

INFORMATICA DI BASE

- 6 crediti -

Docente: Michele Piana

Email: michele.piana@univr.it

URL: <http://www.di.univr.it/~piana>

Ricevimento: Lunedì ore 14:00 – 15:00

Studio: 2.05 (Ufficio del Preside)

Lunedì 10 aprile: ultima lezione prima di Pasqua

Giovedì 13 aprile: primo avviso di chiusura generale

Lunedì 17 aprile: Festa

Martedì 18 aprile: avviso su pagina web di Facoltà'

Mercoledì 19 aprile: secondo avviso di chiusura

Mercoledì 19 aprile: avviso di chiusura per Lettere

RIASSUNTO DELLA LEZIONE PRECEDENTE

Il Sistema operativo (S.O.)

- ✍ **I componenti fisici sono accessibili attraverso software dedicati**

}

Software di sistema o di base

- ✍ **Sono forniti assieme all'hardware**
- ✍ **L'insieme di questi programmi forma il *Sistema Operativo***

FUNZIONI DEL SISTEMA OPERATIVO - III

Il sistema operativo puo' essere inteso come uno strumento che virtualizza le caratteristiche dell'hardware sottostante, offrendo di esso la visione di una macchina astratta piu' potente e piu' semplice da utilizzare di quella fisicamente disponibile

SO: funzionalità

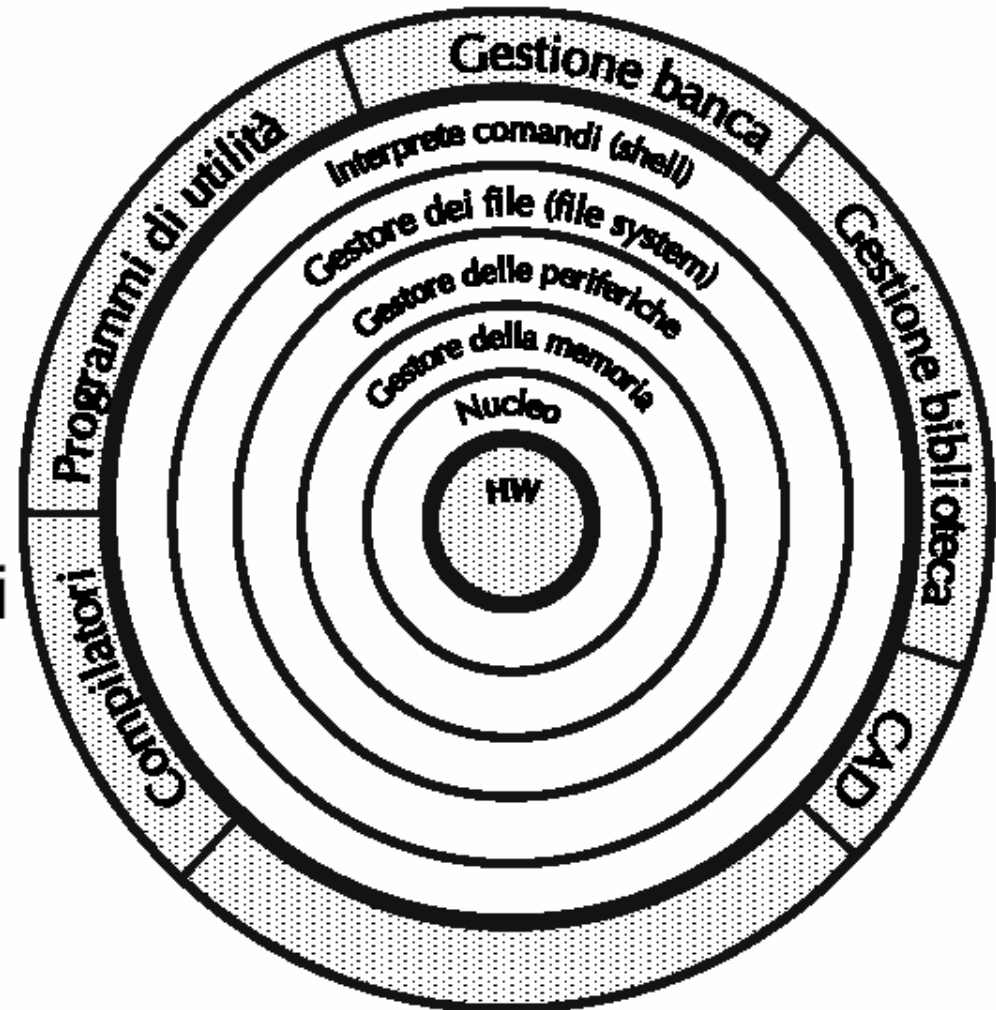
- **SO come GESTORE DELLE RISORSE,**
controlla tutte le risorse del calcolatore e le gestisce in modo efficiente:
 - tiene traccia di chi utilizza la risorse
 - accetta e soddisfa le richieste di utilizzo di una risorsa
 - fa da mediatore fra le risorse che risultano in conflitto.
- **SO come MACCHINA ESTESA:**
 - costituisce la base su cui è possibile scrivere i programmi applicativi.
 - presenta all'utente una macchina estesa più facile da programmare dell'HW sottostante.

