

Approfondimento sul filesystem

Il filesystem è quella parte di sistema operativo che si occupa della organizzazione e memorizzazione dei file.

Le funzioni tipiche di un fs sono:

- identificare in modo unico i file
- nascondere le caratteristiche fisiche di memorizzazione (astrazione)
- fornire metodi di accesso di controllo di accesso alle informazioni dei file
- garantire la persistenza delle informazioni e dei file

Il filesystem deve fare in modo che sia garantita la **nominazione** dei file (*naming*), ovvero il fatto di garantire che un file abbia sempre un *nome* che lo identifica, tramite il quale il sistema operativo lo potrà identificare univocamente.

Oltre al nome, il file presenta altri tipi di *metadati*, come:

- tipo (estensione)
- tempo di creazione
- lunghezza
- ...

In generale un sistema operativo può essere in grado di riconoscere strutture diverse per i file (se ne analizza il tipo) oppure no. In questo ultimo caso il file viene visto come un mero flusso di byte.

1. Directory

Il filesystem deve fare in modo che il nome del file e il suo contenuto/locazione siano collegati in modo persistente. Questa associazione fra nome e locazione è chiamata *descrittore* di file.

Un descrittore associa il nome di un file con le sue proprietà e la sua locazione sul dispositivo fisico di memorizzazione. Quando un file viene creato, ne viene contestualmente creato anche il descrittore.

Le directory, in sostanza, raccolgono **descrittori di file**. Si noti che le directory sono anch'esse file, ma con un contenuto "speciale". La radice non ha descrittore, ma un indirizzo fisico, mentre ciascuna sottodirectory avrà anch'essa il relativo descrittore. Il tipo di file di una directory è, appunto, *directory*. In ciascun descrittore di directory c'è l'indirizzo dal quale partire a cercare i file memorizzati nel blocco di tale directory.

Si noti, dunque, come tutti i file/directory fruibili dal filesystem posseggano il proprio descrittore, oggetto che *indica dove il file si trova fisicamente sul disco*.

Le operazioni possibili su una directory sono:

- ricerca di file;
- elenco dei descrittori di file;
- creazione e cancellazione di file.

Queste ultime operazioni producono effetti collaterali sulla directory: nel caso di creazione viene scritto all'interno di una dir il descrittore (e viceversa nella cancellazione).

1.1 Struttura delle directory

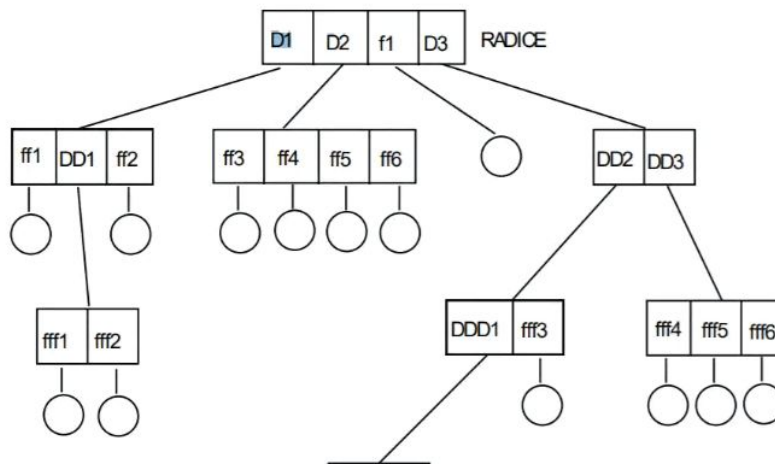
a) Struttura ad albero

La strutturazione ad albero è **gerarchica**. In questo caso l'utente crea delle *subdirectory* che hanno come *padre* la directory di livello gerarchico maggiore.

Ciò significa che questa struttura ha **un'unica radice** e ogni file ha un unico *pathname* (ovvero un unico percorso alla radice). In ogni istante è possibile lavorare dal punto di vista della *directory corrente*, in modo tale da non dover utilizzare il pathname completo.

Ovviamente questa struttura è percorribile in modo ricorsivo.

ESEMPIO DI DIRECTORY AD ALBERO



I nomi dei file possono essere di due tipi:

- nome assoluto (il cammino viene eseguito dalla radice), come ad esempio: `/home/macca/Desktop/file.png`;
- nome relativo alla directory corrente: prendendo l'esempio precedente, supponiamo di trovarci nella directory `/home/macca/Desktop`. Il nome relativo sarà semplicemente `file.png`.

b) Struttura a grafo aciclico

È una struttura ad albero con *più radici*, ottenendo quindi un **grafo**.

Lo stesso file, dunque, può essere riferito (eventualmente con nomi relativi diversi) da directory diverse. È la struttura base per un sistema operativo multiutente.

Questa struttura nasce per dare agli utenti la possibilità di condividere risorse.

