Sistemes de fitxers

Sistemes Operatius 2

Grau d'Enginyeria Informàtica

Introducció

La informació dels fitxers i directoris s'emmagatzema habitualment a un disc. Com a usuaris

- Treballem amb fitxers i directoris.
- Estem acostumats a treballar amb els fitxers com si fossin dades emmagatzemades de forma "lineal" al disc.

El sistema de fitxers s'encarrega de gestionar de forma transparent tota aquesta informació associada a fitxers a directoris:

- Com s'emmagatzema la informació associada als fitxers? S'emmagatzemen realment de forma "lineal" al disc?
- Com s'emmagatzema la informació associada als directoris?
- Com es gestiona l'espai lliure de disc?

Introducció

L'objectiu d'aquest tema se centra en veure els principis dels sistemes de fitxers:

- El problema d'emmagatzemar, recuperar i manipular informació és un problema molt general i hi ha moltes possibles solucions.
- No hi ha una forma "correcta" d'implementar un sistema d'arxius. Tot depèn del tipus suport d'emmagatzematge de què disposem així com el tipus aplicacions a les quals està dirigit.

Introducció

El suport d'emmagatzematge ha evolucionat molt al llarg dels darrers anys

- El diskette (1970–1999, màxim 200MB)
- El disc òptic (DVD, CD-ROM)
- El disc dur magnètic (els habituals a un ordinador)
- El disc d'estat sòlid (SSD, USB, ...)
- La cinta magnètica (utilitzats per fer *backups*).
- Els discos distribuïts en múltiples ordinadors en una xarxa

Cada suport pot necessitar un tipus específic de sistemes de fitxers. En aquestes transparències ens centrarem en sistemes d'ús general.

Concepte de sector i bloc

Un concepte important en un sistema d'arxius és la mida de bloc:

- Bloc o sector del disc: mínima unitat que el disc pot llegir o escriure (típicament 512 bytes).
- Bloc del sistema d'arxius: és la mínima unitat que el sistema d'arxius (del sistema operatiu) llegeix o escriu. Tot el que fa el sistema d'arxius està compost d'operacions sobre blocs.

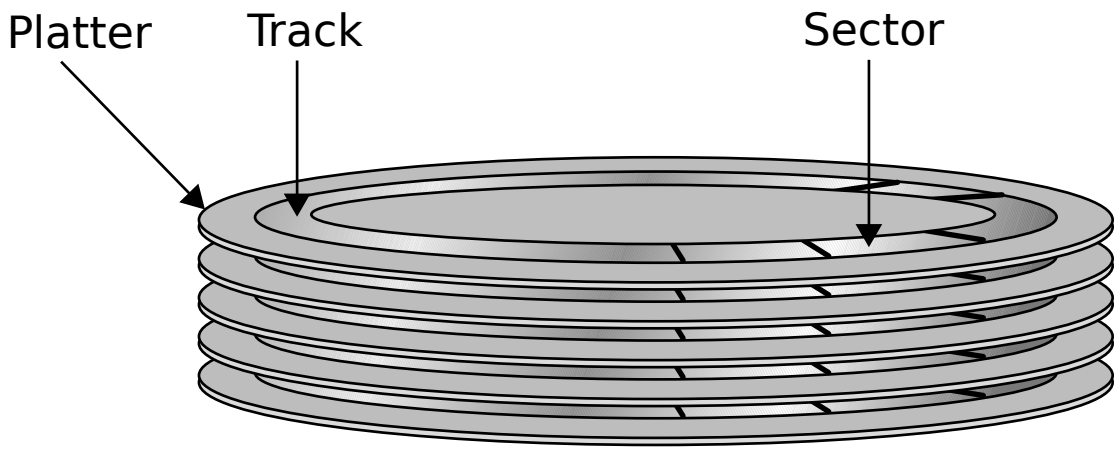
  Un bloc del sistema d'arxius té la mateixa mida o més gran (en múltiples sencers que són potència de 2) que la mida del sector de disc. Són comunes mides de bloc de 1024, 2048, 4096 o 8192 bytes.

  Una bona mida de bloc per Intel és 4096 bytes. Per què penseu que és una bona mida?

El disc dur magnètic

Diagrama d'un disc

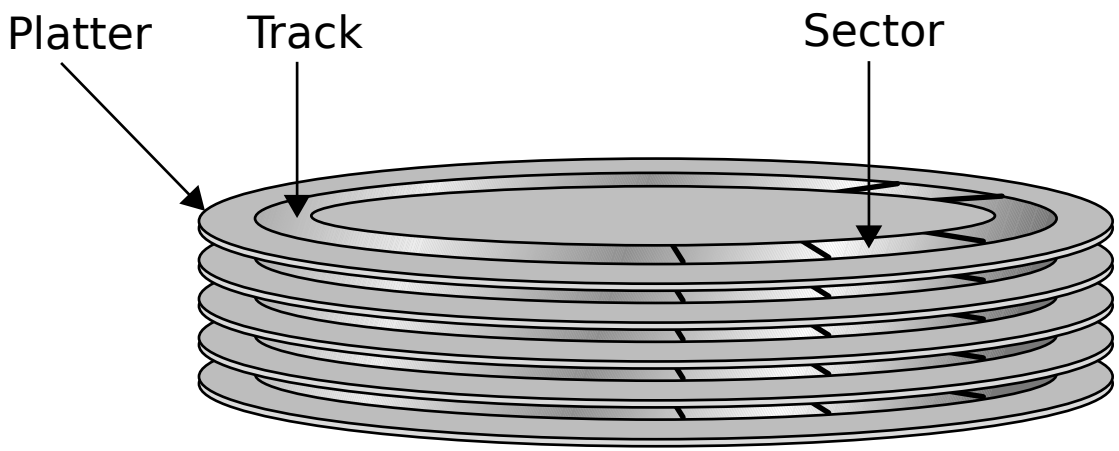
- Un disc dur magnètic està format per diversos discos (*platters*) col·locats un a sobre l'altre. Els discos es poden escriure per les dues bandes.
- Hi ha un únic braç que llegeix els discos amb un capçal de lectura per cada banda del disc. Només es permet una operació de lectura o escriptura en un instant determinat: no es pot llegir o escriure de múltiples discos a la vegada.



El disc dur magnètic

Diagrama d'un disc

- La quantitat mínima que es pot llegir o escriure a un disc és el sector o bloc, que típicament té una mida de 512 bytes.
- Llegir o escriure sectors contigus és una operació molt més ràpida (+ 10 vegades) que fer-ho sobre sectors no contigus. Moure el braç és una operació molt lenta.



El disc dur magnètic

Existeixen multitud de discos de marques diferents, cadascú amb la seva geometria (nombre de discos, nombre de sectors, etc).

- El disc disposa d'un controlador de disc, integrat dins del maquinari. El gestor de disc (*driver*) es comunica amb el controlador per gestionar les operacions a realitzar.
- Els controladors actuals utilitzen l'anomenat adreçament lògic: el gestor de disc veu el disc com un dispositiu d'adreçament lineal. En accedir a una adreça lògica específica, el controlador s'encarrega de traduir l'adreça a la posició física corresponent al disc.

