
Architettura del Calcolatore

Emilio Di Giacomo e Walter Didimo

Computer e programmi

- Il computer è una macchina programmabile
- Grazie alla possibilità di eseguire programmi diversi, il computer può svolgere compiti molto diversi tra loro:
 - eseguire complessi calcoli matematici,
 - redigere documenti,
 - progettare edifici o automobili,
 - navigare nel Web,
 - effettuare transazioni bancarie,
 - riprodurre brani musicali,...

Che cos'è un programma?

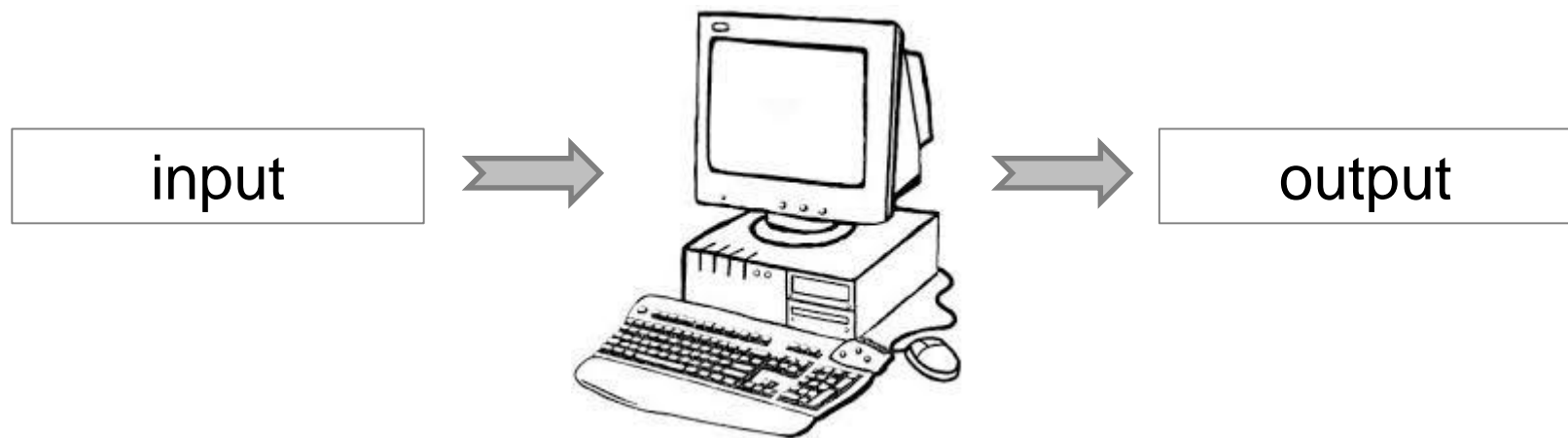
- Un programma è una sequenza di istruzioni
- Ogni istruzione è un'operazione semplice che il computer sa eseguire:
 - un'operazione aritmetica
 - la memorizzazione di un numero
 - ...
- Eseguendo tutte le istruzioni che costituiscono un programma, il computer è in grado di svolgere compiti complessi

Hardware e Software

- Un computer è dunque costituito da due macro-elementi:
 - hardware: insieme di tutti i componenti fisici del computer
 - software: insieme dei programmi che richiedono all'hardware di svolgere compiti specifici

Informazione

- Per eseguire le varie attività richieste dall'utente, un calcolatore elabora e memorizza informazioni
 - riceve dell'informazione sotto forma di dati di input
 - elabora tale informazione manipolando i dati
 - produce nuova informazione che viene fornita in risposta sotto forma di dati di output



Informazione

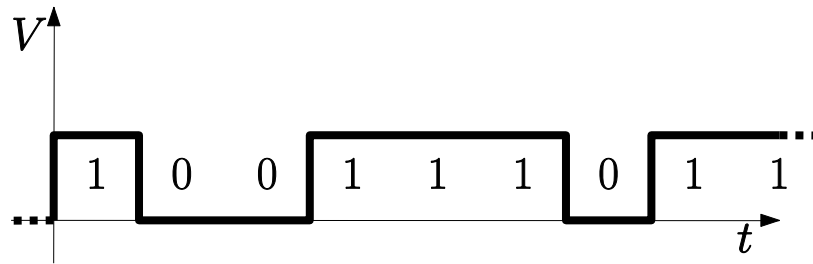
- L'informazione deve essere rappresentata in una forma che il calcolatore è in grado di interpretare
- Si usano sequenze di simboli ognuno dei quali può assumere due valori possibili (convenzionalmente 0 e 1)
- Ognuno di tali simboli è chiamato bit
- Il termine bit deriva dall'inglese binary digit (cifra binaria)

Bit e byte

- Il bit è l'unità minima di informazione
 - Con un bit possiamo rappresentare una tra due informazioni
 - Con due bit una tra quattro
 - Con tre bit una tra otto e così via
- Un'altra unità di informazione è il byte, che consiste di 8 bit
- Il byte viene usato come unità di misura della memoria

Bit e circuiti

- Nei circuiti i bit vengono rappresentati tramite segnali elettrici che assumono uno tra due diversi livelli di tensione

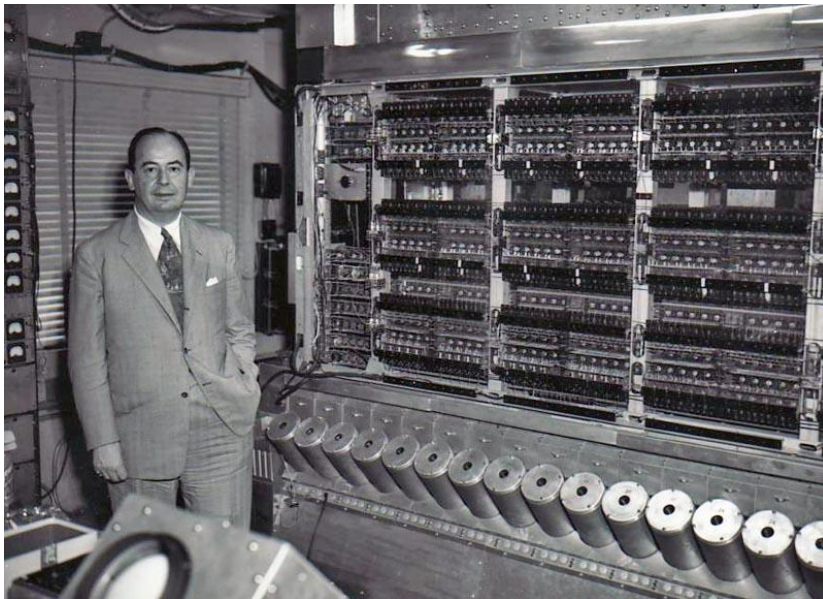


- Anche la memorizzazione dei bit viene realizzata tramite circuiti che possono assumere uno tra due stati

