

FILE SYSTEM

I moduli del Sistema Operativo

Nella dispensa precedente abbiamo visto che il Sistema Operativo è un programma scritto allo scopo di rendere disponibili le risorse di calcolo e i dispositivi periferici collegati al computer.

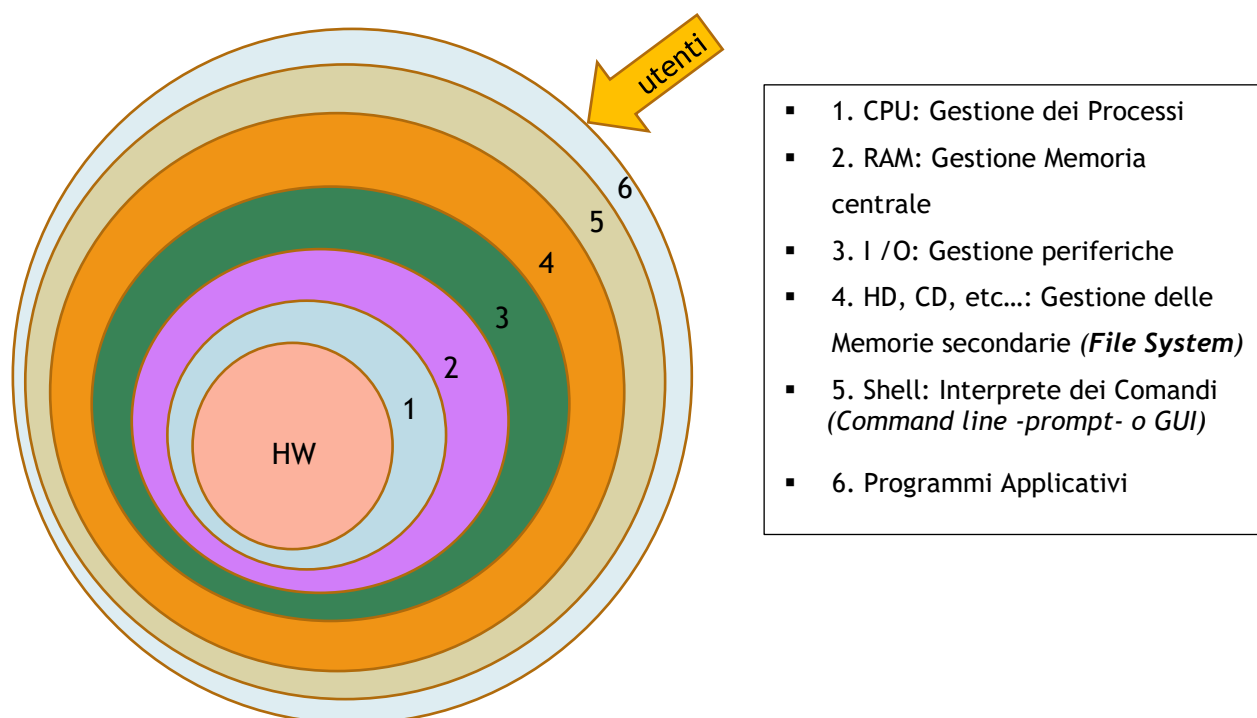
Le richieste che giungono al sistema operativo dall'utente vengono acquisite e gestite rispettando una politica gerarchica, basata su priorità e organizzazione modulare dei servizi.

L'Organizzazione modulare

Il Sistema Operativo viene rappresentato come una successione di strati logici costruiti sopra la macchina hardware, ciascuno dei quali rappresenta un livello di macchina virtuale.

Gli strati si avvicinano secondo un modello detto "a cipolla" in cui i livelli più bassi sono i più vicini all'hardware, mentre i livelli più alti sono più vicini all'utente, che ne richiede implicitamente l'attivazione attraverso una richiesta di servizio, presa in carico dal Sistema operativo.

