

# Sistemi operativi: Lezione 12

Antonio Rocchia

April 26, 2022

## Contents

<b>1</b>	<b>Il file system</b>	<b>1</b>
<b>2</b>	<b>Organizzazione del file system</b>	<b>2</b>
2.1	Struttura logica	2
2.1.1	File	2
2.1.2	Attributi di un file	2
2.1.3	Descrittore del file	2
2.1.4	Direttorio o Directory	3
2.1.5	Operazioni sui direttori	3
2.1.6	Tipi di implementazione per i direttori	3
2.1.7	Partizioni	3
2.2	Accesso	3
2.2.1	Operazioni sui file	4
2.2.2	Tabella dei file aperti	4
2.2.3	Struttura interna dei file	4
2.2.4	Metodi di accesso	4
2.2.5	Accesso sequenziale	5
2.2.6	Accesso diretto	5
2.2.7	Accesso ad indice	5
2.2.8	File system e protezione	5
2.2.9	Liste di accesso e gruppi	5
2.3	Organizzazione fisica	6
2.3.1	Metodi di allocazione	6
2.3.2	Allocazione contigua	6
2.4	Dispositivo virtuale	6

## 1 Il file system

Il file system è la componente del SO che fornisce agli utenti un meccanismo di accesso alle informazioni allocate in memoria di massa.

Il file system mette a disposizione delle astrazioni per rendere più intuitivo l'uso di quella risorsa:

- File: unità logica di memorizzazione
- Direttorio: insieme di file che consente all'utente di raggruppare file e/o altri direttori
- Partizione: Consente di associare più file system ad un dispositivo fisico.

Questa astrazione permette di interfacciarsi con un file system in modo omogeneo con qualsiasi tipo di dispositivo. (es. PC, Penne USB, etc...)

## 2 Organizzazione del file system

Il file system è rappresentata da un insieme di componenti qui rappresentati in ordine dal più alto livello di astrazione al più passo

1. Applicativi
2. Struttura logica
3. Accesso
4. Organizzazione fisica
5. Dispositivo virtuale
6. Hardware

### 2.1 Struttura logica

Fornisce una rappresentazione astratta delle informazioni memorizzate sul file system basata su file, directory e partizioni.

Ad esempio possiamo cambiare il nome di un particolare file, metterlo in una o più directory.

#### 2.1.1 File

Un file è un contenitore di informazioni. Il file system offre un sistema di *naming* che consente di individuare il file attraverso uno o più (come in linux) *nomi simbolici* che può essere usato nell' invocazione di comandi.

#### 2.1.2 Attributi di un file

Un file in generale ha i seguenti attributi:

- indirizzo, puntatore/i verso i blocchi di memoria secondaria in cui il file è effettivamente allocato.
- dimensione.
- data ed ora di creazione e ultima modifica.
- tipo del file che consente di classificarli.
  - di testo
  - eseguibili
  - ...
- utente proprietario, in un sistema multi-utente
- protezione e diritti di accesso.

#### 2.1.3 Descrittore del file

Il sistema deve poter salvare gli attributi di ogni file in opportune strutture dati collocate in memoria secondaria, in modo da rendere questi attributi permanenti anche per le sessioni successive.

Questa struttura si chiama *Descrittore del file*, *I-node in Unix*. I descrittori del file di ogni file sono salvati in una seconda struttura, *I-list in Unix*

In alcuni sistemi l'estensione inclusa nel file rappresenta il suo tipo. Questo aiuta il file system a gestire quel tipo di informazioni. In Unix non esiste questo tipo di classificazione.

### 2.1.4 Direttorio o Directory

Consente di raggruppare file. Una directory può contenere più file ed è rappresentata nel file system da un opportuna struttura dati.

Questa struttura associa al nome di ogni file contenuto i puntatori che consentono di localizzare quel file nella memoria di massa.

### 2.1.5 Operazioni sui direttori

Operazioni realizzate nella struttura logica.

