Le seguenti slides sono un libero taglia-incolla delle dispense del corso “Informatica di Base” tenuto da Alessandro Mazzei nell'A.A 2005/2006 e disponibili alla pagina:

http://www.di.unito.it/~mazzei/bioInfoCourse.html.

Software

- il computer è una macchina programmabile
- l'hardware da solo non è sufficiente a far funzionare l'elaboratore e -> è necessario introdurre del software ovvero...

- ...un insieme di programmi che permettono di trasformare un insieme di circuiti elettronici in un oggetto in grado di svolgere delle
  - delle operazioni di natura diversa e
  - per diversi tipi di utenti

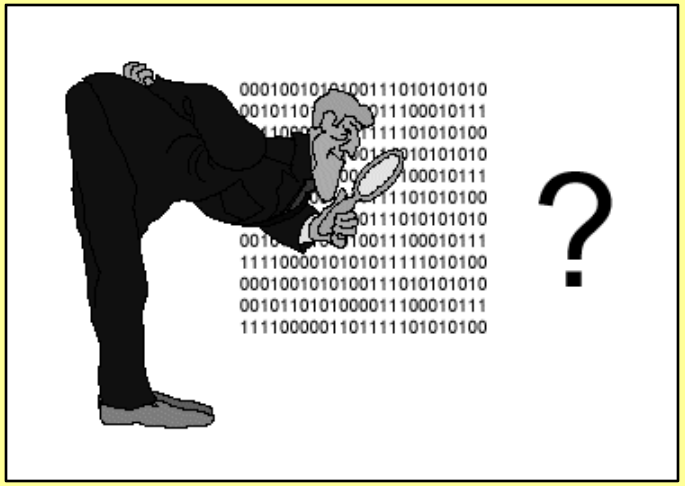
Software

- Il software (= i programmi) è costituito da istruzioni che dicono alla macchina hardware cosa deve fare.
- Una programmazione diretta della macchina hardware da parte degli utenti è davvero difficile

  Qual è il problema?

- l'utente dovrebbe conoscere l'organizzazione fisica dell'elaboratore e il suo linguaggio macchina (***improbabile!***)
- ogni programma dovrebbe essere scritto utilizzando delle sequenze di bit (0-1) (***inumano!***)
- l'hardware cambia da macchina a macchina! Un programma scritto per la macchina A non funzionerebbe se eseguito sulla macchina B -> ogni piccola differenza hardware comporterebbe una riscrittura del programma (***non portabilità!***)

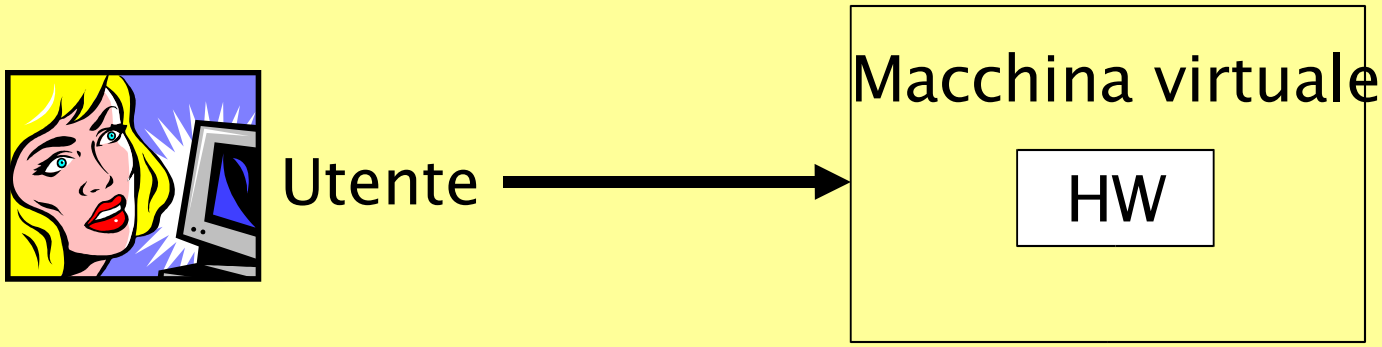
Software



- Questo non è accettabile!
- Occorre fornire all'utente un modo per:
  - astrarre dalle caratteristiche fisiche della macchina
  - avere un linguaggio semplice di interazione con la macchina
  - riuscire a interagire più o meno allo stesso modo con macchine con caratteristiche hardware un po' diverse
  - avere un insieme di programmi applicativi per svolgere compiti diversi

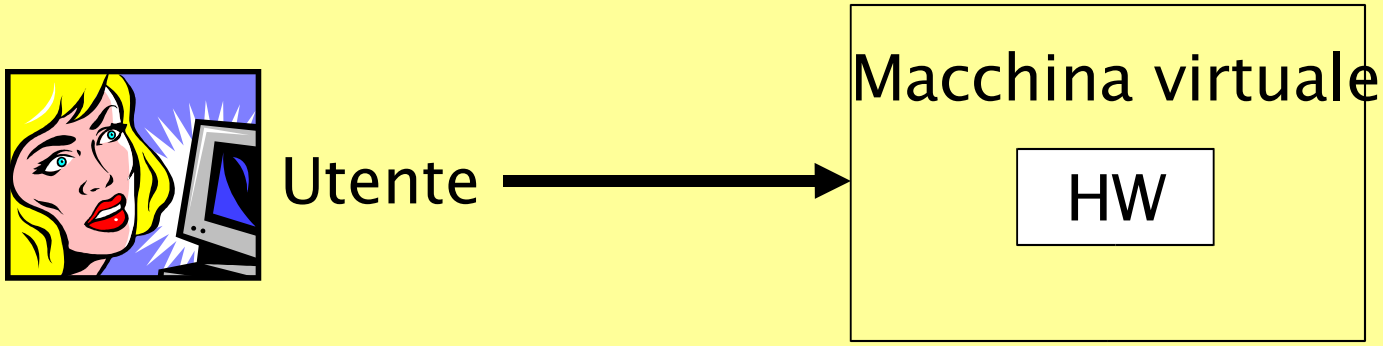
Macchina virtuale

- Nei moderni sistemi di elaborazione questi obiettivi vengono raggiunti grazie alla definizione di una **macchina virtuale** che viene realizzata al di sopra della macchina hardware reale
- Virtuale in quanto essa non esiste fisicamente ma viene realizzata mediante software: il **software di base o di sistema**, che fornisce all'utente una visione sul calcolatore astratta con cui interagire



Macchina virtuale

- L'utente interagisce con la macchina virtuale grazie ad un opportuno **linguaggio di comandi**
  - La macchina virtuale si preoccupa della **traduzione** di ogni comando impartito dall'utente nella sequenza di comandi che realizzano la stessa funzione e sono riconosciuti dalla macchina fisica sottostante

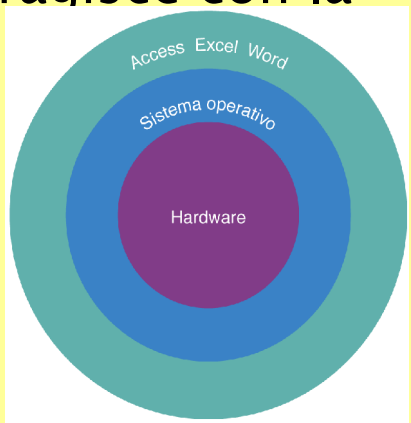


Software di base

- Gli strumenti software che permettono all'utente (e ai programmi applicativi) di gestire le risorse fisiche e interagire con l'elaboratore in modo semplice sono parte della macchina virtuale e ci riferiamo ad essi come **software di base**;
- Tre tipi di software di base
  - **sistema operativo**
  - **Drivers (?!)**
  - **Utility (?!)**
- Sistemi operativi: la componente software principale, presente in qualsiasi computer
  - Esempio di sistema operativo in laboratorio: Windows XP,
  - Unix

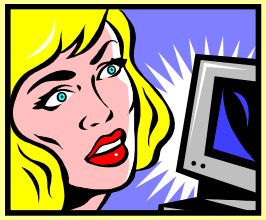
Sistema operativo

- E' costituito da programmi per la gestione e delle operazioni più elementari di un computer che contribuiscono in modo determinante a creare la macchina virtuale
- Ha due obiettivi principali:
  1. Gestione efficiente delle componenti fisiche dell'elaboratore (CPU, memoria, periferiche) in modo da sfruttarne al meglio le potenzialità
  2. Creazione di un ambiente virtuale per facilitare l'interazione uomo-macchina, ovvero creazione dell'ambiente di lavoro in cui l'utente interagisce con la macchina
- Consente l'esecuzione del software applicativo sul computer

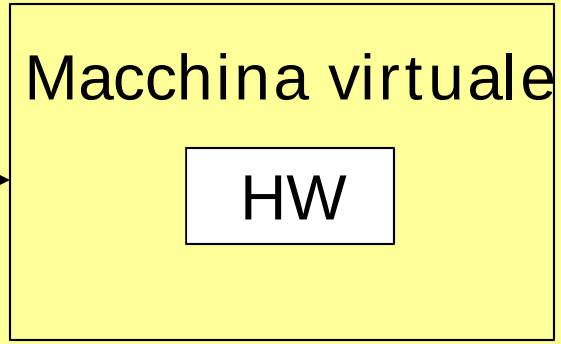


Sistema Operativo

- Ob 1. Gestione efficiente delle componenti fisiche dell'elaboratore: CPU, memoria, periferiche) in modo da sfruttarne al meglio le potenzialità
- Il sistema operativo realizza una macchina virtuale con cui l'utente interagisce, che può essere **più potente** di quella fisica
- L'effetto di questa gestione efficiente è che l'utente ha l'impressione di interagire con una macchina con più potenza di calcolo e più memoria di quella che realmente ha



