

# Laboratorio informatico

Flaminia Luccio

Dipartimento di Informatica  
via Torino 155, Mestre (Ve)

[luccio@unive.it](mailto:luccio@unive.it)

<http://www.dsi.unive.it/~luccio>

[http://www.dsi.unive.it/~luccio/DOTTOGIURBIS08/  
corsodotto.html](http://www.dsi.unive.it/~luccio/DOTTOGIURBIS08/corsodotto.html)

# Contenuti

- Concetti generali e terminologia
- I sistemi operativi: le funzionalita' di base di linux
- Pacchetto OpenOffice
- Latex

# Riferimenti

<http://gca.btbs.unimib.it/brandizi/mysit>

<http://www.di.unipi.it/~susanna/IG02/>

# Informatica

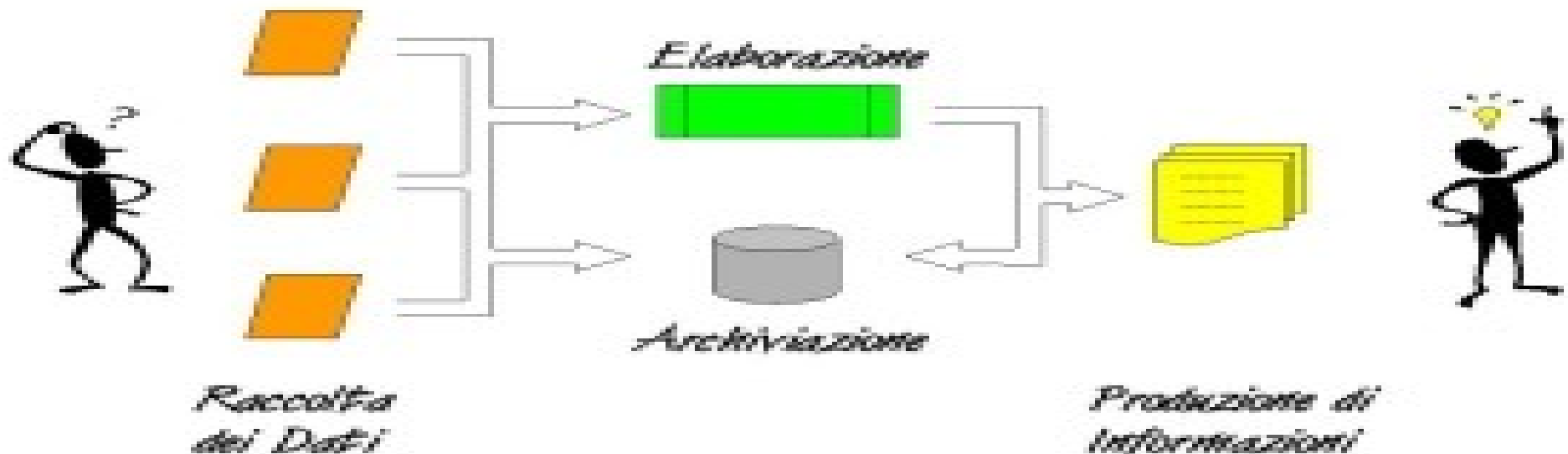
*information électronique ou automatique* =  
**informatica**

gestione automatica dell'informazione mediante un  
elaboratore elettronico (calcolatore)

# Informatica

L'informatica indica l'insieme dei processi e delle tecnologie che rendono possibile la **creazione**, la **raccolta**, l'**elaborazione**, l'**archiviazione** e la **produzione** o diffusione dell'informazione.

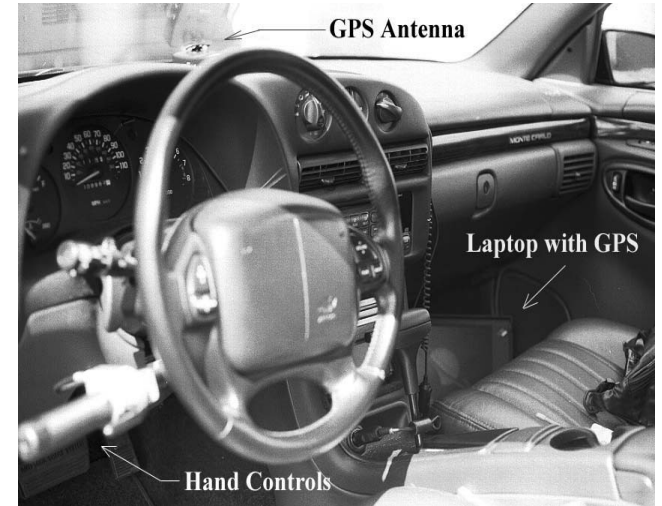
## Dai dati alle informazioni



# Informatica

- E' una disciplina scientifica che studia:
  - i calcolatori: che debbono eseguire “meccanicamente” operazioni ripetitive
  - gli algoritmi: che definiscono procedure “meccaniche” per la soluzione di problemi
  - le applicazioni: per sviluppare sistemi di supporto in varie aree

# I calcolatori o “computer”





# I calcolatori

- I calcolatori sono *macchine per elaborare l'informazione*
- Come funziona un calcolatore?
  - si inseriscono **dati** (INPUT)
  - i dati si elaborano tramite **programmi**
  - si restituiscono risultati (OUTPUT)

# Hardware vs Software

- *Hardware (HW)* Equipaggiamento “**fisico**” del calcolatore, costituito da componenti elettrici, meccanici ed elettronici, che fanno svolgere al calcolatore un insieme di funzioni generiche
- *Software (SW)* Equipaggiamento “**logico**” del calcolatore, costituito da programmi che utilizzano i componenti dell'hardware integrandoli al fine di eseguire compiti specifici.

# Software

- Vogliamo avere dei calcolatori **programmabili**, cioè in grado di eseguire funzioni diverse decise attraverso istruzioni fornite dagli utenti
- Vogliamo un meccanismo per **astrarre** dall'organizzazione fisica della macchina, l'utente non deve conoscere il “linguaggio macchina”
- Vogliamo utilizzare senza problemi macchine con HW diverso
- Vogliamo utilizzare un semplice linguaggio di interazione con la macchina; utilizzare programmi applicativi diversi...

