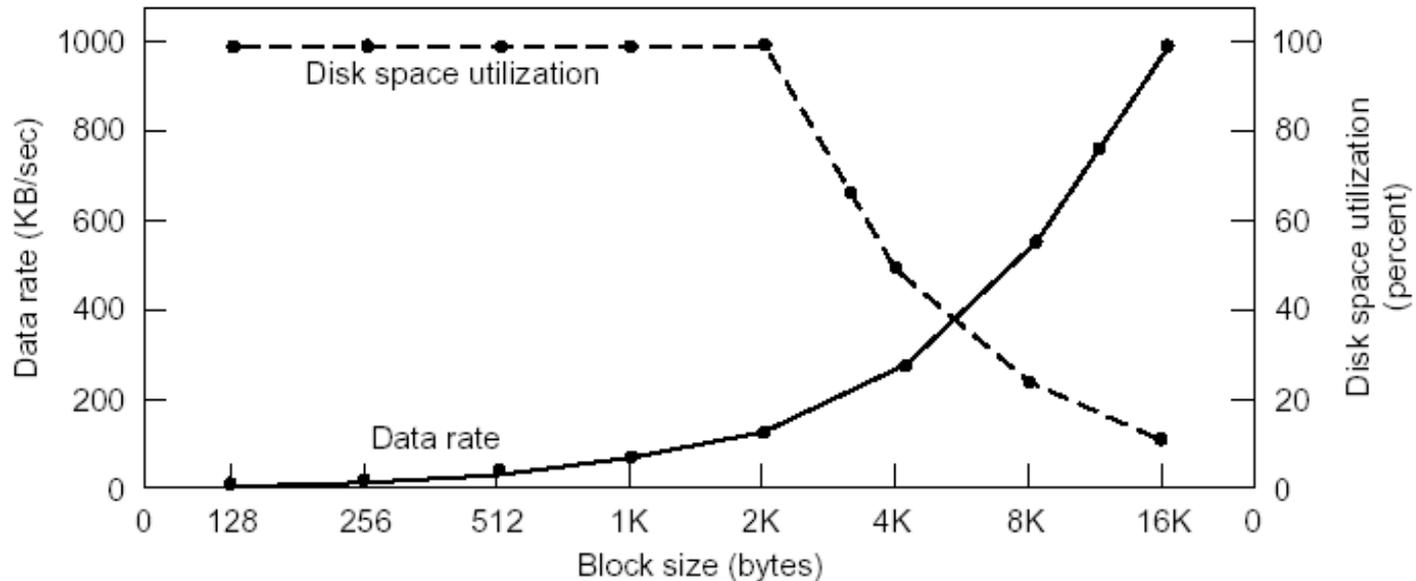


File system II

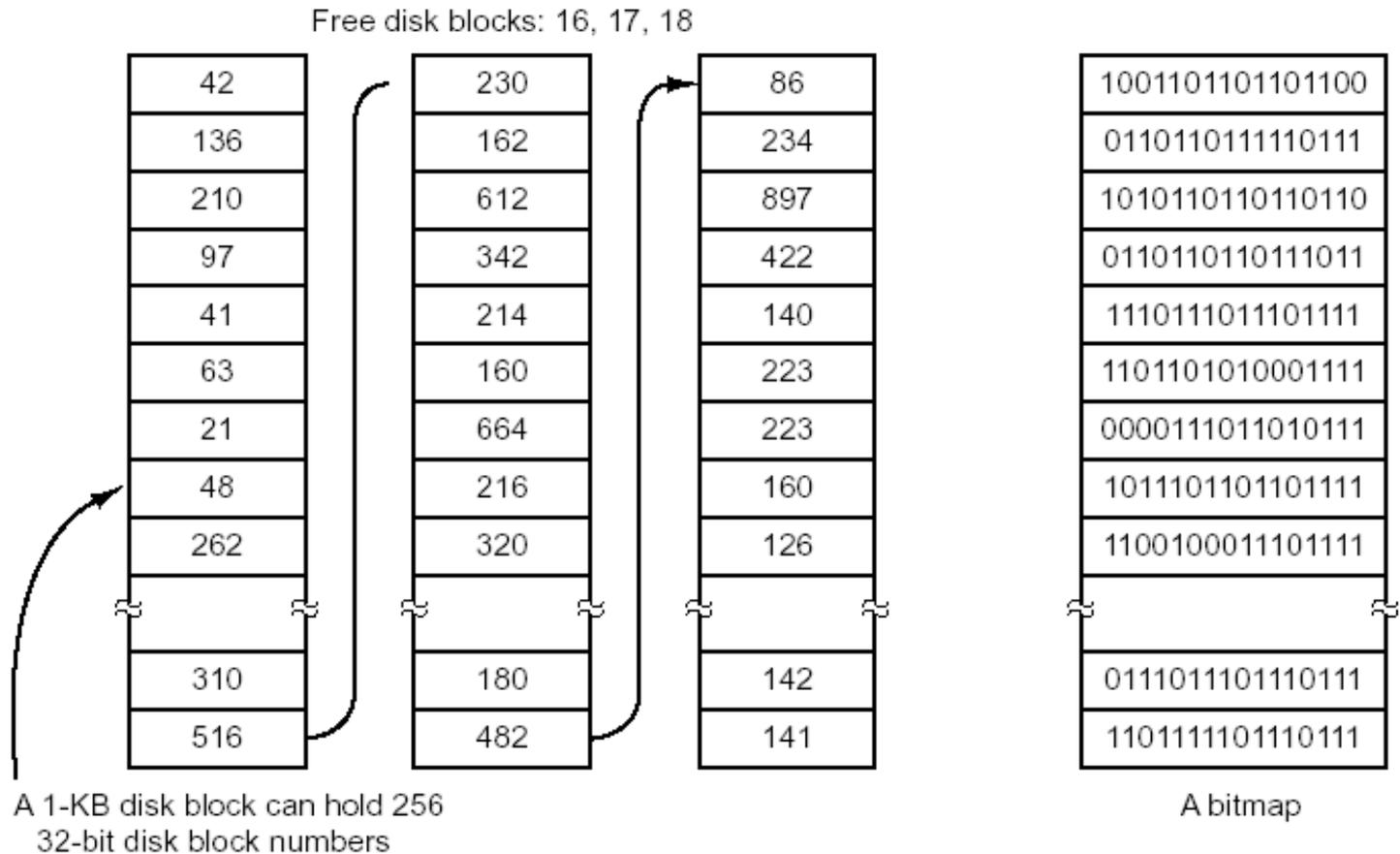
Sistemi Operativi
Lez. 20

Gestione spazi su disco

- Esiste un trade-off, tra spreco dello spazio e velocità di trasferimento in base alla dimensione del blocco fisico



Gestione spazio su disco (1)



Per un disco di 256 GB, con blocchi da 1K ho bisogno di 1.052.689 blocchi per contenere la free-list dei 2^{28} blocchi

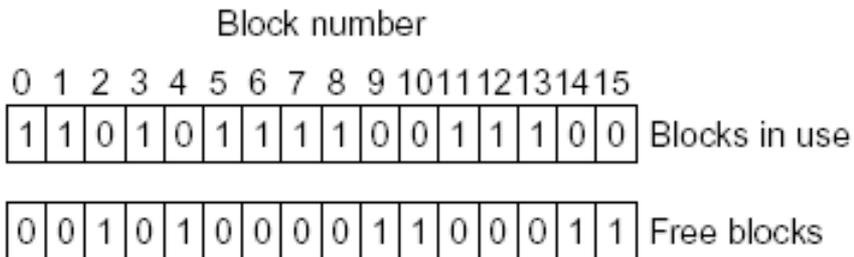
FS Reliability

- **Problema:** i dispositivi di memorizzazione hanno un indice di affidabilità abbastanza basso, e sono tra i primi componenti che si guastano di un sistema
- **Soluzione:** il FS oltre ad alcune tecniche di recovery automatico, strumenti di back-up e recovery per consentire all'utente di ovviare ad eventuali danni
 - Back-up incrementali: solo i dati modificati dall'ultimo back-up sono ricopiati