Componenti hardware di un calcolatore

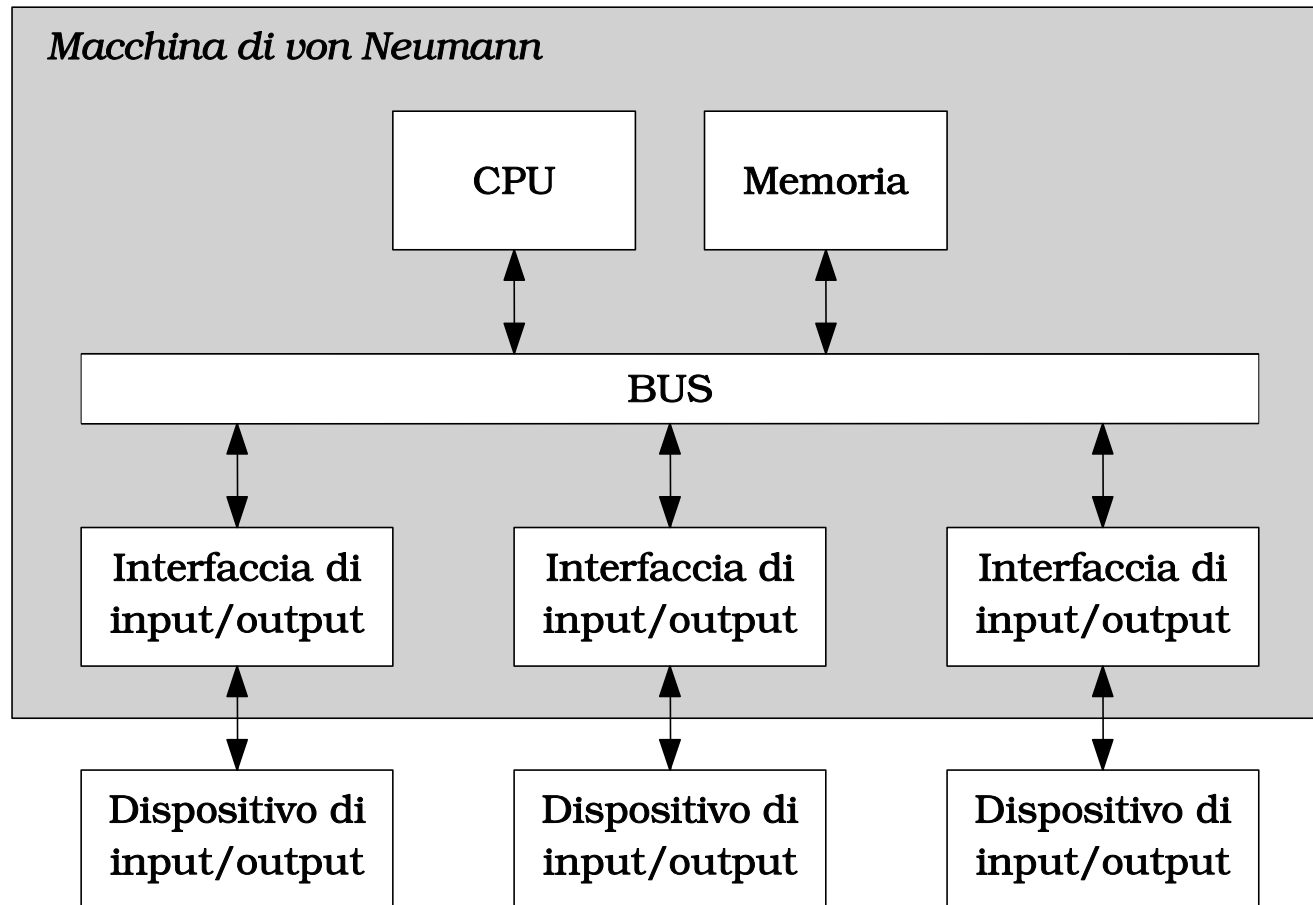
L'architettura di von Neumann

- I diversi calcolatori esistenti differiscono molto tra di loro dal punto di vista dell'hardware
- ciò nonostante quasi tutti si basano su un'architettura comune chiamata architettura di von Neumann (o macchina di von Neumann)



John
von Neumann
(1903-1957)

L'architettura di von Neumann



L'architettura di von Neumann

- CPU (Central Processing Unit): è il dispositivo che esegue le elaborazioni. È costituito da due componenti:
 - ALU (Arithmetic Logic Unit): esegue calcoli aritmetici e logici
 - Unità di controllo: interpreta le istruzioni da eseguire e coordina le attività degli altri componenti
- Memoria: memorizza dati e programmi. È organizzata in celle di lunghezza fissa (ad es. 8 bit), identificate da un indirizzo numerico