# Software

- Questo si raggiunge definendo una macchina virtuale sopra l'HW
- struttura a cipolla, diversi livelli di astrazione

# Il Software

- Software si distingue in quello di **sistema** e quello **applicativo**:
  - di **sistema** per la gestione delle risorse del sistema (es SO); per il supporto nella stesura di programmi applicativi; per la comunicazione con l'utente
  - applicativo** programmi destinati all'utente finale: di videoscrittura, fogli elettronici, posta elettronica, browser, ...

# Il Software

- Componente “algoritmica”
- L'**algoritmo** e' codificato con un “programma” in un particolare linguaggio di programmazione (Pascal, C, C++, Java,...)

# Gli algoritmi

- Intuitivamente un algoritmo e' una procedura per risolvere un problema in un numero finito di passi
- Piu' formalmente: un algoritmo e' un insieme ordinato di passi che descrivono i **dati** che devono essere utilizzati e determinano la sequenza di **azioni elementari** che devono essere svolte per risolvere un problema
- Un algoritmo deve essere: preciso e non ambiguo

# Esempio di algoritmo

- Problema: preparazione della besciamella
- Il “cuoco-computer” parte da degli *INPUT*, cioè i dati (ingredienti), li elabora con delle *RISORSE* (pentole, fornelli) e produce un *OUTPUT* (besciamella)





# Esempio di algoritmo

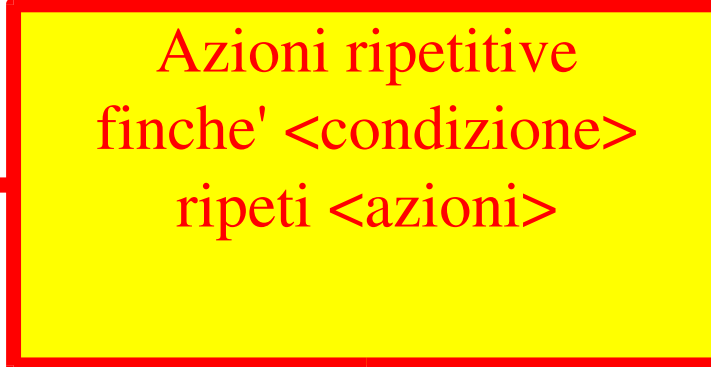
## Besciamella

### Ingredienti e strumenti (dati):

- \* 1/2 etto di burro, 1/2 etto di farina, 1/2 litro di latte, 1/2 cucchiaino di sale, 5 grammi di pepe e noce moscata
- \* 1 casseruola, 1 setaccio, 1 coltello, 1 cucchiaio, 1 fornello

### Preparazione (azioni):

- \* Setaccia la farina
- \* metti la farina nella casseruola con 50 cc di latte freddo
- \* **se** il burro e' in un pezzo unico **allora** taglialo a pezzetti
- \* aggiungi alla casseruola il burro e 300 cc di latte
- \* accendi il fuoco
- \* **finche'** non bolle **ripeti** mescola
- \* **se** vuoi piccante
  - allora finche'** non piccanti **ripeti** aggiungi pepe
  - altrimenti** aggiungi noce moscata
- \* aggiungi 150 cc di latte
- \* cuoci per 20 minuti



# Programmi

- Un **programma** e' la **codifica** di un algoritmo in un particolare linguaggio comprensibile al calcolatore
  - "implementa", cioe' realizza l'algoritmo
  - specifica aspetti quale l'insieme dei dati in input e la visualizzazione dei risultati (output)
  - puo' dover tenere conto dell'architettura (HW) della macchina

# Rappresentazione dell'informazione

- Come si rappresenta l'informazione in un calcolatore?
  - con la notazione **binaria** o **digitale**, un linguaggio di 0 e 1
- L'entita' minima dell'informazione e' il **bit** (*binary digit*) 0 o 1
- 1 **Byte** = 8 bit. Il Byte e' l'unita' di misura della memoria

# Rappresentazione dell'informazione

- Il calcolatore capisce solo successioni di **bit**. I circuiti elettronici (come lo scaldabagno o la lampadina) possono essere solo aperti o chiusi.