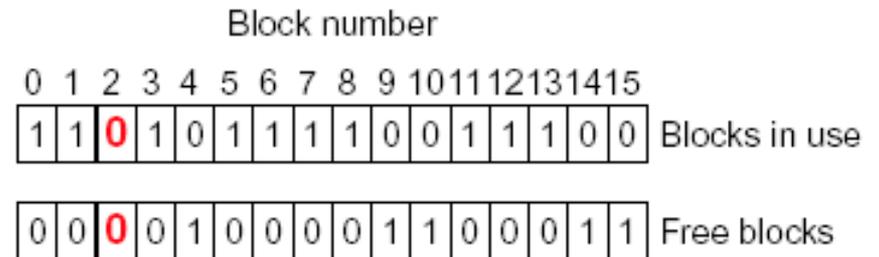
Consistenza FS

- Periodicamente il file system si fa carico di verificare la consistenza delle strutture dati di riferimento
- Per fare quest'attività fa ricorso ad un programma (fsck in Unix e chkdsk in Windows) che sono eseguiti periodicamente in fase di boot, o dopo un guasto o lo spegnimento accidentale di un computer

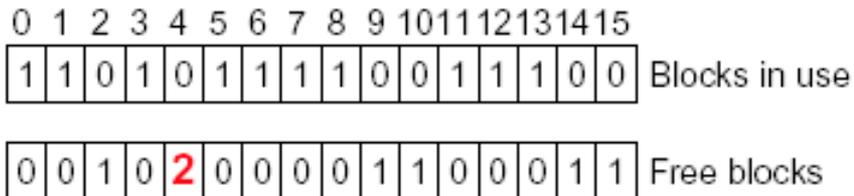
Controllo consistenza dei blocchi



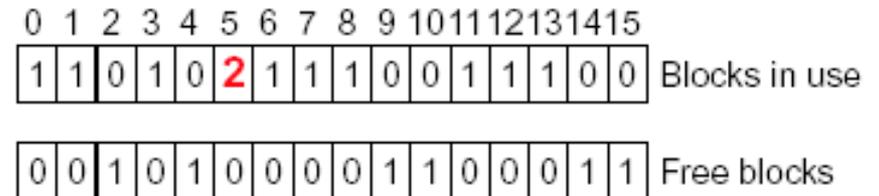
(a)



(b)



(c)



(d)

- a) FS consistente, b) Missing block, c) double free block, d) double used block

