

# Il software

la parte contro cui si può solo imprecare



(continua)

---

---

---

---

---

---

---

---

## Funzioni principali del sistema operativo

- Avvio dell'elaboratore
- Gestione del processore e dei processi in esecuzione
- Gestione della memoria principale
- Gestione della memoria virtuale
- Gestione della memoria secondaria
- Gestione dei dispositivi di input / output
- Interazione con l'utente

---

---

---

---

---

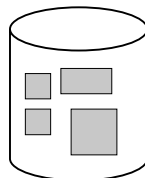
---

---

---

## Gestione della memoria secondaria

- Abbiamo visto che i supporti di memoria secondaria servono per **memorizzare** programmi e dati in modo permanente
- Ora vediamo il **file system**, ovvero quella parte del sistema operativo che si occupa di gestire e strutturare le informazioni memorizzate su questi supporti



---

---

---

---

---

---

---

---

**Gestione della memoria secondaria**

**Il file system:  
organizzazione logica dei file**

---

---

---

---

---

---

---

---

**Cos'è un file?**

*A collection of data or information that has a **name**, called the **filename**. Almost all information stored in a computer must be in a file. There are many different types of files: data files, text files, program files, directory files, and so on. **Different types of files store different types of information**. For example, program files store programs, whereas text files store text.*

[www.webopedia.com](http://www.webopedia.com)

---

---

---

---

---

---

---

---

**Cos'è un file?**

- Un file è una collezione di informazioni e deve avere un **nome logico** (è buona norma scegliere nomi significativi)
- Un file ha anche una **estensione**  
Esempi **.doc .txt .html .jpg .gif**
- Le estensioni sono **importanti** perchè di solito indicano il **tipo** del file (e danno informazioni sul programma applicativo in grado di manipolarlo)

---

---

---

---

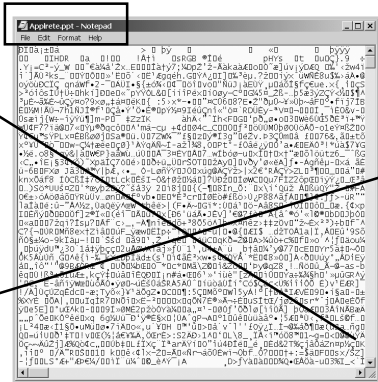
---

---

---

---

**Tipi di file .ppt**



---

---

---

---

---

---

---

---

---

---

**Tipi di file .ppt**



---

---

---

---

---

---

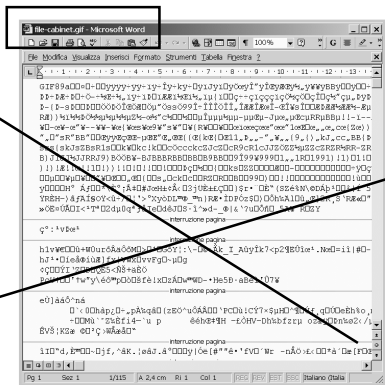
---

---

---

---

**Tipi di file .gif**



---

---

---

---

---

---

---

---

---

---

**Tipi di file .gif**




---

---

---

---

---

---

---

---

**Tipi di file .html**




---

---

---

---

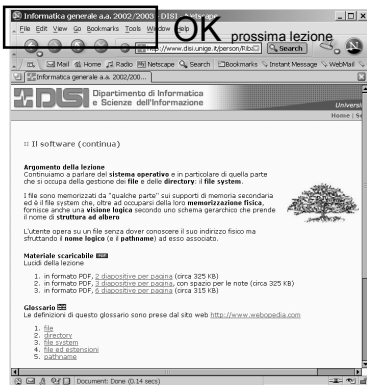
---

---

---

---

**Tipi di file .html**




---

---

---

---

---

---

---

---

### **Nomi dei file**

- Ogni sistema operativo ha delle **convenzioni** per la scelta dei nomi da associare ai file
  
- Consigli
  1. è meglio usare nomi **significativi**
    - mio.doc, pippo.doc ☹
    - voti.xls, tesi.doc ☺
  
  2. è meglio **non** usare caratteri speciali e lo spazio bianco
    - prima pagina.html, mia foto.jpg ☹
    - index.html, montagne.gif ☺

