

# Fonaments dels Sistemes Operatius (FSO)

Departament d'Informàtica de Sistemes i Computadores (DISCA)

*Universitat Politècnica de València*

Bloc Temàtic 3: Gestió d'Arxius

Unitat Temàtica 8

## Implementació de Directoris i Protecció

fSO

DISCA



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

- **Objectius**

- Estudiar el concepte de **directori**
- Entendre els conceptes d'enllaç físic i enllaç simbòlic
- Analitzar les **tècniques** de gestió de l'espai lliure de disc
- Descriure el mecanisme de protecció de la informació utilitzat en sistemes UNIX

- **Bibliografia**

- A. Silberschatz, P.B. Galvin: “Fundamentos de Sistemas Operativos”, McGraw-Hill, 7ª ed. 2006 (Capítulos 10 y 11)

- **Contingut**
  - **Concepte de directori**
  - Implementació de directoris
  - Enllaços o referències a arxius
  - Gestió de l'espai de disc
  - Protecció

- Arquitectura del sistema d'arxius: Visió Usuari

## Nivell Usuari

Abstraccions d'Arxiu i Directori

### Biblioteques Usuari (per a operar amb arxius)

Interfície amb les crides al sistema sobre fitxers i directoris

#### Operacions sobre arxius:

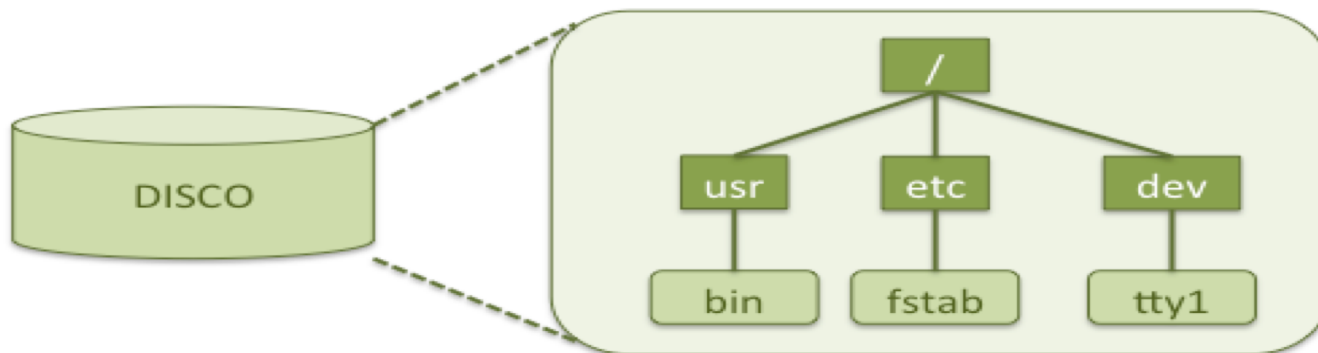
- **Obrir i tancar** arxius
- **Llegir/Escriure** sobre arxiu
- **Posicionar-se** dins d'un arxiu

#### Operacions sobre directoris:

- **Crear/Esborrar entrades** a directori
- **Renomenar** arxius
- **Cercar** per nom
- **Recòrrer** el sistema d'arxius

### Visió jeràrquica

Organització jeràrquica en arxius i directoris



- **Un directori és un arxiu**
  - un tipus abstracte de dades
  - l'element necessari per a organitzar la informació emmagatzemada en memòria secundària
- **Objectius**
  - **Localitzar** ràpidament un **arxiu** a partir del **nom** associat
  - Implementar un esquema de noms convenient per a l'usuari
  - L'usuari pot establir les seues pròpies agrupacions d'arxius
  - **Protecció**.- el propietari pot controlar les operacions permeses a cada usuari
    - Sobre directoris.- crear o esborrar entrades, llistar contingut, cercar

- **Operacions sobre directoris**

- El sistema operatiu ha d'oferir un conjunt de crides bàsiques per a treballar amb directoris