Utente



Esempio: i sistemi multi-utente

- **Ho un sistema multi-utente (una macchina utilizzabile da più utenti contemporaneamente)**
- **E' realizzato con una sola macchina hardware potente collegata e tanti piccoli terminali collegati ad essa che sfruttano le sue risorse (memoria, potenza di calcolo, stampante)**

- **Il SO nasconde all'utente la presenza di altri utenti, creando per lui una sorta di macchina virtuale in cui lui ha l'impressione di lavorare da solo e di avere a disposizione un sistema dedicato solo alle proprie elaborazioni**

Sistema operativo

- Ob 2: Creazione di un ambiente virtuale per facilitare l'interazione uomo-macchina ovvero creazione dell'ambiente di lavoro in cui l'utente interagisce con la macchina

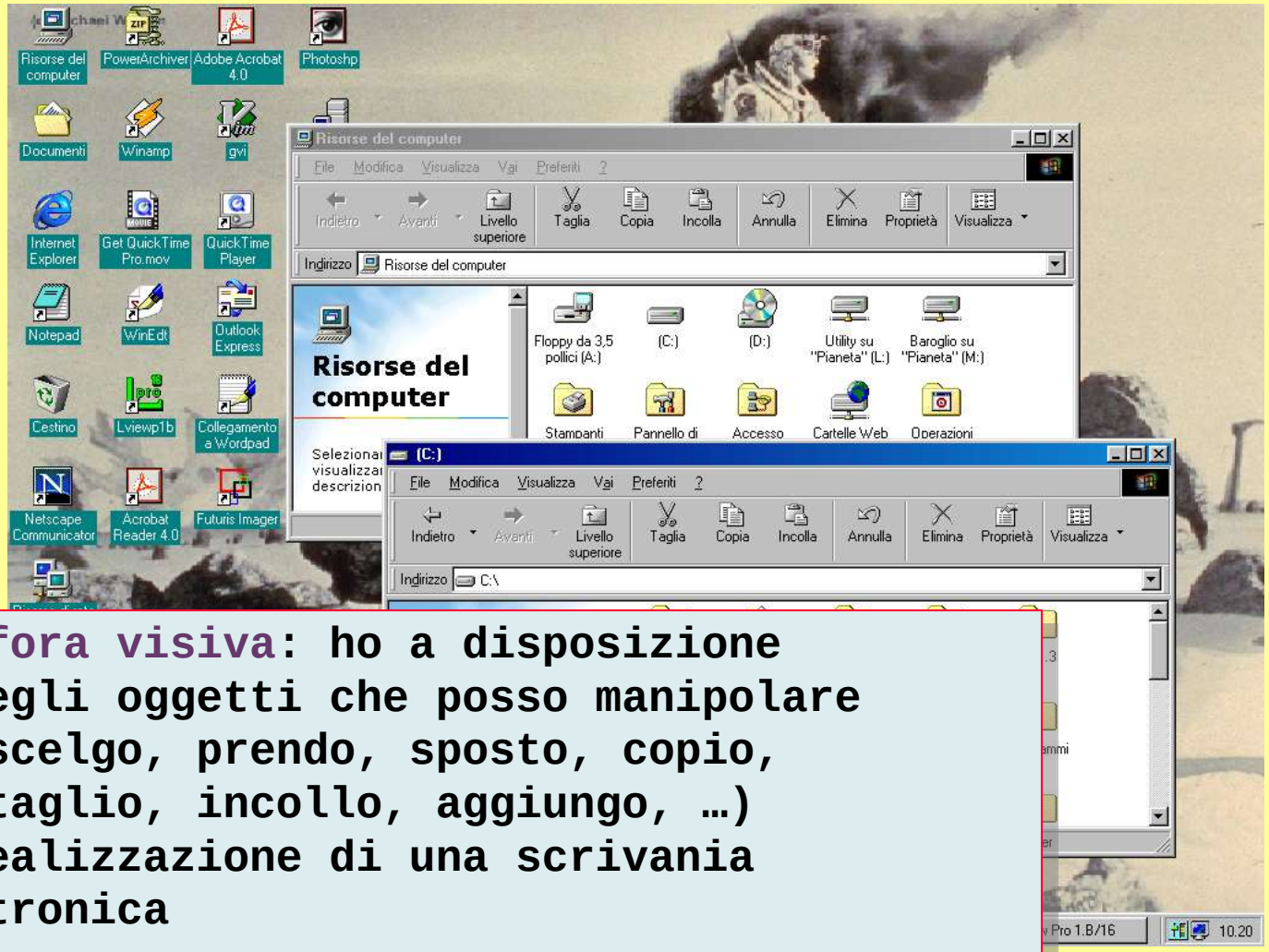
- Il sistema operativo ha il compito di creare un **interfaccia intuitiva verso l'utente** che fornisca un linguaggio di comandi semplice per l'interazione con la macchina

Interfaccia utente

- Consente al computer di interpretare le richieste dell'utente all'interno delle varie applicazioni tramite vari comandi
- Esistono vari tipi di interfacce che variano rispetto al modo in cui l'utente può comunicare con la macchina -> vari modi di permettere all'utente di dare comandi
- Once upon a time ... solo interfacce a comandi (no mouse): l'utente doveva digitare istruzioni di tipo testuale per dare un comando in input al computer:
  >>copy a:\pippo.doc c:

- Interfacce grafiche (GUI): i comandi sono impartiti mediante l'interazione via il mouse. Il clic del mouse su un'icona viene tradotto in una opportuna sequenza di istruzioni che il calcolatore esegue per soddisfare la richiesta dell'utente

Esempio Windows – Accendo il computer:
l'interazione con un'interfaccia grafica



Metafora visiva: ho a disposizione
degli oggetti che posso manipolare
(scelgo, prendo, sposto, copio,
taglio, incollo, aggiungo, …)
-> realizzazione di una scrivania
elettronica

(successiva all'introduzione del **mouse**)

GUI: www.webopedia.com

- Graphical User Interface: A program interface that takes advantage of the computer's graphics capabilities to make the program easier to use. Well-designed graphical user interfaces can free the user from learning complex command languages. On the other hand, many users find that they work more effectively with a command-driven interface, especially if they already know the command language. Graphical user interfaces, such as Microsoft Windows and the one used by the Apple Macintosh, feature the following basic components:
  - pointing device (puntatore)
  - icons (icone)
  - desktop (scrivania)
  - windows (finestre)
  - menu

GUI (continua...)

- **pointing device :** A device, such as a mouse or trackball, that enables you to select objects on the display screen.
- **icons :** Small pictures that represent commands, files, or windows. By moving the pointer to the icon and pressing a mouse button, you can execute a command or convert the icon into a window. You can also move the icons around the display screen as if they were real objects on your desk.
- **desktop :** The area on the display screen where icons are grouped is often referred to as the desktop because the icons are intended to represent real objects on a real desktop.

GUI (continua...)

- **windows**: You can divide the screen into different areas. In each window, you can run a different program or display a different file. You can move windows around the display screen, and change their shape and size at will.
- **menus** : Most graphical user interfaces let you execute commands by selecting a choice from a menu.

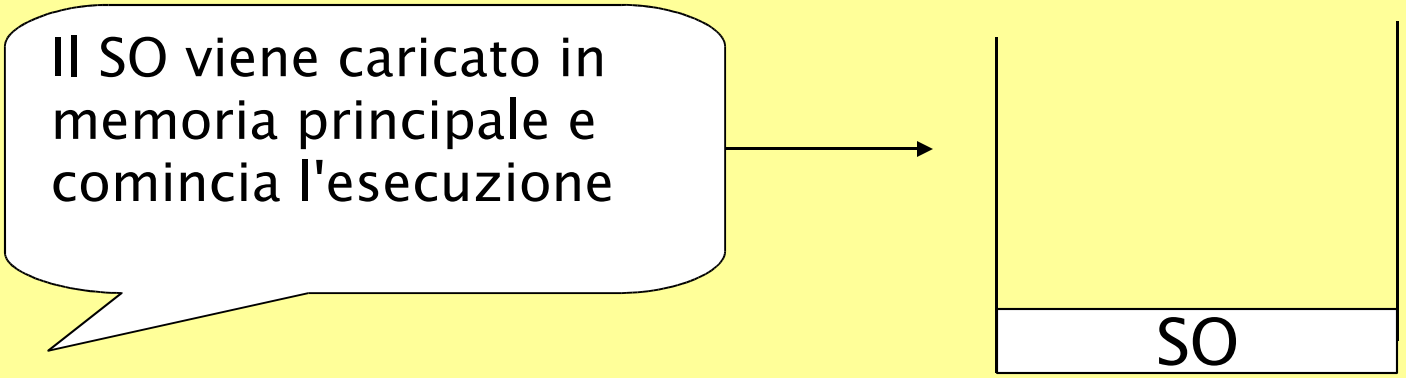
Funzioni principali del sistema operativo