Ciascuno strato costituisce un modulo che è in grado di svolgere le funzioni relative, **indipendentemente dagli altri livelli.**



Ciascuno modulo interagisce direttamente con uno specifico tipo di risorsa attraverso una collezione di routine, al fine di attivare quei meccanismi fisici necessari nel processo di risposta alla richiesta utente.

Ogni modulo deve far in modo che la sua struttura interna sia modificabile senza che gli altri moduli se ne accorgano e deve essere visibile solo attraverso le proprie primitive, cioè

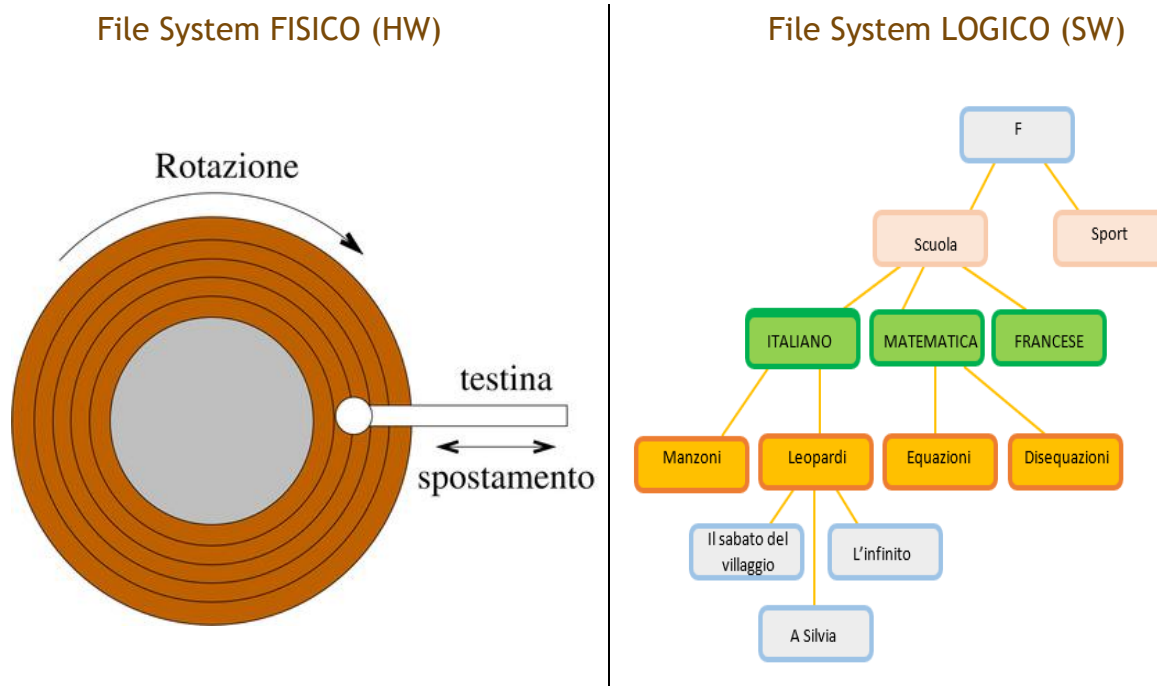
attraverso le procedure standard per mezzo delle quali si possono utilizzare i servizi di un modulo.

File System

Uno dei livelli del modello a cipolla (*onion skin*) è il File System, deputato all'organizzazione dei dati nelle memorie secondarie.

Il file system (abbreviazione: FS) è dunque il modulo del sistema operativo che **implementa il meccanismo con il quale i file sono posizionati e organizzati sui dispositivi di archiviazione** (memorie di massa come disco rigido, CD-ROM), mascherando all'utente la complessità fisica di tali dispositivi.

In sintesi, il FS consente ad un utente di organizzare i propri file, che da un punto di vista fisico non sono altro che una sequenza di byte salvati nei settori del disco, in una struttura gerarchica (ad albero) costituita da tante directory (folder o cartelle) annidate, facendo così diventare il dispositivo fisico un oggetto virtuale.