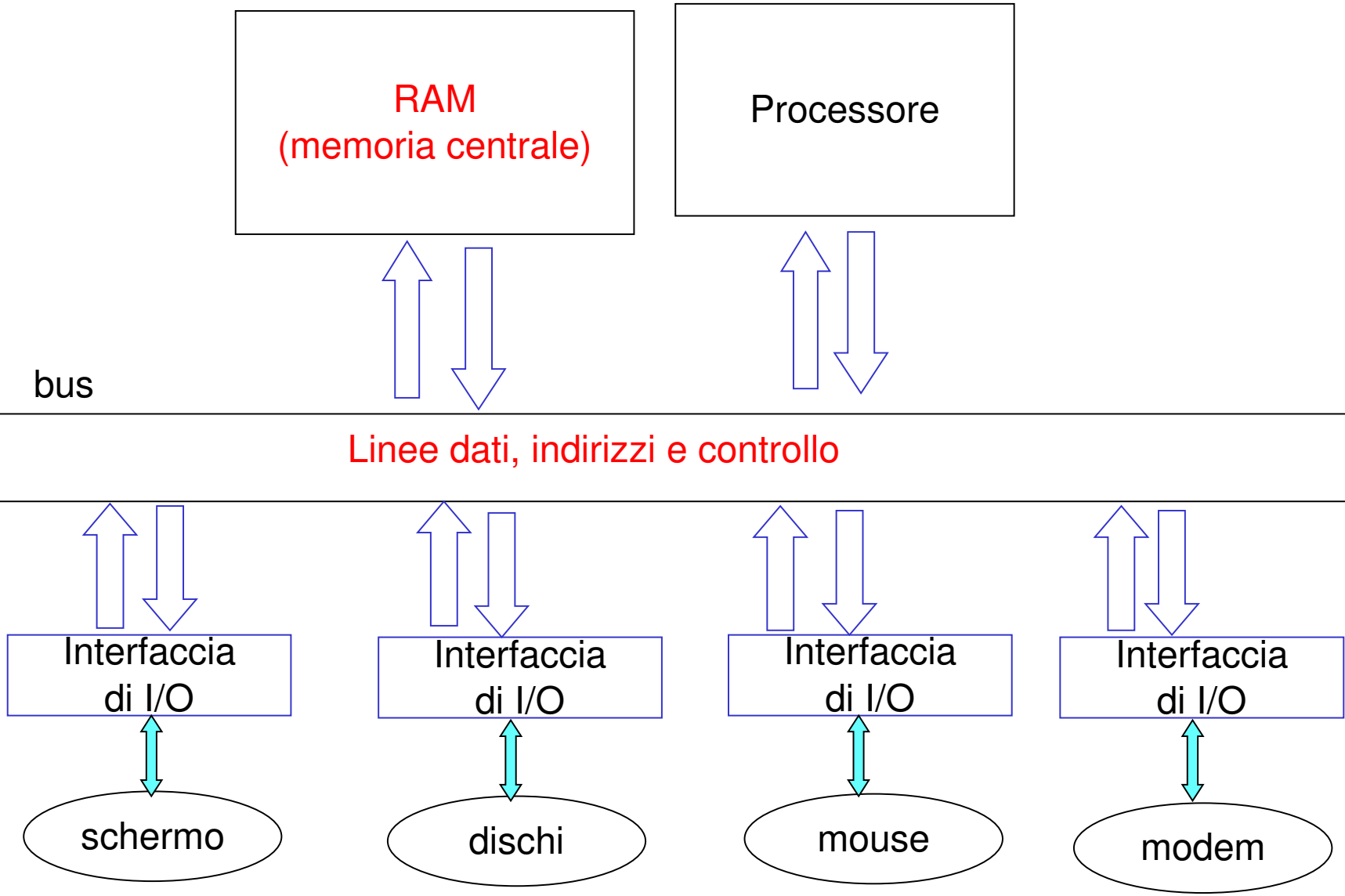
## La numerazione binaria

- E' possibile rappresentare i numeri e le loro operazioni adottando la *numerazione binaria*
- La numerazione binaria è un calcolo che fa uso solo di due cifre: 0 e 1 invece delle dieci che si usano nella numerazione decimale

## Diverse grandezze

<b>Grandezza</b>	<b>Valore</b>	<b>Circa</b>
1 KiloByte (KB)	$2^{10} = 1024$ Byte	$\approx 10^3$
1 Mega (MB)	$2^{20} = 1024^2 = 1024$ KB	$\approx 10^6$
1 Giga (GB)	$2^{30} = 1024^3 = 1024$ MB	$\approx 10^9$
1 Tera (TB)	$2^{40} = 1024^4 = 1024$ GB	$\approx 10^{12}$
1 Peta (PB)	$2^{50} = 1024^5 = 1024$ TB	$\approx 10^{15}$

# Struttura di un calcolatore



# Operazioni elementari eseguibili da un processore

- Dati numerici (binari) :
  - operazioni matematiche
  - confronti con 0
  - operazioni logiche o booleane: AND, OR, XOR, NOT
- Dati numerici e non
  - letture/scritture nella memoria, letture e scritture nei registri di controllo dei dispositivi
  - uguaglianza e disuguaglianza di due rappresentazioni



# Operazioni elementari eseguibili da un processore

- Operazioni di ‘controllo di flusso’ :
  - salti condizionati (corrispondono alla combinazione di istruzioni di controllo *se ( ... ) vai al passo X* )
  - salti incondizionati (corrispondono alle istruzioni di controllo *vai al passo X* )
  - chiamate di un sottoprogramma (CALL)
  - terminazione del programma (RETURN)

# Operazioni elementari eseguibili da un processore

- Esistono sul mercato processori ‘compatibili’ cioè in grado di eseguire lo stesso insieme di istruzioni (facilita la portabilità dei programmi in linguaggio macchina)
  - es : Intel Pentium, IBM-Cyrix, AMD

