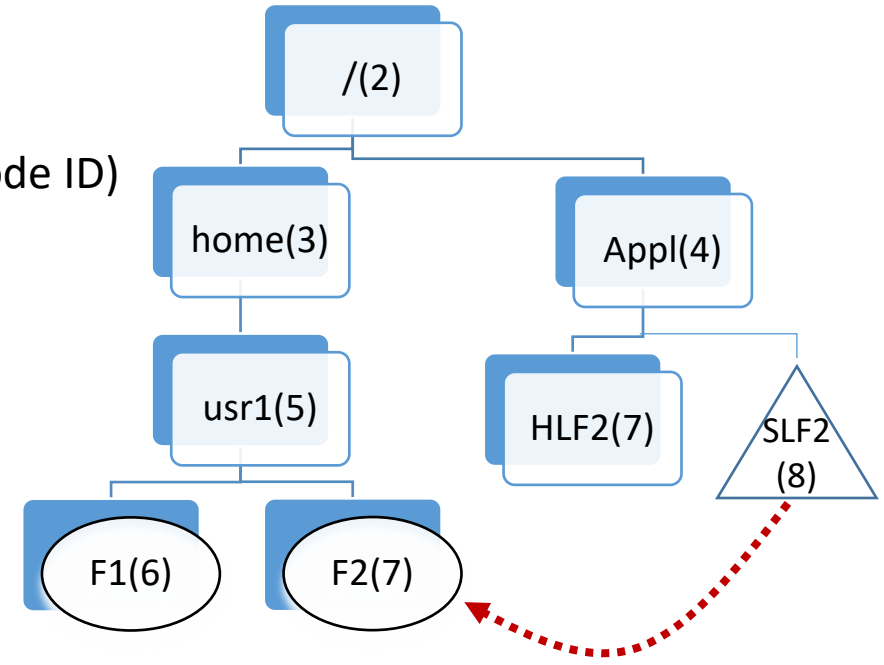
Directori: /home/usr1



Directoris

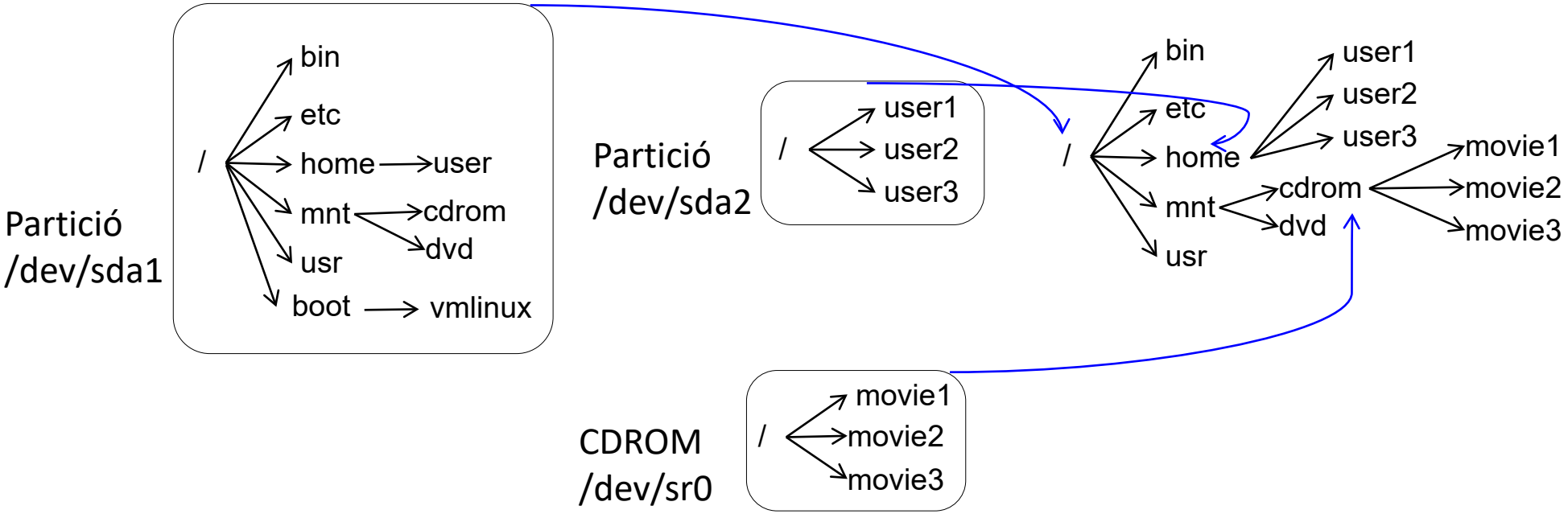
- Directoris están organitzats en grafs
  - Un fitxer pot ser accedit des de diferents directoris