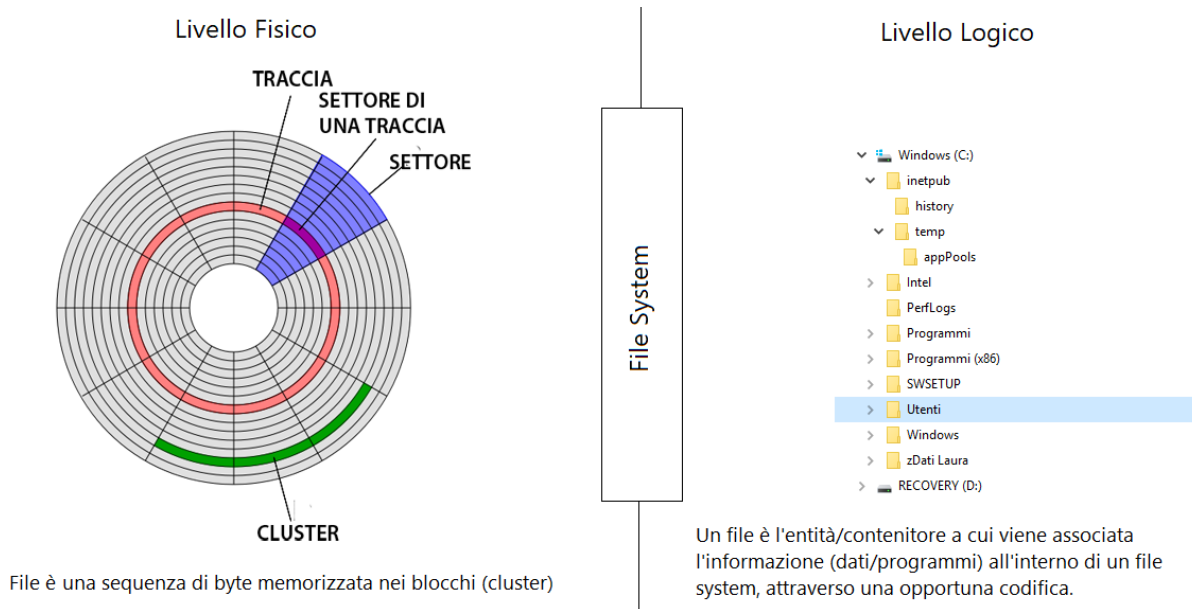
Tra i dispositivi di archiviazione secondaria (o di massa), i più famosi sono gli hard-disk (HD) magnetici, costituiti da blocchi di dimensione fissa (generalmente 512 byte), organizzati in settori.

Le uniche operazioni disponibili sul dispositivo di memorizzazione sono la lettura e la scrittura di uno o più blocchi. Tali operazioni avvengono ad opera di una testina che si muove in senso radiale rispetto al disco.

I dispositivi di memorizzazione secondaria, indipendentemente dalla tecnologia di realizzazione, (magnetici, SSD, ottici ecc), si presentano al sistema operativo come array di blocchi di dimensione fissa.

Il file system realizza un livello di astrazione, facendo sì che il disco appaia all'utente come un dispositivo virtuale, organizzato in una struttura gerarchica di cartelle, all'interno delle quali sono memorizzati i file.

La figura seguente schematizza i concetti esposti



L'immagine descrive la suddivisione del disco in tracce concentriche.

Un'altra suddivisione riscontrabile è quella in settori, ovvero in spicchi, rappresentati nell'immagine in colore viola.

I blocchi (blocchi fisici) non sono altro che le aree del disco di intersezione tra una traccia e un settore. I file vengono memorizzati nei cluster che sono costituiti, a loro volta, da blocchi fisici contigui.

Deframmentazione

Se il file da memorizzare ha dimensioni maggiori rispetto al cluster, il suo contenuto viene suddiviso in porzioni dimensionate al cluster. I cluster che si riferiscono ad uno stesso file possono trovarsi in posizioni non contigue.