# Linguaggi

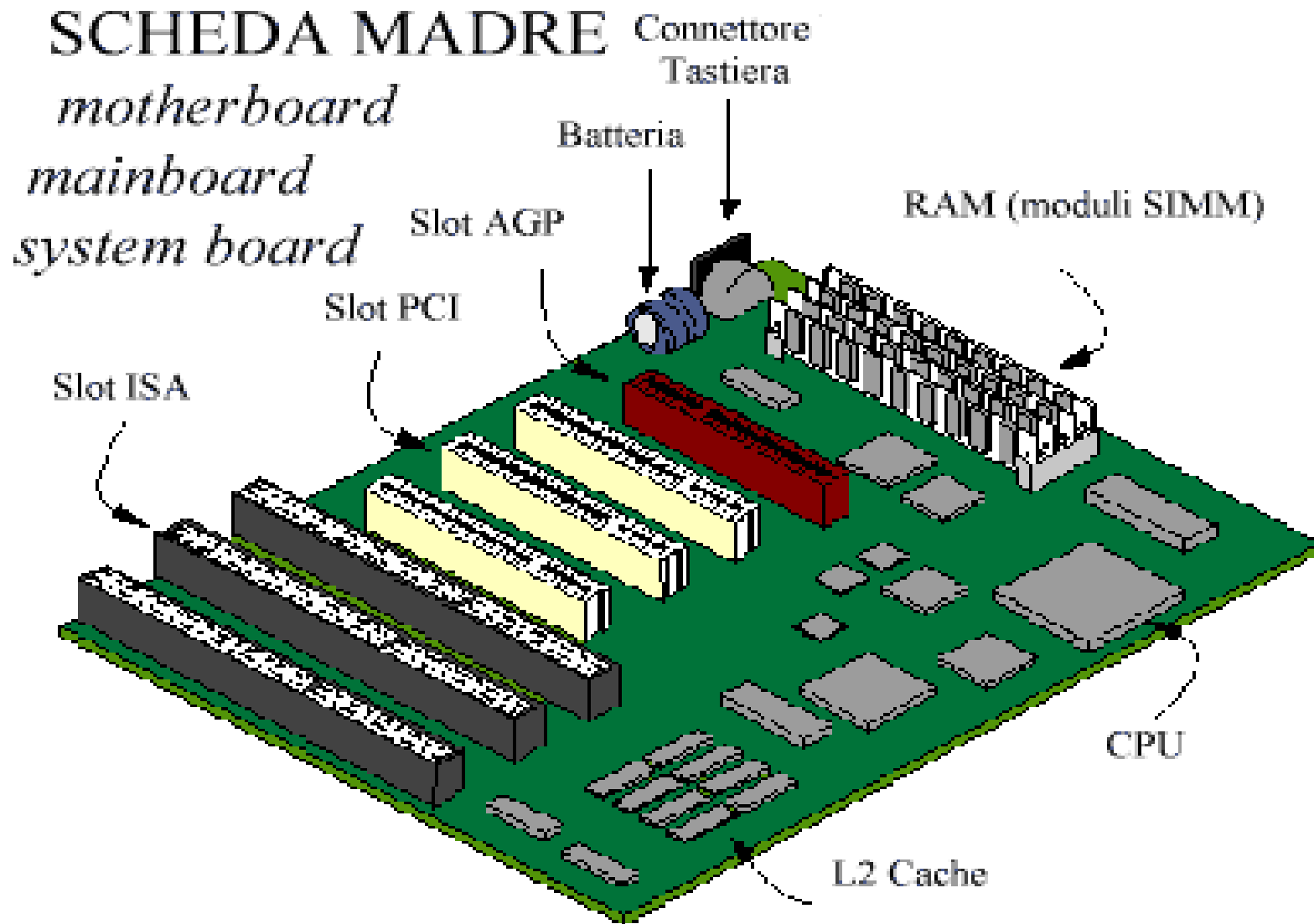
- Linguaggio macchina :  
linguaggio comprensibile direttamente dal processore della macchina (binario)
- Assembler :  
versione simbolica del linguaggio macchina in cui i nomi delle operazioni e degli operandi sono indicati con codici simbolici
- Linguaggi ad alto livello : Pascal, C, C++, Java, Fortran, ...

# Le memorie

- I calcolatori hanno supporti di memorizzazione di più tipi :
  - *una memoria centrale, tipo RAM* : contiene i programmi durante la loro esecuzione ed i dati relativi. E' una memoria relativamente piccola (qualche centinaio di MB) e veloce, volatile, a costo medio alto.
  - *una o più memorie di massa* (dischi, nastri magnetici, CD ROM, ecc.): mantengono tutti i dati ed i programmi in attesa di essere eseguiti in modo persistente. Sono memorie grandi (decine di GB), persistenti e lente.
  - *Una o piu' memorie cache*: piccole (circa 1 MB), volatili e veloci.

# I PC moderni

- Scheda madre (motherboard).