- Tipus de links
  - Hardlinks
    - Relació directa entre nom simbòlic i ID de I-Node (name→I-node ID)
  - **Softlinks**
    - Un **nou fitxer** que té una ruta
      - Similar a un accés directe
  - Pros/Cons/restriccions entre ells



Muntar

- Fer accesibles els continguts d'una partició en el Virtual File System
  - E.g.: mount -t ext4 /dev/sda2 /home // muntar la partició en /home
    mount -t iso9660 /dev/cdrom /mnt/cdrom // muntar el CDROM
    unmount /dev/sda2 // desmuntar la partició

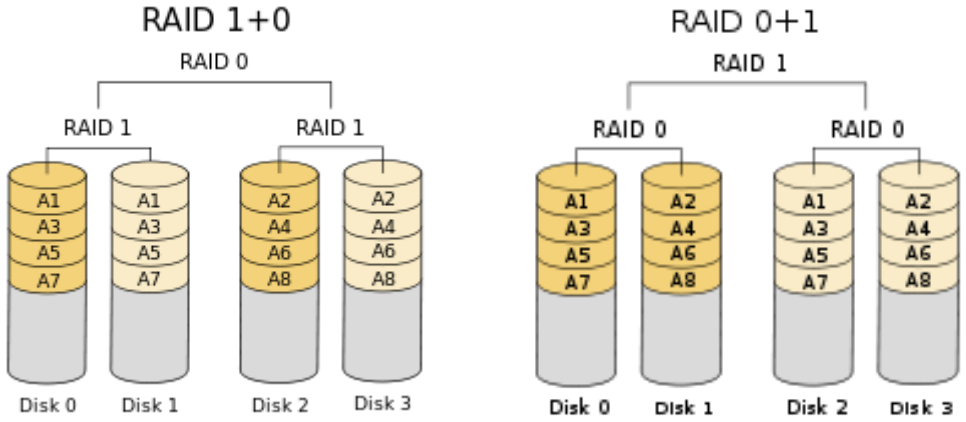


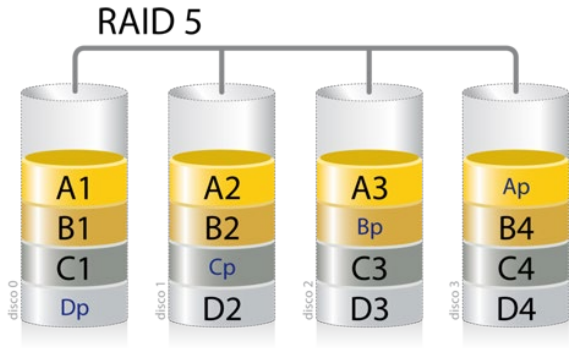
Journaling

- Sistema de fitxers basat en transaccions
  - Enregistra els canvis a realitzar en el sistema de fitxers encara no completats
  - Ho enregistra en un fitxer especial com si fos un “diari” (journal)
    - Té un área dedicada en el sistema de fitxers
- En cas de fallo en el sistema o tall de corrent elèctrica…
  - El sistema de fitxers pot tornar enrere i desfer un canvi no completat, minimitzant la probabilitat d’errors
- Alguns sistemes de fitxers que implementen Journaling
  - Ext3, Ext4, ReiserFS, XFS, JFS

RAID: Redundant Array of Independent Disks

- Tecnologia de virtualització d'emmagatzematge que combina múltiples dispositius físics en un únic de lògic
  - Software driver vs controlador hardware
  - Impacte en el rendiment i capacitat
- Varis alternatives (es poden combinar)
  - RAID 0: Stripping → dades distribuïdes
  - RAID 1: Mirroring → dades replicades
    - RAID 1+0 vs RAID 0+1
  - RAID 5: amb blocs de paritat distrbuïts





Bibliografia

- Computer Systems – A Programmer's perspective (3rd Edition)
  - Randal E. Bryant, David R. O'Hallaron, Person Education Limited, 2016
  - https://discovery.upc.edu/permalink/34CSUC_UPC/11q3oqt/alma991004062589706711
    - Chapter 10

- Operating system concepts (John Wiley & Sons, INC. 2014)
  - Silberschatz, A.; Galvin, P.B.; Gagne, G
  - http://cataleg.upc.edu/record=b1431631~S1*cat

- Operating systems: internals and design principles (Prentice Hall, 2015)
  - Stallings, W
  - http://cataleg.upc.edu/record=b1441252~S1*cat