Soffermiamoci sulle…

- **Principali funzioni del sistema operativo**:
  - **Bootstrap**: configurazione e accensione della macchina
  - **Gestione del processore e dei task**: un solo processore e multi-tasking
  - **Gestione della memoria primaria:** una sola memoria e molti programmi diversi caricati su di essa
  - **Gestione delle informazioni in memoria secondaria -> Gestione dei File:** per consentire da parte dell'utente l'archiviazione e il reperimento dei dati sfruttando i dispositivi di memoria di massa

Bootstrap

- Il sistema operativo viene mandato in esecuzione al momento dell'accensione della macchina -> AVVIO
- Vengono caricati in memoria principale i programmi del sistema operativo relativi alla **gestione delle risorse** hardware (processore, memorie, periferiche,…) e il programma che crea un **interfaccia intuitiva verso l'utente**: ossia fornisce un linguaggio di comandi per l'interazione con il sistema

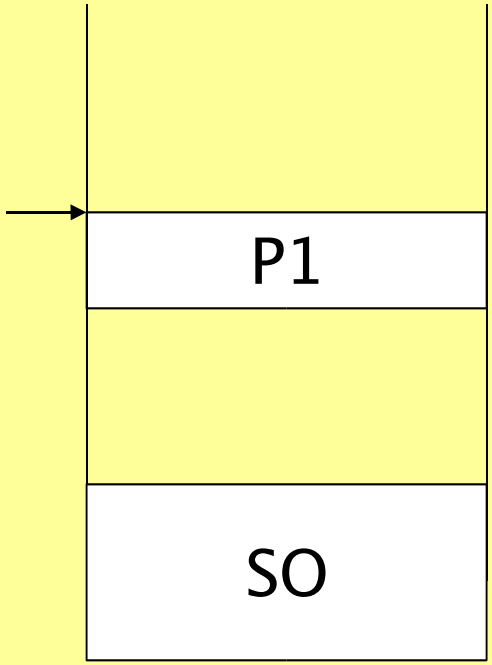


Bootstrap

- Inizializzazione del sistema per tener conto delle risorse hardware collegate
- Esecuzione di programmi antivirus che verificano l'eventuale presenza di virus sul disco dell'elaboratore
- Virus = programmi pirata che possono essere trasferiti sul nostro elaboratore quando copiamo un nuovo programma in memoria -> e-mail

Gestione del processore e dei task

- La corretta e efficace gestione del processore è uno dei compiti più importanti del SO
- Il processore esegue programmi, elaborando in modo sequenziale istruzioni
- Esempio di Windows: quando si clicca due volte sull'icona di un programma, il sistema operativo
  - cerca il programma sull'hard disk
  - copia il programma in memoria principale
  - ***comunica*** al processore l'indirizzo in memoria principale della prima istruzione del programma
- Programma in esecuzione: processo

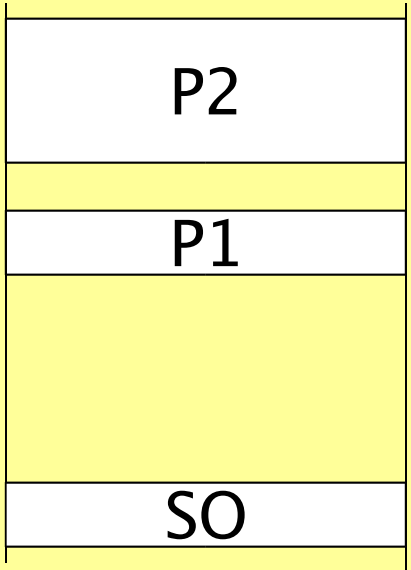


Gestione del processore e dei task

- Il microprocessore di solito è uno solo: significa che deve eseguire un programma per volta?
- No!
- Sistemi mono-tasking versus sistemi multi-tasking

Gestione del processore e dei task

- Per computer operante in multi-tasking si intende un computer che può eseguire più programmi contemporaneamente -> multiprogrammazione
- L'idea è quella di mantenere in memoria primaria diversi programmi oltre al sistema operativo

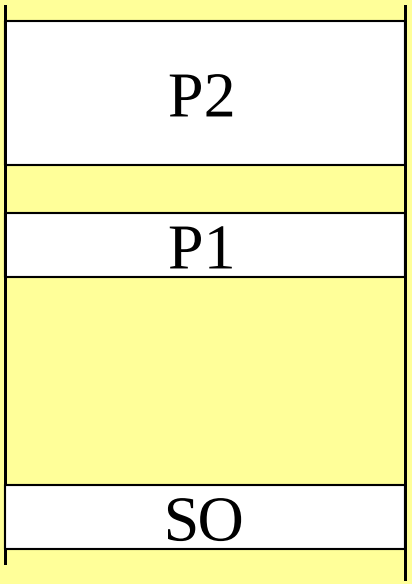


Gestione del processore e dei task

- Osservazione: un programma tipicamente consta di istruzioni di lettura scrittura dati da/su memoria o altre periferiche e di istruzioni che richiedono l'intervento della CPU. MA la CPU è molto più veloce dei dischi e delle altre periferiche -> passa la maggior parte del suo tempo in attesa del completamento delle operazioni demandate a questi dispositivi -> stato inattivo
- Idea: quando durante l'esecuzione di un programma la CPU è nello stato inattivo la si può sfruttare per eseguire parte di un altro programma
- Quando un processo si ferma (per esempio in attesa di un dato dall'utente) la CPU può passare ad eseguire le istruzioni di un altro processo
- Il sistema operativo si occupa di gestire l'**alternanza** tra i processi in esecuzione

Gestione del processore e dei task

- Alternanza veloce di esecuzioni da P1 a P2 gestita dal sistema operativo: il processore in ogni istante considera sempre solo un programma, ma salta dall'esecuzione di istruzioni di P1 all'esecuzione di istruzioni di P2 frequentemente ->
- l'utente ha l'impressione di un'esecuzione simultanea, si può dire che il sistema operativo simula un quasi parallelismo (pseudo-parallelismo) nell'accesso alle utenti o più programmi



- Sistema a finestre offerto dal sistema operativo Windows e multiprogrammazione: in ogni finestra apro un programma diverso

Esempio: multiprogrammazione e finestre

- Grado di multiprogrammazione: misura di quanti programmi posso aprire contemporaneamente

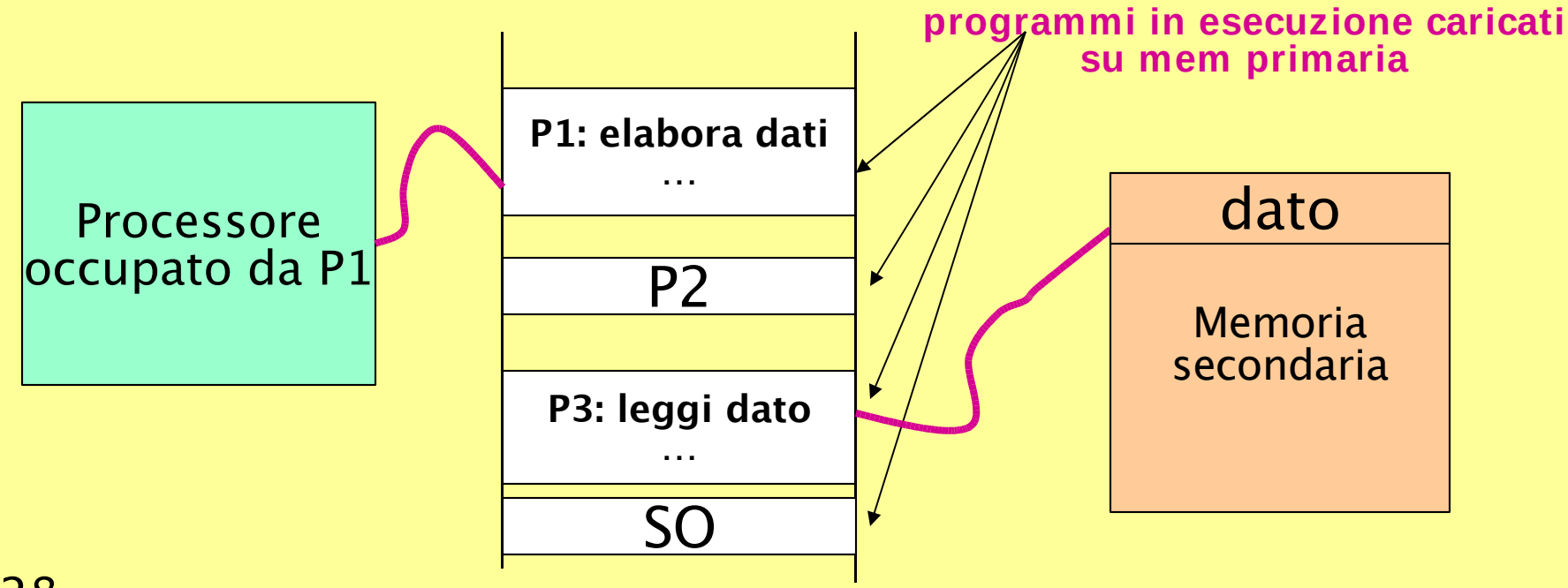
Multiprogrammazione e SO

- Aspetti critici trattati dal sistema operativo nella gestione dell'alernanta fra processi:
  - Interattività: tempi di risposta accettabili per l'utente
  - Protezione: bisogna evitare che durante l'esecuzione di più programmi in contemporanea ci siano interferenze e che un programma possa alterare l'altro
  - Temporizzazione: bisogna evitare che un solo programma monopolizzi l'intero sistema per un tempo eccessivo
  - Efficienza: sfruttare al meglio le risorse del sistema facendole utilizzare in parallelo da programmi diversi quando possibile:

Multiprogrammazione e SO

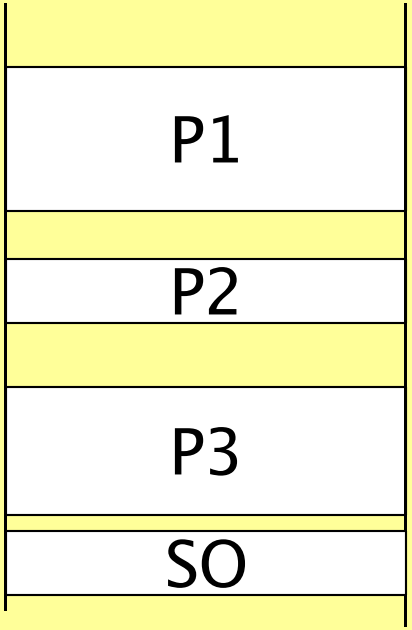
- Efficienza: sfruttare al meglio le risorse del sistema facendole utilizzare in parallelo da programmi diversi quando possibile:

Es. mentre un programma P1 occupa il processore per un attività di elaborazione, non posso fare usare il processore al programma P2 che richiede anch'esso di elaborare dei dati MA se c'è un programma P3 che in quel momento prevede un'operazione di lettura o scrittura in memoria secondaria -> la memoria è libera!



Gestione della memoria

- Al crescere della multiprogrammazione o della multiutenza, cresce il numero di programmi e dati che devono risiedere contemporaneamente in memoria -> ho bisogno di sempre più spazio in memoria primaria
- La memoria primaria ha dimensioni limitate (costosa a causa della tecnologia relativa alla velocità d'accesso); dim: fra i 128 e i 256 MB



Gestione della memoria

- Gestione del sistema operativo che fornisce diversi meccanismi che mi permettono di astrarre dalle dimensioni reali della memoria centrale -> creazione di una **memoria virtuale**
- Come? per es. riducendo la necessità di spazio tenendo in memoria solo una parte dei programmi/dati (quelli attualmente in uso):
- **Tecnica della paginazione e paginazione a richiesta:**

i programmi da eseguire sono suddivisi in “pagine ideali” tutte uguali poste in certe zone della memoria secondaria; le pagine vengono caricate in memoria primaria solo quando occorre

File System

- E' quella parte del SO destinato a gestire e strutturare le informazioni memorizzate sui supporti di memoria secondaria
- Operazioni di base supportate dal File System:
  - Recupero dei dati precedentemente memorizzati
  - Eliminazione dei dati obsoleti
  - Modifica/aggiornamento dei dati preesistenti
  - Copia dei dati (da disco rigido a dischetto, da un calcolatore all'altro...)

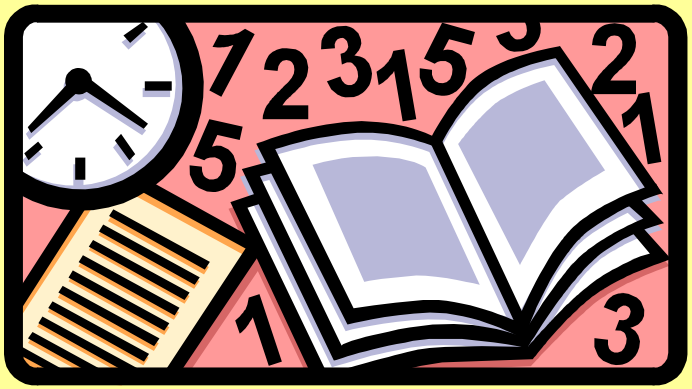
Dati in memoria: concetto di file

- I dati della memoria di massa vengono strutturati e gestiti in modo efficiente mediante la loro organizzazione in FILE

- Un file viene usato per memorizzare
    1. Programmi (utilizzati per avviare un'applicazione)
    2. Dati (documenti, immagini, suoni….)

Dati in memoria: concetto di file

- A livello di memorizzazione fisica possiamo vedere il disco fisso come un quaderno con tante pagine su cui un utente scrive delle relazioni (file)



- File: unità di memoria, agli occhi dell'utente; per facilitare la manipolazione dei dati si attua un meccanismo di strutturazione che permette di aggregare le informazioni elementari secondo una chiave logica e assegnare loro un nome unificante;

File system

- Il sistema operativo si occupa di registrare la posizione di tutti i file all'interno della memoria secondaria
- Inoltre deve fornire una visione astratta dei file su disco in modo che l'utente abbia la possibilità di
  - **identificare** ogni file con un nome astraendo dalla sua posizione nella memoria
  - avere un insieme di operazioni per lavorare sui file
  - **strutturare** i file, organizzandoli in sottoinsiemi secondo le loro caratteristiche, per avere una visione "ordinata" delle informazioni sul disco
  - effettuare l'accesso alle informazioni mediante operazioni ad alto livello, che non tengono conto del tipo di memorizzazione. Esempio: si deve accedere allo stesso modo ad un file memorizzato sul disco rigido oppure su un CD-ROM

Caratteristiche di un file

- Nome
- Dimensione (lunghezza)
  - Unità di misura: byte
- Data ultima modifica
- Attributi:
  - read only
  - hidden
  - system
  - archive

Identificazione di un file: nome

- Ogni file ha
  - un nome lungo fino ad un massimo di 255 caratteri (spazi compresi), con l'esclusione dei seguenti:

    / \ ? : * " < > |

    N.B. Per compatibilità con altri Sistemi Operativi è meglio non usare spazi nei nomi dei file -> usate _ (underscore)

  - un'estensione (3 caratteri)
    che identifica il **contenuto del file** e quindi anche **l'applicazione** con il quale può essere aperto (es. Il file pippo.txt è un file di testo e può essere aperto con un programma per gestione dei testi)

.exe | .doc | .xls | .txt |
.html | .avi | .gif | .bmp | .jpg

Identificazione di un file: nome

- Quindi un nome è composto da più parti separate da un punto: identificativo + estensione
  - es.
    - patti.doc,
    - Patti.doc,
    - v_patti.doc
    - graficiEsame.xls
- Standard DOS:
  - 8 + 3 caratteri
  - Ammessi solo caratteri alfanumerici
- Estensione facoltativa e non vincolante

Estensioni

- Esempi:
  - DOC — Word
  - XLS — Excel
  - PPT — PowerPoint
  - TXT, DAT — Testo
  - EXE,COM,DLL — Pr. Eseguibili
  - INI — file di install
  - PDF — Acrobat
  - MBD — Access
  - BMP — Bitmap
  - GIF, JPG, TIF — Immagini
  - AVI, MOV — Video
  - HTML — pag. Web
  - ZIP, Z — Zip file

Identificazione delle unità disco

- La memoria di massa di solito include più unità disco (hard-disk, floppy, etc) su cui i file possono essere situati -> per identificare il luogo dove sono situati i file in questa visione logica della memoria occorre dare a ogni disco del sistema un nome composto da una lettera seguita dai ":"
- A: l'unità floppy (RIPASSO: consente di salvare piccole quantità di dati su un supporto di memorizzazione esterno)
- C: hard-disk
- D: può essere il CD-ROM o una seconda unità disco
- F: G: H:

Operazioni sui file

- Un insieme di operazioni minimale, presente in tutti i sistemi:

- **creazione** di un file
- **cancellazione** di un file
- **copia** 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

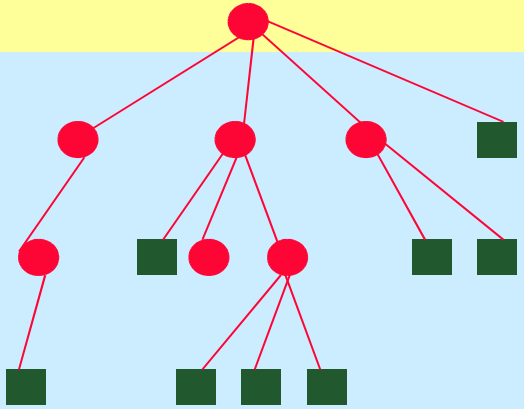
Organizzazione gerarchica dei file

- Il numero di file che devono essere memorizzati su un disco può essere estremamente elevato
- Si ha quindi la necessità di mantenere i file in una **forma ordinata**
- Un unico spazio (contenitore) di file è scomodo perché le operazioni di ricerca e di creazione di un nuovo file diventano onerose
  - Non è possibile avere due file con lo stesso nome
- L'idea: **raggruppare** i file in sottoinsiemi. Questi sottoinsiemi di file vengono memorizzatiall'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

Directory e file

- Ogni unità disco viene suddivisa in **cartelle** (**directory**)

Ad es. dall'unità C: partono una serie di cartelle (directory) e sotto cartelle organizzate in **maniera gerarchica** nelle quali sono salvate le applicazioni e i dati.