Vantaggi di un SO

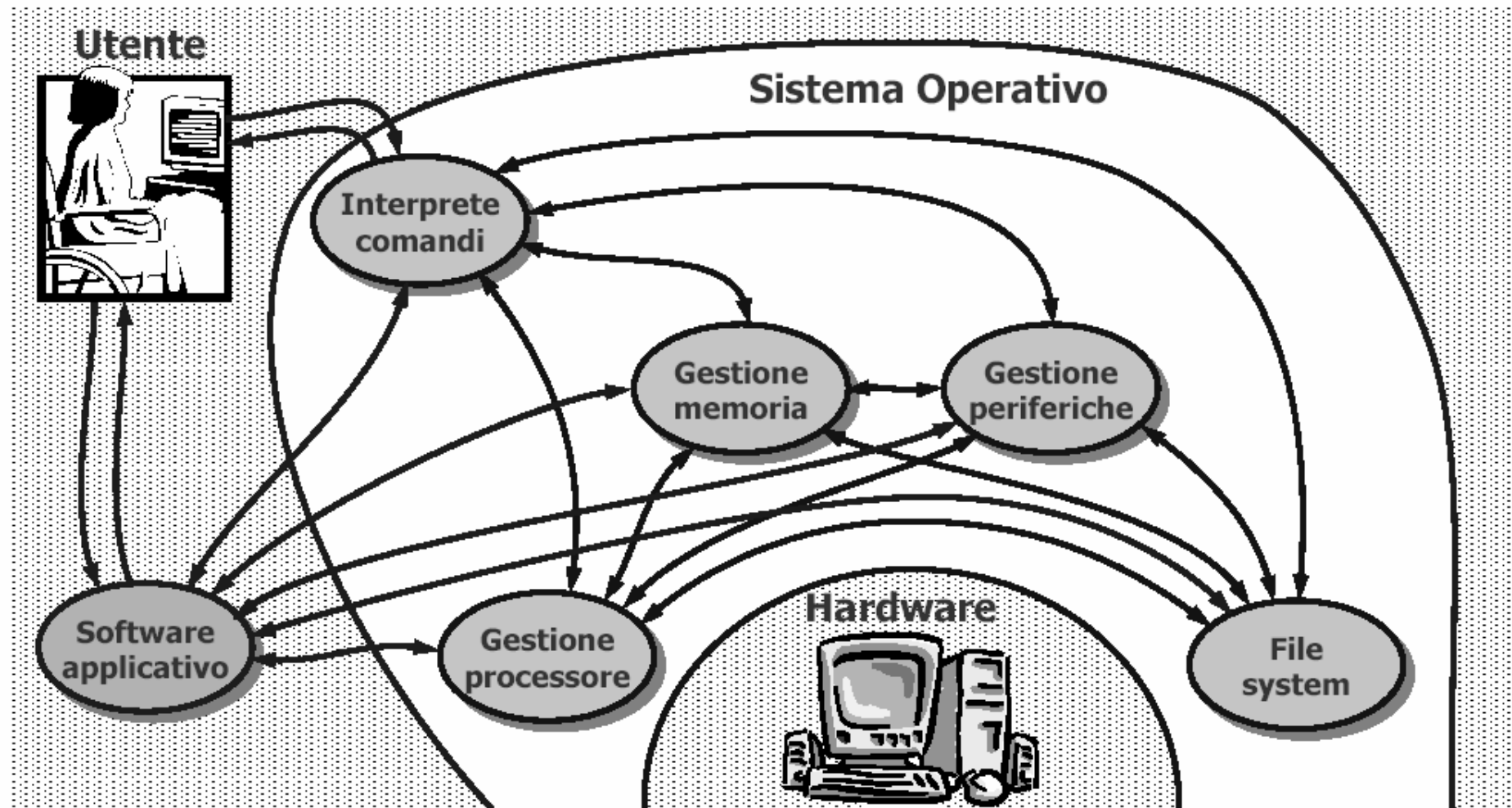
- Sono legati alla possibilità di definire modalità standard di interfaccia con i dispositivi fisici, cosicché sia possibile:
 - sviluppare programmi in modo semplice, modulare ed indipendente dallo specifico calcolatore su cui viene fatto funzionare il sistema operativo;
 - aggiornare il software di base e l'hardware in modo trasparente ai programmi applicativi e all'utente, ossia senza che vengano influenzati dall'operazione.

Organizzazione di un SO

- Gerarchia di "macchine virtuali"
- La visione della macchina virtuale a livello **n** è quella fornita dall'HW e dagli strati del SO fino all'ennesimo (incluso)



Elementi di un SO



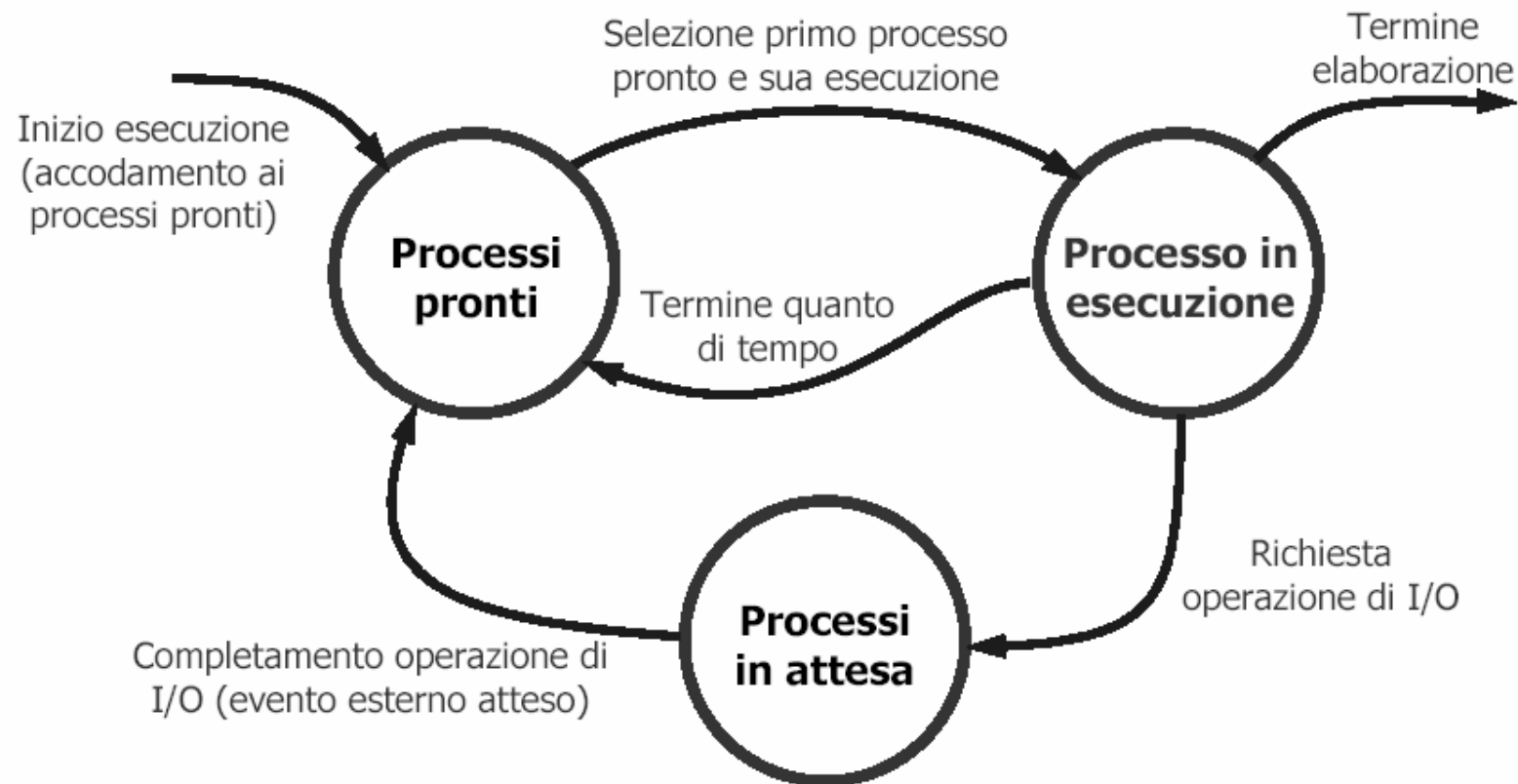
MULTIPROGRAMMAZIONE - I

- **nel sistema sono presenti diversi programmi, ognuno con il proprio tempo di elaborazione e propri tempi di attesa per le operazioni di i/o**
- **per evitare che la CPU venga utilizzata in modo esclusivo da parte di un solo programma, il tempo viene suddiviso in unita' elementari, dette quanti, da assegnare, secondo opportune politiche, a tutti i programmi**
- **Round-robin: assegnare a rotazione la disponibilita' di un quanto di tempo della CPU ai vari programmi presenti contemporaneamente in memoria**