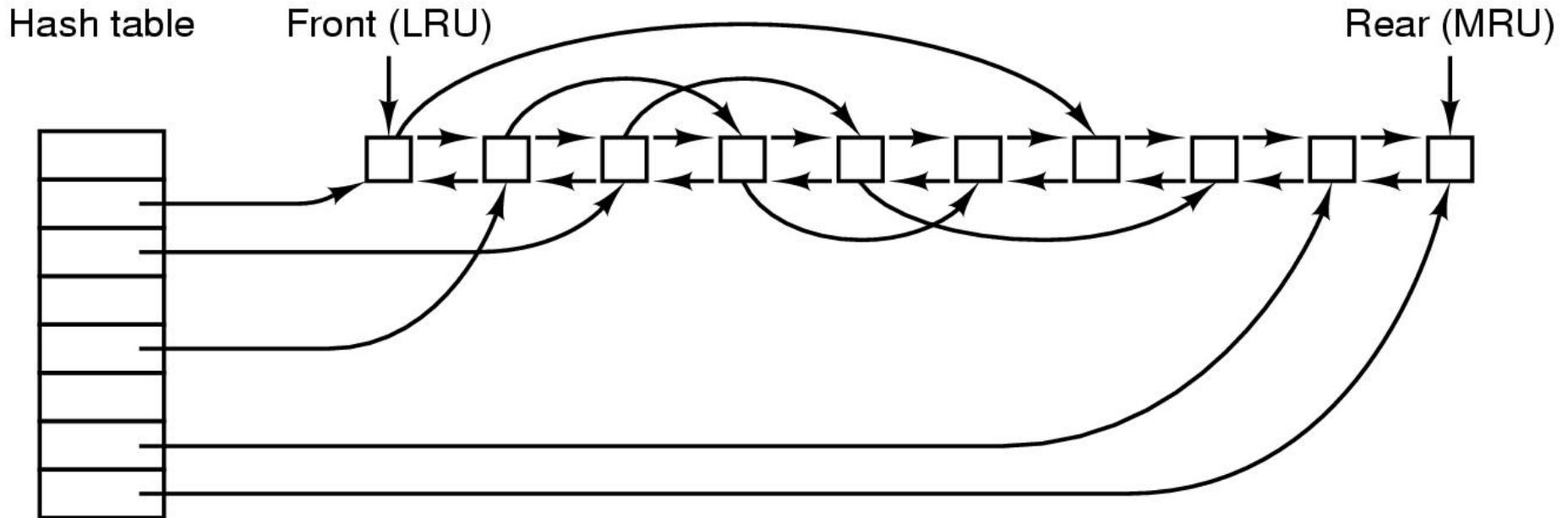
Buffer cache

- Quando si effettua un'operazione su disco i dati contenuti su uno o più blocchi sono portati in memoria
- La buffer cache è un'area di transito della memoria centrale dove questi blocchi restano per un certo periodo al fine di ottimizzare i tempi di trasferimento tra disco e memoria centrale

Buffer cache

- Problema: la buffer cache è volatile, cosa succede in caso di spegnimento di un sistema dei dati in buffer cache?
- Due strategie:
 - Windows: dopo una write riscrivere immediatamente il blocco su disco (write-through)
 - UNIX: riscrive i blocchi di buffer cache modificati ogni 30 secondi

Gestione buffer cache



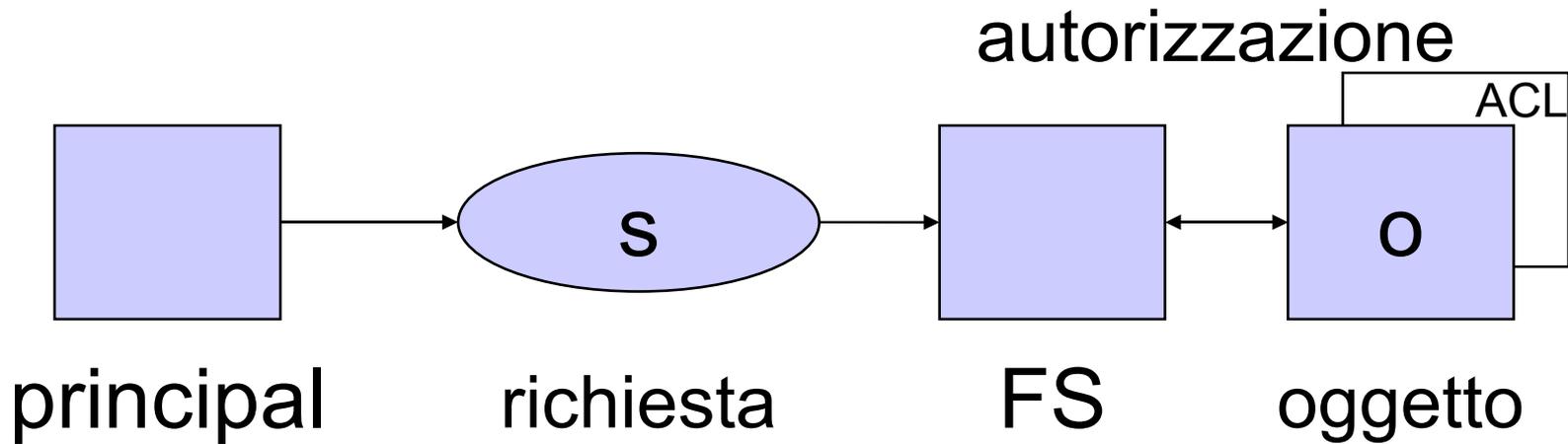
- La buffer cache è gestita tramite una tabella hash con liste linkate per la gestione delle collisioni
- A loro volta tutti i blocchi di buffer cache sono gestiti anche attraverso una lista doppia, ordinata secondo un criterio prossimo a LRU

