

## IL SISTEMA OPERATIVO

Insieme di programmi che opera *al di sopra della macchina fisica, mascherandone le caratteristiche e fornendo agli utenti funzionalità di alto livello.*

PROGRAMMI UTENTE
INTERPRETE COMANDI
FILE SYSTEM
GESTIONE DELLE PERIFERICHE
GESTIONE DELLA MEMORIA
GESTIONE DEI PROCESSI (NUCLEO)
HARDWARE

## IL SISTEMA OPERATIVO

- Il sistema operativo gestisce le risorse della macchina fisica sottostante e fornisce all'utente l'astrazione di *macchina virtuale*
  - lo strato di Gestione dei processi gestisce l'unità di elaborazione, ossia la CPU
  - lo strato di Gestione della memoria gestisce la memoria centrale
  - lo strato di Gestione delle periferiche gestisce i dispositivi periferici e le loro connessioni con la CPU
  - Il file system è l'organizzazione logica dei file sulla memoria di massa
  - l'interprete comandi permette di interpretare i comandi di alto livello

## FUNZIONI DEL SISTEMA OPERATIVO

Attraverso il S.O. il livello di interazione fra utente ed elaboratore viene elevato:

- senza S.O.: sequenze di bit
- con S.O.: comandi, programmi, dati

- Il S.O. **traduce le richieste dell'utente** in opportune **sequenze di istruzioni**, a loro volta trasformate in **valori e impulsi elettrici** per la macchina fisica.
- I sistemi operativi si sono evoluti nel corso degli ultimi anni :
  - interfacce testuali: MSDOS, Unix
  - interfacce grafiche: Macintosh, Windows, ...

## INTERFACCE TESTUALI

- Le operazioni vengono compiute scrivendo dei **comandi** su una riga dello schermo (**riga di comando**).
- In genere tale riga viene individuata mediante una sequenza di caratteri detta **prompt**.
- Il sistema operativo MS-DOS ha un'interfaccia testuale. Il prompt in genere è del tipo:

C:\>

## INTERFACCE TESTUALI

---

- Le utilità del sistema operativo sono disponibili attraverso un **linguaggio comandi** direttamente interpretato da un **processore dei comandi**.
  - I processori dei comandi sono evoluti da semplici riconoscitori di comandi a supporti per ambienti più complessi, quasi veri e propri linguaggi di programmazione
- **I comandi** possono essere classificati in:
  - comandi manipolazione dischi e dei file;
  - comandi di interazione con le risorse del sistema;
  - comandi di utilità e ausilio per lo sviluppo di un programma

## INTERFACCE GRAFICHE

---

- Le operazioni vengono compiute selezionando oggetti grafici mediante l'utilizzo di uno strumento di puntamento ("mouse").
  - **Point and click**: si porta prima il puntatore del mouse sull'oggetto e poi si preme uno dei tasti sul mouse.
- **Interfacce intuitive**: gli oggetti vengono rappresentati mediante disegni (**icone**) che ne richiamano la funzione.
- Molti sistemi operativi offrono la possibilità di interagire via interfacce sia grafiche sia testuali.

## METAFORE

---

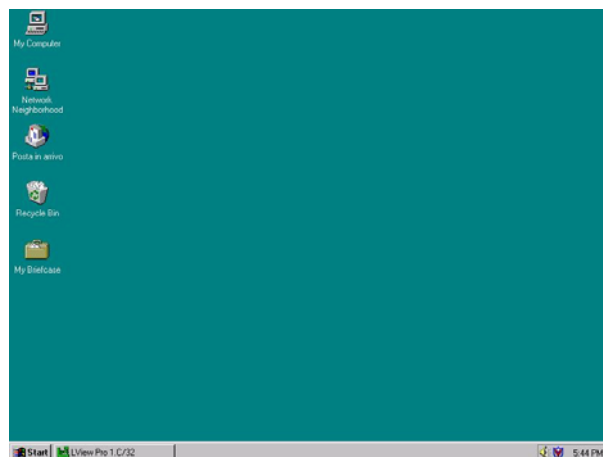
- Per rendere più intuitivo l'impiego della macchina, le interfacce grafiche riproducono sul computer il modo tradizionale di lavorare mediante metafore.