- **Crear entrada**

- Requereix tindre espai lliure per a crear-la

- **Esborrar entrada**

- Allibera l'espai en disc associat a l'arxiu i esborra l'entrada de directori associada ej. unlink(nom)

- **Cercar per nom**

- Normalment la recerca es realitza de forma seqüencial

- **Llistar contingut del directori**

- Permet visualitzar el contingut dels directoris

- **Renomenar fitxer**

- Modifica l'entrada a directori

- **Recòrrer sistema de fitxers**

- Permet situar-se en qualsevol punt de la jerarquia de directoris

- **Contingut**
  - Concepte de Directori
  - **Implementació de directoris**
  - Enllaços o referències a arxius
  - Gestió del espai de disc
  - Protecció

- **Estructura de directoris**

**Directori** -> manté l'associació de noms a arxius

## Estructura de directoris

### Plana

➤ Tots els arxius en un únic directori

- Possibles colisions entre noms
- No permet agrupar per usuaies/temes

### Jeràrquica

➤ Organització en nivells (arbre, graf) amb profunditat arbitrària

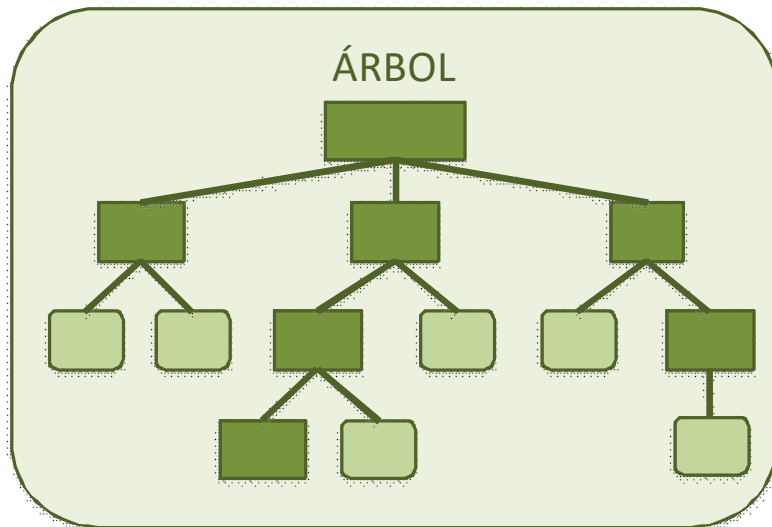
- Permeten crear agrupacions arbitràries
- Permeten muntar/desmuntar altres sistemes d'arxius



## • Estructura jeràrquica de directoris

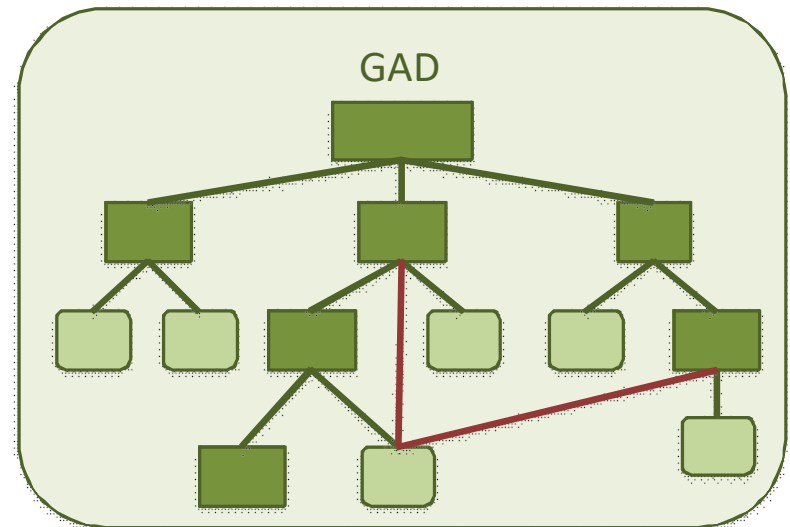
### Arbre

- Recerca i agrupació eficients
- Un únic nom absolut per fitxer = recorregut des de l'arrel fins el node en concret
- Nom relatiu = recorregut des de directori actual al node



