Gestione dei file in UNIX

Index

Introduction

In Unix ci sono sei tipi di file:

- normale
- directory
- speciale (mappano su nomi di file i dispositivi di I/O)
- named pipe (per far comunicare i processi tra loro)
- hard link (collegamenti, nome di file alternativo)
- link simbolico (il suo contenuto è il nome del file cui si riferisce)

Inode

Inode sta per *”index node”* (ispirato al metodo di allocazione indicizzato, con dimensione fissa dei blocchi) e rappresentano i metadati di un file
Questa struttura dati contiene le informazioni essenziali per un dato file (che rientra in quelli sopra elencati, anche la directory) e un numero che permette di accedere più facilmente ad un inode (inode number, quando si cancella un file l’inode number può essere riutilizzato).

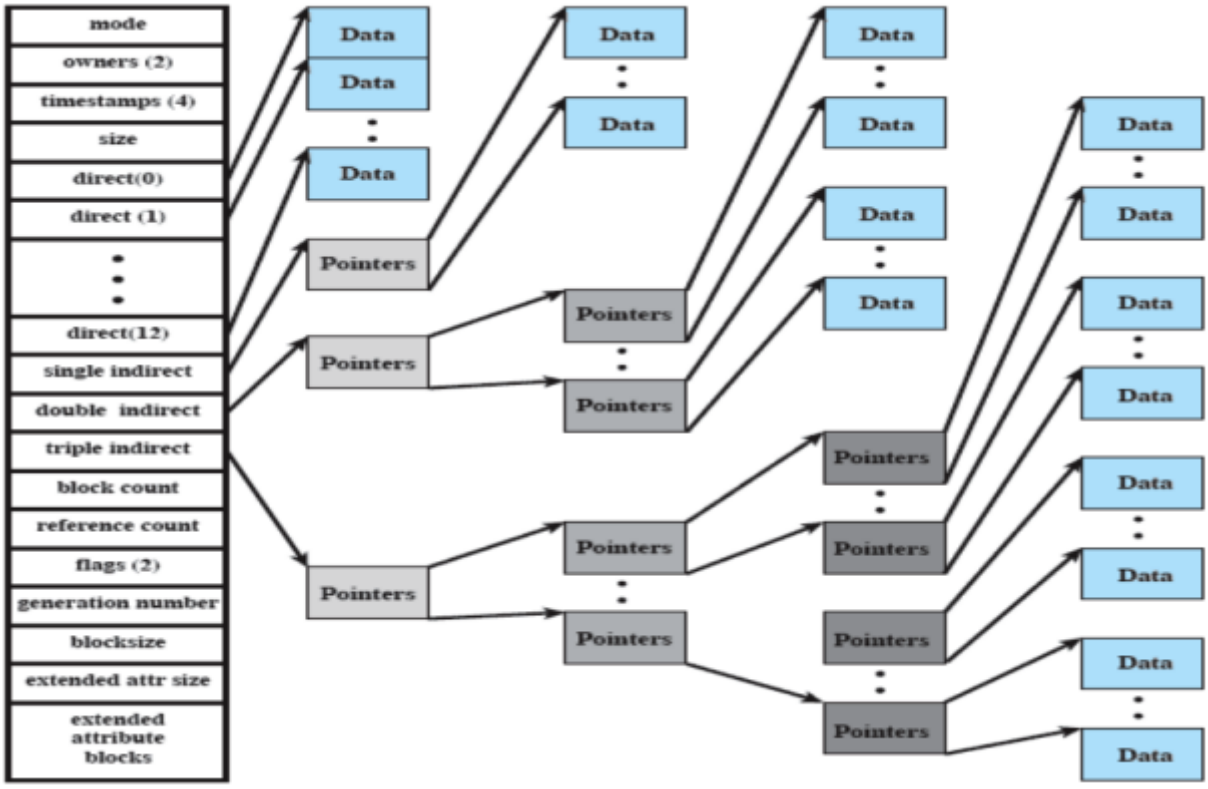
> **Info**
>
> Un dato inode potrebbe essere associato a più nomi di file (anche se di solito un inode, un file) tramite un hard link

Tutti gli inode si trovano in una zona di disco dedicata (*i-list*), viene però mantenuta dal SO una tabella di tutti gli inode corrispondenti a file aperti in memoria principale

Inode in Free BSD

All’interno di un inode sono contenuti:

