



2. Organizzazione fisica dei file

Con riferimento all'organizzazione fisica dei file su memoria di massa, in Linux il termine **filesystem** viene usato per indicare ogni partizione del disco: ogni filesystem viene gestito in modo separato per mezzo di un **super-block** (super-blocco) e di una serie di **i-node** (nodi indice).

Il super-blocco descrive lo stato complessivo del filesystem, mentre ogni i-node è associato a un file, e contiene informazioni quali per esempio la tabella dei blocchi di disco associati a quel file.

L'area che contiene gli i-node si chiama **i-list** (lista degli indici).

Il danneggiamento del super-block può causare l'inaccessibilità a gran parte del filesystem (che coincide con la partizione di disco).

Il super-blocco è localizzato a una distanza fissa dall'inizio della partizione del disco che contiene il filesystem.

Per evitare la perdita di dati che può accadere quando vengano rovinati i blocchi della partizione che contengono il super-blocco, questo viene duplicato in altre posizioni all'interno della partizione, e queste copie possono essere lette e utilizzate dal sistema operativo, per ricostruire il filesystem, se viene rilevato un errore nella prima copia.

Il super-blocco di un filesystem viene registrato nel momento della creazione del filesystem stesso, e non viene più modificato successivamente; esso contiene i parametri che descrivono il filesystem, che sono:

- dimensione del filesystem
- dimensione dell'area riservata per i-list
- numero massimo di blocchi di dati
- numero massimo di file che esso può contenere
- dimensione del blocco di dati
- lista dei blocchi liberi
- numero degli i-node.

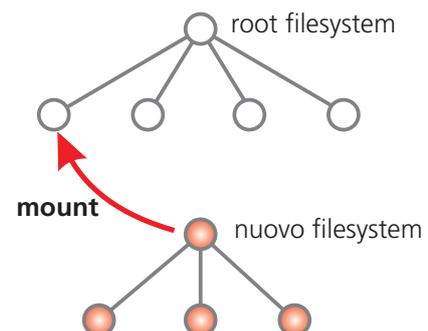
La creazione di un filesystem è realizzata dall'Amministratore del sistema con il comando **mkfs**, specificando

- la partizione dove viene creato il filesystem
- il numero di blocchi fisici assegnati al filesystem
- il numero di i-node nella i-list.

Ogni filesystem, per poter essere utilizzato, deve essere prima sottoposto all'operazione di **mounting** (montaggio), attraverso il comando **mount**.

Esso genera un insieme di informazioni in memoria centrale, per consentire alle procedure del sistema operativo di accedere ai file dello specifico filesystem. L'operazione a rovescio viene realizzata dal comando **umount**.

In pratica, alla partenza del sistema, il sistema operativo utilizza il filesystem fondamentale (*root filesystem*) che contiene tutte le routine che consentono l'avvio e la gestione del sistema. Ogni operazione di *mount* estende la struttura ad albero del filesystem radice, collegando la radice del nuovo filesystem ad un nodo del *root filesystem*.





Il punto dell'albero dove il nuovo filesystem è collegato si chiama **mount point**.

Questa è la modalità attraverso la quale il sistema operativo Linux consente all'utente di utilizzare una partizione del disco, un dischetto, un CD, un nastro o un'unità di memoria stick USB.

Il comando *mount*, eseguito senza parametri, visualizza i filesystem già montati, indicando per ciascuno la directory alla quale è stato collegato.

Un file in generale consiste in uno o più blocchi di dati contenenti qualsiasi tipo di dati (una directory è un particolare tipo di file).

A ogni file è associato un i-node; inoltre gli i-node vengono identificati tramite un numero chiamato **i-number**; con un i-number è possibile identificare univocamente un file all'interno del filesystem.

Allo stesso i-number possono poi essere associati diversi nomi di file, attraverso il meccanismo di **link**, detto anche *hard link* e attivato dall'utente tramite il comando **ln**: in questo modo più nomi di file corrispondono al contenuto di un solo file registrato su disco.

L'*i-node* è formato da 64 byte e contiene informazioni riguardanti:

- tipo di file (normale, speciale, directory)
- permessi, indicati con bit di protezione r (*read*), w (*write*), x (*execute*)
- numero dei link
- nome del proprietario
- nome del gruppo
- dimensione del file in byte
- puntatori ai blocchi del file o ad altri puntatori
- data dell'ultimo accesso
- data dell'ultima modifica
- data di creazione.

Un i-node ha una dimensione limitata e ha lo spazio solo per un piccolo numero di puntatori diretti ai blocchi di dati; i blocchi successivi sono referenziati in modo indiretto.

L'accesso indiretto ai blocchi di dati è realizzato attraverso alcuni puntatori di accesso indiretto in ogni i-node.

Quando la dimensione di un file richiede più blocchi di quelli che possono essere indirizzati dai puntatori diretti, viene allocato un blocco che non fa parte del file, ma viene usato per contenere puntatori diretti ai blocchi successivi del file (**indirizzamento indiretto singolo**).

