

## Implementazione del File System

- Struttura del File System
- Implementazioni delle directory
- Implementazione dei file
  - Allocazione contigua
  - Allocazione concatenata
  - Allocazione basata su indice
  - File System Implementation
- Gestione dei blocchi liberi

Basato parzialmente sulle slide originali di Silberschatz, Galvin and Gagne

Implementazione del File System

1

---

---

---

---

---

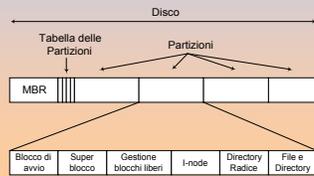
---

---

---

## Struttura del File System

- Il file system è memorizzato su disco
- Il disco può essere suddiviso in partizioni ognuna contenente un proprio file system
- Il partizionamento del disco avviene mediante la formattazione di alto livello



Implementazione del File System

2

---

---

---

---

---

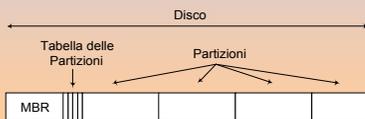
---

---

---

## Struttura del File System

- MBR (Master Boot Record)
  - Contiene programma di avvio
  - La fine del MBR contiene la tabella delle partizioni
- Tabella delle partizioni
  - Contiene punto di inizio e fine di ogni partizione
  - Una sola partizione è marcata come attiva
  - E' la partizione da cui verrà caricato il SO



Implementazione del File System

3

---

---

---

---

---

---

---

---

## Struttura del File System

- Blocco di avvio
  - Contiene un programma che viene eseguito in fase di bootstrap e serve a caricare il SO
  - Ogni partizione contiene il Blocco di Avvio anche se non contiene il SO (potrebbe contenerne uno)
- Superblocco
  - Contiene informazioni sul file system
    - Numero magico che identifica il FS
    - Numero di blocchi del FS
    - ...
- Gestione per lo spazio libero
  - Strutture dati per la gestione dei blocchi liberi
- I-node
  - Nei SO che utilizzano gli i-node questi sono raggruppati in una parte del disco
- Directory radice
- File e directory

Implementazione del File System 4

---

---

---

---

---

---

---

---

---

---

## Bootstrap

- Viene eseguito il programma di avvio contenuto nel **MBR**
  - Localizza la partizione attiva dalla tabella delle partizioni
  - Legge il primo blocco (blocco di avvio) e lo esegue
- Viene eseguito il programma del **Blocco di Avvio**
  - Localizza il SO nella partizione attiva
  - Carica in memoria il SO
  - Cede il controllo al SO

Implementazione del File System 5

---

---

---

---

---

---

---

---

---

---

## Struttura delle directory

pippo.c	attributi
esempio.c	attributi
temp.c	attributi
esercizio.c	attributi

pippo.c		↑	attributi
esempio.c		↑	attributi
temp.c		↑	attributi
esercizio.c		↑	attributi

Attributi memorizzati nell'elemento di directory (Windows)

Attributi memorizzati negli i-node (UNIX)

Implementazione del File System 6

---

---

---

---

---

---

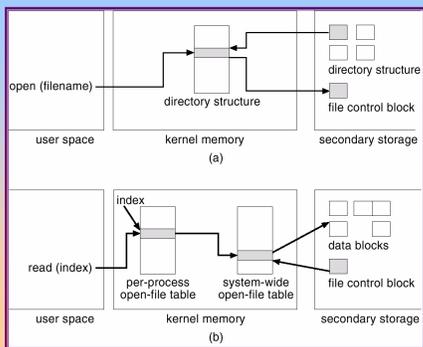
---

---

---

---

## Utilizzo dei file



Implementazione del File System

7

---

---

---

---

---

---

---

---

## Implementazione delle directory

### Lista lineare

- Gli elementi della directory sono collegati a lista
- Creazione di un file
  - Scansione della lista per vedere se il nome è già in uso
- Cancellazione
  - Ricerca dell'elemento corrispondente al file da cancellare e rilascio dello spazio allocato
    - Eliminazione dell'elemento di directory
    - Impostazione di un bit a 0 (elemento non valido)
    - Inserimento in una lista di elementi liberi
    - Copiatura dell'ultimo elemento in quello liberato
- Ricerca
  - Scansione lineare della lista
  - Lista ordinata
    - Complica la fase di creazione e cancellazione dei file