- Tipo e modo di accesso del file
- Identificatore dell’utente proprietario e del gruppo cui tale utente appartiene
- Tempo di creazione e di ultimo accesso (lettura o scrittura)
- Flag utente e flag per il kernel
- Numero sequenziale di generazione del file
- Dimensione delle informazioni aggiuntive
- Altri attributi (controllo di accesso e altro)
- Dimensione
- Numero di blocchi, o numero di file (per le directory)
- Dimensione dei blocchi
- Sequenze di puntatori a blocchi



Per file piccoli (di grandezza massima di $13 \cdot \text{dimensione di un blocco}$), i dati sono puntati direttamente da `direct`
Infatti i blocchi `direct` puntano ai blocchi stessi contenenti i dati, mentre i puntatori indiretti puntano ai cosiddetti **"blocchi di indirizzamento"** ovvero quei blocchi in cui sono contenuti gli indirizzi dei blocchi in cui si trovano i dati veri e propri. Questi vengono utilizzati sono nel momento in cui i puntatori diretti non sono sufficienti

---

Allocazione di file

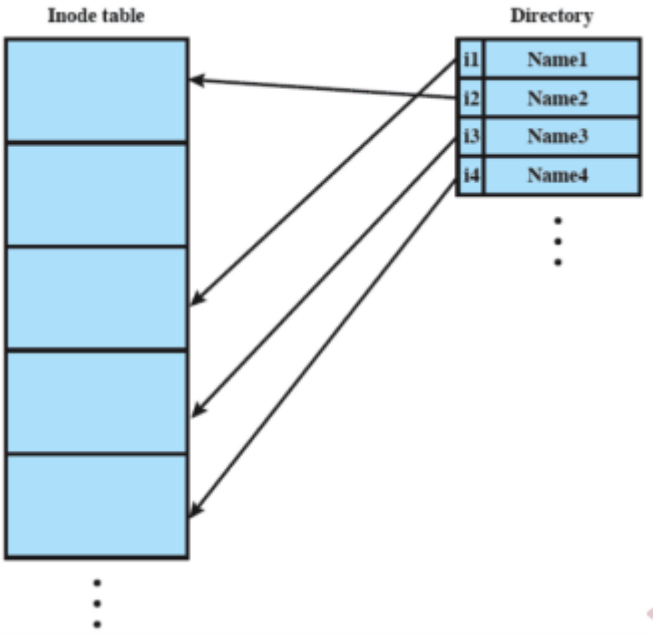
L'allocazione di file è dinamica e viene fatta a blocchi (quindi potenzialmente non contigui). Attraverso l'indicizzazione si tiene traccia dei blocchi dei file (parte dell'indice è puntata dall'inode) all'interno dei puntatori indiretti, mentre si punta ai dati veri e propri attraverso i `direct` (per file piccoli)

Level	Number of Blocks	Number of Bytes
Direct	12	48K
Single Indirect	512	2M
Double Indirect	$512 \times 512 = 256K$	1G
Triple Indirect	$512 \times 256K = 128M$	512G

Inode e directory

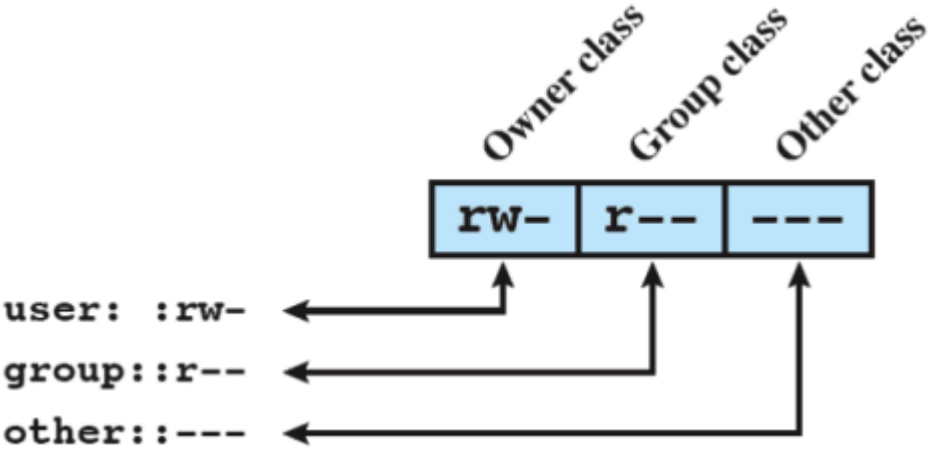
Anche le directory sono file. Infatti al loro interno è contenuta una lista di coppie (nome file, puntatore ad inode), ma alcuni di questi file possono essere a loro volta delle directory, formando così una struttura gerarchica