Controllo degli accessi

Autorizzazione

- Un'entità attiva, chiamata 'principal' o 'soggetto', richiede accesso ad un entità passiva, chiamata oggetto
- Autorizzazione: il FS decide se autorizzare o meno l'accesso
- Il FS basa la sua decisione sulla base delle politiche di sicurezza definite dal proprietario dell'oggetto

Autenticazione & Controllo degli accessi



Terminologia di riferimento

- Soggetti/Principal: entità attive
 - utenti, processi, ...
- Oggetti: entità passive
 - file o risorse
- Operazioni di Accesso: accesso a file,

Quali Approcci

- È possibile realizzare un sistema di controllo degli accessi
 - Definendo le operazioni che ogni principal può svolgere sui diversi oggetti
 - Definendo le operazioni che possono essere svolte su ciascun oggetto?
- Tipicamente il secondo approccio è seguito nell'ambito dei sistemi operativi
- Il primo nell'ambito dei DBMS o dello sviluppo di applicativi

Unix

- File
 - read: da un file
 - write: su un file
 - execute: un file
- directory
 - read: lista dei file
 - write: crea o rinomina file della directory
 - execute: cerca file nella directory

Matrice degli accessi

- Consente di specificare per ogni coppia soggetto-oggetto le operazioni di accesso consentite
- S ... insieme dei soggetti
- O ... insieme degli oggetti
- A ... insieme delle operazioni di accesso
- Matrice: $M = (M_{so})_{s \in S, o \in O}$
- Ogni elemento della specifica le operazioni che il soggetto s può eseguire sull'oggetto o

Matrice degli Accessi

	O1	O2	O3	S1	S2	S3
S1	Own		Execute		Write	
S2		Read Write				
S3	Append			Own		Execute
S4		Read		Read execute		

Matrice degli Accessi

- La matrice degli accessi è:
 - Un concetto astratto,
 - Un utile strumento teorico
 - Non adatta per essere direttamente implementata
 - Poco adatta per la gestione della sicurezza

Capability

- Focus sul soggetto:
 - Diritti di accesso memorizzati con i soggetti
 - capability \equiv righe della matrice degli accessi
- Soggetti possono assegnare diritti ad altri soggetti
- Soggetti possono assegnare il diritto di assegnare diritti