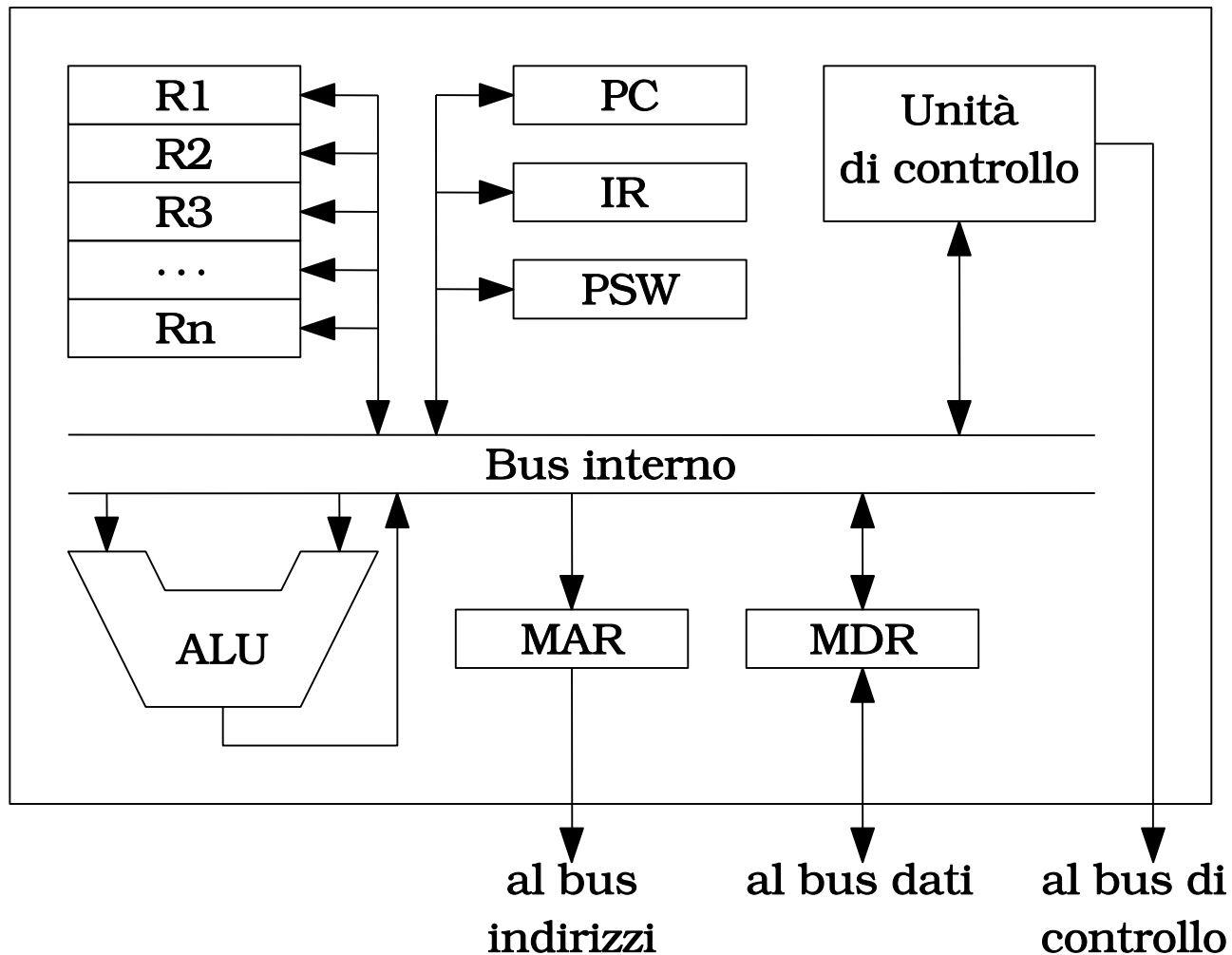
L'architettura di von Neumann

- Interfacce di Input/Output (I/O): consentono al calcolatore di comunicare con le periferiche di I/O (mouse, tastiera, monitor,...) che sono considerate esterne al sistema
- Bus: canale di comunicazione tra i vari componenti. Logicamente composto da:
 - bus indirizzi
 - bus dati
 - bus di controllo

Processore (CPU)

- La CPU viene realizzata nei calcolatori come microprocessore, un dispositivo elettronico estremamente complesso
- Un microprocessore è in grado di eseguire un insieme di istruzioni che costituiscono il suo linguaggio macchina
- Le istruzioni sono molto semplici:
 - istruzioni aritmetiche
 - istruzioni logiche
 - confronti
 - letture e scritture da/in memoria

Architettura interna della CPU



Registri della CPU

- Oltre alla ALU e all'unità di controllo, la CPU contiene una serie di registri
- I registri sono celle di memoria utilizzate dalla CPU per le sue attività
- Registri generali R1...Rn: deposito temporaneo dei dati
- Program Counter (PC): memorizza l'indirizzo della prossima istruzione da eseguire
- Instruction Register (IR): memorizza l'istruzione in esecuzione

Registri della CPU

- Memory Address Register (MAR): nelle operazioni di lettura/scrittura della memoria, memorizza l'indirizzo della cella da leggere/scrivere
- Memory Data Register (MDR): nelle operazioni di lettura/scrittura della memoria, memorizza il dato da scrivere in memoria o riceve il dato letto dalla memoria
- Program Status Word (PSW): memorizza una serie di informazioni sull'ultima operazione eseguita dalla ALU

Bus interno e Clock

- Tutti gli elementi interni della CPU comunicano tramite un bus interno
- Tutte le attività della CPU sono sincronizzate dall'orologio di sistema (clock):
 - ad ogni ciclo di clock la CPU esegue una qualche attività
 - la frequenza di clock (misurata in Hz - Hertz) è una misura della velocità del processore
 - $1 \text{ Hz} = 1 \text{ volta al secondo}$

Esecuzione delle istruzioni

- Le istruzioni del linguaggio macchina sono sequenze di bit
- Un'istruzione ha un codice operativo (opcode) che indica il tipo di operazione ed eventualmente degli operandi
- L'esecuzione di un'istruzione avviene in tre fasi:
 - fase di fetch: caricamento dell'istruzione dalla memoria
 - fase di decode: decodifica dell'istruzione
 - fase di execute: esecuzione dell'istruzione

Esecuzione delle istruzioni

- Fase di fetch:
 - il contenuto del PC viene scritto nel MAR
 - l'istruzione viene letta dalla memoria e scritta nel MDR
 - il contenuto del MDR viene trasferito nell'IR
 - il PC viene incrementato
- Fase di decode:
 - sulla base del codice operativo (opcode) dell'istruzione la CPU capisce il tipo di operazione da eseguire e come interpretare gli operandi