MULTIPROGRAMMAZIONE - II

- **Quanto di tempo troppo lungo: il parallelismo tende a scomparire e la sequenzializzazione dei programmi viene degradata**
- **Quanto di tempo troppo corto: se diventa dello stesso ordine del tempo di commutazione fra programmi (circa 100 ms per Unix) il sistema perde efficienza**

Diagramma a tre stati



CONTENUTO DELLA LEZIONE

- ✍ Ancora processi
- ✍ Gestione della memoria
- ✍ Gestione delle periferiche
- ✍ File system
- ✍ Applicazioni

CONTEXT SWAPPING - I

Domanda cruciale: come avviene il passaggio di stato tra un processo e l'altro?

Definizione: si dice contesto di un processo, l'insieme dei dati che rappresentano lo stato di un processo: situazione della memoria, contenuto dei registri, livello di priorit ...

CONTEXT SWAPPING - II

Transizione di stato per un processo:

- **Quando un processo viene interrotto (ovvero esce dal suo stato di esecuzione), il sistema operativo salva il suo contesto in una struttura della memoria centrale detta descrittore del processo**
- **Quando il processo viene ripreso (torna in esecuzione) il sistema operativo provvede a ripristinare il suo contesto recuperando i dati dal descrittore**

CONTEXT SWAPPING - III

Il 'context swapping' e' il fenomeno per cui un processo P1 in esecuzione viene sostituito da un processo P2 pronto

Il SO provvede a:

- salvare il contesto di P1 e gestirne l'evoluzione (da attesa a pronto)**
- ripristinare il contesto di P2**

CONCORRENZA TRA PROCESSI

Vantaggi:

- **impegnare in modo trasparente una o piu' CPU**
- **aumentare le prestazioni della CPU**
- **condividere la stessa risorsa fisica fra diversi utenti in modo trasparente ma controllato**
- **accedere contemporaneamente agli stessi dati raccolti in un database comune e centralizzato**

PROBLEMI: STARVATION

Diversi processi potrebbero accedere alla stessa risorsa ma solo alcuni di essi vi riescono, a causa della politica di gestione praticata dal sistema operativo

Esempio 1: il sistema operativo decide di assegnare la CPU sempre al processo con priorit  piu' alta. Ne segue che un processo con priorit  bassa non puo' mai accedere alla CPU

Esempio 2: assegnare ai tesisti la precedenza nell'uso delle stampanti: nei periodi di tesi i non-tesisti non stamperebbero mai

PROBLEMI: BLOCCO CRITICO

Un intero gruppo di processi rimane bloccato senza che la risorsa da loro contesa venga utilizzata da alcuno di essi

Esempio: il processo A detiene la stampante. Il processo A richiede i risultati del processo B per poter terminare le elaborazioni e, analogamente il processo B attende i risultati di A. Ne consegue che ne' A ne' B utilizzeranno la stampante (vincolo circolare)

Interazioni tra processi

- Le **interazioni** fra processi sono classificabili in:
 - **indesiderate e (spesso) impreviste:**
i processi **competono** per le risorse che servono per completare l'esecuzione (causando eventualmente problemi quali blocco critico e starvation);
 - **desiderate e previste:**
legate a meccanismi di **cooperazione** fra processi che hanno l'obiettivo di giungere alla soluzione di un problema complesso.
- La **gestione delle interazioni** fra i processi implica
 - la **sincronizzazione** fra le varie attività che ogni singolo processo deve svolgere in modo parallelo rispetto agli altri
 - la **comunicazione**, ovvero una modalità per lo scambio dei dati fra i processi
- **Modalità di funzionamento** dei processi:
 - **in foreground**, quando il processo è abilitato all'interazione con l'utente;
 - **in background**, quando il processo non è in grado, almeno temporaneamente, di interagire direttamente con l'utente; questo è lo stato in cui si trovano parecchi dei processi relativi alle funzioni interne del sistema operativo
 - sono attivati automaticamente all'accensione;
 - sono chiamati **daemon** in Unix e **agenti** o **servizi** in altri sistemi.

Gestore della memoria

- Applica tecniche per gestire il conflitto fra dimensione della memoria fisica e spazio complessivo richiesto dai programmi che devono essere eseguiti in modo concorrente e dai relativi dati.
- Combina le seguenti strategie:
 - consentire il caricamento di un programma a partire da un indirizzo qualunque della memoria;
 - ridurre la necessità di spazio tenendo in memoria solo una porzione dei programmi e dei dati;
 - condividere parte delle istruzioni (codice eseguibile) fra diversi processi corrispondenti a uno stesso programma.
- Il gestore della memoria
 - garantisce ai vari processi uno **spazio di indirizzamento virtuale** in cui lavorare, che può essere superiore alla memoria fisica presente nel calcolatore
 - mette in atto dei meccanismi di protezione che tutelano la privacy dello spazio di lavoro assegnato a ogni processo.

SWAPPING

Nei sistemi a multiprogrammazione la memoria centrale non ha mai dimensioni tali da poter contenere tutti i programmi da eseguire in modo concorrente

Nella memoria di massa c'è un'area di swap in cui il SO trasferisce parte del contenuto della memoria centrale

Nello swapping vengono trasferiti su disco i dati relativi a processi in stato di attesa o di pronto

Gestore delle periferiche

- Comunicazione tra l'ambiente CPU-RAM ed i dispositivi esterni.
 - Asincronicità tra ambiente e calcolatore
 - Gestione dell'accesso contemporaneo al calcolatore da parte di diverse periferiche.
- Mascherare ai processi l'esistenza di un numero limitato di risorse.
 - Esempio: stampa da più processi (es. due word ed un excel) su di una unica stampante.
- Mascherare ai processi la differenza tra risorse dello stesso tipo (o di tipo simile)
 - Esempio: stampante laser da un plotter e da un terminale video

Gestione periferiche I/O

- Comandi **ad alto livello** per accedere alle periferiche che usano meccanismi quali:
 - **i controller**, dispositivi hardware per effettuare le operazioni di trasferimento dati;
 - dipendono dalle caratteristiche fisiche delle periferiche che gestiscono
 - l'interfaccia per la gestione di un mouse è sicuramente diversa da quella utilizzata per controllare il funzionamento di un lettore di CD-ROM.
 - **i driver**, programmi software per la gestione delle periferiche;
 - mascherano le caratteristiche specifiche dei controller,
 - forniscono un insieme di primitive ad alto livello per la gestione delle operazioni di ingresso/uscita utilizzabili dai programmi applicativi e dagli utenti.
- I sistemi operativi comprendono i driver per la gestione delle periferiche più comuni:
 - tastiera, video, mouse, ...
 - stampanti, scanner, ...
- Ogni aggiunta o modifica alla configurazione standard comporta l'installazione di software aggiuntivo (driver aggiuntivi).