Problem

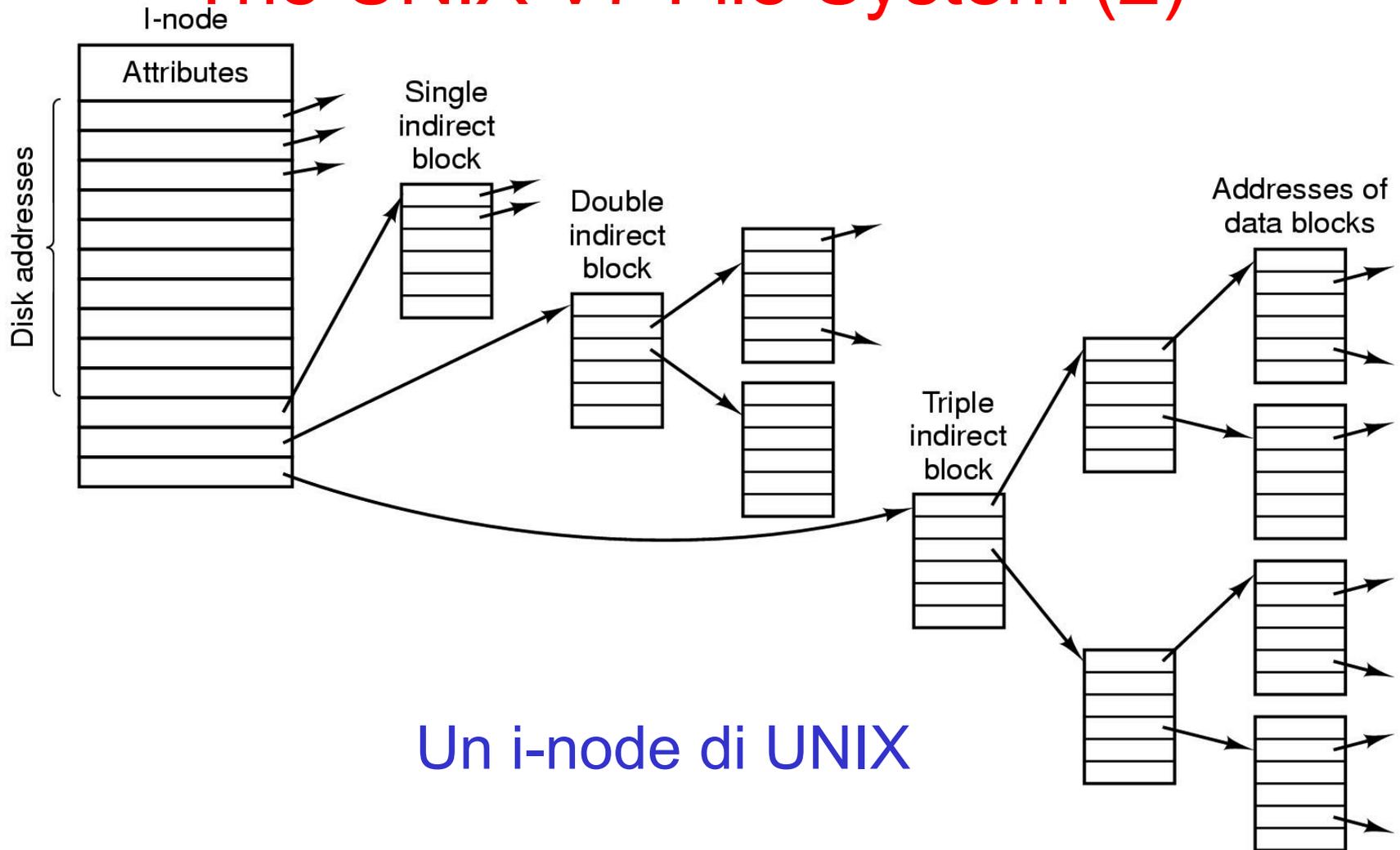
- Come verifico chi può accedere ad uno specifico oggetto?
- Come revoco un diritto di accesso ad un oggetto?

Access Control List (ACL)

- Focus sugli oggetti:
 - I diritti di accesso sono memorizzati insieme ad un oggetto
 - ACLs \equiv colonne delle matrice degli accessi
- Implementate nella maggior parte dei SO commerciali, anche se in versioni semplificate
- Problema:
- Come identifichiamo i diritti di accesso di un soggetto?
- Nella definizione di politiche di sicurezza è più naturale fare riferimento ai soggetti