- Ad esempio, le interfacce attualmente più diffuse sono basate sulla metafora della **scrivania** (o **desktop**): lo schermo del computer rappresenta la scrivania, sulla quale si trovano tipicamente icone che rappresentano strumenti di impiego comune (casella della posta di ingresso, il cestino...), e i documenti (o file) su cui stiamo lavorando correntemente e collegamenti alle applicazioni che ci servono in questo momento.

## SCRIVANIA

---

- Scrivania di Windows 95/NT

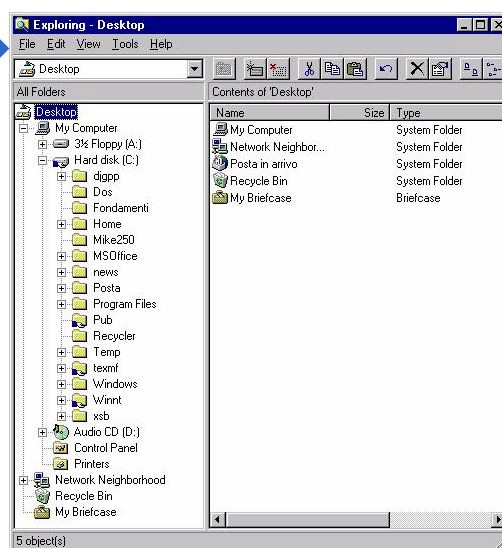


## FINESTRE

- Quando lanciamo una applicazione, una porzione rettangolare dello schermo (chiamata **finestra** o **window**) verrà riservata all'applicazione. L'input e l'output dell'applicazione verrà visualizzato in tale finestra.
- Mediante le finestre è possibile mandare in esecuzione più applicazioni contemporaneamente:
  - a ciascuna applicazione è riservata una finestra separata
  - in ogni istante, ci sarà una sola finestra attiva corrispondente all'applicazione con la quale si sta interagendo
  - le finestre possono essere spostate, ridimensionate e chiuse (termine dell'applicazione).

## FINESTRE

BARRA DEI  
MENU'



BARRA DEL  
TITOLO

BARRA DEGLI  
UTENSILI

## MEMORIZZAZIONE DI INFORMAZIONI: IL FILE

- Dal punto di vista dell'utente, il file è l'unità logica di memorizzazione delle informazioni.
- Un file è caratterizzato da:
  - un nome
  - una dimensione (in byte)
  - degli attributi
  - una data
- Il nome di un file è costituito da:

Identificativo  
univoco

**nomefile.estensione**

Facoltativa:  
identifica il tipo di  
informazioni  
contenute nel file

## TIPI DI FILE

- File di testo: sono file che contengono testo ASCII  
**LETTERA.TXT**

Spettabile Ditta,  
dal vostro catalogo datato  
Ottobre 1994

- File eseguibili: sono file che contengono "codice macchina"

**MIOPROG.EXE**

Б\\_^¥†©÷ÓμÆ@â¢%|ð‡f  
©™âÆ@μ¢‡f¥^¥®ð\*œ\_μ¢\_  
Б¬\_^Ÿ®ð‡%™ª¿¿μ¢%|ð‡f  
±¤‡£÷™â

## TIPI DI FILE

- File (di testo) contenenti programmi

**PROVA.C**: contiene un programma di prova scritto in linguaggio C (“PROVA” è la parte *nomefile* e “C” è la *estensione*).

```
main()
{ int i;
  i = 5;
  printf("Valore di i: %d\n", i);
}
```

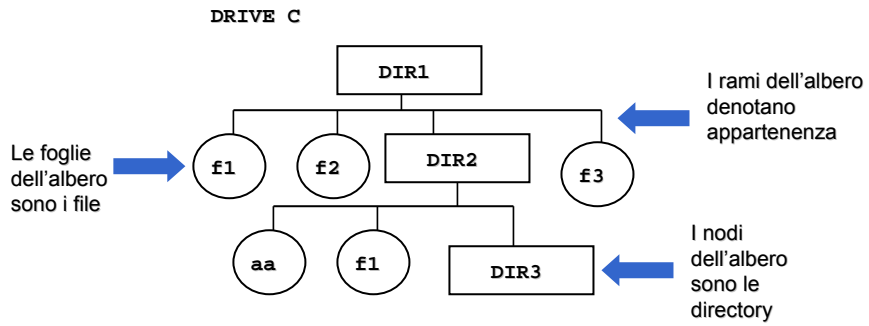
- File di dati o file binari: contengono informazione strutturata  
**STIPENDI.DAT**