### Graf Acíclic Dirigit (ej. Unix)

- Estructura bàsica en arbre, però permet compartir fitxers i directoris
- Divesos noms absoluts (diverses rutes des de l'arrel) per a un mateix node
- No permet cicles



- **Contingut d'un directori**
  - La informació d'un directori està organitzada en registres
  - Un directori conté un conjunt de registres denominats **entrades de Directori**
  - Té una entrada per cada arxiu existent al directori
- **Entrada de Directori**
  - **Contingut** de cada entrada es dependent de la implementació del sistema d'arxius
  - Nom + referència als atributs (ej. Unix)
  - Nom + atributs + referència a dades (ej. Windows)
- **Ubicació** de les dades de directoris en disc
  - Centralitzada (àrea dedicada en disc)
  - En arxius (adequada per a sistemes jeràrquics)

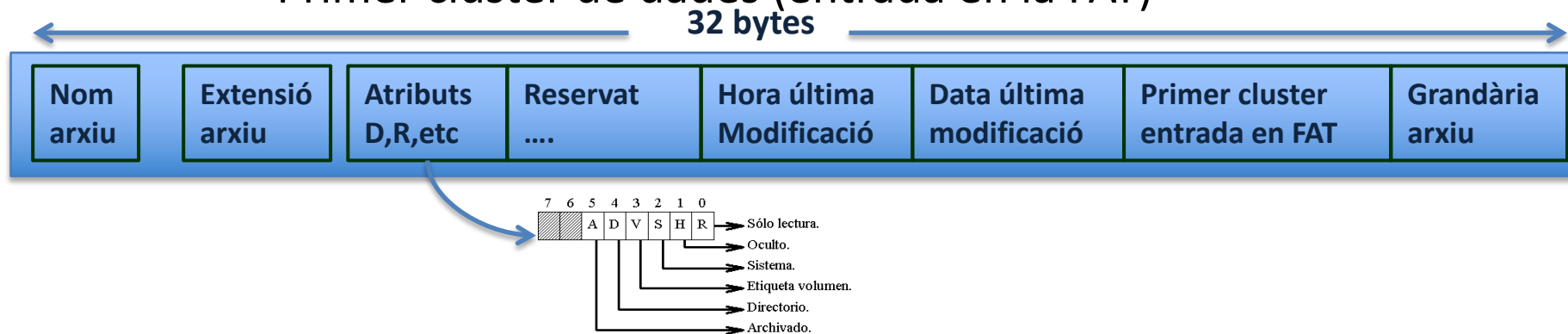
- **Directoris en MS-DOS → FAT**

- Característiques

- Directori arrel en lloc conegut i grandària constant
- La resta de directoris es gestionen com un arxiu

- Una entrada de directori ocupa 32 bytes

- Nom del fitxer + extensió (8 bytes + 3bytes )
- Permisos, tipus, etc.
- Data, hora, talla, etc.
- Primer cluster de dades (entrada en la FAT)



## Estructura d'un disc MS-DOS



- **Directoris en sistemes UNIX**

- Directoris implementats com a un tipu d'**arxiu**
- Les dades d'un directori estan estructurades en una taula con dues columnes: nombre de inode i nom

## Entrada de directori

Nombre node-i	Nom d'arxiu
---------------	-------------

- Cada entrada de directori correspon a un arxiu
- L'entrada de nom "." correspon al directori actual, apunta a si mateix
- L'entrada de nom ".." correspon al directori pare
- Nombre de inode = índex per a localitzar el i- node en el vector de nodes-i que emmagatzema el sistema en el disc

## Contingut d'un directorio

Entrada del  
directori arrel



Node-i	Nom arxiu
1	.
1	..
3	dev
4	bin

- **Mecanisme d'accés per nom**

- En el disc existeix una secció dedicada a emmagatzemar un **vector de nodes-i** (nodes índex)
  - Node-i = atributs + localització de les dades de l'arxiu
  - En el vector de nodes-i hi ha un node-i per arxiu
- Nom absolut:
  - La recerca de l'arxiu comença des del directori arrel
  - El directori arrel sempre té assignat un node-i fixe( MINIX node-i 1)
  - ejemplo: /a/b/c
- Nom relatiu
  - La recerca de l'arxiu comença des del directori actual
  - Exemple: b/c