Esecuzione delle istruzioni

- Fase di execute:
 - l'unità di controllo attiva i diversi componenti della CPU per eseguire l'istruzione
 - Esempio ($R1 \leftarrow R1 + R2$), cioè "*scrivi in R1 la somma dei contenuti dei registri R1 e R2*":
 - il contenuto del registro R1 viene trasferito in ingresso alla ALU
 - il contenuto del registro R2 viene trasferito in ingresso alla ALU
 - la ALU esegue l'operazione di somma
 - il risultato viene scritto nel registro R1 (sovrascrivendo il precedente valore)

Memoria

- La memoria è costituita da un numero finito di celle o locazioni
- Ogni cella
 - è in grado di memorizzare una sequenza di bit di lunghezza fissa (di solito 8)
 - è identificata in modo univoco da un indirizzo numerico
- Sebbene le celle abbiano la dimensione di un byte, di solito la CPU opera con dati di dimensione maggiore:
 - 4 byte (= 32 bit) o 8 byte (= 64 bit) nei moderni calcolatori

Memoria

- La dimensione dei dati su cui tipicamente opera la CPU viene chiamata parola (word)
- La dimensione della parola è una caratteristica di una specifica architettura
- Tipicamente hanno la stessa dimensione della parola:
 - il bus dati
 - i registri della CPU
 - la lunghezza delle istruzioni

Capacità della memoria

- La caratteristica più importante della memoria è la sua capacità, cioè la quantità di informazione che può contenere.
- Essa dipende da:
 - dimensione di una cella di memoria
 - numero di celle
- La capacità viene misurata in byte
- Poiché la dimensione di una cella è tipicamente di un byte, la capacità è di solito pari al numero di celle

Capacità della memoria

- Il numero di celle in una memoria è di solito pari ad una potenza di 2
- Infatti, poiché le celle hanno un indirizzo numerico, se si usano k bit per l'indirizzo si hanno 2^k possibili indirizzi
- Poiché la capacità delle memorie è tipicamente molto grande essa viene espressa tramite i multipli del byte (KB, MB, GB, TB, ecc.)

Quanto vale un kilobyte?

- Rispetto ai multipli del byte esiste una certa confusione
- Poiché $2^{10}=1024\approx 1000$ e poiché le memorie hanno dimensioni che sono di solito potenze di due, si è soliti indicare con 1KB (un kilobyte) una quantità di memoria pari a 1024 byte
- Analogamente si usa:
 - 1MB (un megabyte) per indicare $2^{20}=1048576\approx 1000000$ byte
 - 1GB (un gigabyte) per indicare $2^{30}=1073741824\approx 1000000000$ byte
 - ...

Quanto vale un kilobyte?

- I prefissi chilo (k), mega (M), giga (G), ecc. sono definiti nel Sistema Internazionale delle misure e corrispondono a potenze di 10
- Il loro uso per indicare potenze di 2 è scorretto e fonte di confusione
- Per questo motivo la **IEC** (International Electrotechnical Commission) e la **IEEE** (Institute of Electrical and Electronics Engineers) hanno definito dei prefissi alternativi che corrispondono a potenze di 2

Prefissi per multipli binari

Fattore	Simbolo	Nome	Nome esteso	Corrispondente SI	Fattore SI
2^{10}	Ki	kibi	kilobinary	kilo (k)	10^3
2^{20}	Mi	mebi	megabinary	mega (M)	10^6
2^{30}	Gi	gibi	gigabinary	giga (G)	10^9
2^{40}	Ti	tebi	terabinary	tera (T)	10^{12}
2^{50}	Pi	pebi	petabinary	peta (P)	10^{15}
2^{60}	Ei	exbi	exabinary	exa (E)	10^{18}
2^{70}	Zi	zibi	zettabinary	zetta (Z)	10^{21}
2^{80}	Yi	yobi	yottabinary	yotta (Y)	10^{24}

- L'uso di tali prefissi stenta ad affermarsi e sono ancora ampiamente utilizzati i prefissi decimali

Memoria principale

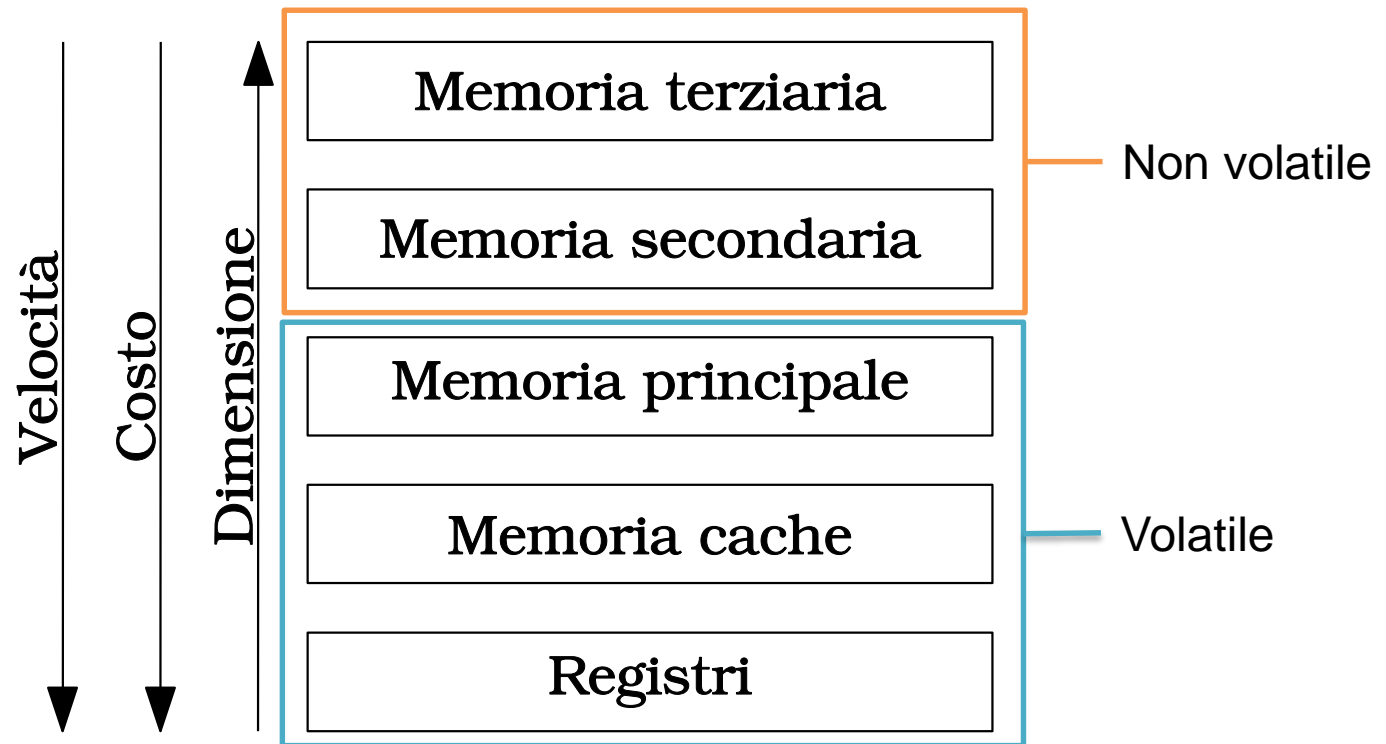
- La memoria di cui abbiamo parlato finora è quella che viene chiamata memoria principale
 - anche indicata come RAM (Random Access Memory)
- La RAM è la memoria di lavoro della CPU
- In essa si trovano i programmi in esecuzione e i dati ad essi necessari
- La RAM è una memoria volatile:
 - allo spegnimento del computer i dati in essa memorizzati vengono persi