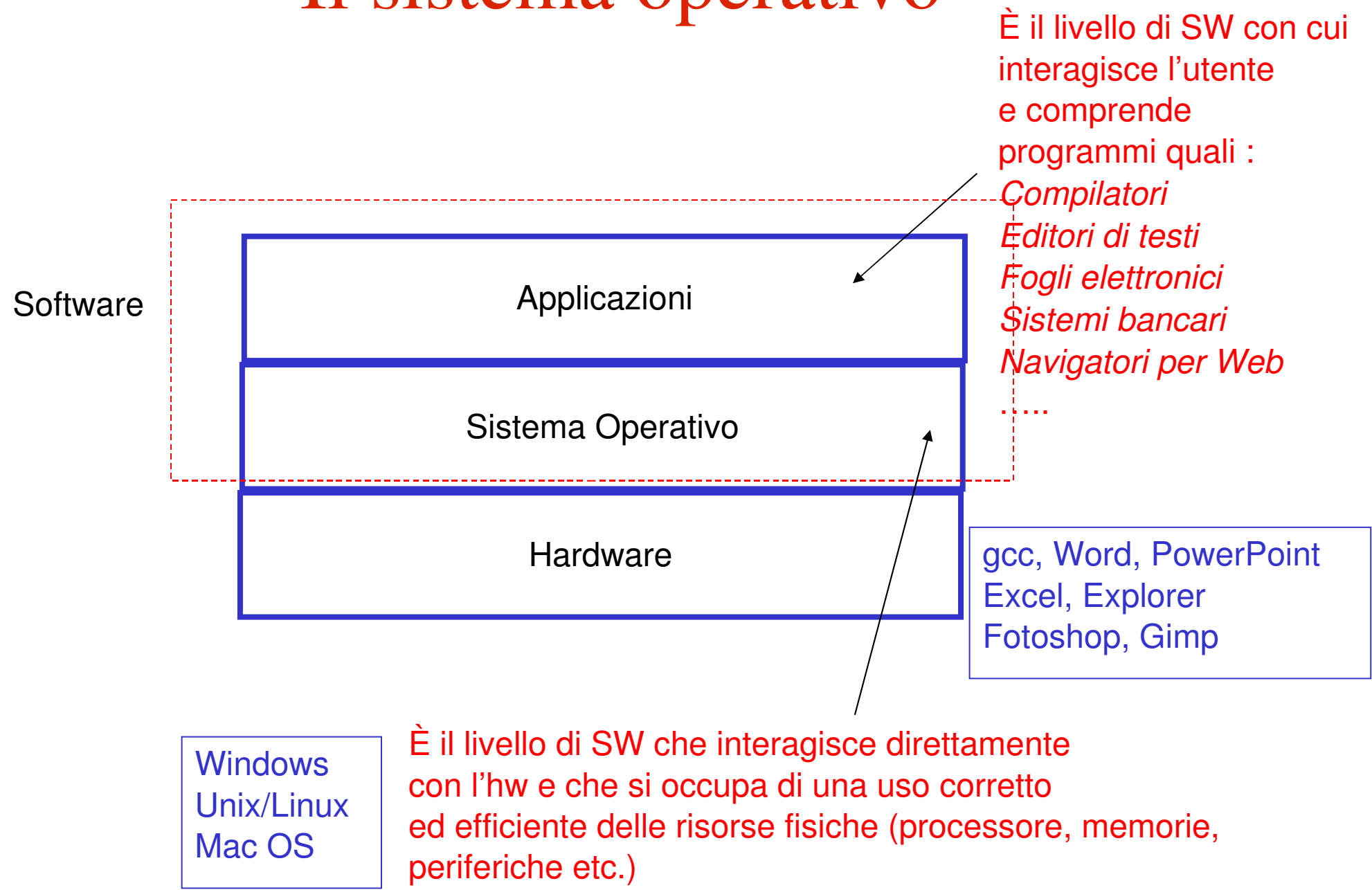


# I PC moderni

- Connessione alle periferiche



# Il sistema operativo



# Quali sono le funzioni di un SO ?

- Esegue applicazioni
- Facilita l'accesso ai dispositivi di I/O (modem, reti, video, ecc.)
- Archivia dati e programmi :
  - mette a disposizione dell'utente una visione astratta della memoria secondaria (il **file system** basato sulle astrazioni: *file/archivi e folder/cartelle*).



# Quali sono le funzioni di un SO?

- Gestisce le risorse
  - ripartisce le risorse disponibili fra le varie applicazioni (processore, RAM, periferiche)
  - evita che ci siano malfunzionamenti dovuti all'uso contemporaneo di risorse
  - ottimizza le prestazioni
- Gestisce malfunzionamenti del sistema
  - rileva e gestisce situazioni anomale

# Quali sono i diversi SO?

Esistono vari tipi di SO:

- **Microsoft Windows**: molto diffuso.
- **Linux+Unix**: SO freeware (gratuito) e “open-source” (SW di cui è reso pubblico il codice sorgente, che può quindi essere modificato).  
Grandi potenzialità gestionali, difficile da usare.
- **Mac OS X**: SO sviluppato dalla Apple per i calcolatori Macintosh.

# Accesso

- Ogni utente, indipendentemente dal sistema ha un
- *Account*, cioè uno spazio personale su disco. Il nome dell'account è in genere anche la *login* per accedere.
- Una *password* (chiave segreta).

# File System

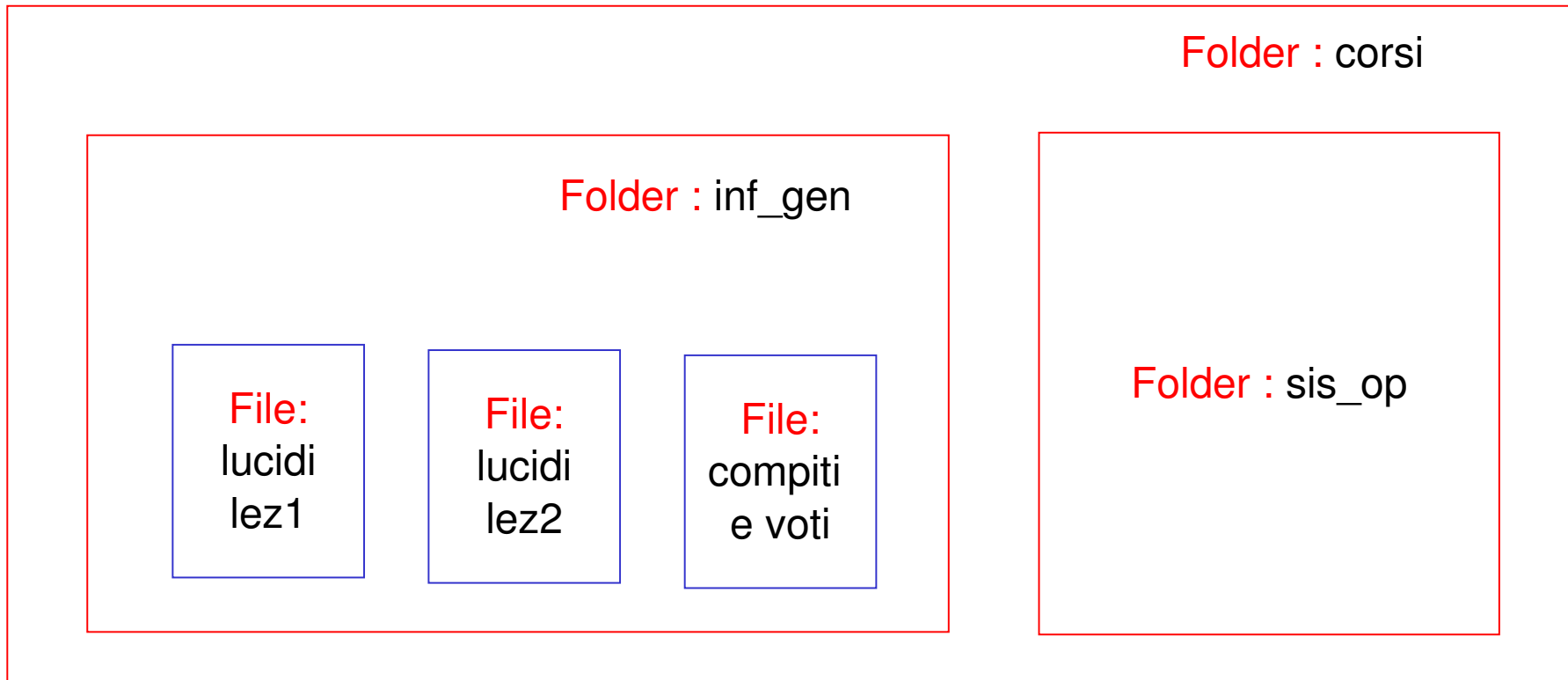
- È la parte del SO che
  - permette di memorizzare dati e programmi in modo persistente
  - permette di organizzare dati e programmi in modo da renderne agevole la localizzazione da parte dell'utente umano
  - può essere modificata per cancellare dati obsoleti, aggiornare l'organizzazione etc ..