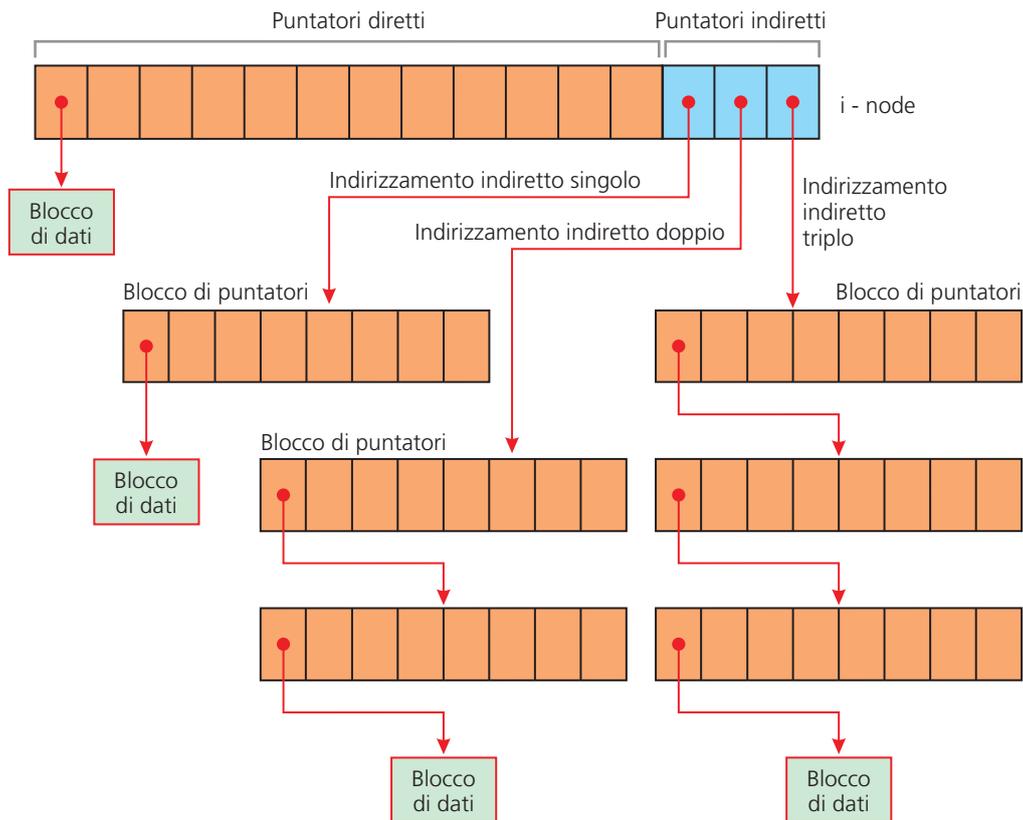
Se il file è particolarmente grande, un blocco pieno di puntatori diretti può non bastare, in questo caso sarà necessario usare uno o due livelli di indirizzamento (**indirizzamento indiretto doppio o triplo**).

Un puntatore indiretto doppio di i-node punta a un blocco che contiene puntatori indiretti singoli, ciascuno dei quali punta ai blocchi contenenti puntatori diretti.

Così il sistema operativo può seguire una catena fino a 4 di questi puntatori per accedere ai blocchi di un file molto grande.

Per esempio, si supponga che ogni i-node contenga 15 puntatori, 12 per puntatori diretti ai blocchi del file e 3 per gli indirizzamenti indiretti, ogni blocco fisico sia formato da 4 kbyte (4096 byte) e che ogni puntatore sia rappresentato con 4 byte. Quindi ogni blocco di puntatori indiretti contiene 1024 byte.

Con il solo indirizzamento diretto la dimensione massima di un file è di 49.152 byte.



Se il file è grande, il 13.o puntatore punta a un blocco di 1024 puntatori diretti (*indirizzamento indiretto singolo*): la dimensione massima diventa $12 \times 4096 + 1024 \times 4096 = 4.243.456$ byte (4 MB circa).

Se il file ha dimensioni ancora maggiori il 14.o puntatore verrà utilizzato per indirizzare un blocco contenente 1024 puntatori a blocchi che contengono a loro volta 1024 puntatori a blocchi di dati (cioè i blocchi vengono indirizzati in modo *indiretto doppio*): la dimensione massima del file diventa $12 \times 4096 + 1024 \times 4096 + 1024 \times 1024 \times 4096 = 4.299.210.752$ byte (4 GB circa).

Se il file richiede uno spazio ancora più grande viene usato il 15.o puntatore e il meccanismo di *indirizzamento indiretto triplo*, cioè quest'ultimo puntatore dell'*i*-node punta a un blocco che contiene puntatori a blocchi di indirizzamento indiretto doppio: lo spazio massimo indirizzabile per un file diventa $12 \times 4096 + 1024 \times 4096 + 1024 \times 1024 \times 4096 + 1024 + 1024 \times 1024 \times 4096 = 4.402.345.721.856$ byte (4 TB circa).