Memoria secondaria

- Per memorizzare programmi e dati in maniera permanente, il computer è dotato di una memoria secondaria
- La memoria secondaria è rappresentata dall'hard disk o da memorie a stato solido
- La memoria secondaria si differenzia da quella principale perché:
 - non è volatile
 - è più grande
 - è più economica
 - è più lenta

Gerarchia di memoria

- Più in generale nel calcolatore esistono diversi tipi di memoria che si differenziano per dimensione, costo, velocità



Gerarchia di memoria

- Memoria cache: memoria piccola e veloce che memorizza i dati utilizzati più recentemente. Permette di ridurre i tempi di accesso a memoria
- Memoria terziaria: usata per la memorizzazione off-line dei dati (CD-ROM, DVD, Nastri,...)

Interfacce di I/O

- Per poter comunicare con l'esterno il calcolatore è dotato di periferiche di input e output:
 - dispositivi per la memorizzazione secondaria e terziaria (hard disk, masterizzatori, ecc.)
 - dispositivi di comunicazione (schede di rete, modem, ecc.)
 - dispositivi per l'interazione con l'utente (tastiera, mouse, monitor, ecc.)
- le interfacce di I/O (dette anche controllori) fanno da tramite tra la CPU e dispositivi esterni

Interfacce di I/O

- I controllori sono dotati di registri che la CPU può leggere e scrivere per comunicare con il controllore e quindi con il dispositivo di I/O
- Due modalità di comunicazione:
 - I/O mappato in memoria: i registri vengono visti dalla CPU come se fossero celle di memoria; la lettura/scrittura avviene per mezzo delle operazioni di lettura/scrittura in memoria
 - apposite istruzioni per scrivere o leggere i registri dei controllori

Interrupt

- Quando un controller deve segnalare alla CPU che ci sono dei dati in arrivo da una periferica di input invia un segnale di interrupt
- In questo modo la CPU viene “avvertita” che ci sono dei dati da gestire
- La CPU interrompe l’esecuzione in corso e gestisce l’interrupt

Direct Memory Access

- In alcuni casi un controllore può comunicare direttamente con la memoria senza l'intervento della CPU
- Si parla di modalità DMA (Direct Memory Access)
- La modalità DMA viene usata soprattutto per dispositivi che trasferiscono grandi quantità di dati (dispositivi di memorizzazione secondaria o terziaria)

Il sistema operativo

Hardware e software

- Abbiamo visto che le due macro componenti del calcolatore sono:
 - hardware: insieme di tutti i componenti fisici del computer
 - software: insieme dei programmi che richiedono all'hardware di svolgere compiti specifici
- Si distingue tra:
 - software di base: programmi necessari al funzionamento del calcolatore, principalmente il sistema operativo
 - software applicativo: programmi che l'utente utilizza per i suoi scopi

Sistema operativo

- Il sistema operativo (SO) controlla e coordina l'uso dell'hardware da parte dei programmi applicativi
- Offre gli strumenti per utilizzare correttamente ed efficientemente le risorse di calcolo
- Fornisce un ambiente all'interno del quale gli altri programmi possono lavorare in modo utile
- Nelle prossime slide descriveremo i diversi componenti del SO

Kernel

- Il kernel (nucleo) è il cuore del SO
- Raggruppa le funzionalità fondamentali strettamente connesse all'hardware sottostante
- Tali funzionalità sono eseguite in modalità privilegiata (modalità kernel):
 - le altre funzionalità sono eseguite in modalità utente, cioè con delle restrizioni rispetto all'uso dell'hardware

Kernel

- Esistono diverse filosofie su quali servizi debbano far parte del kernel e quali no
 - sistemi con kernel monolitico: la maggior parte delle funzioni fanno parte del kernel
 - sistemi a microkernel: il kernel contiene poche funzionalità essenziali
- I sistemi a microkernel sono di solito più sicuri e più semplici da sviluppare e mantenere, ma hanno prestazioni inferiori

Scheduler e processi

- Uno dei compiti più importanti del SO è la gestione dell'esecuzione dei programmi
- Quando un programma viene avviato il SO lo carica in memoria e gli assegna una serie di risorse tra cui una certa quantità di memoria
- Un programma in esecuzione viene chiamato processo
 - un processo è caratterizzato dallo stato corrente della CPU (valore del PC, contenuto dei registri, ecc.) e della memoria ad esso riservata