AVANTI LUIGI	1/2/93
L. 2.145.000	
BOSSI PIERO	6/2/93
L. 2.000.000	

## FILE SYSTEM

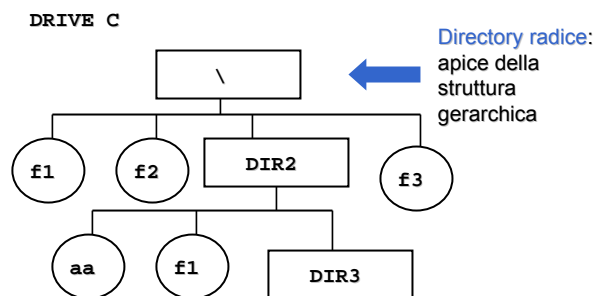
- È l'organizzazione dei file su disco, come vengono strutturati e memorizzati.
  - Ogni unità (disco) contiene almeno una *directory*, ossia un “contenitore” in cui memorizzare nome, posizione e altre informazioni di ogni file residente sul disco stesso.
  - Il termine **drive** è una abbreviazione di *disk drive* e significa *unità*.
- La *directory* principale (*directory* radice o root) viene creato al momento della *formattazione* del disco. Altre *directory* possono essere create dagli utenti secondo una *struttura gerarchica*.

## FILE SYSTEM



- Non ci sono limiti teorici al grado di innestamento delle directory.

## FILE SYSTEM





## IDENTIFICAZIONE DI FILE E DIRECTORY

---

- Ogni file o *directory* è identificato da un percorso (*path*) seguito dal suo nome.
- Il percorso è **assoluto** se è riferito alla radice: rappresenta il percorso che si deve seguire a partire dalla radice per giungere alla foglia voluta (file o *directory*).
- Il percorso è **relativo** se è riferito alla *directory corrente* (e al *drive corrente*) rappresenta il percorso che si deve seguire a partire dalla *directory* corrente per giungere alla foglia voluta.

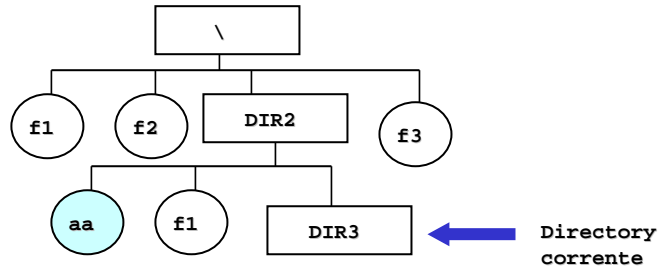
## DRIVE E DIRECTORY CORRENTE

---

- Quando si lavora su un calcolatore, ci si “posiziona” in un determinato punto del file system caratterizzato univocamente da due informazioni:
  - *il drive corrente*
  - *la directory corrente*
- Si definisce *drive corrente* il drive di lavoro, che verrà considerato come punto di partenza in tutte le operazioni.
- Si definisce *directory corrente* la directory di lavoro, che verrà considerata come punto di partenza in tutte le operazioni.