ALBERO

I file vengono collocati in cartelle proprio come i singoli fogli di carta vengono collocati negli archivi.

Directory e file

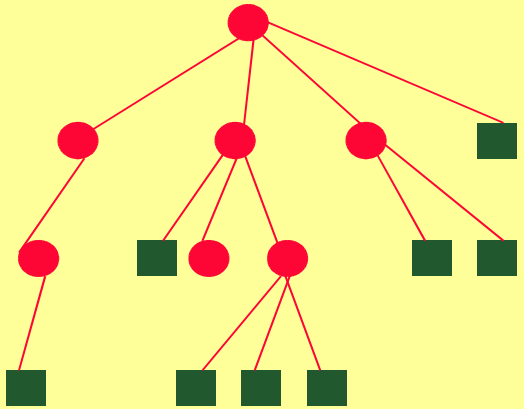
- Le directory costituiscono una **struttura ad albero**
- L'utente può spostarsi sui rami dell'albero

per decidere in che area immagazzinare l'informazione contenuta in un file o per cercare un file memorizzato in precedenza

- Una directory è una scatola

(punti rossi) che può contenere altre directory

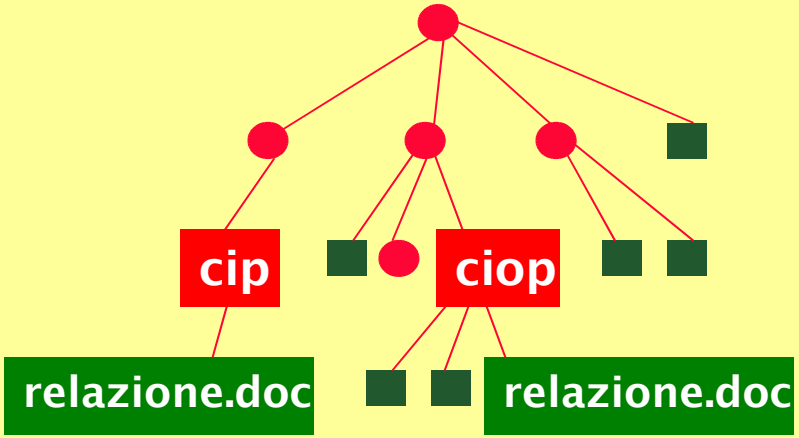
- file: foglie dell'albero (quadratini verdi)



ALBERO

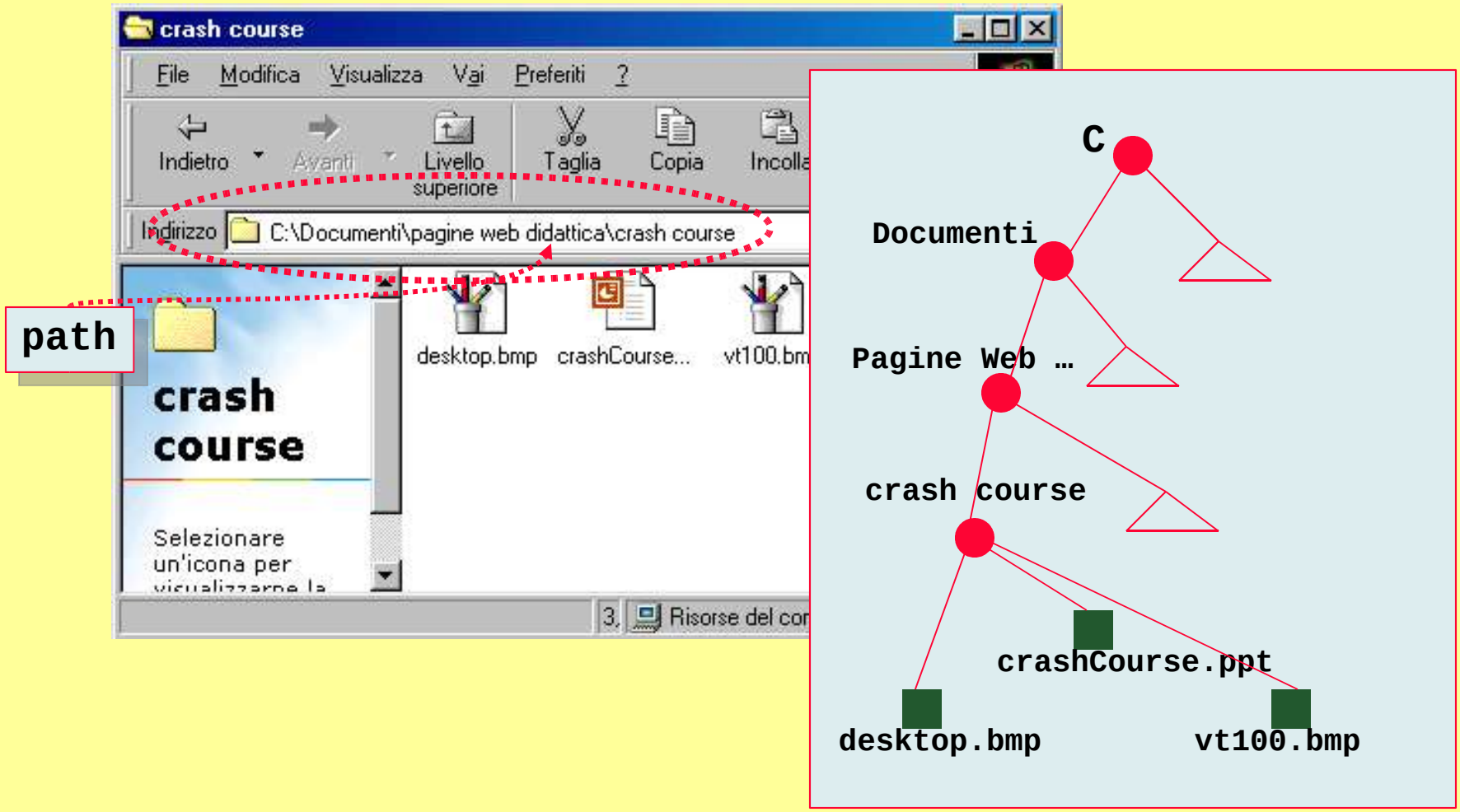
Individuazione di un file

- Se non esiste la strutturazione in directory, tutti i file possono essere identificati mediante il loro nome (univoco)
- 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

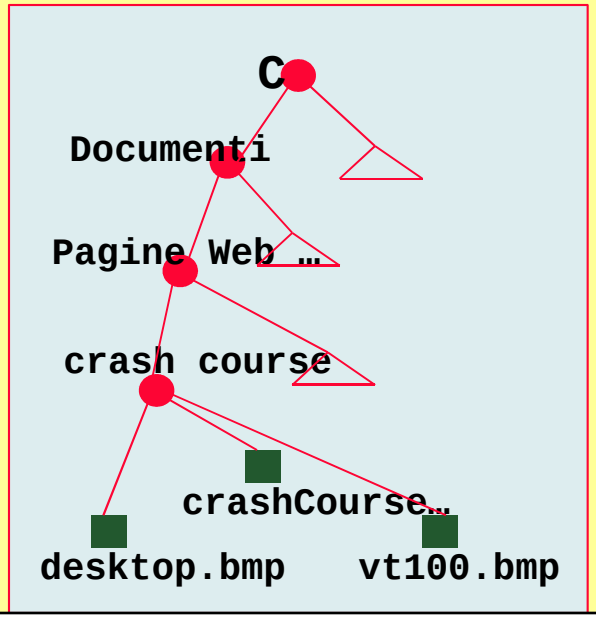


ALBERO

Path/directory



Path/directory



C: \Documenti\Pagine Web\crash course\crashcourse.ppt
…vediamo insieme altri esempi

Una sequenza di questo tipo può essere vista come il cammino che si deve compiere per raggiungere un determinato file a partire dalla radice dell'albero, ed è chiamata **pathname**

File System

Presenta all'utente l'organizzazione logica dei file e gestisce le operazioni che è possibile fare su di essi