Scheduler e processi

- Diversi processi possono essere in esecuzione contemporaneamente, ma esiste una sola CPU
- Il SO si occupa di suddividere l'uso della CPU tra i processi in esecuzione:
 - ogni processo opera come se avesse la CPU completamente a sua disposizione
 - l'utente ha l'impressione che i processi vengano eseguiti in contemporanea
- Il componente del SO che ripartisce l'uso della CPU tra i processi si chiama [scheduler](#)

Scheduler e processi

- Lo scheduler mantiene una coda dei processi in attesa di usare la CPU
- Assegna la CPU ad un processo per un breve intervallo di tempo
- Quando il tempo scade, la CPU viene assegnata ad un altro processo
- Quando un processo viene interrotto si salvano alcune informazioni sul suo stato ([contesto](#))
- Quando un processo viene avviato viene ripristinato il suo contesto

Scheduler e processi

- Lo scheduler deve
 - massimizzare l'uso della CPU
 - minimizzare i tempi di attesa dei processi
- In particolare i processi che richiedono l'interazione con l'utente non devono restare in attesa troppo a lungo

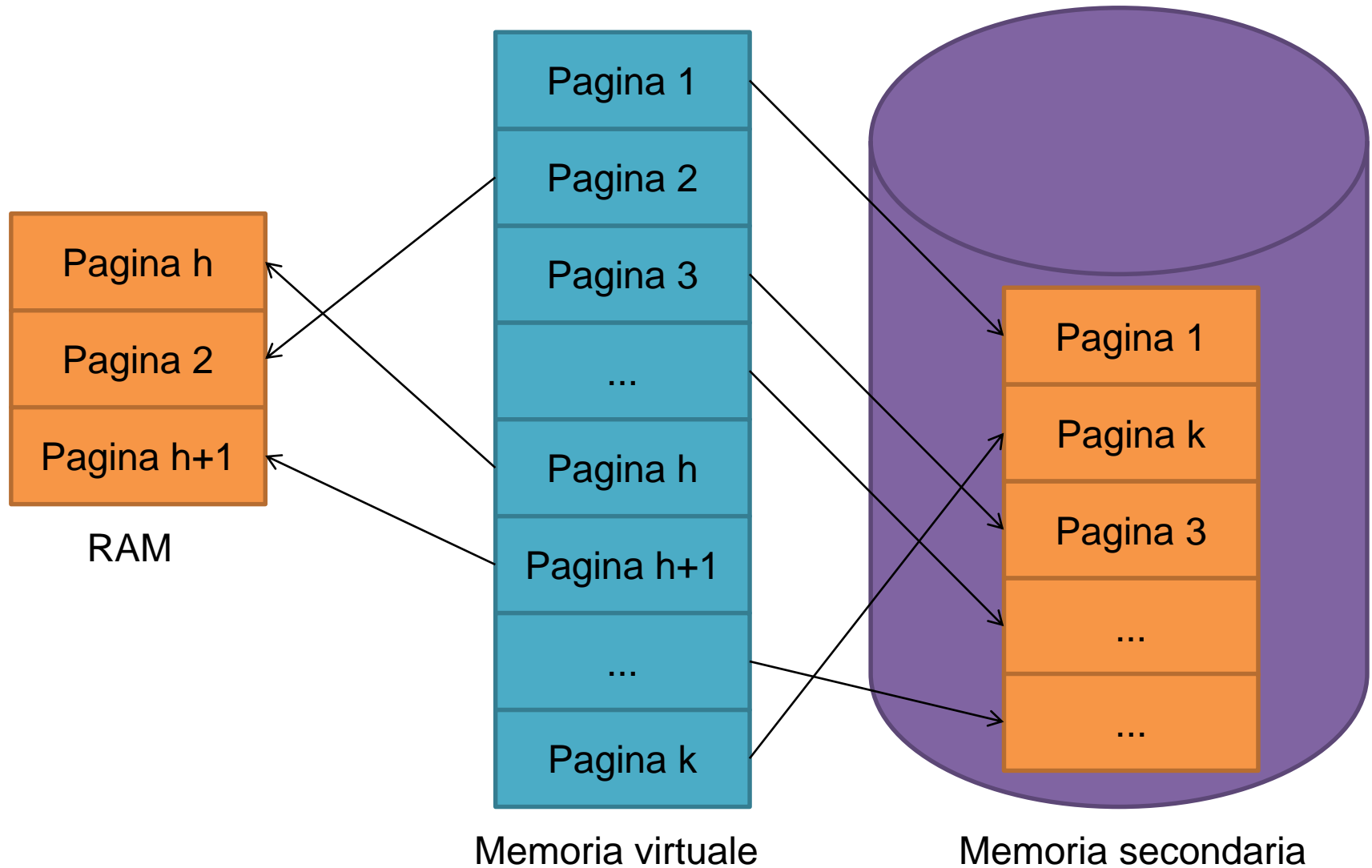
Gestore della memoria

- Ad ogni processo viene riservata una certa quantità di memoria
- I processi in esecuzione in un certo momento potrebbero aver bisogno di una quantità di memoria maggiore di quella disponibile
- La tecnica che permette di utilizzare più memoria di quella a disposizione si chiama [memoria virtuale](#)

Memoria virtuale

- Ogni processo ha a disposizione uno spazio di memoria virtuale con un certo spazio di indirizzi
- Sia lo spazio di memoria virtuale che la RAM sono divisi in pagine
- In ogni momento soltanto alcune delle pagine dello spazio virtuale si trovano in RAM
- Le restanti si trovano in memoria secondaria che funge da memoria di appoggio
- Il gestore della memoria si occupa di spostare le pagine dalla memoria secondaria alla RAM (e viceversa)

Memoria virtuale



Memoria virtuale

- Se un processo richiede di accedere ad una cella la cui pagina non è in RAM, il gestore della memoria:
 - se necessario sposta una pagina dalla RAM alla memoria secondaria
 - recupera la pagina richiesta dalla memoria secondaria
 - la carica in RAM
- Il gestore della memoria si occupa anche di mappare gli indirizzi virtuali negli indirizzi fisici