---

---

---

---

---

---

---

---

### **Il file system**

- Mediante il file system il sistema operativo fornisce una visione **astratta** dei file su disco e permette all'utente di
  - **identificare** ogni file mediante il suo nome
  
  - **operare** sui file mediante opportune operazioni
  
  - effettuare l'accesso alle informazioni grazie ad **operazioni di alto livello**, che non tengono conto del tipo di memorizzazione (si deve accedere allo stesso modo ad un file memorizzato sul disco rigido oppure su un CD-ROM)

---

---

---

---

---

---

---

---

### **Il file system**

- ... l'utente deve avere la possibilità di
  - **strutturare** i file, organizzandoli in sottoinsiemi secondo le loro caratteristiche, per avere una visione "ordinata" e strutturata delle informazioni sul disco
  
  - in un sistema multi-utente, inoltre, l'utente deve avere dei meccanismi per **proteggere** i propri file, ossia per impedire ad altri di leggerli, scriverli o cancellarli

---

---

---

---

---

---

---

---

### ***Il file system***

- Un insieme di operazioni minimale, presente in tutti i sistemi, è il seguente
  - **creazione** di un file
  - **cancellazione** di un file
  - **copia o spostamento** di un file
  - **visualizzazione** del contenuto di un file
  - **stampa** di un file
  - **modifica** del contenuto di un file
  - **rinomina** di un file
  - **visualizzazione delle caratteristiche** di un file

---

---

---

---

---

---

---

---

### ***Organizzazione gerarchica dei file***

- Il numero di file che devono essere memorizzati su un disco può essere molto elevato
- Si ha quindi la necessità di mantenere i file in una **forma ordinata**
- Un unico spazio ("contenitore") di file è scomodo
  - La scelta dei nomi diventa difficile poiché non è possibile avere due file con lo stesso nome nello stesso contenitore
  - Le operazioni di ricerca dei file diventano onerose



---

---

---

---

---

---

---

---

### ***Organizzazione gerarchica dei file***

- L'idea è quella di **raggruppare** i file in sottoinsiemi
- Questi sottoinsiemi di file vengono memorizzati all'interno di contenitori dette **cartelle (directory)**
- I nomi dei file sono **locali** alle directory  
Si possono avere due file con lo stesso nome purché siano in due directory diverse

---

---

---

---

---

---

---

---

### Organizzazione gerarchica dei file

- Per organizzare gerarchicamente i file, il sistema operativo deve fornire all'utente un insieme di operazioni per
  - **creare** una nuova directory
  - **rimuovere** una directory
  - **ridenominare** una directory
  - **elencare** il contenuto di una directory
  - **copiare** o **spostare** i file da una directory ad un'altra

---

---

---

---

---

---

---

---

### Organizzazione gerarchica dei file

- Gli elaboratori sono dotati di più unità di memoria secondaria
- DOS e Windows usano dei nomi per distinguere le unità
- In Unix/Linux la gestione è trasparente all'utente che conosce solo il nome del file, e non si interessa dell'unità dove esso è memorizzato



---

---

---

---

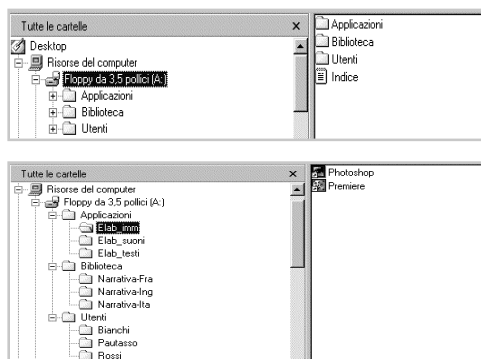
---

---

---

---

### Organizzazione gerarchica dei file



---

---

---

---

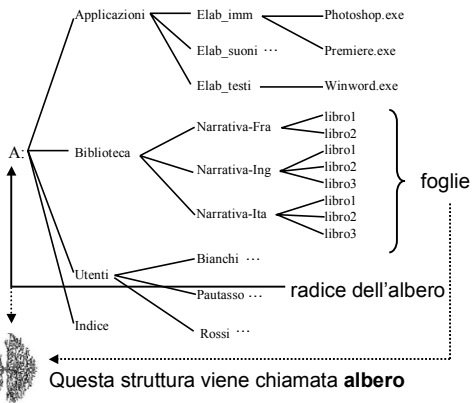
---

---

---

---

### Organizzazione gerarchica dei file




---

---

---

---

---

---

---

---

### Organizzazione gerarchica dei file

- Senza una strutturazione in directory, tutti i file possono essere identificati mediante il loro nome
- Nel caso di un'organizzazione gerarchica a più livelli il nome non è più sufficiente ad identificare un file
- Per individuare un file o una directory in modo univoco si deve allora specificare **l'intera sequenza di directory che lo contengono, a partire dalla radice dell'albero (pathname)**