Plug&Play

- I sistemi operativi più recenti sono dotati di funzioni di **Plug&Play (PnP)** che permettono la configurazione automatica dei driver:
 - all'attivazione il sistema operativo scandisce ed esamina tutte le periferiche collegate al sistema;
 - le periferiche si fanno riconoscere specificando quali driver servono;
 - il sistema operativo installa gli opportuni driver per la loro gestione.
- Un sistema **PnP** consente di aggiungere (**plug**) nuove periferiche al sistema che possono essere utilizzate (**play**), senza necessità di intervento da parte dell'utente per la selezione e l'installazione dei driver.

Gestione memoria di massa

- **Obiettivo:**
presentare all'utente l'organizzazione logica dei dati e le operazioni che è possibile compiere su di essi.
- Operazioni di base di un file system:
 - **recupero** di dati precedentemente memorizzati;
 - **eliminazione (cancellazione)** di dati obsoleti;
 - **modifica/aggiornamento** di dati preesistenti;
 - **copia** di dati (e.g. da HD a FD) per backup o per il trasferimento;
 - ...
- I servizi vengono forniti sia ai **programmi applicativi** che direttamente agli **utenti**.

File system

➤ **FILE:**

- contenitore logico di informazioni (dati o istruzioni);
- oggetto a “lunga vita”, per conservare le informazioni anche dopo la terminazione del processo che lo ha generato.

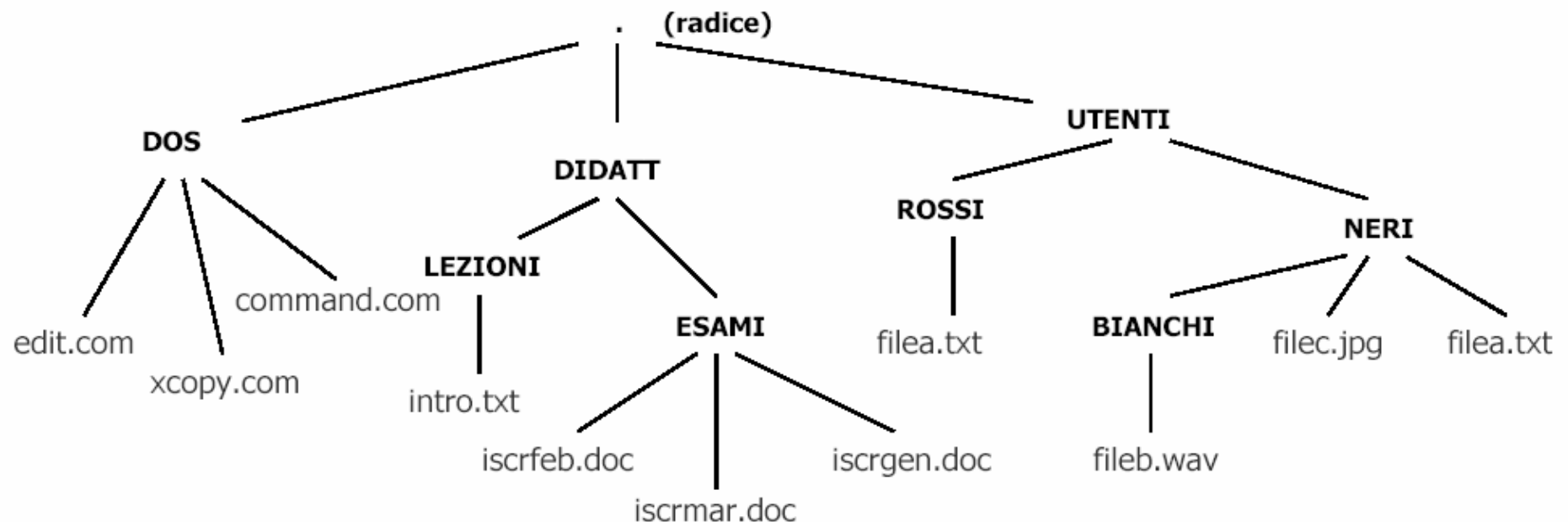
➤ Per ogni file:

- Identificatore (**nome.estensione**)
- Periferica (**drive**) e percorso sulla periferica
- Data creazione
- Dimensione
- Posizione effettiva dei dati nella memoria di massa
- Altre informazioni
 - applicazione che consente all'utente di “usare” il file
 - data di ultima modifica
 - diritti di accesso al contenuto del file
 - ...