Node-i	Nom arxiu
6	.
1	..
10	b

Node-i	Nom arxiu
1	.
1	..
3	dev
4	bin
6	a

Node-i	Nom arxiu
10	.
6	..
20	c

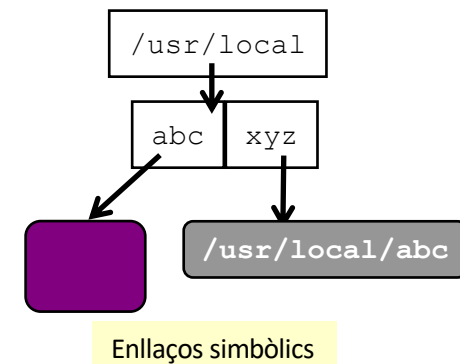
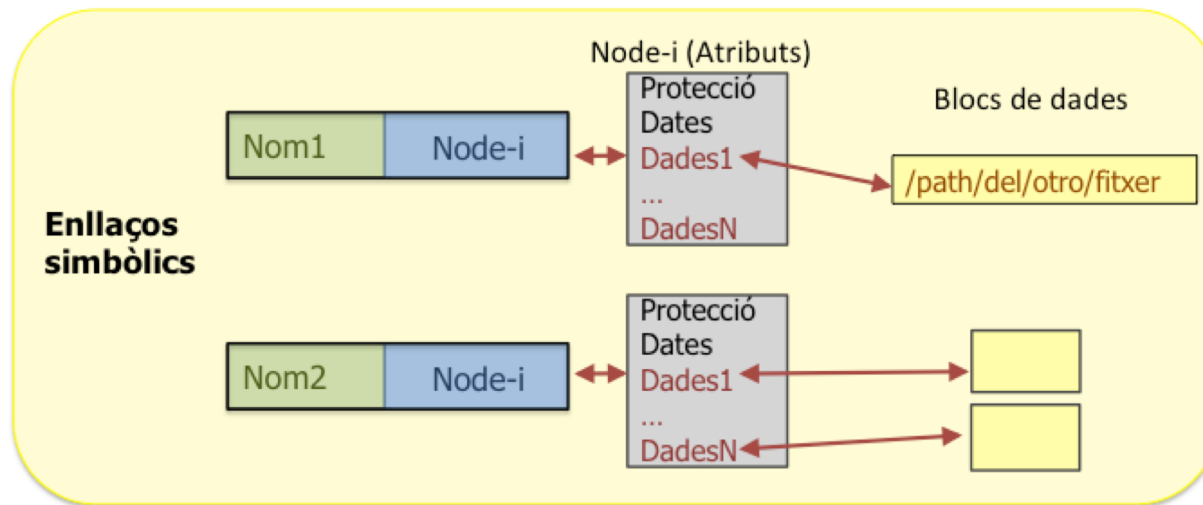
```

Mentre queden elements
  Si es tracta d'un directori
    Comprova permisos,
    localitza element en aqueste
    directori,
    obtenir node-i
Retornar node-i final
    
```

- **Contingut**
  - Concepte de Directori
  - Implementació de directoris
  - **Enllaços o referències a arxius**
  - Gestió del espai de disc
  - Protecció