Implementazione del File System

8

---

---

---

---

---

---

---

---

## Implementazione delle directory

### Tabella hash

- Tabella + Funzione hash
- Funzione hash
  - $h(\text{nomefile}) = \text{puntatore all'elemento del file}$
  - Possibilità di collisioni
  - La funzione hash dipende dalle dimensioni della tabella
  - Cambiare dimensione comporta problemi
- Gli elementi della tabella sono generalmente liste lineari di elementi di directory
  - Si risolve il problema delle collisioni
  - Si svincola la funzione hash dalle dimensioni della tabella
  - Le ricerche sono più lente (scorrimento della lista)

Implementazione del File System

9

---

---

---

---

---

---

---

---

## Implementazione dei file

- Un file viene implementato mediante un certo numero di blocchi di disco in cui sono memorizzati i dati
- Metodi di allocazione
  - Allocazione contigua
  - Allocazione concatenata
  - Allocazione indicizzata

Implementazione del File System

10

---

---

---

---

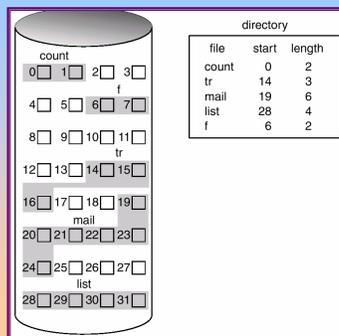
---

---

---

---

## Allocazione contigua



Implementazione del File System

11

---

---

---

---

---

---

---

---

## Allocazione contigua

- Ogni file occupa un insieme contiguo di blocchi
- Metodo di allocazione semplice
  - È necessario conoscere solo l'indirizzo del primo blocco e il numero di blocchi
- Consente facilmente l'accesso casuale
- Il file non può crescere
- Frammentazione
  - Interna
  - esterna

Implementazione del File System

12

---

---

---

---

---

---

---

---

## Allocazione basata su Extent

- Alcuni FS usano un modello modificato di allocazione contigua
- I blocchi vengono allocati sotto forma di **extent**
  - ↳ Un **extent** è un insieme di blocchi contiguo
- Un file è costituito da uno o più extent collegati mediante lista concatenata

Implementazione del File System

13

---

---

---

---

---

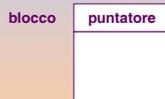
---

---

---

## Allocazione concatenata

- Ogni file è costituito da un insieme di blocchi collegati mediante lista
- I blocchi si possono trovare in qualsiasi posto del disco



Implementazione del File System

14

---

---

---

---

---

---

---

---

## Allocazione concatenata

- Gestione Semplice
  - ↳ è necessario conoscere l'indirizzo di partenza
- Gestione efficiente dello spazio disco
  - ↳ Si elimina il problema della frammentazione esterna
  - ↳ Rimane il problema della frammentazione interna (ineliminabile)
- Overhead dovuto ai puntatori
- Accesso casuale impossibile
  - ↳ Simulato attraverso accesso sequenziale