A seguito di numerose cancellazioni e memorizzazioni, sul disco si vengono a creare molteplici frammentazioni, caratterizzate da notevoli distanze, sulla superficie del disco, tra cluster relativi allo stesso file e dallo spazio libero suddiviso in tante piccole porzioni di spazio non contigue.

In questa condizione le testine dovrebbero percorrere numerosi spostamenti al fine di ricostruire il contenuto di un file, con conseguenti aumenti in termini di tempo di accesso e prestazioni del sistema.

Le utilità di deframmentazione ricompattano lo spazio libero in modo da permettere una memorizzazione dei file su cluster contigui.

Partizionamento

Pur avendo un solo disco, è possibile suddividerlo, a livello logico, in più partizioni (o unità logiche), a seconda delle esigenze dell'utente, che le visualizzerà come separate.

Ciascuna unità logica (o root) viene indicata con una lettera e i due punti.

La lettera 'A' e 'B' ("A:" e "B:") in passato erano usate per il floppy disk (oggi non si usano più), mentre per l'hard disk, cd-rom, dvd, chiavette usb si usava la "C:", "D:", "E:" e così via.

Nella nostra figura le partizioni del disco sono due, associate alle unità "C:" e "D:".

Formattazione e Partizionamento

Per poter essere usato, un disco ha sempre bisogno di subire un processo di formattazione a basso livello (o

formattazione fisica).

Questa formattazione, fatta di solito dal costruttore dell'HD, permette l'associazione di un numero identificativo a ciascun settore.

In fase di formattazione fisica, è anche possibile scegliere la dimensione dei blocchi fisici (il valore standard è di 512 byte).

Il SO sottopone poi l'HD ad una **formattazione logica**, necessaria per creare e gestire sull'HD il File System del SO.

Il SO crea quindi la lista dei blocchi liberi e una directory iniziale (root), da cui si dipartiranno tutte le altre.

Se la formattazione logica si articola in più unità logiche, otterremo un partizionamento su più root differenti.

File path

Ogni file all'interno del file system è individuato da un percorso (path) che descrive la struttura gerarchica da percorrere per raggiungerlo, partendo dall'unità logica (**root**) in cui è stato caricato.

Per esempio:

C:\informatica\compiti\verifica.docx

In questo caso "C:\informatica\compiti\" è il percorso, dalla root, per arrivare alla cartella compiti in cui è contenuto il file verifica.docx.

Quando il percorso è riferito all'unità logica ("C:" in questo caso) si parla di **path assoluto**.

È anche possibile esprimere lo stesso percorso in modo relativo ad un'altra directory.

Per esempio, supponendo di voler esprimere il percorso di prima rispetto alla cartella "informatica", il **path relativo** sarà:

\compiti\verifica.docx

File Name

Tornando al concetto di file, occorre precisare che si tratta di un oggetto astratto, che contiene dati rappresentati secondo una specifica codifica (ad es.: ASCII; UTF-8; istruzioni in linguaggio macchina per un programma; png; mp3).

Al file sono sempre associati un nome ed un'estensione.

Nell'esempio precedente

- il nome è *verifica*
- l'estensione è *.docx*

L'estensione indica la tipologia di file in oggetto e serve al SO per capire con quale programma dovrà aprire il file. L'estensione dunque non determina la codifica dei dati, ma serve al SO per capire quale tipologia di programma associarvi.

Ad esempio, potrei avere un file con estensione “.png” (immagine); se anche gli cambiassi l'estensione in “.txt” rimarrebbe ugualmente un'immagine. A questo punto però, se cercassi di aprirlo col doppio click, il SO lo aprirebbe con il notepad (programma di default associato ai “.txt”) e ciò che otterrei sarebbe una serie di caratteri senza senso che corrispondono ai caratteri ascii associati ai byte dell'immagine (a volte non si tratta nemmeno di caratteri stampabili).

Operazioni sui file

Attraverso la mediazione del FS, l'utente può eseguire diverse operazioni sui file e sulle cartelle:

- Creazione
- Modifica
 - Copia
 - Rinomina
 - Sposta
- Eliminazione
- Interrogazione e modifica Proprietà