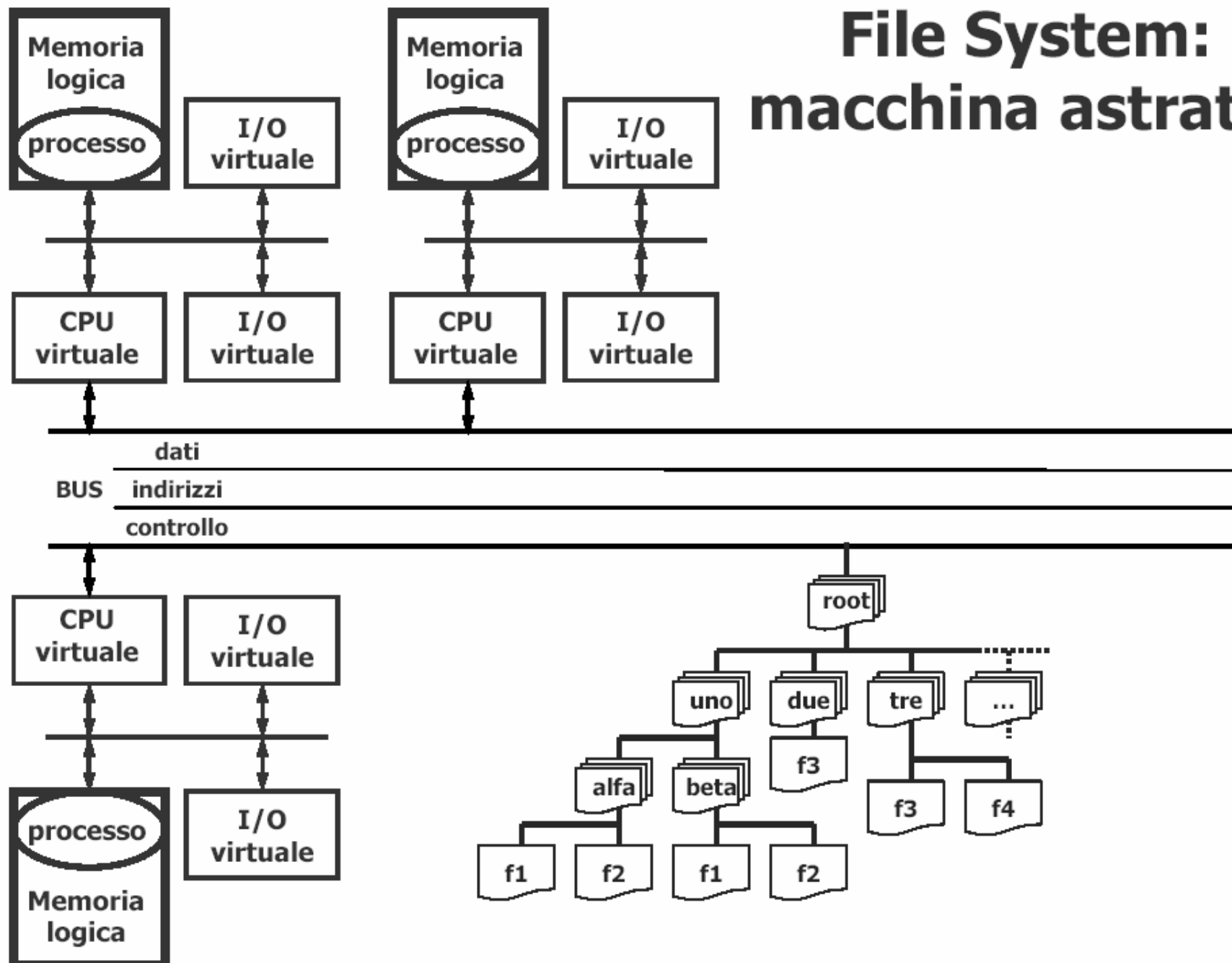
File

- I nomi dei file sono in genere composti da due parti:
 - nome (vero e proprio), che viene assegnato dall'utente
 - estensione, associata al programma che ha generato il file e consente quindi di identificare la tipologia dei dati contenuti nel file
 - i file eseguibili hanno estensioni exe,
 - i file generati da Word hanno estensione doc,
 - i file di testo generici hanno estensione txt.
- Ogni sistema operativo pone dei vincoli sulla lunghezza dei filename e sui caratteri di cui possono essere costituiti
 - MS-DOS imponeva una lunghezza massima di 8+3 caratteri per nomi ed estensioni
 - Windows ha un limite di 254 caratteri (compreso il path)
- I file sono generalmente organizzati in cartelle (directory) e sottocartelle in una gerarchia ad albero (o, al limite, a grafo aciclico).

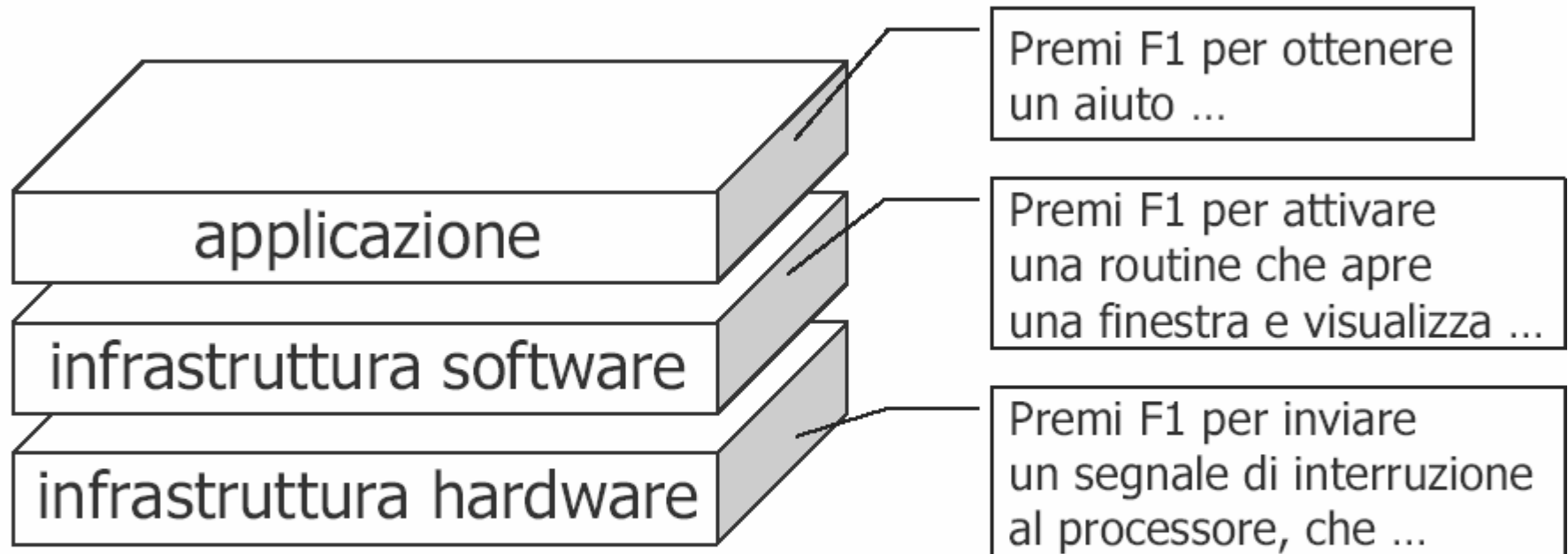
Un esempio di struttura



File System: macchina astratta



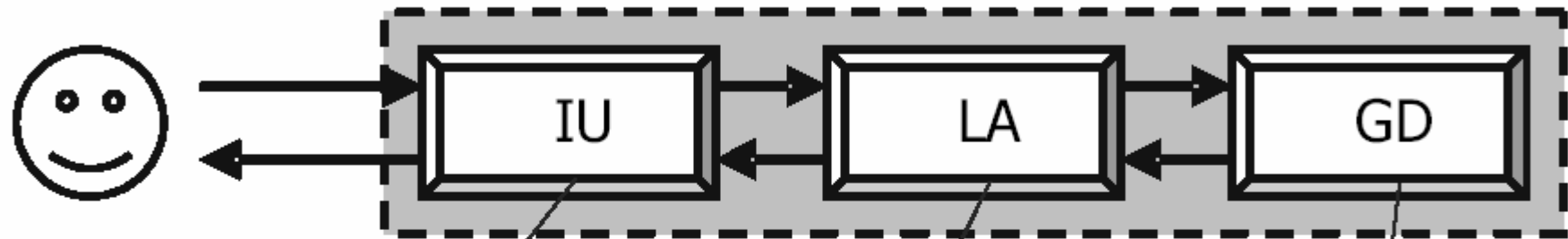
L'organizzazione "per livelli" dei sistemi informatici



... sono "punti di vista" diversi, e complementari, sullo stesso sistema...