Implementazione del File System

15

---

---

---

---

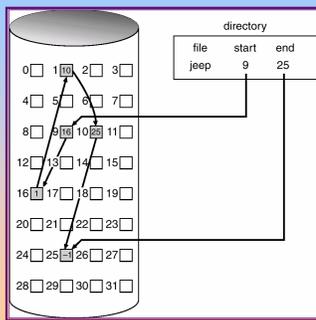
---

---

---

---

## Allocazione concatenata



Implementazione del File System

16

---

---

---

---

---

---

---

---

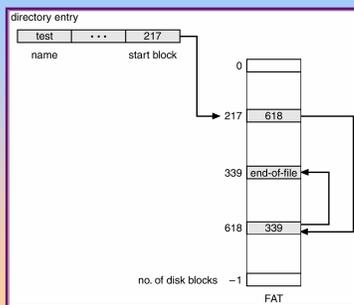
---

---

---

---

## File-Allocation Table (FAT)



Implementazione del File System

17

---

---

---

---

---

---

---

---

---

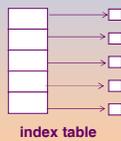
---

---

---

## Allocazione indicizzata

- Tutti i puntatori vengono raggruppati all'interno di un unico blocco (blocco indice)
- Il blocco indice contiene i puntatori ai blocchi di dati



Implementazione del File System

18

---

---

---

---

---

---

---

---

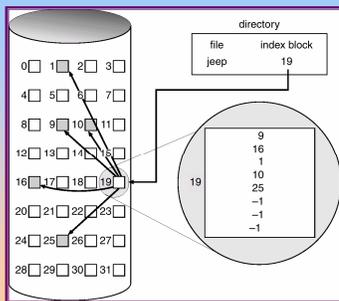
---

---

---

---

## Allocazione indicizzata



Implementazione del File System

19

---

---

---

---

---

---

---

---

## Allocazione indicizzata

- Necessita di un blocco particolare (blocco indice)
- Permette l'accesso casuale
- Elimina la frammentazione esterna ma introduce l'overhead relativo al blocco indice
  - Un file (anche breve) è costituito da almeno 2 blocchi

Implementazione del File System

20

---

---

---

---

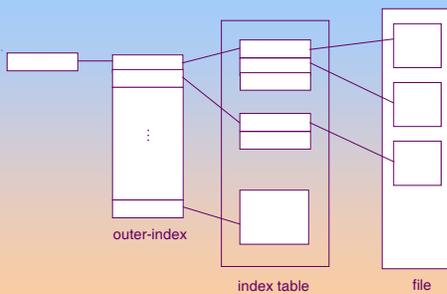
---

---

---

---

## Allocazione indicizzata



Implementazione del File System

21

---

---

---

---

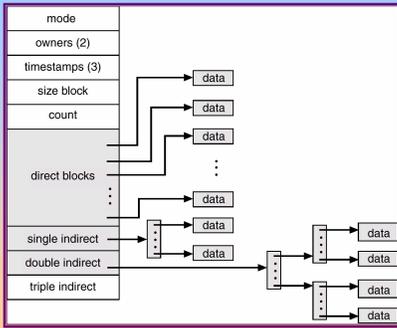
---

---

---

---

### Combined Scheme: UNIX (4K bytes per block)



Implementazione del File System

22

---

---

---

---

---

---

---

---

---

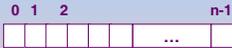
---

---

---

### Gestione dei blocchi liberi

- Vettore di bit ( $n$  blocchi)



bit[ $i$ ] = 0  $\Rightarrow$  block[ $i$ ] libero  
 bit[ $i$ ] = 1  $\Rightarrow$  block[ $i$ ] occupato

Implementazione del File System

23

---

---

---

---

---

---

---

---

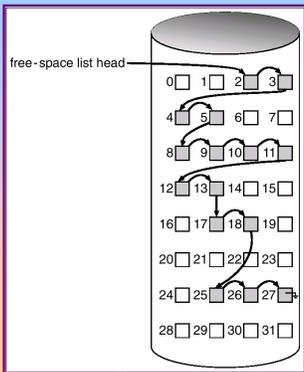
---

---

---

---

### Gestione dei blocchi liberi



Implementazione del File System

24

---

---

---

---

---

---

---

---

---

---

---

---