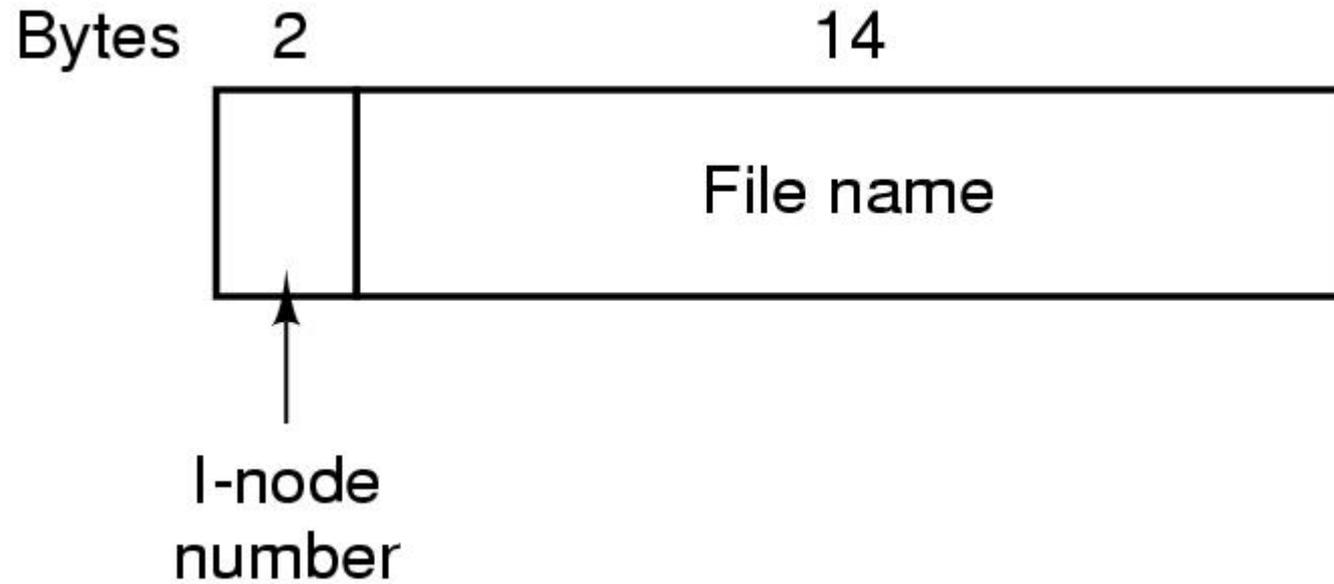
FS di UNIX

The UNIX V7 File System (2)



Un i-node di UNIX

Entry di Directory



Lettura dei dati da file

- Per accedere ai dati di un file è necessario trasferire i dati presenti nell'i-node in memoria
- Le soluzioni a questo punto possono essere due:
 - in una tabella di sistema
 - Facilita la condivisione delle strutture dati
 - In una tabella privata del processo
 - Svincola le attività di un processo su un file da quelle che possono essere svolte da un altro file
 - Spreco memoria