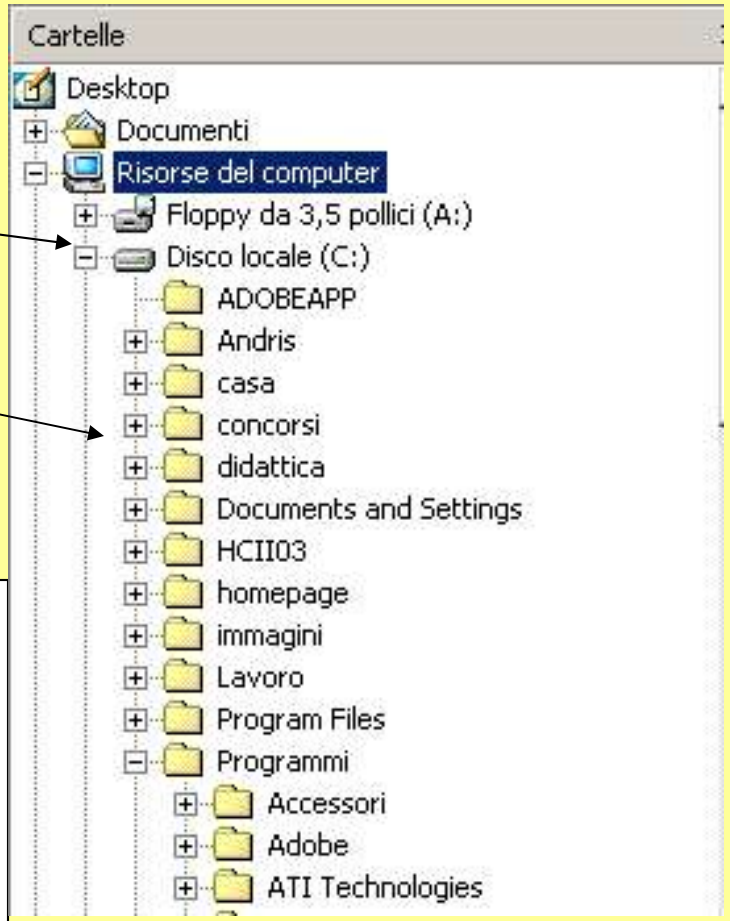
- **offre una vista** della memoria secondaria come insieme di unità logiche di memoria (file)
- **gestisce** la memoria secondaria vista come luogo di memorizzazione di insiemi di file

Gestione della gerarchia

- Es. di Windows:

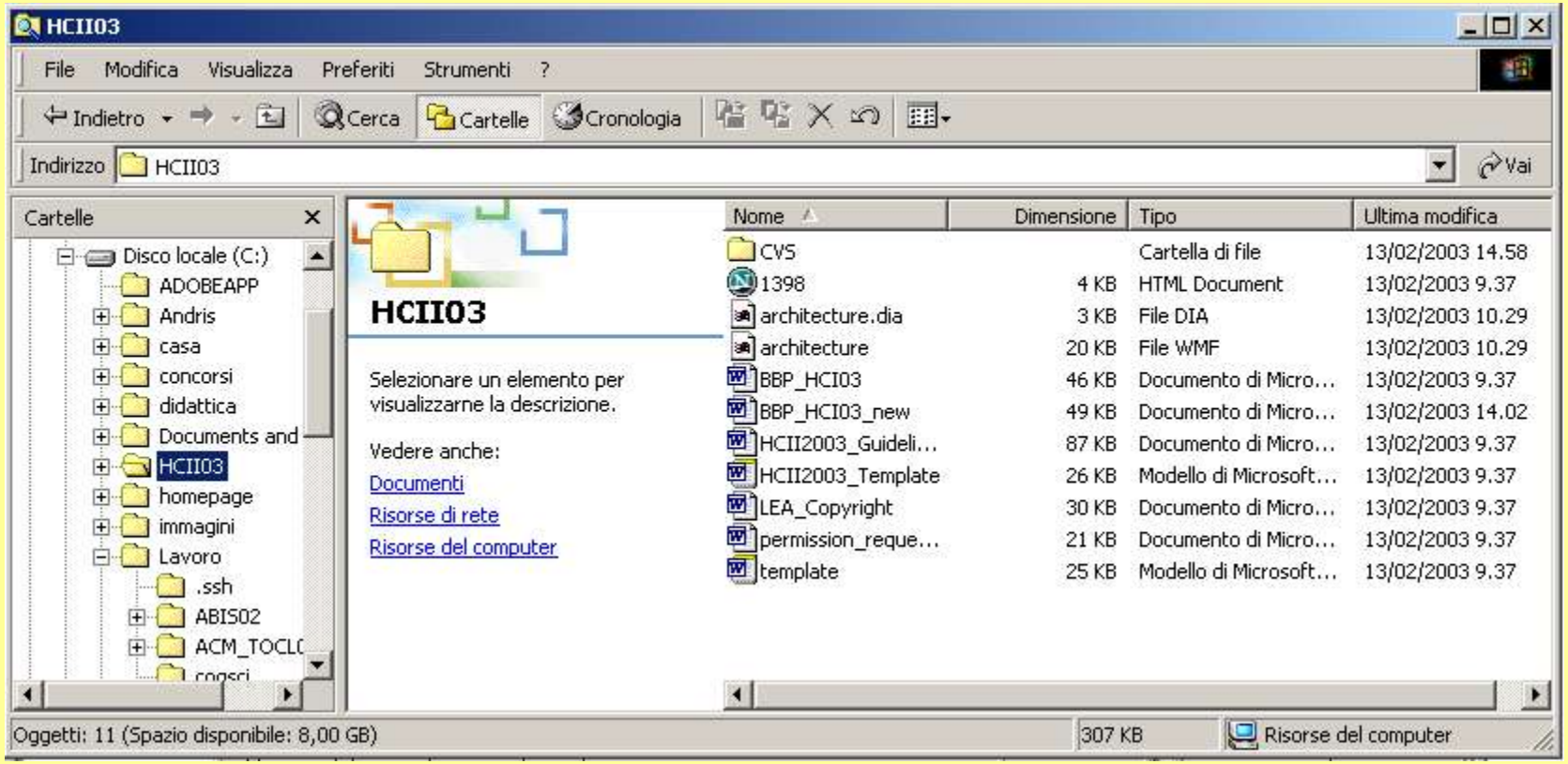
dischi

cartelle



Viene offerta una vista logica
sulle unità disco e sulle informazioni
in esse contenute che corrisponde
al concetto di organizzazione dell'utente

Gestione risorse



File system

Riassumendo...

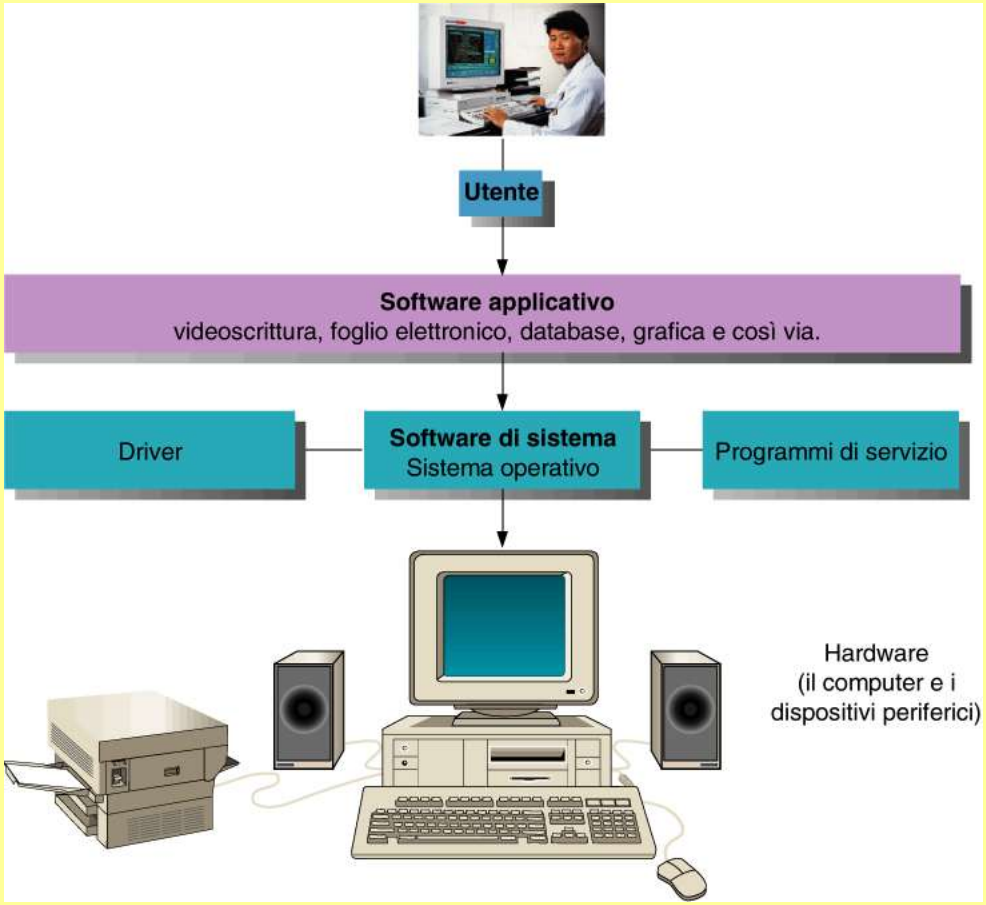
- Funzioni principali di quella parte di SO detta file system:
  - allocazione spazio fisico sulla memoria
  - gestione delle dinamiche di creazione, cancellazione, spostamento di file
  - gestione dell'organizzazione gerarchica dei file in cartelle

- Gestione di file system multipli (partizioni)
- Gestione di file system distribuiti (rete)

Software

- Riassumendo...
- Software di sistema o di base
  - Viene fornito a corredo dell'hardware
  - Programmi speciali per eseguire operazioni di base che determinano in generale il comportamento del computer e la sua facilità d'uso da parte di un utente che eventualmente non ne conosce la struttura fisica
  - Consente l'esecuzione del software applicativo
- Software applicativo
  - Programmi per svolgere operazioni specifiche: editare testi, creare fogli elettronici etc. -> lab: pacchetto Microsoft Office
  - Richiede la presenza del software di sistema