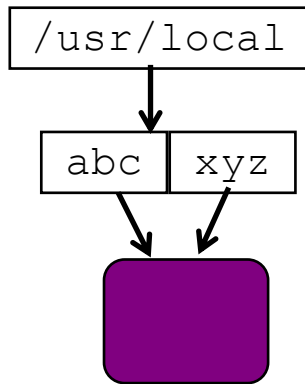
## • Enllaços Simbòlics

- En Unix es denominen **enllaços lògics**, en Windows fitxers d'**accés directe**
- A és un arxiu de tipus 'enllaç simbòlic' que enllaça amb l'arxiu B
  - Entre els atributs de A s'indica el tipus de l'arxiu (link)
  - El SO interpreta les dades de A com un path per a accedir a l'arxiu B
  - El SO redirigeix les lectures i escriptures sobre A per a que s'accedisca a les dades de B
  - A nivell de protecció, s'apliquen els permisos del fitxer B, no els de A
- B pot estar en un altre sistema d'arxius (ej. muntat remotament)
- ¿Què passa si l'arxiu B s'esborra o es desplaça a una nova ubicació?
  - En alguns sistemes (ej MacOS) el propi SO corrigeix el path
  - En altres sistemes (ej Linux) l'enllaç deixa de funcionar (queda orfe)
- Esborrar l'arxiu enllaç simbòlic, esborrar A, no afecta al referenciat (B)



## • Enllaços Físics

- Dos o mes entrades de directori contenen el mateix nombre de inode
  - Un únic arxiu al qual s'accedeix per diverses rutes (diversos noms)
- Cada inode manté un comptador amb el nombre de referències al mateix
  - El fitxer només s'esborra físicament a l'eliminar su últim nom
- Només resulta vàlid dins d'un mateix sistema de fitxers

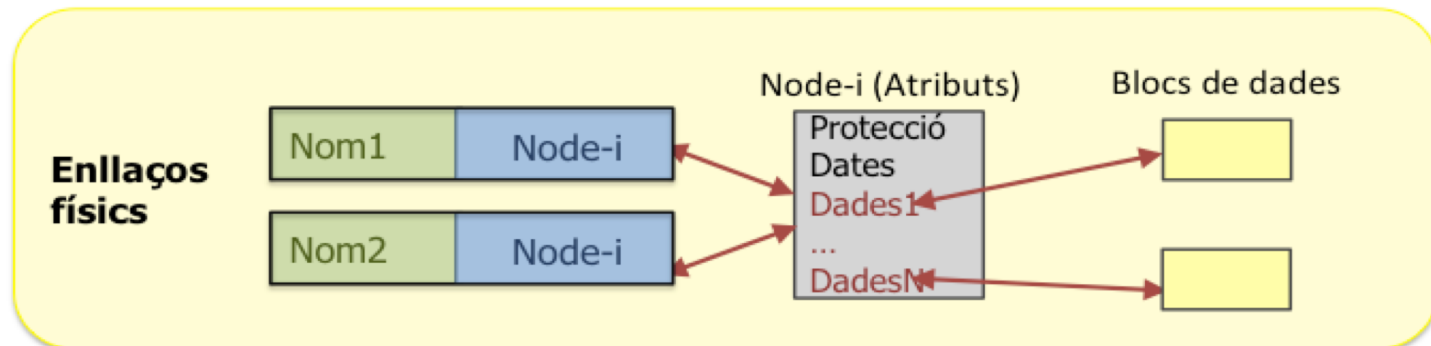


Enllaços físics

inode 10
....
Nº enllaços=2
....
Pter triple ind.