---

---

---

---

---

---

---

---

### Organizzazione gerarchica dei file

- Ad esempio il file libro1 di narrativa italiana è univocamente identificato dal pathname  
A:/Biblioteca/Narrativa-Ita/libro1
- La directory Pautasso di Utenti è identificata dal pathname  
A:/Utenti/Pautasso
- Il carattere "/" viene usato come separatore in sistemi Unix-like, in Windows si usa il carattere "\"

---

---

---

---

---

---

---

---



### Organizzazione gerarchica dei file

- Un altro modo di individuare un file (o una directory) è basato sul concetto di **posizione relativa** (**pathname relativo**)
- In questo caso per individuare un file o una directory in modo univoco si deve specificare **l'intera sequenza di directory che lo contengono, a partire dalla posizione corrente**

---

---

---

---

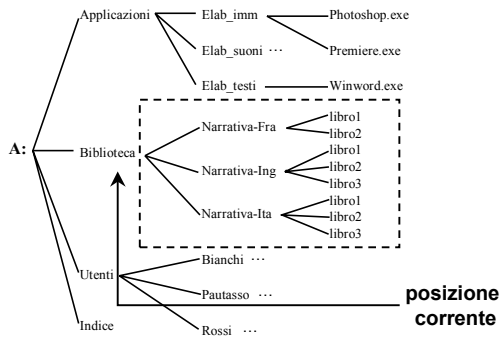
---

---

---

---

### Organizzazione gerarchica dei file



---

---

---

---

---

---

---

---

### Organizzazione gerarchica dei file

- Se la posizione corrente è A:/Biblioteca, il file libro1 di narrativa italiana è univocamente identificato dalla sequenza  
Narrativa-Ita/libro1
- Se la posizione corrente è A:/Utenti, la directory Pautasso è identificata dalla sequenza  
Pautasso
- Per raggiungere un determinato file a partire dalla posizione corrente si utilizza quindi il **pathname relativo**

---

---

---

---

---

---

---

---

**Come si cambia directory?**

- In Windows è facile, basta **clikkare** sulle icone gialle chiamate **Folder** per aprire una nuova directory ed esplorarne il contenuto
- Usando un'interfaccia a carattere (come la shell di Linux) si devono **conoscere** quei **comandi** che permettono di manipolare directory (e file)



---

---

---

---

---

---

---

---

**In laboratorio ...**

- In laboratorio usiamo i comandi per gestire file e directory tramite la shell, ovvero un processo in attesa di comandi
- Ricordiamo che
  - ad ogni comando corretto corrisponde l'esecuzione del programma corrispondente
  - se il comando è errato viene visualizzato un messaggio di errore

---

---

---

---

---

---

---

---

**In laboratorio ...**

- Usiamo anche dei programmi chiamati text-editor che permettono di creare dei file di tipo testo
- Un file di testo è un file che viene memorizzato in formato ASCII
- I file HTML (ovvero i file per il web) sono dei file di testo

---

---

---

---

---

---

---

---

**Gestione della memoria secondaria**

**Il file system:  
organizzazione fisica dei file**

---

---

---

---

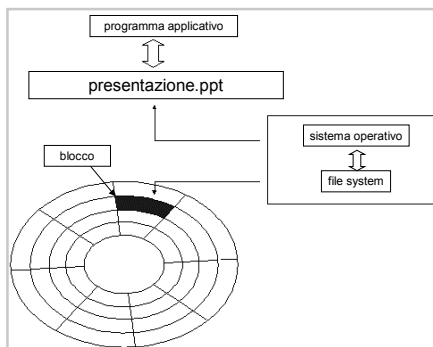
---

---

---

---

**Memorizzazione fisica dei file**



---

---

---

---

---

---

---

---

**Memorizzazione fisica dei file**

- Nel caso della memorizzazione fisica si devono considerare vari aspetti
  - Quali file sono memorizzati su disco?
  - Dove sono?
  - Come si può ottimizzare lo spazio su disco, evitando degli sprechi?