Gestore delle periferiche

- Consente ai programmi applicativi di utilizzare le periferiche ignorando i dettagli degli specifici dispositivi utilizzati
- Ad esempio, diversi programmi applicativi utilizzano le funzioni di stampa
- Chi scrive tali programmi non vuole scrivere codice diverso per ogni possibile stampante presente sul mercato

Gestore delle periferiche

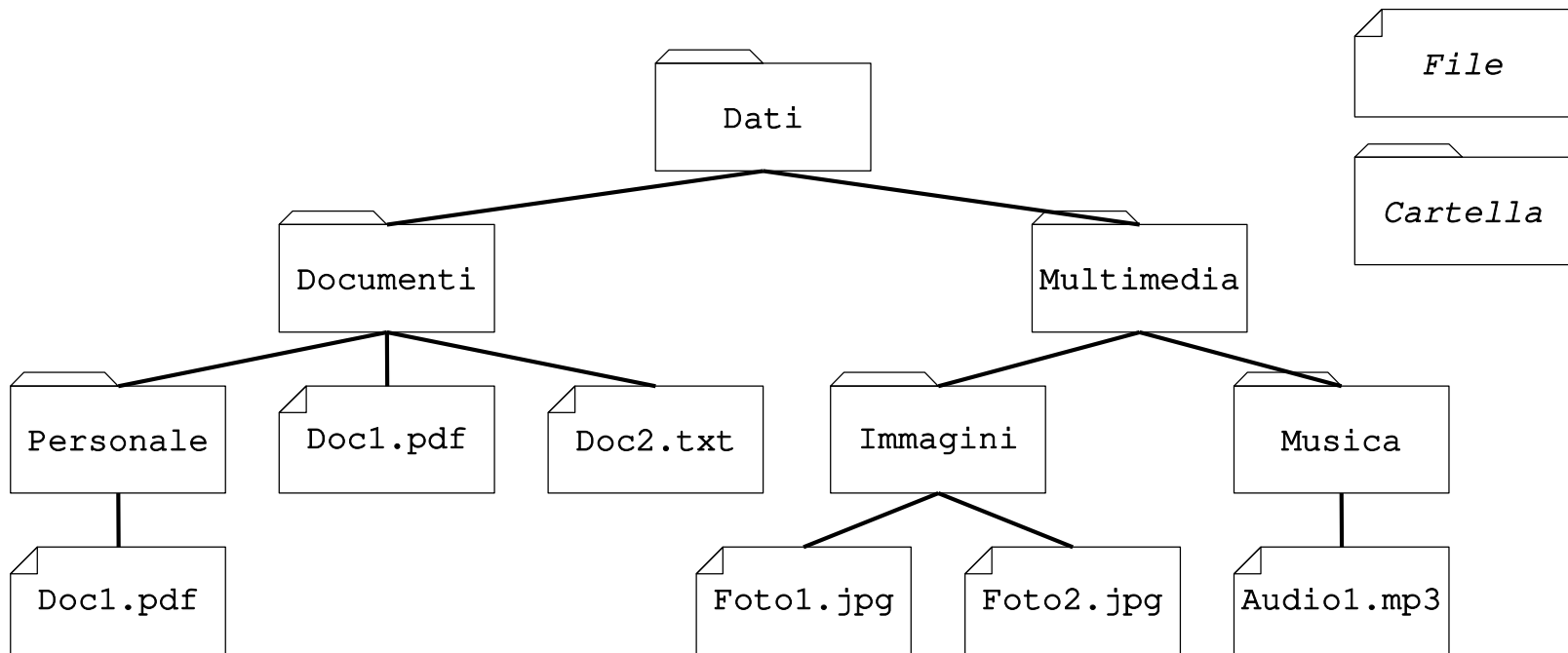
- Il gestore delle periferiche maschera le differenze tra le diverse stampanti reali offrendo ai programmi applicativi una “stampante virtuale” e delle funzioni per controllarla
- I programmi applicativi fanno riferimento a tale dispositivo virtuale
- Le differenze tra i dispositivi reali sono incapsulate in moduli software detti driver che:
 - vengono forniti dai produttori delle periferiche
 - vengono installati nel SO come moduli aggiuntivi del kernel

File system

- Il SO offre all'utente e ai programmi applicativi una rappresentazione logica in cui i dati in memoria secondaria sono organizzati in una struttura gerarchica chiamata [file system](#)
- Dal punto di vista logico i dati sono organizzati in [file](#)
- Un file è una sequenza di byte
 - che codifica una certa informazione di interesse per l'utente (un documento, una foto, un programma,...)
 - che è stata archiviata in memoria secondaria
 - a cui è stato assegnato un nome.

File system

- I file sono raggruppati in cartelle (o directory) che possono essere a loro volta raggruppate in altre cartelle



File system

- I dati archiviati nella memoria secondaria (o terziaria) sono memorizzati in maniera molto diversa da come appaiono nel file system
- Un file può essere spezzato in più blocchi memorizzati in porzioni della memoria secondaria non necessariamente contigue
- Il SO maschera la realtà mostrando i dati come appaiono nel file system
- Il SO offre anche una serie di funzionalità per operare sui file

File system: dettagli

- Come abbiamo visto l'elemento principale del file system è il file
- Un file ha un nome, di solito costituito dal nome vero e proprio più un'estensione:
 - l'estensione è costituita da un punto seguito da pochi caratteri (.txt, .exe, ...)
 - l'estensione di solito indica il tipo di file
- Oltre al nome un file ha altri attributi:
 - dimensione in byte
 - data di creazione e data dell'ultima modifica
 - ...