Un'applicazione ben costruita (e funzionante...!) può essere usata prescindendo in larga misura dalle problematiche infrastrutturali dei livelli sottostanti

L'organizzazione funzionale di un'applicazione informatica

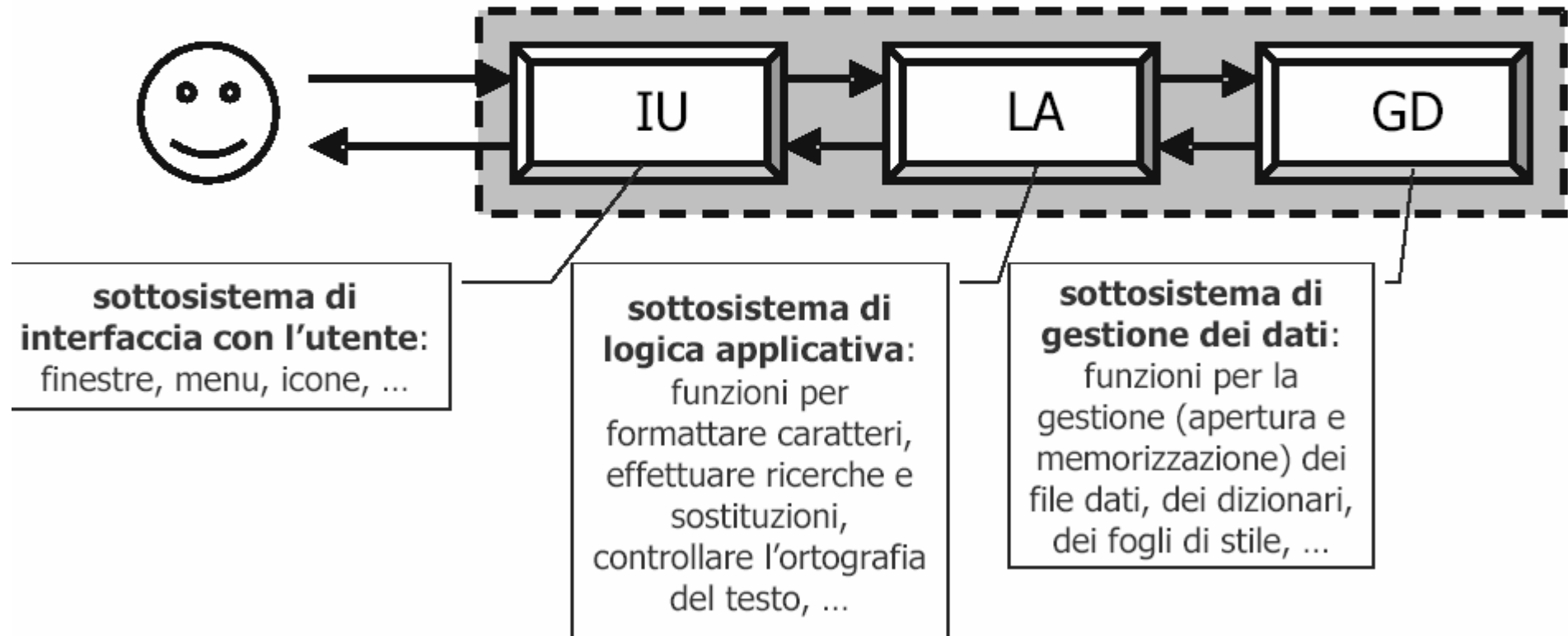


sottosistema di interfaccia con l'utente
(*user interface*, o anche *presentation layer*), incaricato di controllare i comandi e i dati che l'utente fornisce come input al sistema e di presentare come output i risultati dell'esecuzione del programma

sottosistema di logica applicativa
(*business logic layer*), che implementa gli specifici algoritmi di manipolazione dei dati che caratterizzano l'applicazione

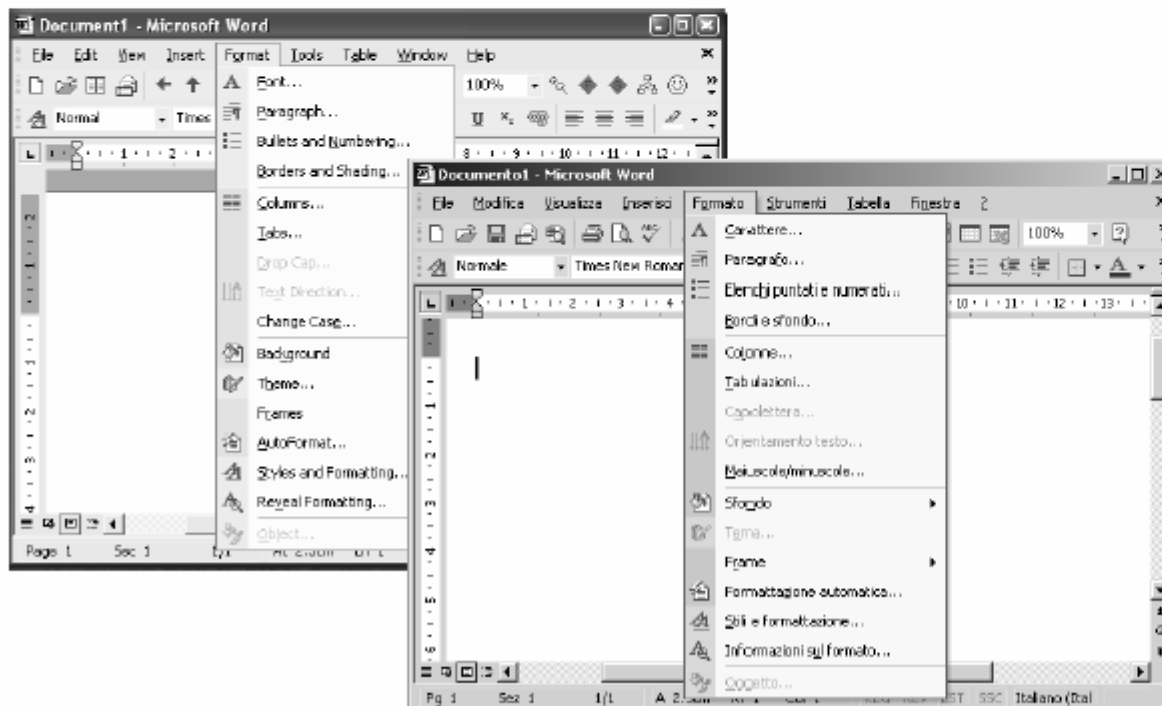
sottosistema di gestione dei dati
(*data layer*), che si occupa dell'organizzazione dei dati, e in particolare della loro memorizzazione e del loro reperimento efficiente

Organizzazione funzionale delle applicazioni: l'esempio dei sistemi di elaborazione di testi



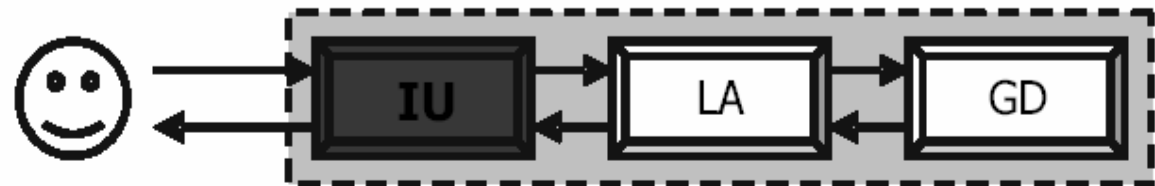
Organizzazione funzionale delle applicazioni: indipendenza funzionale

I tre sottosistemi sono funzionalmente indipendenti l'uno dall'altro...
... si può, per esempio, cambiare IU senza modificare LA e GD



... come nel caso
della localizzazione
delle applicazioni...