# File System

- Le astrazioni fornite sono :
  - il file (o archivio): una sequenza di byte che contiene dati o programmi rappresentati con una opportuna codifica binaria
  - il folder/directory (o cartella) : astrazione che permette di collezionare insieme più file e/o folder

# File System

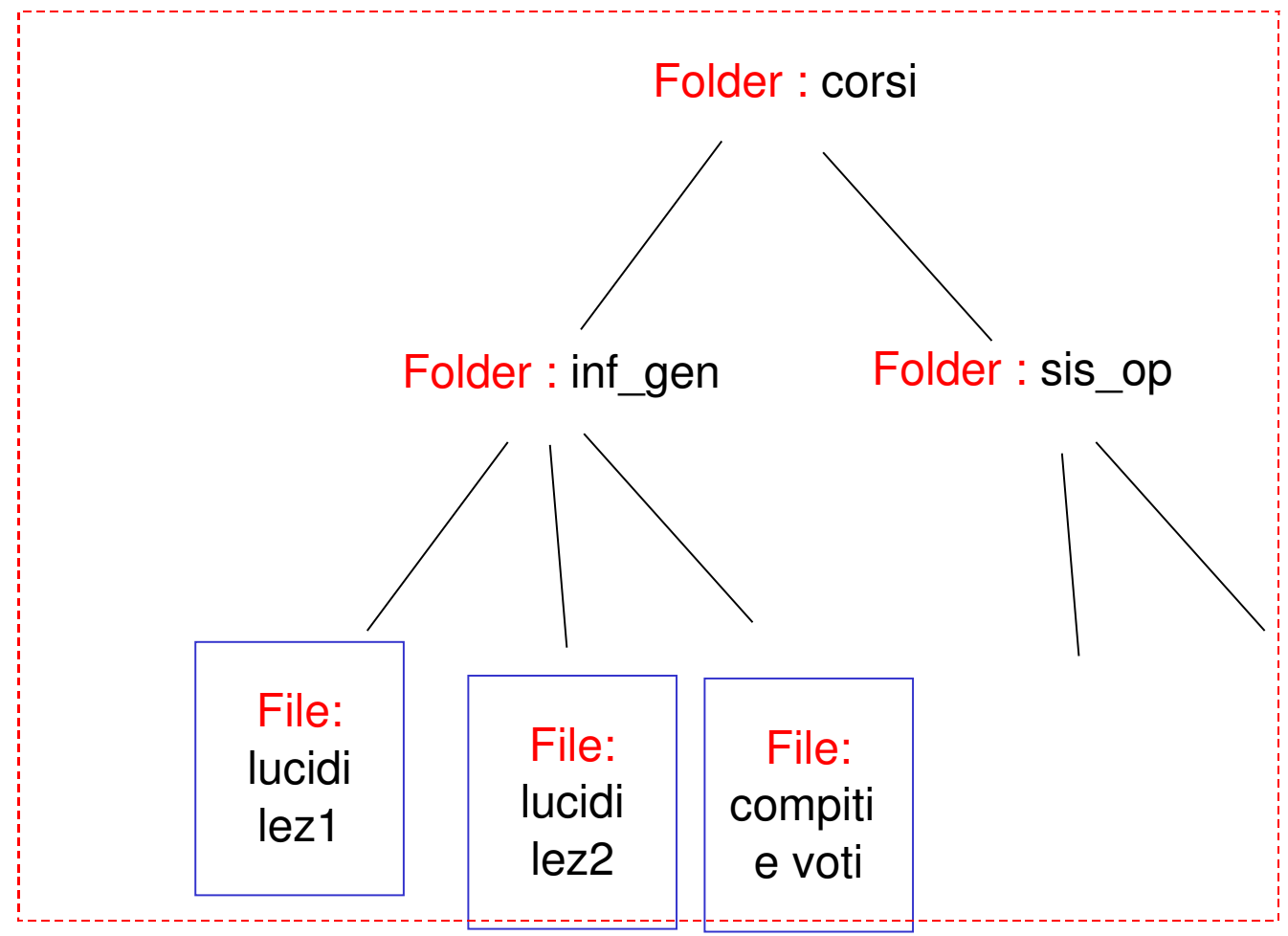
- Esempio : una possibile organizzazione di file



# File System

- La struttura è gerarchica (ad *albero*)