Discos d'estat sòlid

Discos d'estat sòlid (SSD)

- Tenen les mateixes característiques que discs durs tradicionals (sectors, ...) amb l'avantatge que no tenen parts mòbils. Això fa que el seu rendiment sigui molt superior.
- Abans de poder escriure a un bloc de disc cal esborrar-lo (això no passa amb els discs durs), una operació "lenta" en comparació amb la lectura o escriptura d'un SSD.
- Les dades s'emmagatzemen en circuits elèctrics. Els circuits, però, es poden anar degradant al llarg del temps i calen tècniques específiques per assegurar la durabilitat. Cal "evitar" escriure sempre als mateixos sectors!

Requisits del sistema de fitxers: els fitxers i directoris

En un disc volem emmagatzemar fitxers i directoris. Per fer-ho

1. Cal que el sistema de fitxers (del sistema operatiu) emmagatzemi al disc les metadades associades al fitxer: la seva localització al disc, el nom del fitxer, la mida del fitxer, la protecció (drets d'accés), dates de creació i modificació, ...
2. Els sistemes de fitxers habituals permeten organitzar els fitxers en directoris. Els directoris també són metadades i han de permetre emmagatzemar el llistat de fitxers i subdirectoris que conté.

Aquestes metadades és informació que emmagatzema el sistema de fitxers a disc. Podem accedir al contingut d'aquestes metadades fent servir instruccions com "ls", per exemple.

Requisits del sistema de fitxers: operacions sobre fitxers

Quines operacions volem realitzar sobre els fitxers?

- Crear fitxers: el sistema de fitxers hauria de ser capaç de trobar ràpidament espai lliure al disc pel fitxer.
- Escriure a un fitxer: donat el nom d'un fitxer, el sistema de fitxers ha de ser capaç de trobar la localització del fitxer a disc i permetre escriure-hi.
- Llegir d'un fitxer: un cop trobada la localització del fitxer a disc s'han de poder llegir les dades que conté.
- Posicionament en un fitxer: s'ha de permetre posicionar el punt d'escriptura o lectura en qualsevol punt de fitxer.
- Esborrar un fitxer: aquesta operació implica eliminar el fitxer de les metadades del directori així com alliberar tot l'espai que ocupa el fitxer a disc.
- Truncar un fitxer: és una operació que permet alliberar l'espai que ocupa un fitxer sense esborrar-lo.

Requisits del sistema de fitxers: operacions sobre directoris

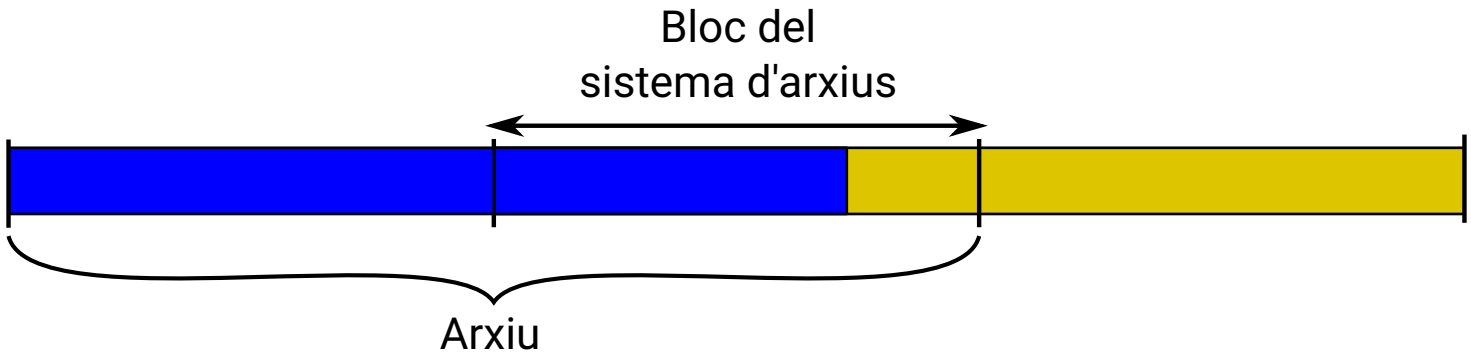
Quines operacions volem realitzar sobre els directoris?

- Buscar un fitxer: poder trobar un fitxer al directori per tal d'accedir a les metadades associades al fitxer.
- Crear un fitxer: permetre afegir nous fitxers a un directori.
- Esborrar un fitxer: permetre eliminar un fitxer d'un directori (i eliminar les metadades associades al fitxer).
- Reanomenar un fitxer: poder canviar el nom d'un fitxer del directori.
- Llistar el contingut d'un directori: poder llistar els fitxers (i subdirectoris) que conté un directori.

Requisits del sistema de fitxers: nivell físic

Als blocs del disc s'han d'emmagatzemar els arxius i les metadades associades.

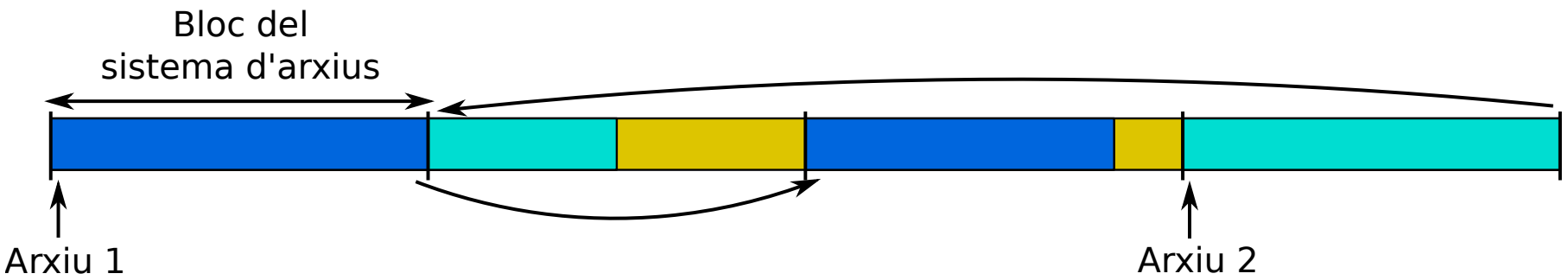
- Aquests "ocupen" un nombre sencer de blocs. Els bytes d'un bloc no utilitzats per un fitxer no poden ser utilitzats per altres arxius.
- No es pot llegir un byte d'un arxiu. Per fer-ho cal llegir tot el bloc del disc, emmagatzemar-lo a memòria i després accedir al byte demanat.



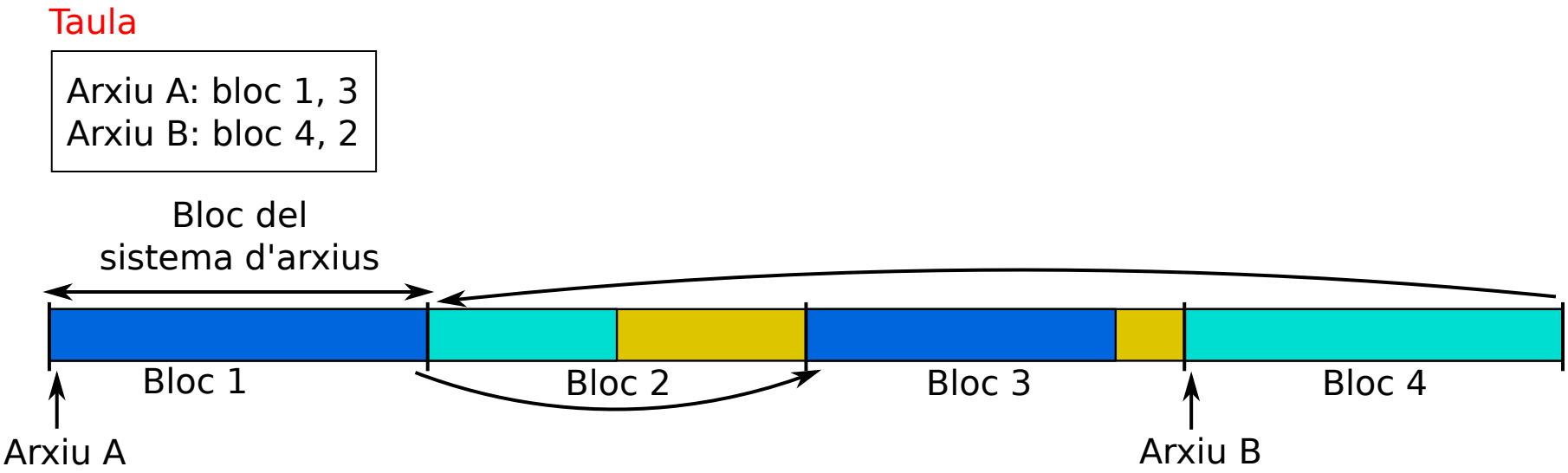
Requisits del sistema de fitxers: nivell físic