- Creazione e cancellazione di directory (mkdir e rmdir)
- Aggiunta e rimozione di file alla directory
- Listing, elenco di tutti i file contenuti nella directory (ls)
- Attraversamento della directory (cd)
- Ricerca di file in directory (find)

### 2.1.6 Tipi di implementazione per i direttori

- **Struttura ad un livello**, una sola directory per ogni file system dove colloco tutti i miei file. Ora obsoleta data la mole di file contenuta in ogni sistema. Uno dei problemi può essere trovare nomi simbolici diversi per ogni file e proteggere ogni file per utenti diversi
- **Struttura a due livelli**. Obsoleta, usata nei primi sistemi multi-utente, una directory principale che contiene una directory per ogni utente. Presenta gli stessi problemi della struttura ad un livello, anche se i meccanismi di protezione sono più facili da realizzare.
- **Struttura ad albero**. Una struttura moderna formata da una directory radice da cui si forma una gerarchia di directory in modo ricorsivo. Esiste un direttorio per ogni utente ed ogni utente può definire un numero arbitrario di livelli.
- **Struttura a grafo aciclico**. Simile alla struttura ad albero, la differenza sostanziale consiste nel poter posizionare un file in più direttori diversi. *Uno stesso file può essere riferito con nomi diversi* (Questo file ha due *link* diversi). Questo sistema è quello implementato dai sistemi Unix.

### 2.1.7 Partizioni

Una partizione è l'astrazione che si associa univocamente ad una struttura logica.

Un disco può ospitare più partizioni. Esiste in oltre la possibilità di collegare a livello logico più partizioni in un'unica gerarchia tramite il meccanismo di mounting. Si sceglie un *punto di mounting* per collegare la radice del file system montato in un nodo del file system che monta.

Dato che il sistema di montaggio sostituisce il nodo scelto con la radice del file system montato, per evitare che questo nasconda dei file che vorremo accedere il sistema operativo mette a disposizione delle directory specifiche (mnt in linux) dove montare altre partizioni

## 2.2 Accesso

Definisce un'unità di trasferimento per leggere e scrivere i file, il *record logico*.

A differenza del *blocco* il *record logico* è un'unità di lettura/scrittura a livello di accesso e non ha nulla a che fare con l'effettiva allocazione delle informazioni. Si occupa di fornire meccanismi per manipolare il file system tramite:

- Metodi di accesso:
  - sequenziale
  - casuale

- indice
- Meccanismi di protezione

### 2.2.1 Operazioni sui file

Il Sistema operativo si occupa di gestire l'accesso *on-line* ai file. Ogni volta che un processo modifica un file, quel cambiamento è immediatamente visibile a tutti gli altri processi.

Le tipiche operazioni sui file sono:

- **Creazione:** allocazione di un file in memoria secondaria e inizializzazione dei suoi attributi
- **Modificare:**
  - **Lettura**
  - **Scrittura**
- **Cancellazione**

Ogni operazione richiederebbe la localizzazione del file sulla memoria secondaria. Quando un utente richiede una di queste operazioni si controlla se tale utente può fare questa operazione e dopo eventualmente si deve effettuare l'operazione stessa. Per farlo abbiamo bisogno delle informazioni contenute nel descrittore del file.

- Indirizzi dei record logici a cui accedere
- Attributi del file (diritti di accesso...)

### 2.2.2 Tabella dei file aperti

Le operazioni di accesso alla memoria secondaria sono molto costose. Per questo nei sistemi operativi moderni, per migliorare l'efficienza, *il file system* mantiene una struttura dati che *registra i file attualmente in uso*.

Viene fatto il *memory mapping* dei file aperti. I file aperti o porzioni di essi sono temporaneamente copiati in memoria centrale (che funge da buffer) per avere una velocità di accesso maggiore.

Quando il file viene **aperto**,

- introduciamo un nuovo elemento nella tabella dei file aperti
- il contenuto del file, o parte di esso, viene *mappato* in memoria centrale.

Quando il file viene **chiuso**,

- Salviamo il file in memoria secondaria
- Eliminiamo l'elemento corrispondente dalla tabella dei file aperti.

### 2.2.3 Struttura interna dei file

A livello di organizzazione fisica il file viene suddiviso in blocchi (o record fisici).