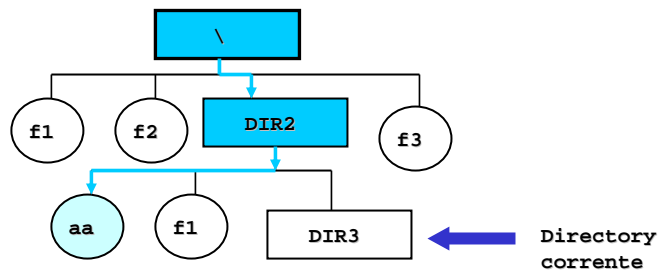
## ESEMPIO: PERCORSO ASSOLUTO

DRIVE C



## ESEMPIO: PERCORSO ASSOLUTO

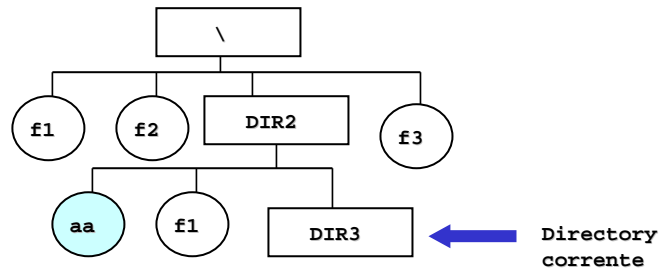
DRIVE C



- NOTA: la directory corrente non viene considerata nel determinare il percorso assoluto

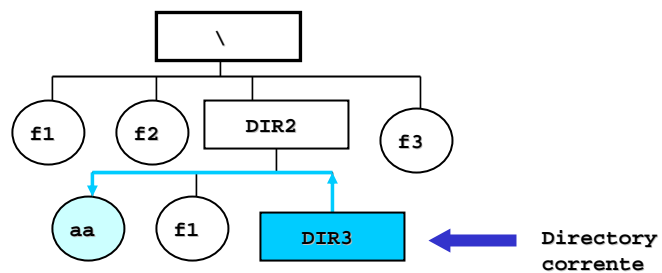
## ESEMPIO: PERCORSO RELATIVO

DRIVE C



## ESEMPIO: PERCORSO RELATIVO

DRIVE C

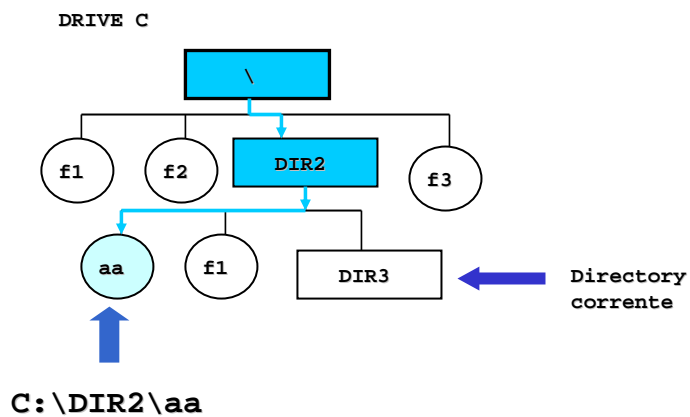


- NOTA: la directory corrente e' il punto di partenza per determinare il percorso relativo

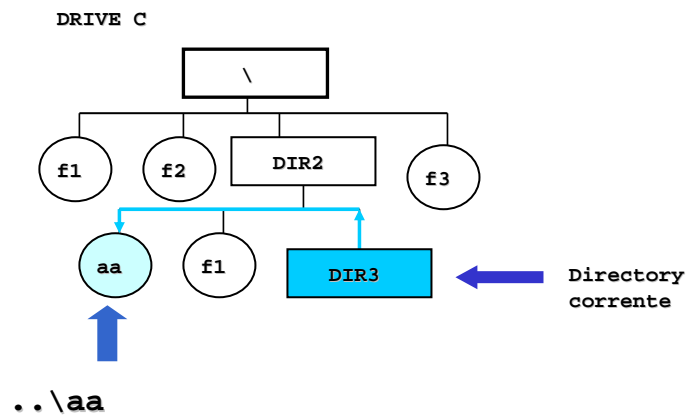
## NOTAZIONI CONVENZIONALI

- Nel sistema MSDOS e Windows 95/98/NT (quando usato con interfaccia comando):
  - la *directory* corrente è identificata dal punto (.)
  - la *directory* padre di quella corrente è identificata dal doppio punto (..)
  - la *directory* radice è indicata con una barra rovesciata (\)
  - il percorso assoluto di un file (o directory) viene indicato specificando nell'ordine il drive su cui si trova il file, il cammino per raggiungerlo inteso come sequenza di directory che si devono percorrere e, infine, il nome del file. Tale sequenza e' separata da simboli \
  - il percorso relativo di un file (o directory) viene indicato specificando il cammino per raggiungerlo a partire dalla directory corrente (che non viene indicata).