Node-i	Nom Arxiu
6	.
1	..
10	Nom1

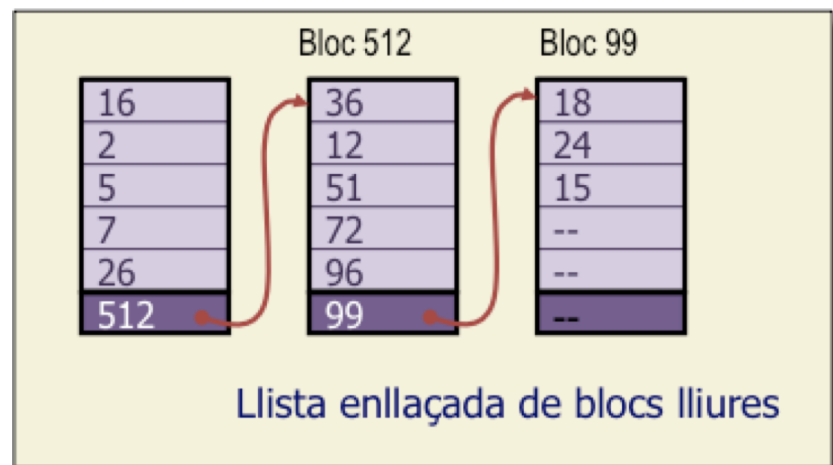
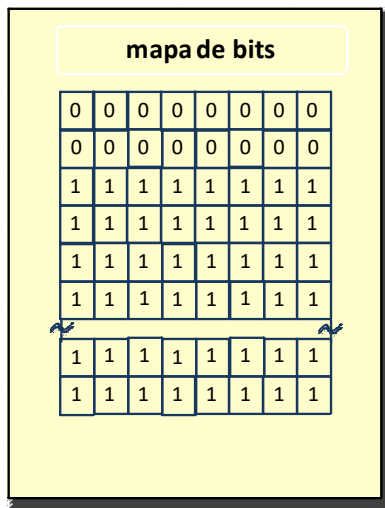
Node-i	Nom Arxiu
20	.
5	..
10	Nom2





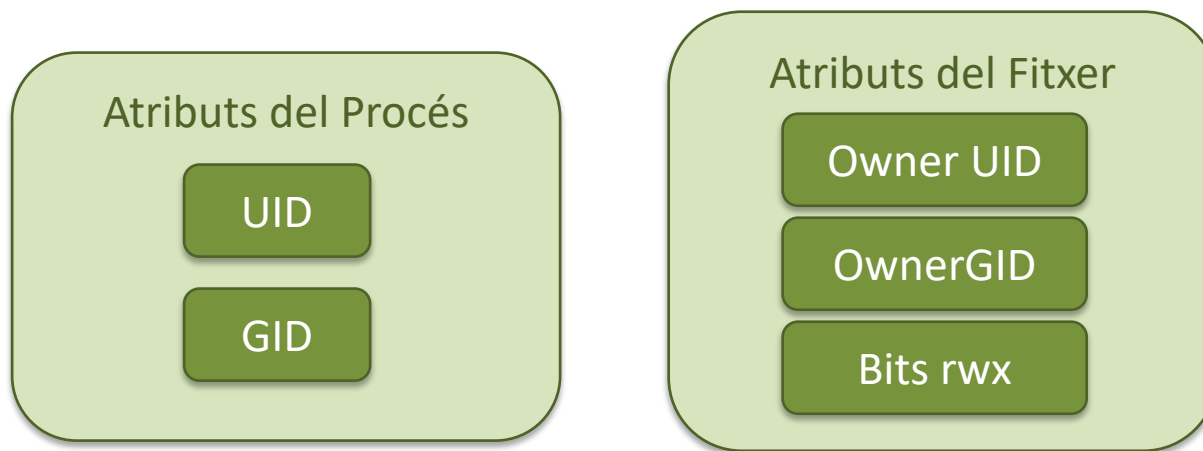
- **Contingut**
  - Concepte de Directori
  - Implementació de directoris
  - Enllaços o referències a arxius
  - **Gestió del espai de disc**
  - Protecció

- La gestió de l'espai lliure de **disc** veu al mateix com a un **vector de blocs**
- En cada moment hem de saber **quins estan lliures**
  - No ens serveix qualsevol bloc
    - Cerquem **contigüitat** (por eficiència posterior)
  - **Mapa de bits**
    - Cada bit representa un bloc en disc (ej. a 1 si bloc lliure)
    - S'emmagatzema en una zona dedicada de disc
    - Recerca eficient de blocs lliures consecutius
  - **Llista enllaçada**
    - En un llocr específic del disc es manté l'índex del primer bloc lliure
    - Cada bloc lliure apunta al següent
  - **Agrupament**
    - Se representen els blocs lliures mitjançant una llista de blocs index
    - Facilita inserir/extraure, complica la recerca blocs lliures consecutius



