

File System

- Tipologie differenti di file per scegliere la formattazione giusta

MAGIC NUMBER: annotazioni all'inizio del file per identificare il modo in cui deve essere aperto

Esempi di magic number:

%PDF	CAFEBAE	(.gif)	oppure un INODE
------	---------	--------	-----------------

All'interno del file è presente un FILE CONTROL BLOCK (FCB) che contiene i METADATI

FILE
[ALFANUMERICI codifiche che corrispondono a caratteri
BINARI file offset, executable ecc...]

FILE SYSTEM LOGICO

FILE (operazioni consentite) (operazioni dell'os)

- ① CREAZIONE → SPAZIO DISCO, DESCRITTORI, METADATI, AGGIORNAMENTO FILE SYSTEM
- ② LETTURA → ID POSIZIONE, ACCESSO, AGGIORNAMENTO METADATI
- ③ SCRITTURA → ID POSIZIONE, ACCESSO, AGGIORNAMENTO METADATI
- ④ CANCELLAZIONE → LIBERO MEMORIA, AGGIORNAMENTO FILE SYSTEM
- ⑤ TRONCAMENTO → ... (trascina una parte del file)
- ⑥ RIPOSIZIONAMENTO → ...
- ⑦ APERTURA → (cerca il file e ottiene il puntatore)
- ⑧ CHIUSURA → (rilascia risorse)

ACCESSO AI FILE → sequenziale
→ diretto
→ a indice



MECCANISMI DI PROTEZIONE

- DIRITTI DI ACCESSO

u g o a
rwx rwx rwx rwx

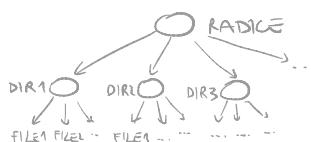
- LOCK
 - consigliati
 - obbligatori

DIRECTORIES

- Donna struttura al file system
- Il file system nasce come **punto**, senza directories. Problemi di sicurezza in computer con più utenti. I nomi di tutti i file del sistema devono essere univoci

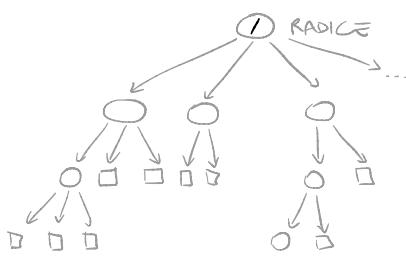


- File system → due livelli



che può portare poche possibilità di articolare il proprio spazio di lavoro e due utenti non possono più lavorare agli stessi file

- File system gerarchico



I nodi interni sono directories, la frontiera è formata da file → directories vuote

Il nome univoco di ogni file è il percorso assoluto che lo collega alla radice

- File system → grafo

simbolici → finici

Aggiunge i LINK che sono collegamenti → file

APPROCCIO DEL CONTEGGIO

- Se cancello un file linkato da qualche parte decremento il conteggio del link. Il file viene rimosso solo quando #link va a 0
- In alternativa si chiede all'utente, se cancella un link simbolico cancella solo il collegamento, se cancella il file dal percorso originale il link rimane DANGLING, collegato → niente

In caso di link finisce qualunque eliminazione rimuove direttamente il file

MOUNT - all'uvio

FILE SYSTEM DISTRIBUITO

- Si trova in macchine collegate in rete
- CLIENT - SERVER

IMPLEMENTAZIONI - disco

PAGINE - FRAME - BLOCCHI

Da blocchi l'accesso è diretto e può essere effettuato solo attraverso il file system.