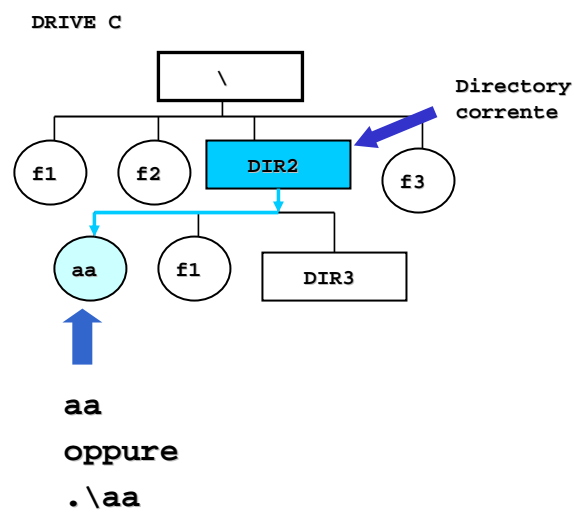
## ESEMPIO: PERCORSO ASSOLUTO



## ESEMPIO: PERCORSO RELATIVO



## ESEMPIO: PERCORSO RELATIVO



## COMANDI SUL FILE SYSTEM

---

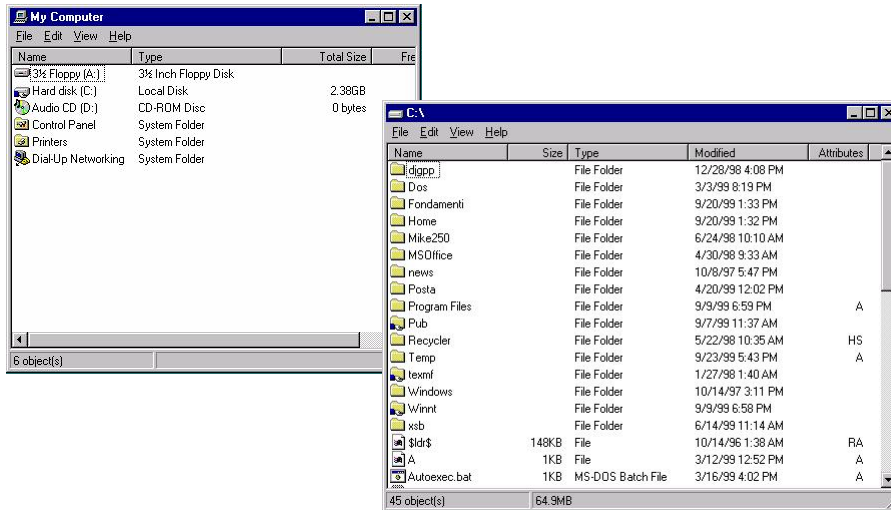
- Esistono comandi che agiscono sul file system per:
  - creare/rimuovere/spostare/rinominare directory
  - creare/rimuovere/spostare/rinominare file
  - visualizzare il contenuto di directory
  - cambiare directory/drive corrente

## SISTEMI A FINESTRE

---

- In Windows tutte le risorse della macchina, non solamente la memoria di massa, sono viste attraverso il file system.
- La radice del file system è rappresentata dal desktop, ogni risorsa offerta dal computer è vista come un elemento della gerarchia e rappresentata con una icona.
- Ogni elemento della gerarchia (o icona) può rappresentare una risorsa oppure un gruppo di risorse (cartella). Il file system può essere esplorato aprendo l'icona My Computer (Risorse del Computer)

## SISTEMI A FINESTRE



## COMANDI SUL FILE SYSTEM

- Esistono comandi che agiscono sul file system per:
  - creare/rimuovere/spostare/rinominare directory
  - creare/rimuovere/spostare/rinominare file
  - visualizzare il contenuto di directory
  - cambiare directory/drive corrente



**NEI SISTEMI A FINESTRE SONO SPECIFICATI TRAMITE IL MOUSE**