- **Contingut**
  - Concepte de Directori
  - Implementació de directoris
  - **Enllaços o referències a arxius**
  - Gestió del espai de disc
  - **Protecció**

- **Concepte de protecció**
  - Mecanisme que s'utilitza per a controlar els accesos que els processos realitzen als recursos del sistema
- **¿Com es realitza la protecció en els sistemes UNIX?**
  - La protecció en UNIX està basada en contrastar els atributs del procés amb els atributs de l'arxiu i determinar si la operació es pot efectuar



- **Atributs de protecció de procés**
  - **Identificador d'usuari**
    - UID real (rUID) Identificador de l'usuari que ha creat el procés
    - UID efectiu (eUID) Identificador de l'usuari per al qual s'executa el procés
  - **Identificador de grup**
    - GID real (rGID) Identificador del grup
    - GID efectiu (eGID) Identificador del grup efectiu
- **Atributs de protecció d'arxiu**
  - **Bits de permís:** son 9 bits de permís en grups de tres (propietari, grup, altres)
  - **Formats :** rwxr\_xr\_x. 0755, 04755, rwsr\_xr\_x
  - Interpretació
    - **Arxius regulars:** lectura, escriptura, execució
    - **Directoris:** llistar contingut, crear o eliminar entrades, cercar
    - **Especials:** lectura, escriptura, -----
  - BITS de SETUID, SETGID

- Assignación d'atributs

- El **fitxer** reb els atributs del procés que el crea

`ownerUID = UID`

`ownerGID = GID`

- El **procés** reb els atributs gràcies al mecanisme d'herència i a la informació arreplegada en la taula d'usuari (/etc/passwd)

`nom:contrasenya:UID:GID:descripció:HOME:shell`

- Un procés pot canviar de UID i GID quan faça `exec()` sobre un fitxer amb el SETUID o SETGID activats

- Si l'executable té el bit de SETUID actiu, el eUID passa a ser el del "ownerUID" del fitxer
- Si l'executable té el bit de SETGID actiu, el eGID passa a ser el del "ownerGID" del fitxer

- EjemploExemple

-rwsr—r-x	1	felip	users	17	Jan	29	09:34	arxi1
-rwxr-sr-x	1	felip	users	223	Jan	29	09:34	arxi2

## • Regles de protecció en Unix

Quan un procés intenta realitzar un accés sobre un arxiu, cal comprovar les següents regles en l'ordre que s'indica

1) Si el procés té **eUID=0** (superusuari), no es realitza cap comprovació i s'admet l'accés

→ Excepte el permís d'execució que ha d'estar establert per a algun domini

2) Si el **procés té eUID** igual al del propietari de l'arxiu es comprova la part del propietari

3) Si no, si el **procés té eGID** igual al del grup del propietari de l'arxiu, es comprova la part del grup.

4) Si cap de les consultes anteriors ha funcionat es comprova la part de la resta d'usuaris (altres)

→ Observa que si es comprova un domini no es realitza cap altra comprovació en la resta de dominis. (dominis: usuari, grup, altres).

si UID = 0

aleshores

permís d'accés concedit

sino

si eUID = ownerUID

aleshores

es concedeixen permisos segons la primera tripletta

**rws rwx r-x**

sino

si eGID = ownerGID

aleshores

es concedeix permisos segons la segona tripletta

**rws rwx r-x**

sino

es concedeixen permisos segons a tercera tripletta

**rws rwx r-x**

- **Exemple:**

- `$ ls -l`

total 7

-rwsr-xr-x 1 felip users 17 Jan 29 09:34 ejec1

-rwxr-sr-x 1 felip users 223 Jan 29 11:03 ejec2

-rw----- 1 felip users 5120 Jan 29 12:00 dades

- El fitxer dades només sólo pot ser accedit (per a lectura o escriptura) per felip.
- El fitxer ejec1 té el bit SETUID fixat (s, en propietari) i permisos d'execució per a els altres dominis. Això permet que quan un procés d'un altre domini l'execute, passe al domini del propietari (felip) i pugui llegir/escriure l'arxiu dades.