Il sottosistema IU – interfaccia utente



Ha il compito di gestire l'interazione del programma con l'utente:

- consente all'utente di specificare gli input al programma
 - restituisce all'utente gli output del programma
- e inoltre:
- fornisce informazioni sullo stato di esecuzione del programma

A causa dell'indipendenza funzionale tra sottosistemi,
le stesse caratteristiche per LA e GD possono essere ottenute
con IU diverse...

Il sottosistema IU – Un esempio /2

IU (opzione 2): interfaccia **a linea di comando** (*command line*),
con modalità interattiva modale

```
>conta2
Conta il numero di occorrenze di un carattere all'interno di una stringa.

Carattere: a
Stringa: Ancora una prova
Distinguere tra minuscole e minuscole (s/n)? n

Numero di occorrenze: 4

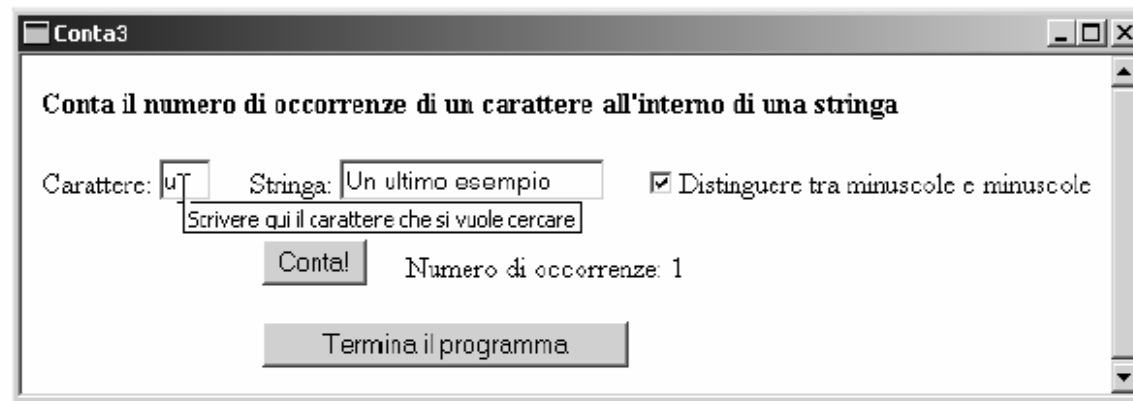
>_
```

Questo programma opera secondo la logica:

1. l'utente mette in esecuzione il programma
2. il programma attiva una sessione domande-risposte per ottenere dall'utente i parametri di input
3. viene eseguita la parte del programma che opera sui parametri di input per ottenere il risultato
4. il risultato prodotto nell'esecuzione viene presentato all'utente come output del programma
5. l'esecuzione termina

Il sottosistema IU – Un esempio /3

IU (opzione 3): interfaccia **grafica** (*graphical user interface*),
con modalità interattiva non modale



Questo programma opera secondo la logica:

1. l'utente mette in esecuzione il programma
2. il programma attiva un'interfaccia utente interattiva non modale e si mette in condizione "di ascolto"
3. fino a che l'utente non seleziona il pulsante per terminare l'esecuzione del programma:
 - a) l'utente immette i parametri di input
 - b) l'utente seleziona il pulsante che mette in esecuzione la parte del programma che opera sui parametri di input per ottenere il risultato
 - c) il risultato prodotto viene presentato all'utente
4. l'esecuzione termina

Graphical user interface (GUI): personalizzazione

Forniscono una molteplicità di strumenti di input...

(p.es. per copiare nell'archivio appunti (*clipboard*) un testo selezionato si può:

(1) con la tastiera premere Ctrl-C

(2) con la tastiera aprire il menu Modifica con il corrispondente tasto scorciatoia e quindi muovere il cursore con i tasti freccia fino a evidenziare la voce Copia premendo quindi Invio

(3) con il mouse aprire il menu Modifica e quindi selezionare la voce Copia

(4) con il mouse aprire il menu contestuale (cursore sopra la selezione e click con il tasto di destra) da cui scegliere la voce Copia

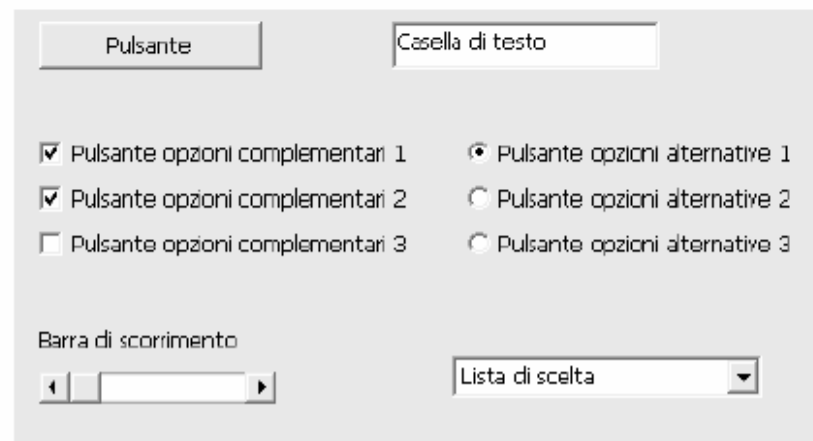
(5) ...