En manipular arxius "veiem" els bytes dels arxius contigus (un byte al darrera l'altre). Però...

- Els arxius (així com les seves metadades) no tenen perquè estar emmagatzemats de forma contigua al disc.
- Idealment els blocs d'un arxiu han de ser contigus a disc, ja que així el rendiment del disc (magnètic) és major.



Fonaments dels dissenys implementats



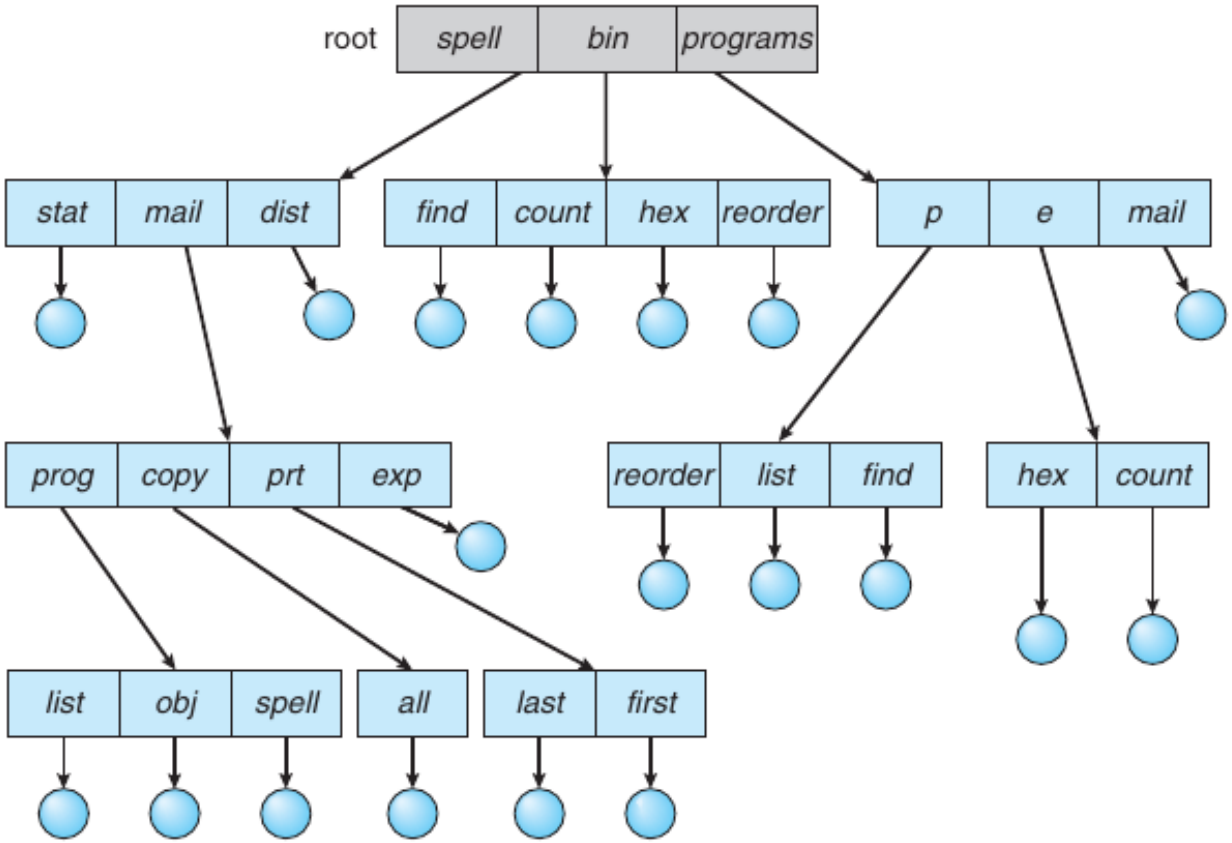
Ens preguntem

- Com implementar el fet que un arxiu està format per blocs no contigus?
- Com accedir de forma eficient a un bloc d'un arxiu i fer-ho escalable per fitxers petits i grans?

Al llarg de l'evolució dels sistemes d'arxius hi ha hagut diverses formes d'implementar-ho. Aquí només veurem els principis.

Fonaments dels dissenys implementats

Com emmagatzemem la informació dels directoris? A l'actualitat estem acostumats a fer servir sistemes d'arxius que permeten fer servir una estructura de directori...



Fonaments dels dissenys implementats

De forma intuïtiva, en un disc cada directori s'emmagatzema com un fitxer amb una estructura similar a una taula. Aquesta taula

- Conté el llistat de fitxers, amb la mida, els permisos, ... així com un "apuntador" cap als blocs que formen part del fitxer.
- Cada directori pot contenir subdirectoris. Per a cada subdirectori hi ha un "apuntador" cap al fitxer que conté els fitxers del subdirectori.

Com gestionar aquestes taules de forma eficient? No és una tasca senzilla...

Fonaments dels dissenys implementats

Com gestionar l'espai lliure al disc?

- Els sistema de fitxers ha de saber quins blocs estan lliures. I si són contigus, millor! Els sistemes de fitxers han d'intentar evitar la fragmentació d'un fitxer.
- L'organització dels fitxers (a un disc magnètic) s'hauria de fer de forma que els directoris (les taules) i els fitxers continguts als directoris estiguin físicament propers. És el que en termes tècnics s'anomena l'heurística de localitat.

Objectiu: disseny d'un sistema de fitxers!

Es disposa d'una gran flexibilitat a l'hora de dissenyar un sistema de fitxers. En dissenyar el sistema,

- Els blocs que formen part del fitxers haurien de ser contigus al disc.
- Disposar d'un lloc al disc on emmagatzemar les metadades dels fitxers: la seva mida en bytes, la data de creació, ... així com els blocs que formen part del fitxer.
- Accés eficient a un bloc d'un fitxer (evitar haver de moure gaire el braç del disc per accedir a cadascun dels blocs que formen el fitxer).
- Ser escalable per donar suport a fitxers petits i grans.

S'han proposat múltiples sistemes de fitxers. Anem a veure els principis bàsics

Principis bàsics

Principis bàsics dels sistemes de fitxers

- FAT (File Allocation Table). Desenvolupat a finals del 1970, utilitzat avui en dia a memòries flash i càmeres digitals on la simplicitat és primordial.
- FFS (Unix Fast File System). Desenvolupat a mitjans de 1980. És la base de molts sistemes de fitxers utilitzats avui en dia (ext3, ext4, NTFS, ...)

