

Il file system di Linux

Con riferimento all'organizzazione fisica dei file su memoria di massa, nei sistemi *Linux* il termine **file system** viene usato per indicare ogni partizione del disco.

Un'installazione tipica di Linux crea due partizioni:

- il file system **root** simboleggiato dal simbolo **/** (barra o *slash*);
- l'**area di swap**, cioè l'area per la memoria virtuale su disco fisso; le sue dimensioni possono variare in base alle dimensioni della RAM (da 1 fino a 2 volte la quantità di RAM fisica disponibile).

Ogni file system viene gestito in modo separato per mezzo di un **super-block** (superblocco) e di un insieme di **i-node** (nodi indice).

Il super-blocco descrive lo stato complessivo del file system, mentre ogni i-node è associato a un file e contiene informazioni come 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-blocco può causare l'inaccessibilità a gran parte del file system (che coincide con la partizione di disco).

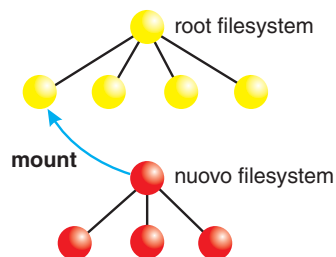
Il super-blocco è localizzato a una distanza fissa dall'inizio della partizione del disco che contiene il file system. Per evitare la perdita di dati, che può accadere quando vengano rovinati i blocchi della partizione che contengono il super-blocco, questo viene duplicato in altre posizioni all'interno della partizione e queste copie possono essere lette e utilizzate dal sistema operativo per ricostruire il file system se viene rilevato un errore nella prima copia.

Il super-blocco di un file system viene registrato nel momento della creazione del file system stesso, e non viene più modificato successivamente.

Esso contiene i parametri che descrivono il file system, che sono:

- dimensione del file system
- 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.

Ogni file system, per poter essere utilizzato, deve essere prima sottoposto all'operazione di **mounting** (montaggio). Essa genera un insieme di informazioni in memoria centrale, per consentire alle procedure del sistema operativo di accedere ai file dello specifico file system. L'operazione a rovescio termina la disponibilità del file system (*unmounting*).



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

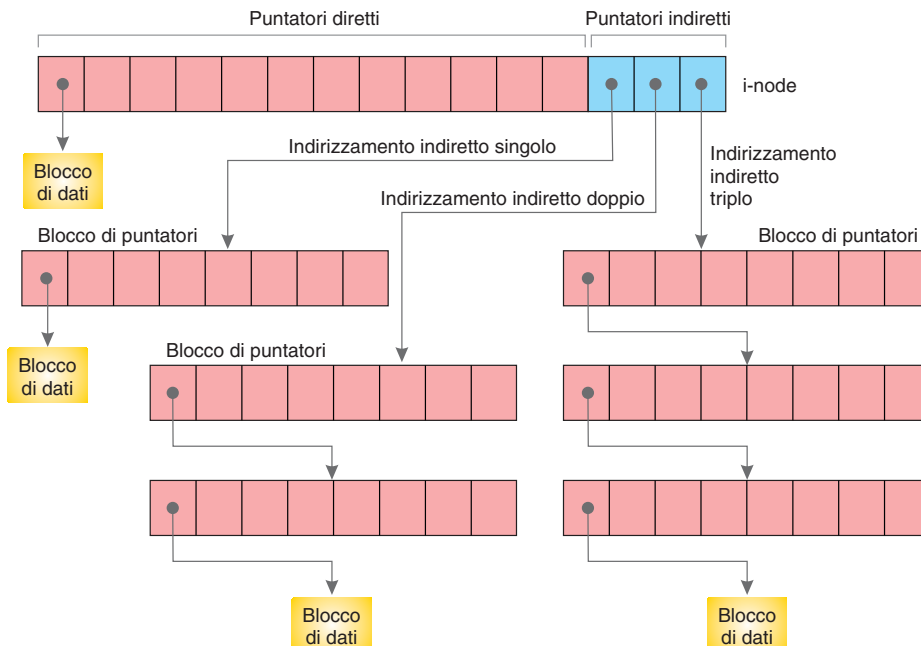
Il punto dell'albero dove il nuovo file system è collegato si chiama **mount point**. Questa è la modalità attraverso la quale il sistema operativo Linux consente all'utente di utilizzare una partizione del disco, un CD, un nastro oppure un disco esterno o una chiavetta sulla porta USB.

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). Ad 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 file system. Allo stesso *i-number* possono poi essere associati diversi nomi di file, attraverso il meccanismo di **link**, detto anche *hard link*: 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.

Poiché 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 puntatori.

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 $12 \times 4096 + 1024 \times 4096 + 1024 \times 1024 \times 4096 + 1024 \times 1024 \times 4096 = 4.402.345.721.856$ byte (4 TB circa).

I tipi di file system più recenti sono dotati di **journaling**, cioè garantiscono in modo più efficiente l'integrità dei dati, in quanto le modifiche ai file e alle directory sono registrate in un diario (*journal*), in modo che, nel caso di chiusura anomala del sistema, come un'interruzione di corrente, i dati possano essere ripristinati.

Il sistema operativo Linux è in grado di accedere a diversi tipi di file system. I più comuni sono:

Nome	Descrizione
ext2	(<i>second extended file system</i>) file system standard per il sistema operativo Linux
ext3, ext4	come il filesystem <i>ext2</i> con l'aggiunta del <i>journaling</i> (questi sono i file system standard di <i>Ubuntu</i>)
fat16, fat32	sono i file system standard per Windows su computer meno recenti
linux-swap	file system per l'area di <i>swap</i>
reiserfs	filesystem dotato di <i>journaling</i> , fa uso di strutture dati particolari per aumentare le prestazioni
ntfs	file system standard per Windows su computer recenti
hfs+	file system standard per computer con Mac OS X.