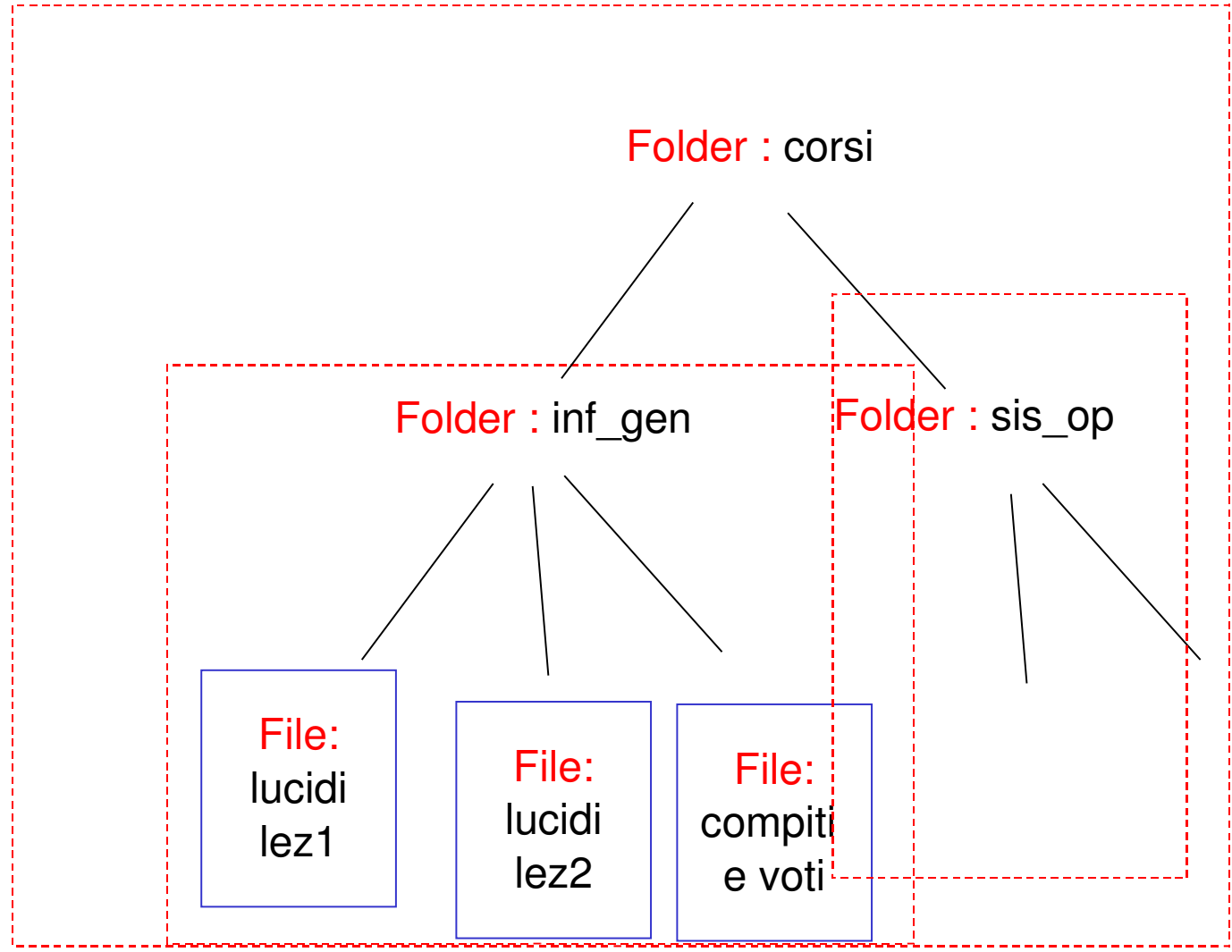
Contenuto del folder 'corsi'



# File System

- La struttura è gerarchica (ad *albero*)

Contenuti dei  
due folder  
*inf\_gen* e *sis\_op*





# File

- Come si crea un file
  - utilizzando una applicazione
    - editor di testi per creare file di testo secondo diversi formati (doc,html,text) es. Word
    - applicazioni per gestire grandi quantità di dati persistenti (*basi di dati*) es. Access
    - fogli elettronici es. Excel
  - dando comandi diretti al SO
    - es. chiedendo di duplicare un file esistente (taglia e incolla ...) via shell o interfaccia grafica

# Comandi Unix di base

- Aprire un terminale
- *ls* elenca il file e le directory all'interno della directory corrente, eccetto i file che iniziano con "."
- *cd pathname* ci si sposta dalla directory corrente (inizialmente la home directory) alla directory definita dal pathname (se esiste).
- *cd* da qualunque punto si torna alla home directory.
- *cd ..* si risale al nodo padre nell'albero.
- *pwd* stampa il nome della directory corrente;

# File

- Nome di un file
  - ogni file ha un nome (es *lez1.doc*, *lez2.ppt* etc)
  - il nome deve essere unico nel folder che contiene il file
  - la parte dopo il punto (.) si chiama ‘estensione’

di solito l’estensione fornisce informazioni sul formato del file e/o sull’applicazione utilizzata per crearlo

*lez2.ppt* estensione

Guardare con ls

# File

- Nome di un file
  - alcuni SO (come Windows) non mostrano normalmente l'estensione
    - *lez1.doc* viene mostrato come *lez1*
  - tuttavia l'estensione è presente e serve al sistema per capire che tipo di file è e quale applicazione attivare quando desideriamo 'leggerlo'
    - es. un doppio click su un file lez1 attiva automaticamente Word per visualizzarlo

# Folder

- Anche i folder sono identificati univocamente da nomi
  - es *inf\_gen*, *corsi*
  - non usano estensioni
    - contengono solo informazioni sui file/folder contenuti

# Folder

- Per creare un folder basta invocare una opportuna funzione del SO
  - es. scegliere una opportuna opzione (*new folder, nuova cartella*) in un menu dell'interfaccia grafica
- Per cancellare file/folder basta invocare una opportuna funzione del SO
  - es. trascinarli con il mouse fino al cestino
- I file/folder possono essere spostati da un folder all'altro

# Comandi Unix di base

- *mkdir* *NomeDirectory* crea una nuova directory di nome *NomeDirectory* nella directory corrente;
- *rmdir* *NomeDirectory* rimuove la directory di nome *NomeDirectory*: prima di farlo è necessario cancellare tutti i file contenuti;
- *mv* *Nomefile NuovaDirectory* sposta *Nomefile* nella directory *NuovaDirectory* cancellando eventualmente il vecchio *NuovoNomefile* se esisteva già

## Comandi Unix di base

- *cp Nomefile NuovoNomefile* crea un nuovo file NuovoNomefile che ha lo stesso contenuto di Nomefile. Se il file NuovoNomefile esisteva già verrà sovrascritto;
- *cp Nomefile NuovaDirectory/NuovoNomefile* copia Nomefile nel file NuovoNomefile della directory NuovaDirectory cancellando eventualmente il vecchio Nomefile se esisteva già;



# Comandi Unix di base

- *rm -i Nomefile* cancella il file Nomefile; l'opzione -i, che può essere omessa, serve per avere una richiesta esplicita della cancellazione dal sistema;
- *rm \** cancella tutti i file della directory corrente;
- *rmdir NomeDirectory* cancella la directory NomeDirectory;

# Sicurezza

- Come evitare che i nostri file siano letti da altri utenti che hanno accesso al sistema?
  - Assegnare ad ogni utente : *login* (nome unico identificativo) e *password* (segreta, abbastanza complessa!)
  - Ogni file appartiene ad un utente
  - Il proprietario può specificare se un file può essere letto o scritto anche da altri
  - Il file system al momento dell'accesso controlla che l'operazione sia effettivamente consentita