Què els diferencia?

- FAT: utilitzen una llista enllaçada per emmagatzemar els blocs que formen part d'un fitxer.
- FFS: utilitzen vectors per emmagatzemar els blocs que formen part d'un fitxer.

Sistemes FAT

Els sistemes FAT

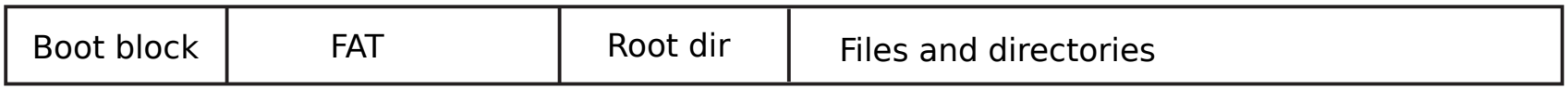
- Desenvolupat per Microsoft a finals del 1970. Va ser utilitzat pel sistemes operatius MS-DOS i les primeres versions de Windows.
- Ha sigut millorat aquests darrers anys per donar suport a la "gran" capacitat d'emmagatzematge dels dispositius dels darrers anys.
- És utilitzat avui en dia als discos i llàpisos USB, a la tarja de memòria de les càmeres fotogràfiques, etc.

Sistemes FAT

En sistemes FAT està estructurat d'aquesta forma (en localitzacions conegudes pel sistema de fitxers).

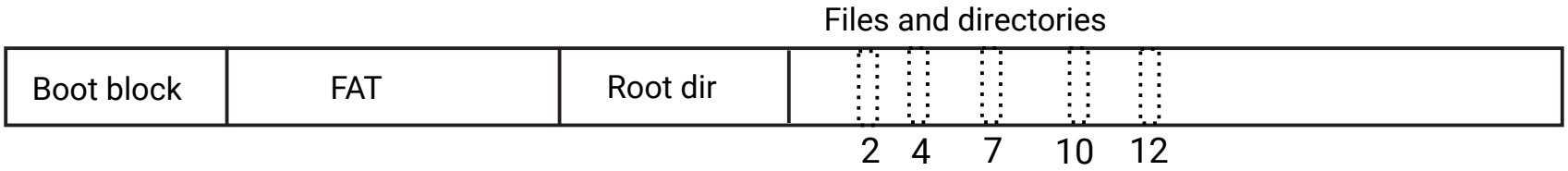
- La taula FAT: és una taula que es farà servir com a llista enllaçada. Emmagatzema la informació sobre els blocs que formen part de cada fitxer.
- El directori arrel: és el (primer) bloc del directori arrel del sistema de fitxers. La seva posició es coneguda pel sistema de fitxers.
- Fitxers i directoris: aquí és on s'emmagatzemen els fitxers i directoris del disc. Es va omplint a mesura que s'hi afegeixen dades.

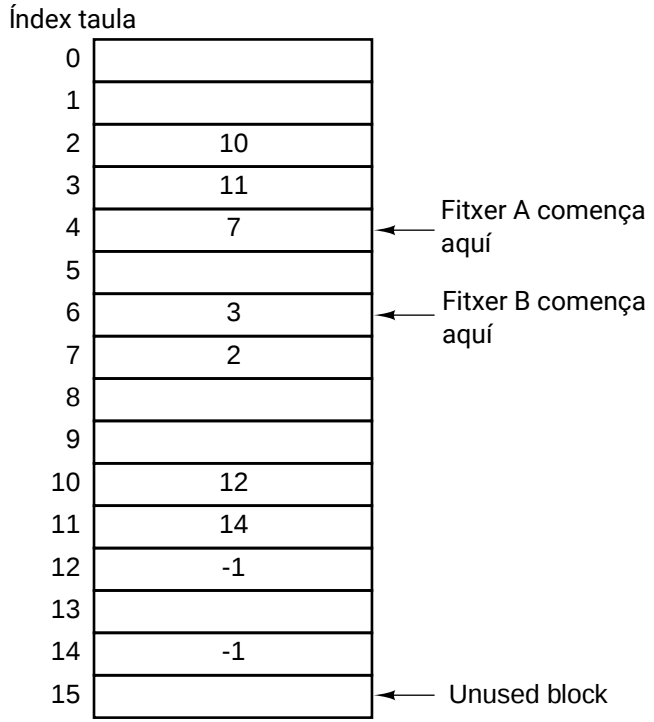
Ara veurem els detalls de cada punt...



La taula FAT (File Allocation Table)

La FAT és una taula de sencers que emmagatzema, en forma de llista enllaçada, els blocs de què es componen els fitxers[1].





Per a l'exemple:

- El fitxer A ocupa els blocs 4, 7, 2, 10 i 12. El fitxer B ocupa els blocs 6, 3, 11 i 14.
- El blocs buits així com el "final" d'un fitxer es marquen amb un codi especial.

[1] La mida de la taula FAT és fixa; depèn de la capacitat del disc.

Els directoris a la FAT

Els directoris s'emmagatzemen a disc com un fitxer: cada directori és un fitxer que emmagatzema una taula (no ordenada) dels fitxers que componen el directori. A la taula:

- Emmagatzemen metadades com la mida en bytes, la seva data de creació, ...
- També s'hi emmagatzema el primer índex a la llista enllaçada. Pel fitxer A, s'hi emmagatzemaria un 4. Pel fitxer B s'hi emmagatzemaria un 6.
- Si hi hagués subdirectoris, hi hauria una entrada per a cada subdirectori que apunta al fitxer (i.e. la "taula") que emmagatzema el contingut del subdirectori.

Fitxer B	Mida en bytes, data creació, etc.	Primer index a la taula FAT
Fitxer A	Mida en bytes, data creació, etc.	Primer index a la taula FAT
Fitxer Z	Mida en bytes, data creació, etc.	Primer index a la taula FAT
Fitxer G	Mida en bytes, data creació, etc.	Primer index a la taula FAT
Fitxer K	Mida en bytes, data creació, etc.	Primer index a la taula FAT

Trobar espai buit i fitxers contigus a la FAT

El sistema de fitxers fa servir un esquema molt senzill

- Per trobar espai buit per un nou fitxer, el sistema de fitxers recorre la taula FAT per veure si hi ha un element no ocupat. Cada bloc lliure es marca amb un "codi" especial.
- S'acostumen a fer servir algorismes molt senzills per emmagatzemar fitxers que ocupin múltiples blocs: se cerca a la taula "el següent" bloc lliure. Això fa que els fitxers a un sistema FAT puguin estar molt fragmentats[2].

[2] Els sistemes FAT disposen d'un programari, un defragmentador, que "mou" les dades al disc per aconseguir que els fitxers estiguin contigus al disc.

Limitacions del sistema FAT

El sistema de fitxers FAT es molt utilitzat per la seva simplicitat (dispositius USB, càmeres digitals, ...). Però té moltes limitacions

- Fitxers fragmentats: l'estratègia utilitzada per aquest sistema de fitxers fa que els fitxers acostumin a estar molt fragmentats.
- Accés aleatori pobre: per poder accedir a un bloc del fitxer, cal recórrer la llista enllaçada de la taula FAT.
- Taula de fitxers d'un directori: el llistat de fitxers d'un directori no està "ordenat" i si el directori conté milers de fitxers... uff...
- Control d'accés pobre: no es permet controlar l'accés a diferents usuaris.
- Seguretat: el sistema FAT no incorpora sistemes de seguretat (com els sistemes moderns) per assegurar, per exemple, la corrupció del disc en cas l'ordinador es pengi.