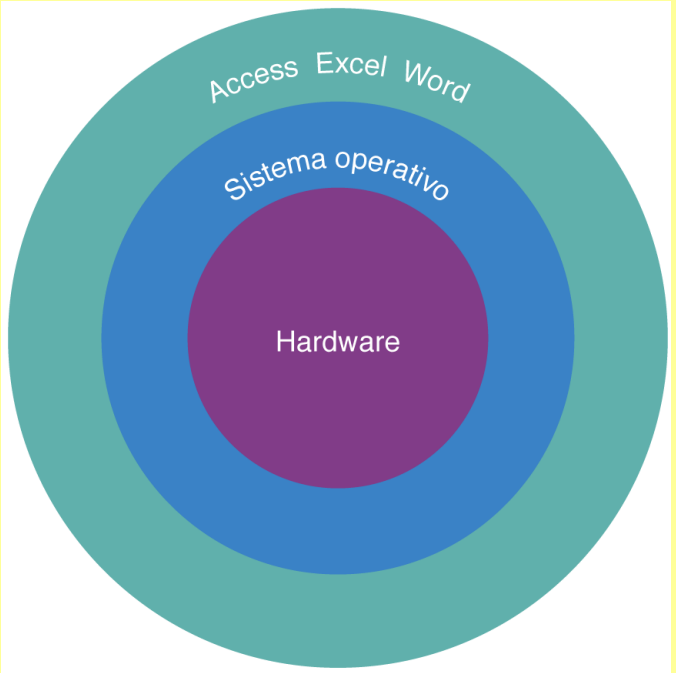
Una visione complessiva



Programmi applicativi e sistema operativo

Sistema Operativo

- indispensabile (senza il SO il computer non funziona)
- dal SO dipendono le funzioni uguali per tutti i programmi
- Esempi di sistemi operativi su PC: Windows 2000, Unix

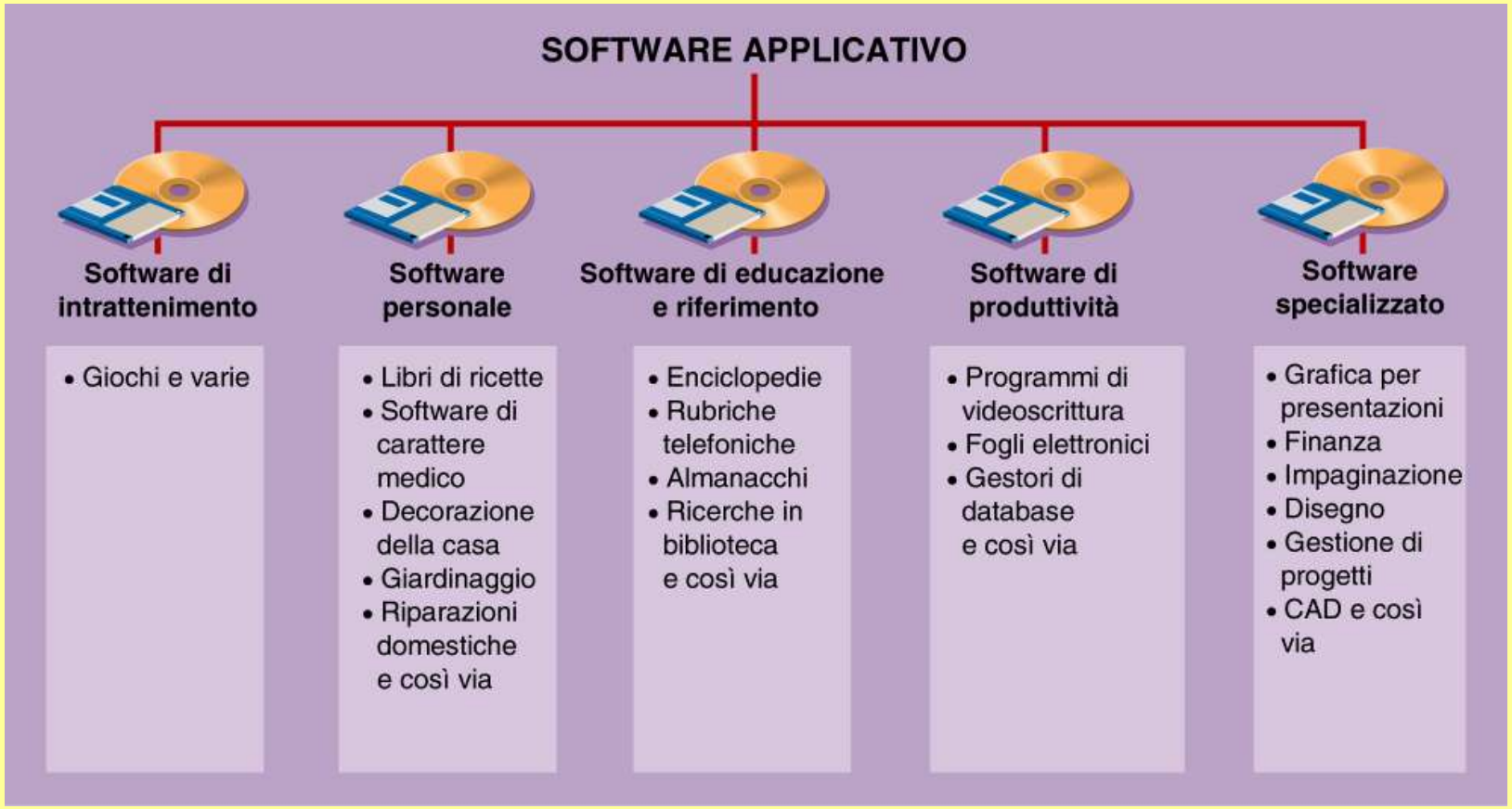


Programmi Applicativi

- opzionali
- necessari per funzioni specifiche (come scrivere, archiviare dati …)
- compatibili con il software di base
- Esempi: World, Excel, Access