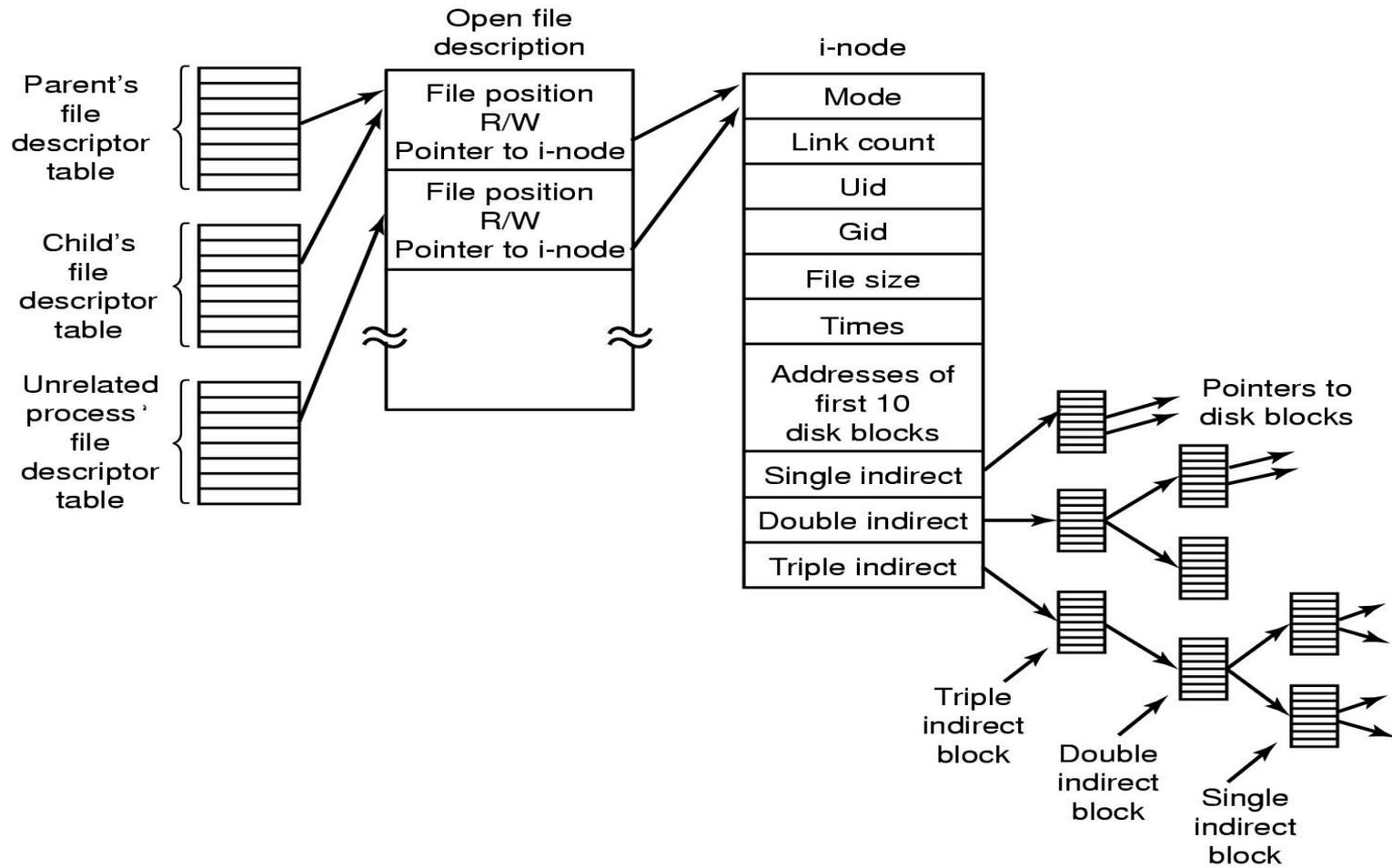
Letture dati

- Considero il I caso devo risolvere il problema di due file che accedono al file in posizioni diverse (dove inserisco il puntatore al file?)
- Posso inserirlo in uno spazio privato riservato ad ogni processo
- Mantengo così una struttura unica con i dati degli i-node
- Rendo più processi che vogliono operare sullo stesso file autonomi, ma ciascuno “vede” le modifiche apportate dall’altro

Lettura dati

- Sorge però un altro problema:
 - Come posso fare in modo che due processi padre e figlio possano condividere in fase di esecuzione lo stesso puntatore a file e cioè ciascuno veda le modifiche che l'altro apporta a questo campo?

UNIX File System (3)



Linux

- Linux supporto un proprio file sistem noto come Ext2
- Ext2 è basato sui principi generali del file system di Unix
- Adotta la bitmap come sistema per memorizzare i blocchi liberi e gli i-node liberi
- Per l'indirizzamento uso lo schema 12+3 invece di 10+3
- Il disco viene diviso in gruppi di blocchi per tenere fisicamente vicini gli i-node con i blocchi dati

The Linux File System