---

---

---

---

---

---

---

---

**Memorizzazione fisica dei file**

- Il disco è suddiviso in due parti
  - **Device directory**
    - contiene informazioni sui file e sulle directory
    - può essere vista come una tabella con un elemento per ogni file (**file descriptor**)
  - **Spazio disco**
    - contiene i file

---

---

---

---

---

---

---

---

**Device directory**

- Per ogni file sono mantenute informazioni diverse, tra cui
  - il **nome** del file
  - la **data di creazione** e di **ultima modifica**
  - la **dimensione** del file
  - il **nome** del **proprietario**
  - la descrizione delle **protezioni**
  - il **tipo** di file

---

---

---

---

---

---

---

---

**Device directory**

- Per ogni directory sono mantenute informazioni diverse, tra cui
  - il **nome** della **directory**
  - la **data di creazione** e di **ultima modifica**
  - il **nome** del **proprietario**
  - la descrizione delle **protezioni**
  - l'**elenco** dei file e delle sotto-directory

---

---

---

---

---

---

---

---

**Come sono memorizzati i file?**

- **Allocazione contigua**
  - ogni file è visto come una entità indivisibile memorizzata in blocchi contigui
  - noto l'indirizzo del primo blocco e il numero di blocchi si accede alle informazioni
- **Allocazione sparsa**
  - i blocchi sono memorizzati separatamente

---

---

---

---

---

---

---

---

**Allocazione sparsa**

- Possiamo vedere il disco fisso come un quaderno con tante pagine su cui un utente scrive delle relazioni (file)



- L'utente ha la necessità di **scrivere, cancellare, modificare** i file

---

---

---

---

---

---

---

---

**Allocazione sparsa**

- **Problemi**
  - Quando un file viene cancellato rimangono delle pagine "vuote"
  - La modifica di un file può dare origine ad una nuova versione più corta o più lunga di quella precedente. Potrebbero esserci dei file che richiedono delle pagine già occupate da altri file

---

---

---

---

---

---

---

---

### Allocazione sparsa

- In analogia con un quaderno “ad anelli”, le pagine possono essere **spostate, aggiunte, cancellate** senza creare sprechi di spazio e senza limitazioni sulla crescita delle dimensioni di un file (l'unico limite è quello di avere pagine bianche)
- In analogia con l'utilizzo di un indice, le **pagine** del quaderno vengono **numerate** e l'indice conterrà (nell'ordine) i numeri delle pagine del file

---

---

---

---

---

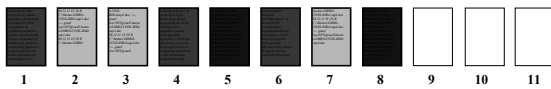
---

---

---

### Allocazione sparsa

relazione1: pag. 1, pag. 4, pag. 6  
relazione2: pag. 2, pag. 3, pag. 7  
relazione3: pag. 5, pag. 8



pagine bianche

---

---

---

---

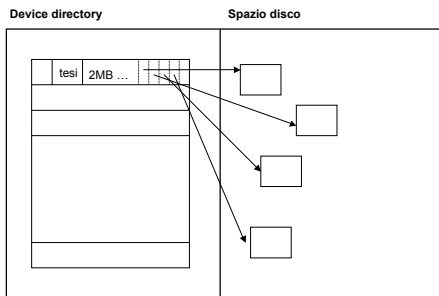
---

---

---

---

### Allocazione indexata



---

---

---

---

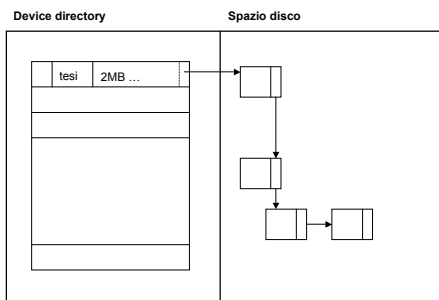
---

---

---

---

## Allocazione linkata



*Di solito si usa una tecnica mista indexata/linkata*

---

---

---

---

---

---

---

---