# Sicurezza

- Il *superutente* (root, administrator)
  - E' l'account dell'amministratore del sistema
  - Può accedere in maniera non ristretta dappertutto
  - I file del sistema operativo sono generalmente dell'utente root e non sono accessibili da nessun altro utente
  - ..... ovviamente il superutente è oggetto di numerosi tentativi di violazione
    - termini come *attack*, *hacker* si riferiscono a questo

# Sicurezza

- I vecchi sistemi Windows (3.1,95,98) non hanno alcun meccanismo di protezione
  - tutto gira in stato kernel
  - da cui la facilità di danneggiare il sistema operativo o i file utente (es. con virus)
  - molto pericoloso in connessioni Internet non sporadiche
- Windows NT, 2000, XP sono protetti
  - è necessario però creare utenti non privilegiati
- Sistemi Unix/Linux protetti fino dagli anni 70

# Comandi Unix di base

- *password* comando per modificare la password corrente. Compare "Old Password" e bisogna digitare quella vecchia, poi compare per due volte "New Password" e bisogna digitare quella nuova;

# Comandi Unix di base

- *ls -la* -l sta per long list e -a per all entries:  
elenca tutti i file e le directory mostrando anche i vari diritti di accesso ai file (i primi 4 sono relativi a "u" ("user"), poi i seguenti 3 a "g" ("group") e gli ultimi 3 ad "o" ("others")), il nome dell'utente, il gruppo a cui appartiene, lo spazio occupato, la data dell'ultima modifica al file/directory, il nome del file/directory.

# Comandi Unix di base

- ***chmod*** *EntitàAzioneDiritto nomefiles* è un comando che permette all'utente di cambiare il di accesso ai suoi file. Azione è "+" per aggiungere o "-" per togliere i diritti e Entità è "u" o "g" o "o", o varie combinazioni, Diritto è "r", o "w" o "x".

# Le reti di calcolatori

- Sono classificate per estensione
  - reti locali *LAN (Local Area Network)*  
consentono di collegare fra loro dispositivi nello stesso edificio o in edifici adiacenti  
es : le rete del centro di calcolo
  - reti metropolitane *MAN (Metropolitan AN)*  
consentono il collegamento di dispositivi nella stessa area urbana  
es : la rete privata dell'Università



# Le reti di calcolatori

## Estensione

- reti geografiche WAN (*Wide AN*)  
collegano dispositivi diffusi su un'ampia area geografica  
es : in genere tutte le reti che devono attraversare suolo pubblico (due sedi distanti della stessa ditta)
- reti di reti (*internetworking*)  
collegano fra loro più reti diverse (in termini sia di hardware che di software) mediante opportuni elementi di interfaccia  
es: Internet

# Il World Wide Web (WWW)

- Insieme di *server* distribuito sulla rete, che permette di accedere a file memorizzati in particolari directory su tutte le macchine collegate
- Per richiedere informazioni ai server Web si usano solitamente dei programmi detti *Web client* (i normali navigatori)
  - i navigatori si preoccupano di interagire con i server seguendo opportuni protocolli
    - generalmente [http](#) (*hypertext transfer protocol*) ma anche [sftp](#) etc.

# Il World Wide Web

- Vediamo cosa accade richiedendo l'accesso a una certa pagina del web

es: <http://www.dsi.unive.it/~luccio/DOTTOGIURBIS08/corsodotto.html>

- `http://www.dsi.unive.it`

è l'indirizzo IP formato simbolico del server web dove si trova l'informazione cercata

- il navigatore traduce questa richiesta

- cioè la trasforma in un messaggio al server con tutti i dettagli necessari e secondo le regole del protocollo specificato (`http`)

`~luccio/DOTTOGIURBIS08/corsodotto.html`

viene inviato al server per individuare il file cercato (**`corsodotto.html`**)

all'interno delle directory `~luccio` e `DOTTOGIURBIS08` (il server sa come trovarle)

# Comandi Unix di base

- *ssh login@indirizzo* Permette di collegarsi in modo "SICURO" alla macchina definita da indirizzo con la login scelta.
- *telnet indirizzo* Permette di collegarsi in modo "INSICURO" alla macchina definita da indirizzo.
- *sftp login@indirizzo* Permette di caricare e scaricare i file in modo "SICURO" sulla/dalla macchina definita da indirizzo.
- *ftp login@indirizzo* permette di caricare e scaricare i file in modo "INSICURO" sulla/dalla macchina definita da indirizzo.

## Comandi Unix di base

- *sftp* > *ls* dice quali file e directory sono sulla macchina di indirizzo (ad es. su quella di informatica a Mestre);
- *sftp* > *lls* dice quali file e directory sono sulla macchina da cui ci si sta collegando (ad es. su quella del laboratorio);
- *sftp* > *cd pippo* permette di entrare nella directory di nome pippo sulla macchina indirizzo;
- *sftp* > *lcd pluto* permette di entrare nella directory di nome pluto sulla macchina da cui ci si sta collegando;

# Comandi Unix di base

- *sftp* > *get mario* permette di prendere il file di nome mario (nella directory in cui si è ) dalla macchina indirizzo e copiarlo sulla macchina da cui ci si sta collegando (nella directory in cui si è );
- *sftp* > *put anna* permette di prendere il file di nome anna dalla macchina da cui ci si sta collegando (nella directory in cui si è ) e copiarlo sulla macchina indirizzo (nella directory in cui si è );
- *sftp* > *quit* permette di chiudere sftp;

# Comandi Unix di base

- *ps* visualizza informazioni sui processi attivi.
- *ps -aux* vede tutti i processi, anche quelli che non compaiono con *ps*
- *CTRL^C* interrompe il programma in corso di esecuzione;
- *kill -9 numeroprocesso* uccide il processo con numeroprocesso (che si vede da *ps*);
- Per uscire dal sistema: *ESCI* oppure *EXIT* oppure *FINE SESSIONE* oppure *LOGOUT*.