Principis bàsics

Principis bàsics dels sistemes de fitxers

- FAT (File Allocation Table). Desenvolupat a finals del 1970, utilitzat avui en dia a memòries flash i càmeres digitals on la simplicitat és primordial.
- FFS (Unix Fast File System). Desenvolupat a mitjans de 1980. És la base de molts sistemes de fitxers utilitzats avui en dia (ext3, ext4, NTFS, ...)

Què els diferencia?

- FAT: utilitzen una llista enllaçada per emmagatzemar els blocs que formen part d'un fitxer.
- FFS: utilitzen vectors per emmagatzemar els blocs que formen part d'un fitxer.

Sistemes FFS (Unix Fast File System)

Els sistemes FFS incorpora idees interessants

- Els blocs que formen part d'un fitxer s'indexen de forma que es pugui treballar de forma eficient tant per fitxers petits com fitxers grans.
- Es permet controlar quins usuaris poden accedir (per lectura, escriptura o execució) a cadascun dels fitxers.
- El sistema inclou heurístiques que aconsegueixen que els fitxers estiguin "repartits" al disc en blocs contigus. D'aquesta forma el braç no s'ha de moure gaire per trobar/llegir un fitxer.

Sistenes FFS (Unix Fast File System)

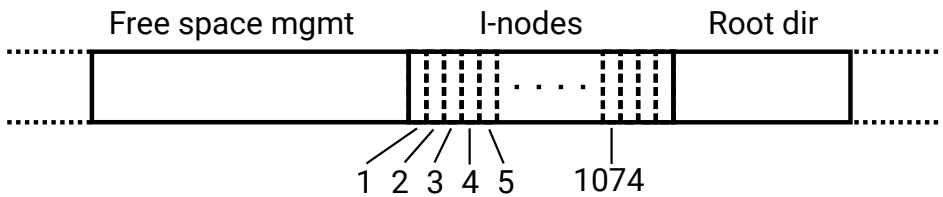
En sistemes Unix, en general, està estructurat d'aquesta forma (en localitzacions conegudes pel sistema de fitxers).

- Els i-nodes es fan servir per emmagatzemar els blocs que componen un fitxer.
- El directori arrel: és el (primer) bloc del directori arrel del sistema de fitxers. La seva posició es coneguda pel sistema de fitxers.
- Al Free space mgmt és on s'emmagatzema la informació els blocs lliures a disc.
- Fitxers i directoris: aquí és on s'emmagatzemen els fitxers i directoris del disc. Es va omplint a mesura que s'hi afegeixen dades.

Sistemes FFS (Unix Fast File System)

Els i-nodes (que emmagatzemen els blocs que formen part d'un fitxer) s'emmagatzema en una posició coneguda del sistema de fitxers.

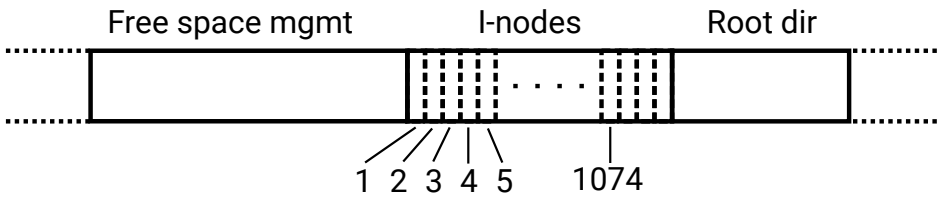
- Cada i-node té una mida fixe (en bytes) i té associat un índex, un valor numèric. El sistema de fitxers en pot localitzar un ràpidament.
- En especificar l'i-node 1074, per exemple, el sistema de fitxers sap localitzar-lo de forma immediata al disc.



Els i-nodes

En un sistema FFS

- Un directori, intuïtivament, és un fitxer (la taula) amb els noms de fitxers que conté i l'i-node que li correspon.
- El nombre d'i-nodes està prefixat! Què implica?[3]



Organització del disc

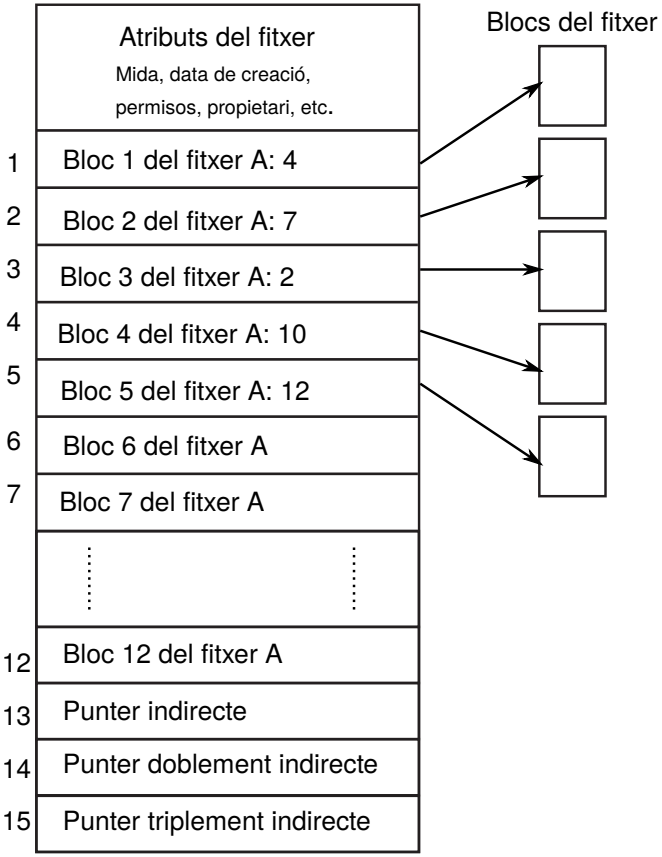
Fitxer B	i-node
Fitxer A	i-node
Fitxer Z	i-node
Fitxer G	i-node
Fitxer K	i-node

Fitxer de directori

[3] En un sistema FFS hi ha un i-node per a cada 16KB d'espai de disc, més que suficient per a les necessitats de l'usuari.

Els i-nodes

Els i-nodes (que tenen mida fixa!) emmagatzemen els atributs del fitxer així com els indexos dels blocs que formen part d'un fitxer.



Els i-nodes

Hi ha un i-node per a cada fitxer. Però, aleshores, com es manegen fitxers de mida diferent?