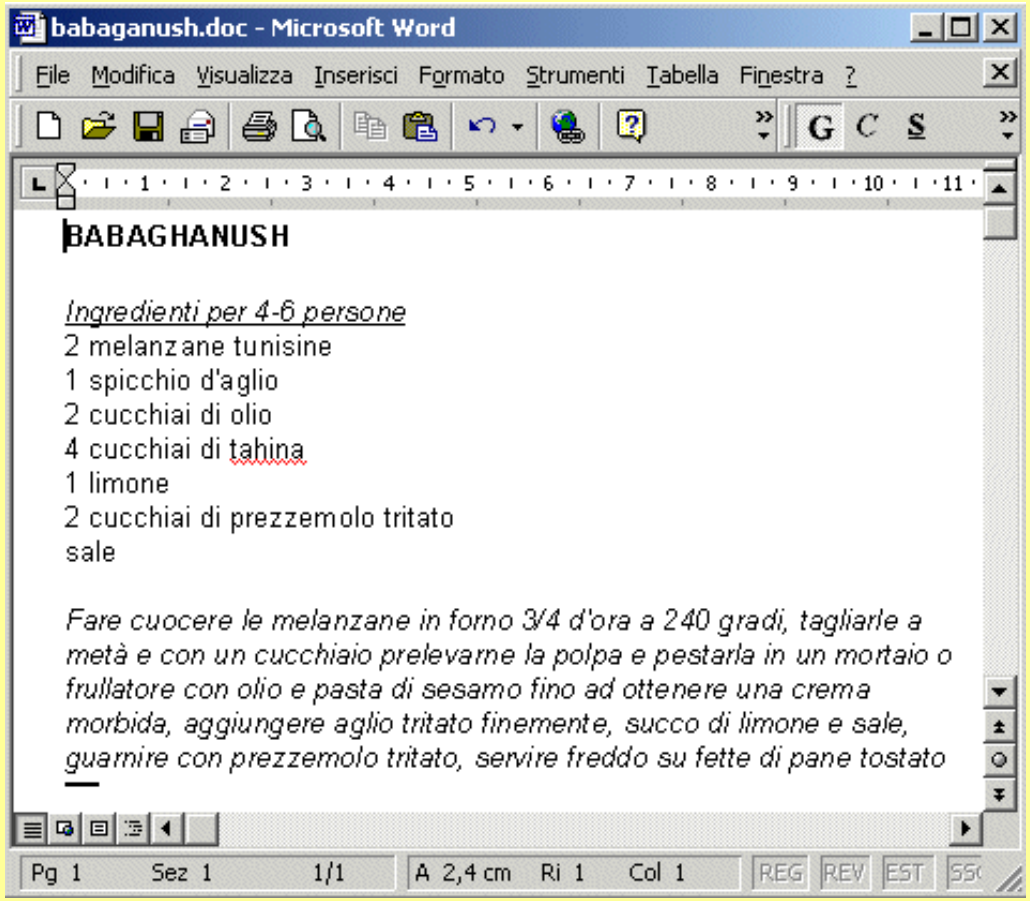
Software applicativo

- Classificazione in base all'uso



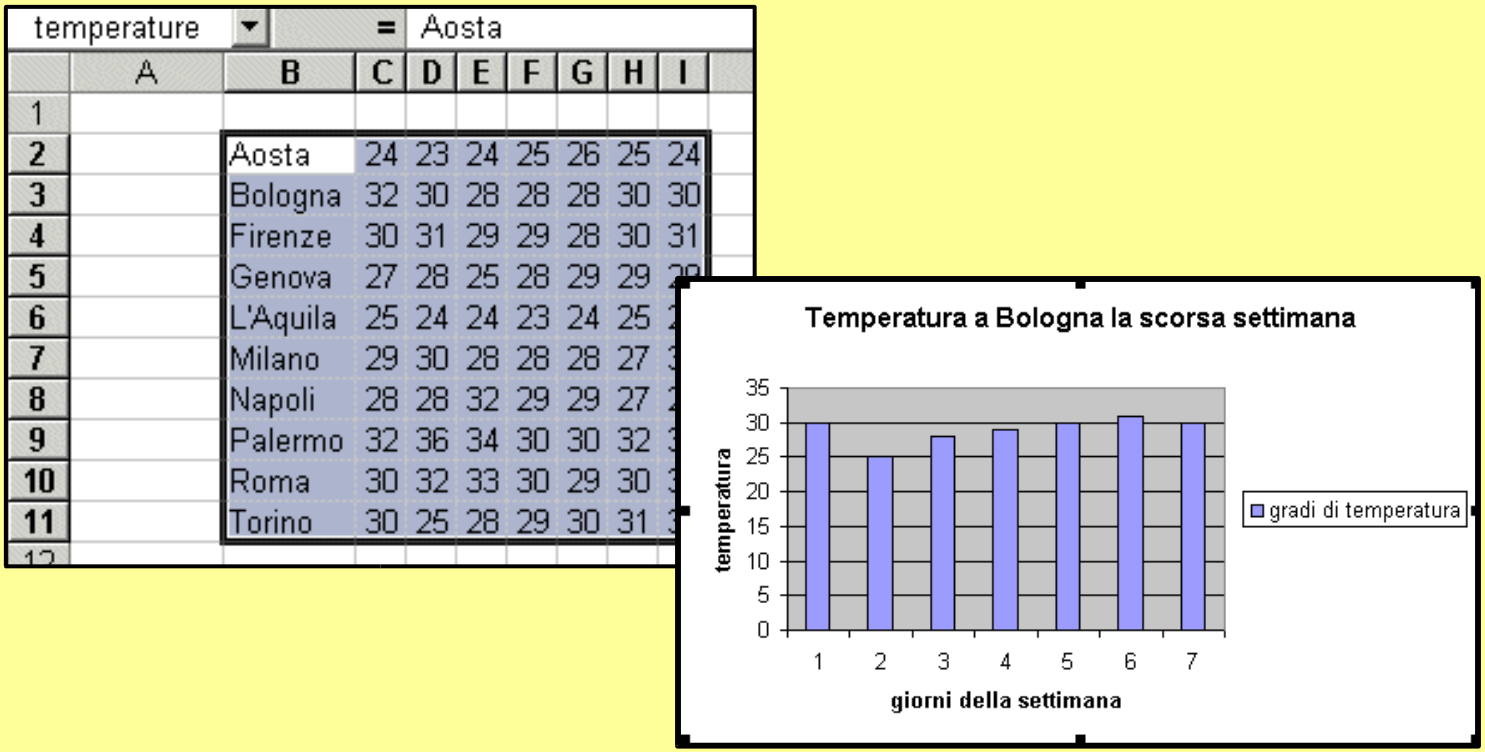
Software di produttività

- Videoscrittura



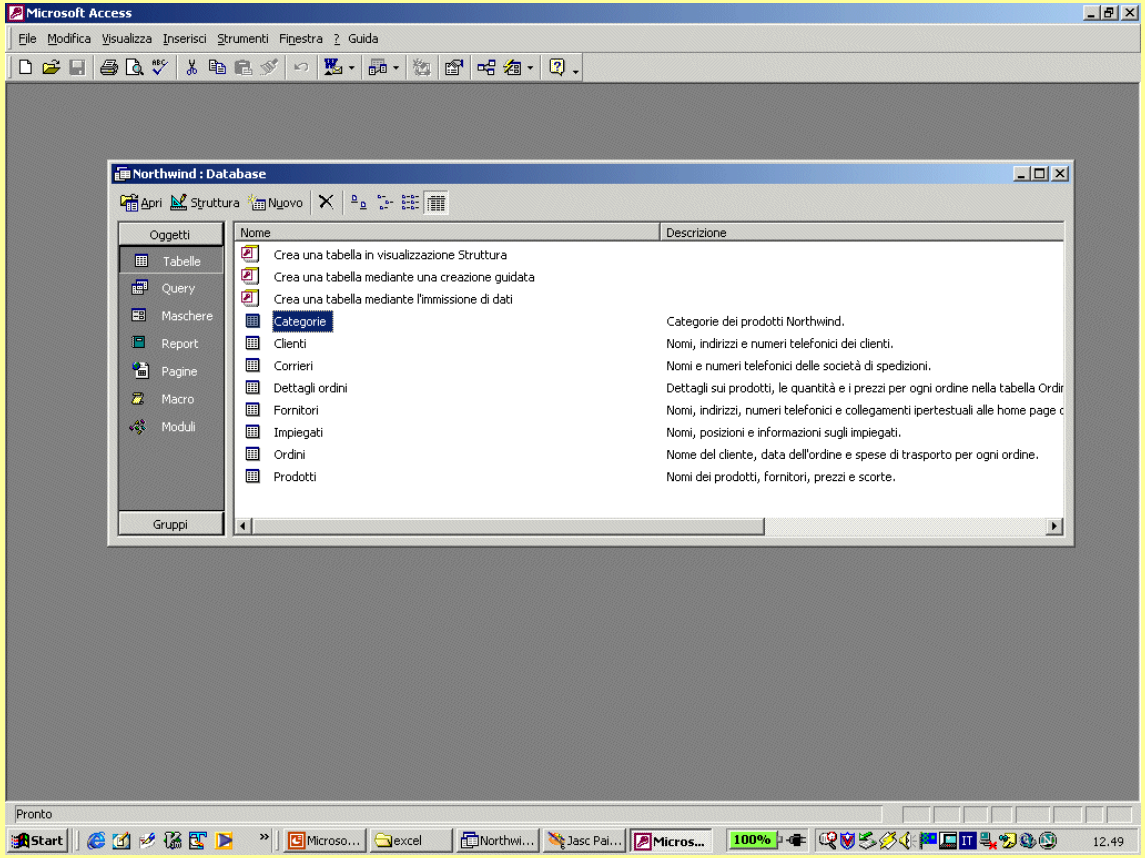
Software di produttività

- Fogli elettronici



Software di produttività

- Database



copyright vs copyleft

- Possibile classificazione interessante (del software in generale e quindi anche degli applicativi):
- **con copyright** o **software proprietario**: è coperto da diritti d'autore -> una volta venutone in possesso non posso copiarlo, **modificarlo**, rivenderlo…
  - Commerciale -> occorre acquistarlo
  - **Shareware** -> gratuito solo per un periodo di prova
  - **Freeware** -> disponibile gratuitamente
- **senza copyright o open source:** software di pubblico dominio non coperto da diritti di autore -> posso modificarlo come voglio, migliorarlo, adattarlo alle mie esigenze...

Software open-source: manifesto

- L'espressione "software libero" si riferisce alla libertà dell'utente di eseguire, copiare, distribuire, studiare, cambiare e migliorare il software. Più precisamente, esso si riferisce a quattro tipi di libertà per gli utenti del software:

1)Libertà di eseguire il programma, per qualsiasi scopo

2)Libertà di studiare come funziona il programma e adattarlo alle proprie necessità (libertà 1). L'accesso al codice sorgente (open-source) ne è un prerequisito.

3)Libertà di ridistribuire copie in modo da aiutare il prossimo

4)Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio. L'accesso al codice sorgente ne è un prerequisito.

(R. Stallman)

Software open-source

- Dalla lettura del manifesto emergono tre concetti fondamentali
  - Accesso al codice sorgente -> la traduzione letterale di s***oftware OPEN SOURCE è proprio software a "codice sorgente aperto"***
  - libertà di copia e ridistribuzione
  - libertà di modifica per adattamento alle proprie esigenze o per migliorare il programma in sé -> ***la disponibilità del codice sorgente, ossia del DNA del software, consente sia la sua libera circolazione, sia processi di modifica, produzione, ridistribuzione, evoluzione e riuso***

Software open-source

- Tipici fraintendimenti rispetto al software libero.
  - Il software libero è spesso gratuito MA non è detto che lo sia. Esistono programmatori che vengono retribuiti per aggiornare, modificare, adattare programmi liberi.
  - Il software gratuito non è per forza libero. Molte aziende divulgano software proprietario gratuitamente per vendere altri prodotti, per attirare nuovi clienti o per porre fuori gioco un concorrente.

L'esempio di Linux

http://www.linux.org/

- Linux è una particolare versione di Unix disponibile gratuitamente;
- Linux è un sistema operativo OpenSource: tutti vi possono accedere, tutti possono copiare programmi per modificarli o per eliminare eventuali malfunzionamenti o vulnerabilità;
- Linux è un software sviluppato in comunità: è il risultato di uno sforzo collettivo di migliaia di persone, non è prodotto da alcuna società commerciale e non si avvale di alcun reparto marketing e sviluppo che imponga delle "regole" ai propri utenti;