... e consentono quindi a ogni utente di adottare
un proprio **stile di uso** dei programmi e, spesso,
di **personalizzarne l'interfaccia**

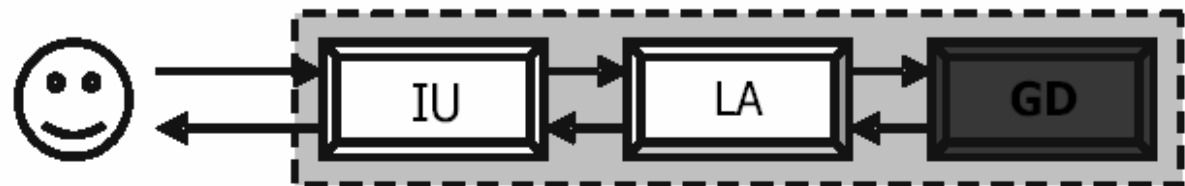
Graphical user interface (GUI): standardizzazione

L'uniformità delle IU consente agli utenti di focalizzare l'attenzione sulle funzionalità del programma e non sull'interfaccia utente (sul **cosa** fare e non sul **come** farlo); sono stati standardizzati:

- l'uso di sequenze di tasti di comando, per esempio F1 per richiamare un aiuto sul programma in esecuzione o Ctrl-C per copiare nell'archivio appunti l'entità selezionata
- la presenza e le modalità di uso dell'archivio appunti, che funziona in modo omogeneo in diverse applicazioni e consente di trasferire dati tra applicazioni
- la presenza e le modalità di uso di funzioni di utilità di impiego generale, come Apri..., Salva con nome..., o Stampa...
- la presentazione grafica e le modalità di uso degli oggetti grafici nelle finestre:



Il sottosistema GD – gestione dei dati



Ha il compito di organizzare i dati gestiti dall'applicazione, in particolare relativamente alla loro memorizzazione e al loro reperimento efficiente

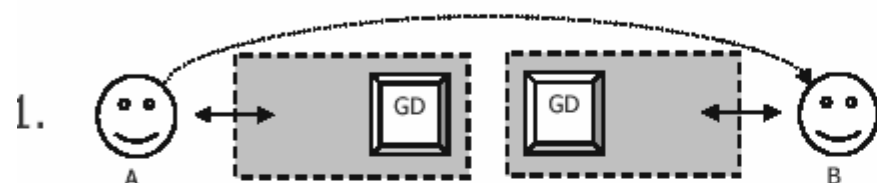
A causa dell'indipendenza funzionale tra sottosistemi, le stesse caratteristiche per IU e LA possono essere ottenute con GD diversi...

Per esempio, una tipica applicazione di elaborazione di testi mette a disposizione diversi formati di file per memorizzare i testi

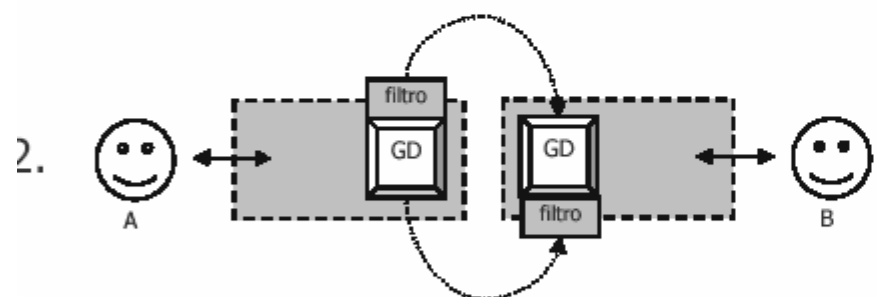
Il sottosistema GD diventa critico quando utenti che usano applicazioni diverse vogliono **scambiarsi dei dati**

Il sottosistema GD – Scambio di dati

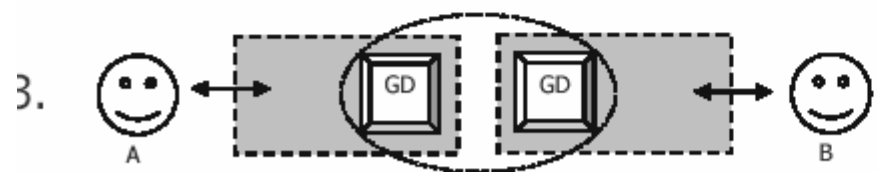
Scenari (l'utente B deve operare su un file inizialmente prodotto dall'utente A):



i programmi di A e B adottano formati di file non compatibili e non offrono strumenti per rendere dati accessibili all'esterno: **B deve riscrivere il testo**



i programmi di A e B adottano formati di file non compatibili ma dispongono di filtri di esportazione e/o di importazione: **A può esportare il testo nel formato del programma di B oppure B può importare il testo dal formato del programma di A**



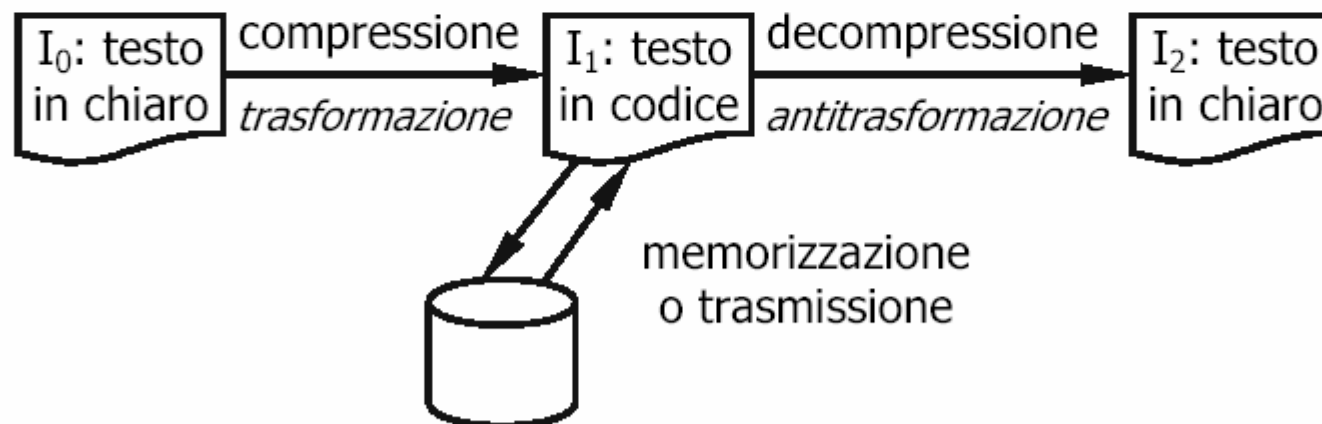
i programmi di A e B adottano uno stesso formato di file e quindi si comportano come se disponessero di uno stesso sottosistema GD: **non ci sono problemi di scambio dei dati**

I servizi di compressione dei dati

nel caso di:

- memorizzazione su un supporto di capacità scarsa
- trasmissione su un canale di capacità scarsa

si impiegano tipicamente **servizi di compressione dei dati**



RIFERIMENTI AL LIBRO

- ✍ Gestione della memoria: paragrafi 6.3, 6.3.1, 6.3.2, 6.3.3
- ✍ Gestione delle periferiche: paragrafo 6.4
- ✍ File system: paragrafi 6.5, 6.5.1, 6.5.2
- ✍ Applicazioni: capitolo 8 (introduzione); paragrafi 8.1, 8.2, 8.2.1