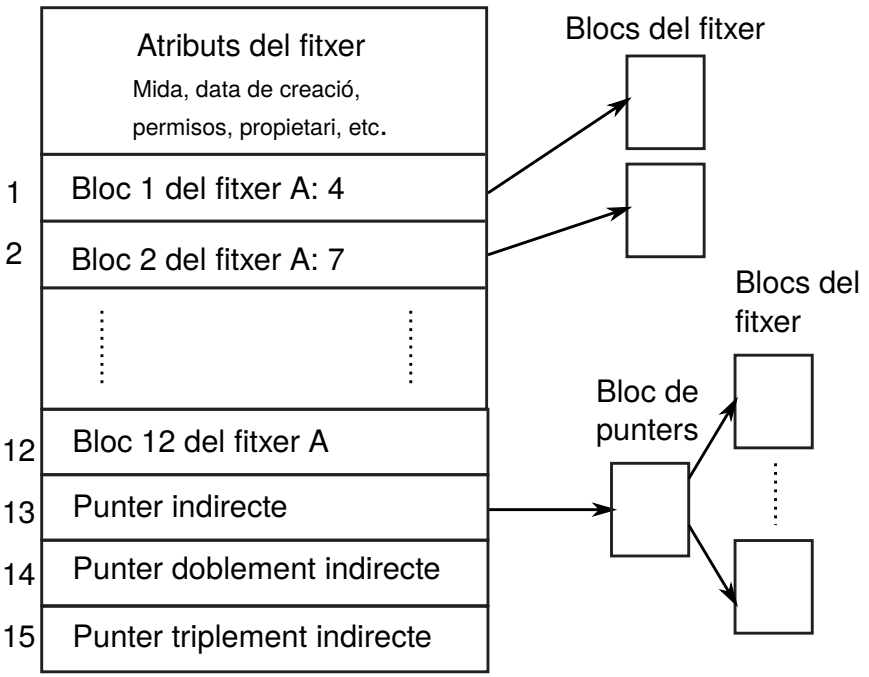
- En un sistema de fitxers amb una mida de bloc de 4096 bytes, si només fem servir els punters directes (vegeu dibuix) implica una mida màxima de fitxer de $12 \times 4096 = 48\text{KB}$. Això era suficient per a molts fitxers – en aquella època – ja que eren petits (i avui en dia també!).
- Què passa si el fitxer es fa més gran que el limit de 48KB?

Cada i-node proporciona diferents tipus de "punters" per permetre emmagatzemar fitxers molt més grans...

- Cada i-node conté un vector de punters cap als blocs que formen el fitxer. En total n'hi ha 15: els 12 primers són punters directes (veieu dibuix anterior), i després conté altres tipus de punters... veiem-ho!

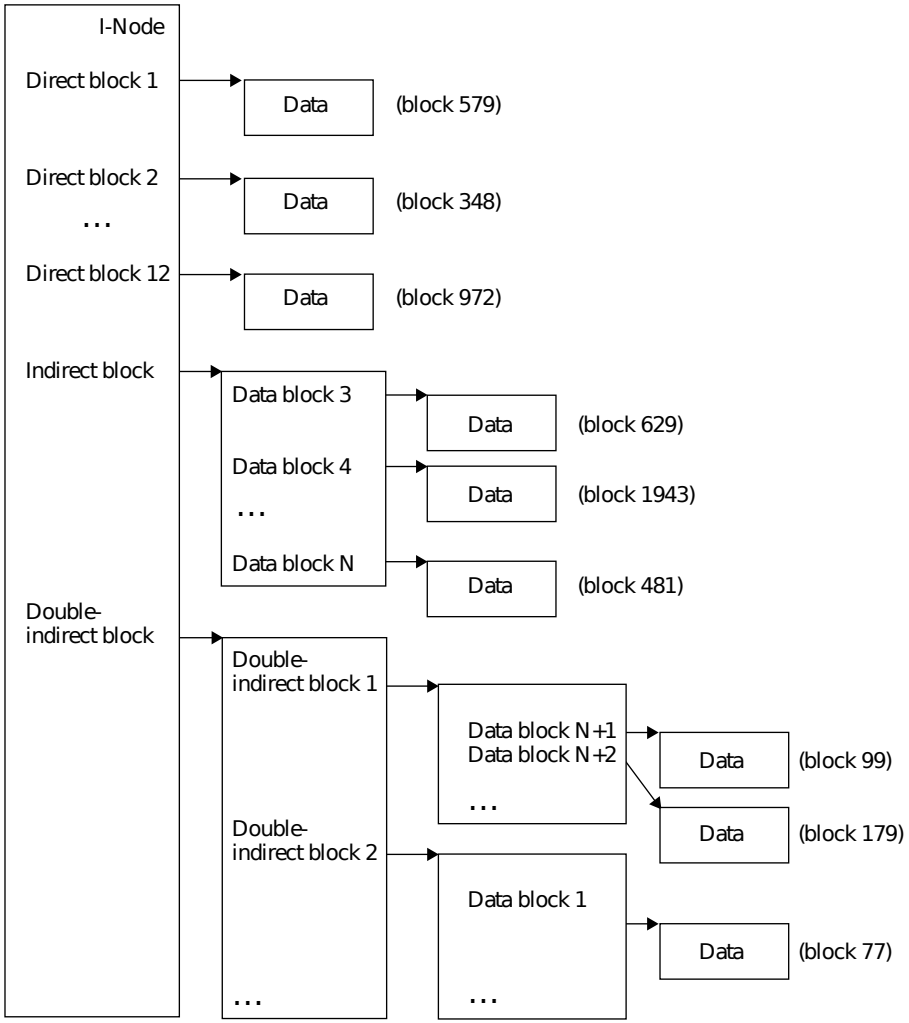
Els i-nodes

Els punters indirectes apunten a un bloc que conté (només) punters directes. Amb blocs 4 KB i punters de 4 bytes, un bloc de punters (indirectes) pot contenir 1024 punters. Això implica ara una mida de fitxer de $48\text{KB} + 1024 \times 4\text{KB} \approx 4\text{MB}$.



Els i-nodes

Els i-nodes, a més de les referències directes i les indirectes també suporten referències doblement indirectes:



Els i-nodes

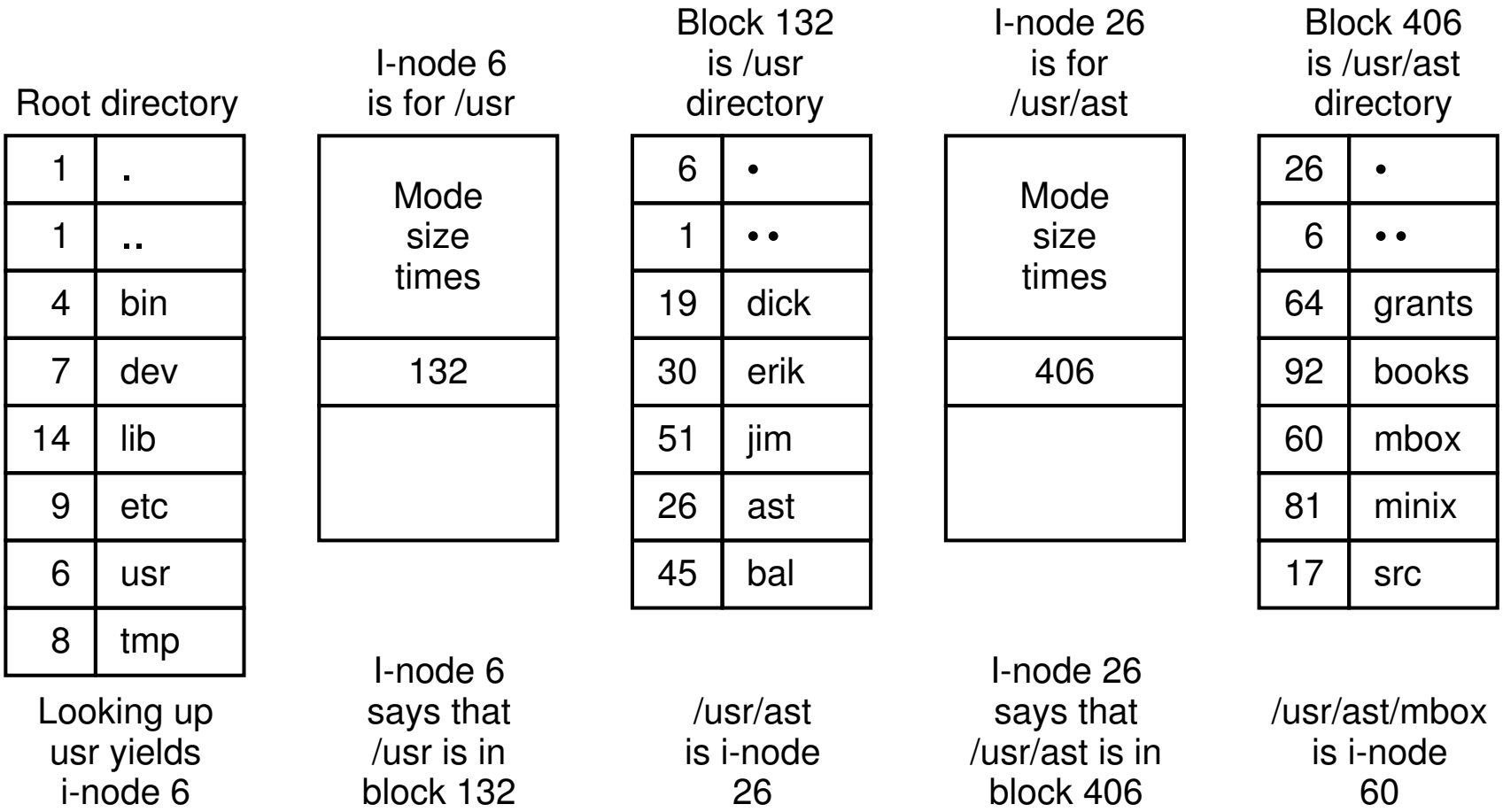
En resum

- Amb 12 punters directes podem apuntar a 48KB de dades.
- Amb 1 punter indirecte (que emmagatzema 1024 punters directes) podem apuntar a 4MB de dades.
- Amb 1 punter doblement indirecte (que pot apuntar a 1024^2 punters directes) podem apuntar a 4GB de dades.
- Amb 1 punter triplement indirecte podem apuntar a 4TB de dades

El sistema FFS (dels anys 80!) utilitza un arbre fixe multinivell asimètric capaç de donar suport a fitxers petits i grans de forma eficient. Múltiples sistemes de fitxers actuals han adoptat aquest enfoc (amb diverses optimitzations).

Els i-nodes: exemple de localització d'un fitxer

Veiem un exemple complet. Suposem que es vol obrir el fitxer `/usr/ast/mbox` del sistema de fitxers. Quins passos s'han de seguir per aconseguir-ho ?

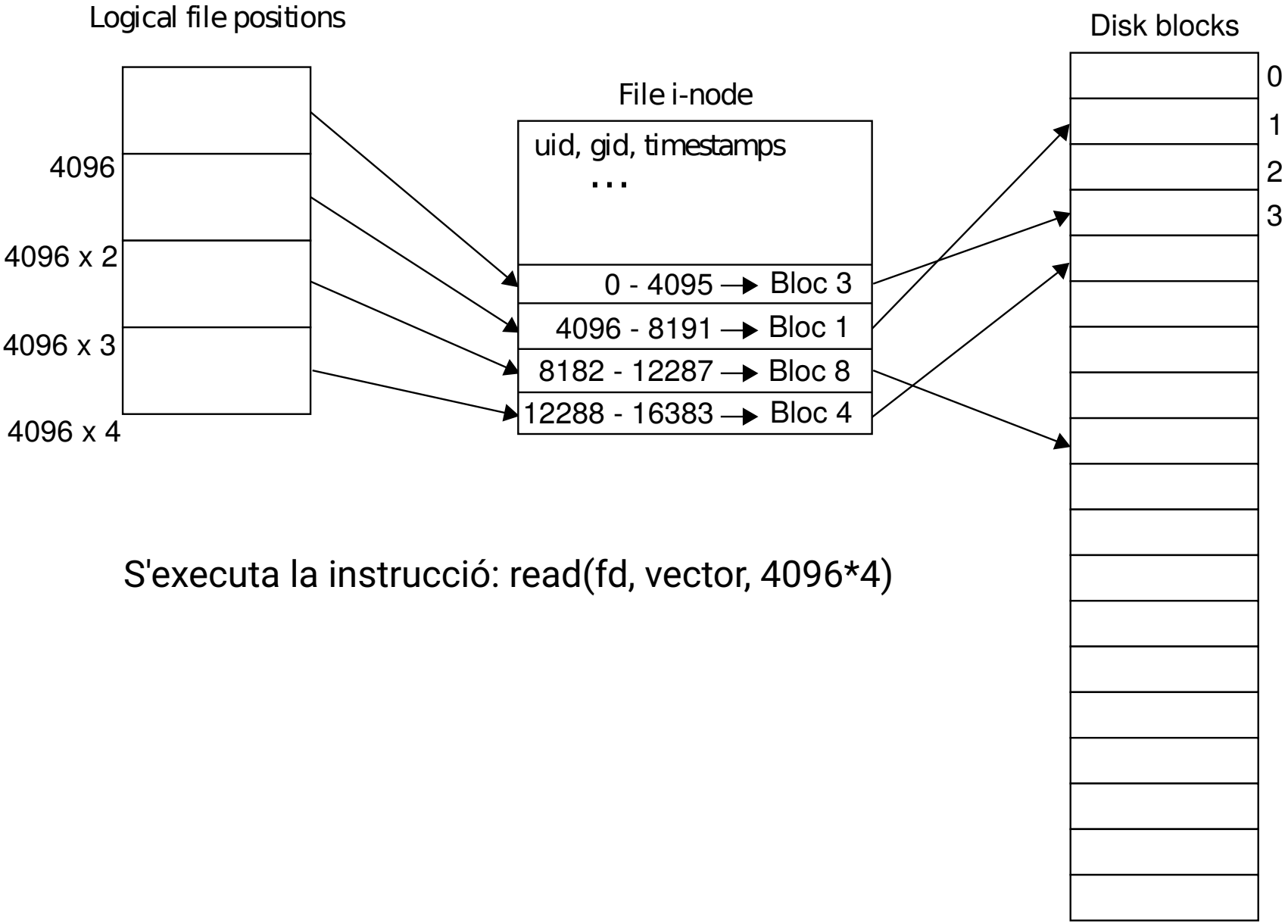


Els i-nodes: exemple de localització d'un fitxer

Passos a seguir

1. El sistema de fitxers coneix la localització del directori arrel. Cal cercar-hi l'entrada `usr`.
2. L'entrada `usr` fa referència a l'i-node 6 en què hi ha indicat que els continguts d'aquest directori es troben al bloc 132.
3. Al bloc 132 es cerca la paraula `ast`.
4. El directori `ast` es troba a l'i-node 26 en què hi ha indicat que els continguts són al bloc 406.
5. Al bloc 406 es cerca la paraula `mbox`.
6. El fitxer `mbox` té associat l'i-node 60. Es llegeix aquest i-node i ja es pot obrir el fitxer! Un cop obert, el sistema operatiu guarda internament l'i-node associat al fitxer.