Il primo blocco del disco è il **BOOT CONTROL BLOCK** (vuoto se il file system non contiene OS) usato come **BOOTSTRAP**

Il secondo è il **VOLUME CONTROL BLOCK** che contiene informazioni sul file system

Successivamente ci sono TOT blocchi che forniscono informazioni sulla struttura delle **directories**, seguiti dai **blocchi di dati**



- Quando i FILE vengono copiati in RAM gli INODE vengono associati (**INORE INODE**) con:
 - stato dell'INODE (LOCKED/ PUNTO DI MOUNT/...)
 - DEVICE NUMBER (identifica il file system)
 - INODE NUMBER
 - CONTATORE DEI RIFERIMENTI

INODE: Numero rappresentativo del file per il SO

ALGORITMO NAMEI PER CALCOLARE INODE

```

#INODE NAMEI(String PATH)
if (first_directory == '/')
    current = root_inode
else
    current = inode_working_directory

repeat
    el = next_element_input
    if (el == NULL) // punta all'ultimo elemento
        return current
    else if (el <= current)
        current = inode_associato_is_el
    else
        error
    while (el == NULL)

```

Ogni INODE è salvato in una posizione $\langle B, D \rangle$ del disco

B: ID blocco

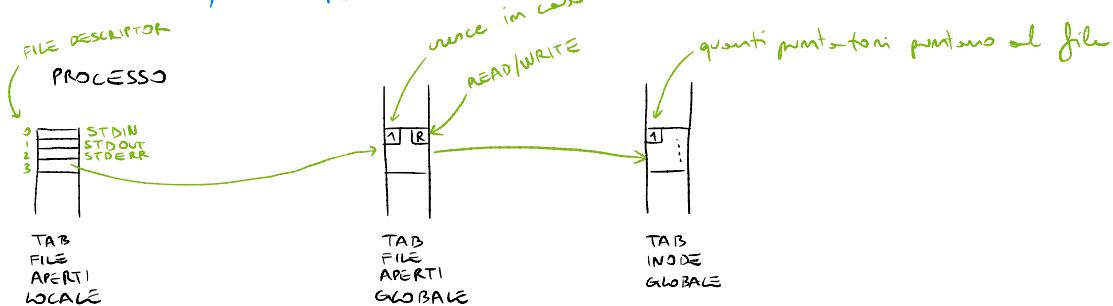
D: ID displacement

$$B = (\text{int}) \frac{\# \text{INODE}}{\# \text{INODE} \times \text{BLOCCO}}$$

mo di INODE contenuti in ogni blocco

$$D = (\text{int}) \# \text{INODE \%} \# \text{INODE} \times \text{BLOCCO}$$

APERTURA / CHIUSURA FILE



IMPLEMENTAZIONE DIRECTORIES

- SEQUENZIALE
- B-TREE di ordine n: ogni nodo contiene tra N e 2N chiavi, se contiene K chiavi sono K+1 successori
- B-TREE di ordine 25 con 10 000 000 di nodi permette di identificare chiavi con solo 4 salti, più complesso ma molto più efficiente

ALLOCAZIONE SPAZIO DISCO AI FILE

- 1 allocazione contigua
- 2 allocazione concatenata (FAT)
- 3 allocazione indirizzata
- 4 allocazione ibrida

① Blochi dati contigui

- + accesso sequenziale e diretto facilitati
- spreco memoria se riserva troppi blocchi per un file piccolo
- bisogna gestire il caso in cui il file cresce

② Possibilità di aggiungere blocchi in maniera dinamica



I blocchi non devono per forza essere concatenati

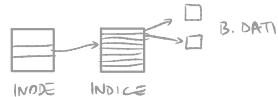
- + favorisce l'accesso sequenziale
- non favorisce l'accesso diretto
- blocchi dati nominati causano il truncamento dei file

FAT : File Allocation Table

Organizza i riferimenti ai blocchi in una tabella.

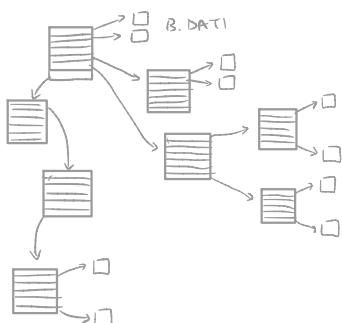
- i riferimenti sono rinvii dai singoli blocchi
- Contiene informazioni sui blocchi liberi

③ Blocco indice



La dimensione del file è limitata → blocco indice x blocco
Combinando i blocchi indice si risolve questo problema

④ Soluzione ibrida



GESTIONE BLOCCHI LIBERI

- Anzio si associa a ciascun blocco un bit, la sequenza indica se un blocco è occupato (1) o libero (0)