Una directory può essere letta da ogni utente ma modificata solo dal sistema operativo



Accesso ai file

Per ogni file, ci sono 3 terne di permessi: lettura, scrittura ed esecuzione. Di queste terne ce ne sta una per il proprietario, una per il suo gruppo (insieme di utenti) e una per tutti gli altri



Gestione file condivisi

Come si fa a gestire i file che devono essere condivisi in più directory?
Fare una copia del file è inutilmente costoso quindi si sono strutturate due alternative: **symbolic links** e **hard link**

I symbolic links contengono solamente il path completo sul file system verso il file (esiste un solo inode del file originale), ma comporta che possano esistere symlink a file non più esistenti
Gli hard link sono dei puntatori diretti al descrittori del file originale (all'inode, questo contiene inoltre un contatore dei file che lo referenziano) e il file condiviso non può essere cancellato finché esiste un link remoto di accesso

---

Regioni del volume

ⓘ **L'immagine non è totalmente rappresentativa, manca i-node list**

Regione boot block

Risulta essere simile al blocco di boot per FAT, contiene infatti le informazioni e dati necessari per il bootstrap (caricamento del sistema operativo)

Regione superblock

Contiene le informazioni sui metadati del filesystem (dimensione partizione, dimensione blocchi, puntatore alla lista dei blocchi liberi, ecc.).
All'interno del disco sono inoltre presenti svariate copie del superblock in modo tale che, in caso di corruzione del principale, sia possibile comunque recuperare il filesystem e ripristinare le informazioni cruciali (la prima copia è in una parte prefissata della partizione, in modo tale da consentire recovery in modo semplice)

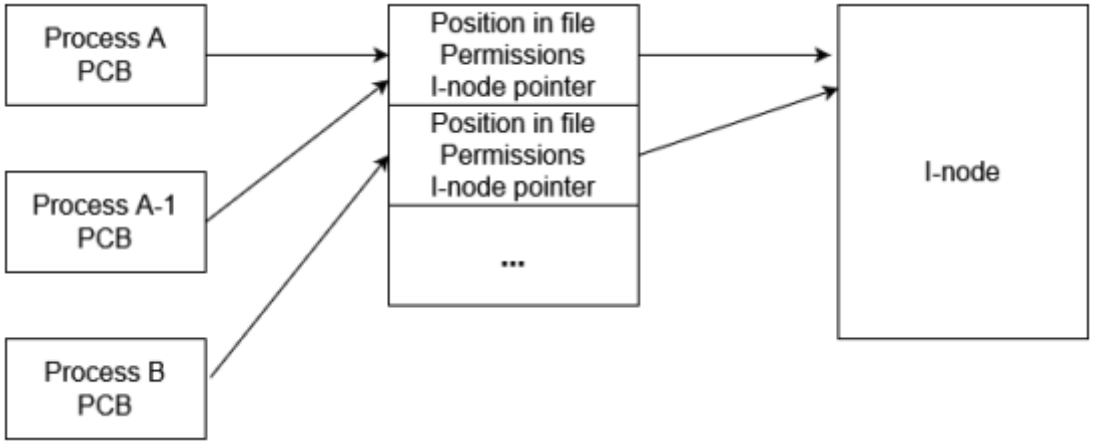
Regione lista degli i-node (i-list)

E' una tabella numerata di descrittori di file (i-node) contenente un i-node per ogni file salvato nel sistema. Ciascun i-node nella lista punta ai blocchi dei file nella sezione dati del volume (data blocks)

Il kernel e gli i-node

Il kernel Unix usa due strutture di controllo separate per gestire file aperti e descrittori i-node:

- puntatori a descrittori dei file attualmente in uso, salvato nel processo control block
- una tabella globale di descrittori di file aperti, mantenuto in un'apposita struttura dati



E Linux?

Linux nativamente supporta **ext2**/**ext3**/**ext4**:

- ext2 → direttamente dai file system Unix originari
- ext3 → ext2 con journaling
- ext4 → ext3 in grado di memorizzare singoli file più grandi di $2\mathrm{TB}$ e filesystem più grandi di $16\mathrm{TB}$

Inoltre supporta pienamente per gli i-node i quali vengono memorizzati nella parte iniziale del file-system
Linux permette anche di leggere e scrivere altri file system, come ad esempio quelli di Windows, ma con FAT non è possibile memorizzare gli i-node sul dispositivo, per questo motivo vengono creati on-the-fly su una cache quando vengono aperti i file