Els i-nodes: exemple de lectura d'un fitxer



S'executa la instrucció: read(fd, vector, 4096*4)

Els i-nodes: exemple de lectura d'un fitxer

A la figura anterior

- Se suposa una mida de bloc de 4096 bytes.
- Un usuari del sistema operatiu "veu" que els bytes (o blocs) que formen un fitxer qualsevol són contigus .
- L'i-node fa el mapat entre els bytes lògics i la posició física a disc.

Exemple

- L'usuari executa la instrucció C

```
read(fd, vector, 4096*4);
```

- El codi del sistema d'arxius s'ha d'encarregar de
  - Dividir la instrucció anterior en els blocs que el componen.
  - Fer la petició de lectura de cadascun dels blocs.
  - Unir cadascun dels blocs llegits en un sol vector.

FFS: resum del que hem vist fins ara

Resumim

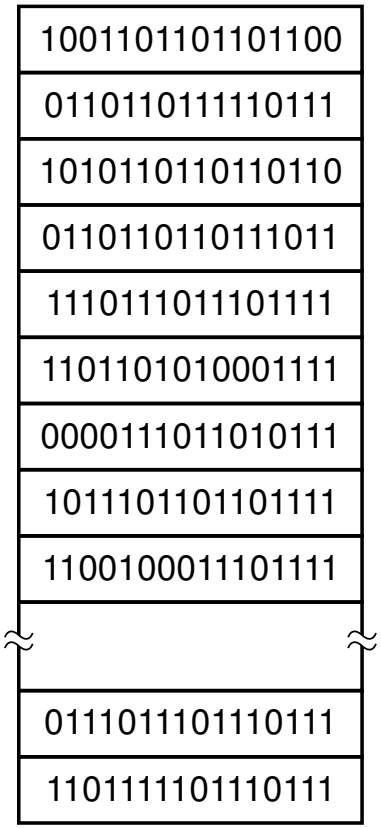
- Cada fitxer té associat un i-node, el qual es pot interpretar com l'arrel d'un arbre que emmagatzema els blocs que formen part del fitxer.
- L'arbre és asimètric i permet tractar de forma eficient fitxers petits i grans (nodes directes, nodes indirectes, nodes doblement indirectes, nodes triplement indirectes).

Falta per veure

- Com es gestiona l'espai lliure?
- Com podem gestionar l'heurística de la localitat dels fitxers?

FFS: com se cerca espai buit?

Per cercar espai buit, el FFS utilitza un *bitmap* emmagatzemat a disc en què cada bloc es representa amb un bit.

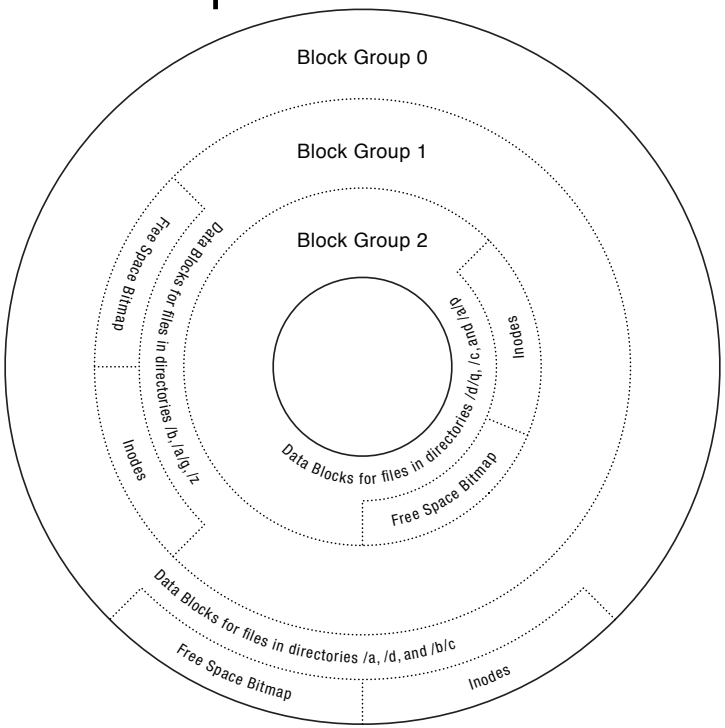


- El *bimap* s'emmagatzema en una posició coneguda pel sistema de fitxers.
- Avui en dia es fan servir, per exemple, estructures d'arbre per localitzar blocs lliures amb la mida demanada.

FFS: heurística de localitat

En el sistemes basats en FFS

- El disc es divideix el grups. Cada grup permet emmagatzemar les metadades (els i-nodes, directoris, el bitmap d'espai buit, i el que faci falta).
- Un determinat directori està associat a un determinat grup. El sistema FFS (intenta) emmagatzemar els fitxers del directori en el grup que li correspon.



Principis bàsics

Hem vist els principis bàsics dels sistemes de fitxers

- FAT (File Allocation Table). Desenvolupat a finals del 1970, utilitzat avui en dia a memòries flash i càmeres digitals on la simplicitat és primordial.
- FFS (Unix Fast File System). Desenvolupat a mitjans de 1980. És la base de molts sistemes de fitxers utilitzats avui en dia (ext3, ext4, NTFS, ...)

El sistema FFS és la "base" dels sistemes de fitxers més moderns. Compareu-lo amb el FAT.

Què és el que s'ha millorat aquests darrers anys?

Millores en els sistemes de fitxers (FFS)

Les millores s'han centrat en el rendiment i la consistència de les dades:

- Els directoris, en comptes de ser una taula no ordenada d'elements, s'ha millorat fent servir (pex) tècniques d'arbre balancejats per emmagatzemar-hi els elements que en formen part.
- Es fan servir tècniques de journalling que assegura la consistència de les metadades en cas de fallada de la llum.
- Hi ha sistemes que utilitzen una tècnica anomenada Copy-on-Write (COW). Permet emmagatzemar, de forma implícita i eficient, còpies de seguretat d'un mateix sistema de fitxers en un disc. Si una còpia falla, es pot "recuperar" una antiga còpia.

Discos d'estat sòlid

Els sistemes que hem vist han estat dissenyats i/o optimitzats per a discos magnètics amb parts mecàniques. Però... i els discos d'estat sòlid?

- No tenen parts mòbils. Això fa que el seu rendiment sigui molt superior que els discos magnètics, especialment per accés aleatori.
- Sistemes de fitxers com FAT els basats en el FFS utilitzats molt avui en dia als discos magnètics no són adequats per aquests tipus de dispositius. Per què? Perquè accedeixen a les mateixes posicions físiques per emmagatzemar les dades, cosa que fa degradar ràpidament un disc d'estat sòlid.

Discos d'estat sòlid

Per augmentar la durabilitat dels discos d'estat sòlid

- A l'actualitat els discos d'estat sòlid poden fer servir els sistemes de fitxers habituals (FAT, FFS, NTFS, ...) perquè utilitzen una tècnica anomenada "wear leveling". Sabeu el que és? Busqueu...
- Es poden fer servir sistemes de fitxers específics per a aquests tipus de dispositius. Són els anomenats log-structured filesystems.

Conclusions

Hem vist l'evolució històrica dels sistemes de fitxers

- La capacitat d'emmagatzematge ha crescut pràcticament de forma exponencial i això ha fet necessari re-dissenyar cada cop aquests sistemes per tal d'incloure-hi les funcionalitats necessàries.
- Tot sistema de fitxers té limitacions i avantatges. Feu servir el sistema de fitxers que us faci falta...
- Els sistemes de fitxers continuen evolucionant... i avui en dia es fan servir molt els sistemes distribuïts! Us imagineu com funcionen?