A livello logico invece le applicazioni vedono i file come un insieme di record logici.

Il record logico è un'unità di trasferimento logico nelle operazioni di accesso al file. Sono di dimensione variabile. Le caratteristiche dei record logici sono indipendenti da quelli dei record fisici, ma in generale i record fisici sono molto più grandi dei record logici.

### 2.2.4 Metodi di accesso

Esistono vari metodi di accesso ai file:

- Accesso sequenziale
- Accesso diretto
- Accesso ad indice

Il metodo di accesso è indipendente dal tipo di dispositivo utilizzato e dalla tecnica di allocazione dei record fisici in memoria secondaria.

### 2.2.5 Accesso sequenziale

In questo tipo di astrazione un file è una *sequenza* di record logici.

Le operazioni di accesso sono del tipo:

- readnext (lettura del prossimo record logico nella sequenza)
- e writenext (scrittura del prossimo record logico nella sequenza)

È necessario registrare la posizione corrente nella sequenza attraverso un I/O pointer. Ogni operazione di lettura/scrittura posiziona il puntatore sull'elemento successivo della sequenza.

Unix utilizza questo tipo di accesso.

### 2.2.6 Accesso diretto

Il file è un insieme di record logici. Si può procedere verso un particolare record logico specificandone il numero.

Le operazioni sono del tipo:

- read(i) lettura del record logico i
- write(i) scrittura del record logico i

Al contrario dell'accesso sequenziale, con l'accesso diretto posso raggiungere una posizione arbitraria nel file senza doverlo necessariamente scorrere l'intero file. Questo è utile quando ho bisogno di estrarre poche informazioni da grossi file, ad esempio in un database.

### 2.2.7 Accesso ad indice

Ad ogni file viene associata una struttura dati che contiene l'indice delle informazioni contenute. In questo tipo di accesso si presume che il file abbia una determinata struttura, si conservano delle chiavi a cui vengono associate dei puntatori ai record logici del file contenenti le informazioni desiderate

### 2.2.8 File system e protezione

Il sistema operativo permette al proprietario del file di controllare quali azioni sono consentite sul file e da parte di quali utenti.

### 2.2.9 Liste di accesso e gruppi

In UNIX ci sono tre modalità di accesso:

- read
- write
- execute

e tre classi di utenti:

1. owner access
2. group access
3. public access

L'amministratore del sistema può creare dei gruppi con nomi unici e assegnare/rimuovere utenti da tali gruppi.

## 2.3 Organizzazione fisica

Si occupa di rappresentare il file system nella memoria secondaria:

- Realizza i descrittori e li organizza.
- Alloca i record logici impaccati i record fisici nella memoria secondaria e si occupa di gestire lo spazio libero.

### 2.3.1 Metodi di allocazione

Il blocco o record fisico è l'unità di allocazione su memoria secondaria. Ogni blocco contiene un insieme di record logici contigui. Ogni blocco può contenere i record logici di al più un file.

Esistono varie strategie di allocazione:

- Allocazione contigua
- Allocazione a lista
- Allocazione ad indice

### 2.3.2 Allocazione contigua

Ogni file è mappato su un insieme di blocchi fisicamente contigui.

I vantaggi di questa strategia sono:

- Localizzare un blocco appartenente ad un file è poco costoso a livello computazionale, basta conoscere la posizione del primo blocco appartenente al file e calcolare la posizione del blocco ricercato rispetto ad esso.
- Si può realizzare l'accesso sequenziale o l'accesso diretto.

Gli svantaggi sono:

- Individuare spazi di memoria libera di blocchi contigui è un'operazione onerosa. Dovrei ispezionare ogni sequenza di blocchi liberi per conoscerne la lunghezza.
- Si può incorrere nella *frammentazione esterna*, man mano che si riempie il disco rimangono zone contigue sempre più piccole, a volte inutilizzabili.

## 2.4 Dispositivo virtuale

Presenta una visione astratta del dispositivo che appare come una sequenza di *blocchi* di dimensione fissa e costante che fungono da unità di allocazione.