

# FFS

Fast File System. Usa indici multilivello, con struttura ad albero non bilanciato. Ad ogni file corrisponde un *inode*, che contiene:

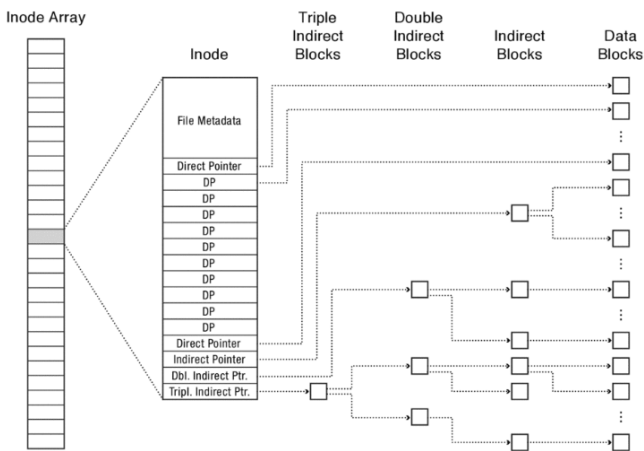
- metadati: proprietario, permessi (u/g/o r/w/x), c/m/atime, ...
- 12 puntatori (*diretti*) ai primi 12 blocchi del file;
- un puntatore (*indiretto*) a un blocco che contiene puntatori ai successivi 1K blocchi (con blocchi da 4 KB e puntatori da 4 byte);
- un puntatore (*doppio indiretto*) a un blocco di puntatori indiretti (1M blocchi);
- un puntatore (*triplo indiretto*) a un blocco di puntatori doppi indiretti (1G blocchi).

Quindi:

- la dimensione massima di un file è più di 4 TB;
- per via dell’asimmetria, la profondità dell’albero cresce con la dimensione anche se la struttura è fissa. Questo garantisce efficienza con file sia piccoli che grandi;
- il grado alto dei nodi interni offre una performance migliore rispetto a e.g. alberi binari, perché minimizza il numero di seek e sfrutta le operazioni con granularità di settore (con una singola lettura su un blocco indiretto possiamo accedere a centinaia di blocchi dati);
- supporta file *sparse*, cioè che contengono regioni vuote. Tali regioni non occupano spazio su disco.

Gli inode sono conservati in un array che si trova in una posizione fissa sul disco, e si identificano con l’indice all’interno dell’array (*inumber*). Le directory contengono associazioni tra nomi e inumber. FFS fa caching in memoria degli inode di file aperti o usati di frequente.

Il tracciamento dello spazio libero avviene con una bitmap. La località è gestita con gruppi di blocchi e spazio riservato.



## Block group placement

Ottimizza il caso tipico in cui si accede in sequenza a dati e metadati di file contenuti nella stessa directory.

- il disco viene diviso in gruppi di blocchi, composti da tracce vicine;
- i metadati (array degli inode e bitmap dello spazio libero) sono distribuiti tra i gruppi – mantengono logicamente la struttura di array, e si può calcolare facilmente la posizione di ciascuna entry;
- i file se possibile vengono allocati nello stesso gruppo della loro directory. Le sottodirectory no, altrimenti sarebbe difficile garantire località per i file della directory;
- all’interno del blocco l’allocazione è con politica first-free. Tipicamente si forma qualche buco nella prima parte del gruppo, seguito da una larga sezione vuota alla fine. Per i file piccoli non è garantita località (ma è accettabile), mentre i file grandi tenderanno a finire per gran parte in una zona contigua alla fine del gruppo.

Questa strategia funziona solo se c’è abbastanza spazio libero. Per questo motivo, FFS riserva una porzione del disco (e.g. 10%), che non viene presentata alle applicazioni.