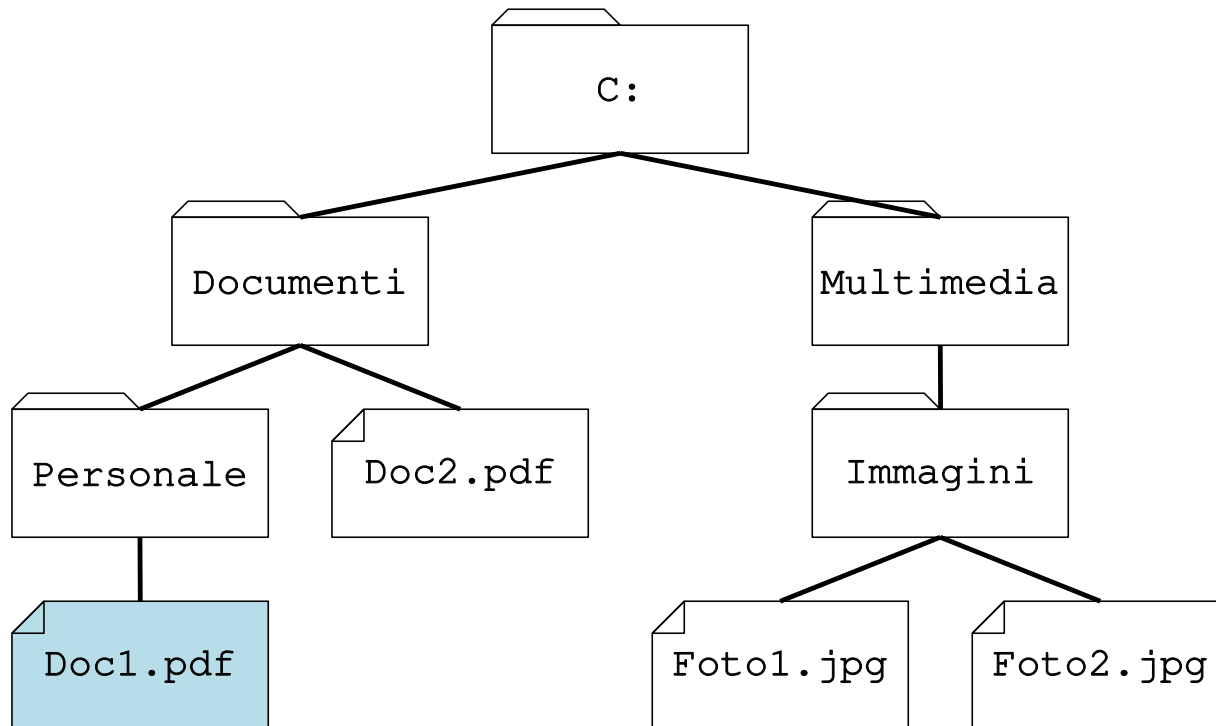
Percorsi

- Un attributo importante per identificare un file è il suo percorso (path) cioè la sua posizione nel file system
- La cartella più in alto del file system (quella che contiene tutti i file e le altre cartelle) è la radice del file system:
 - nei sistemi Unix/Linux la radice è indicata con il carattere slash (/)
 - nei sistemi Windows la radice è indicata da una lettera seguita dai due punti e da un backslash (C:\)

Percorsi

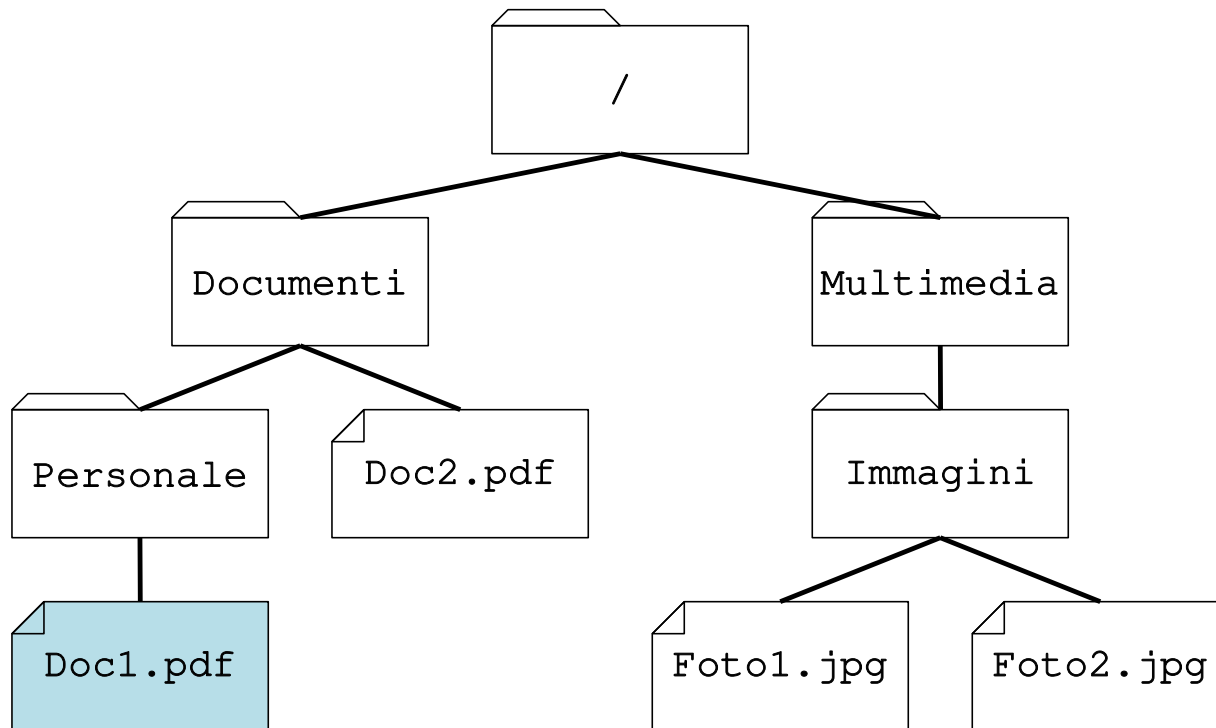
- Il percorso assoluto di un file è la sequenza di cartelle che bisogna attraversare per andare dalla radice del file system al file in questione
- Il percorso assoluto viene indicato elencando le cartelle che lo costituiscono separate da un carattere di separazione:
 - nei sistemi Unix/Linux il carattere slash (/)
 - nei sistemi Windows il carattere backslash (\)

Percorsi: esempi



C:\Documenti\Personale\Doc1.pdf

Percorsi: esempi

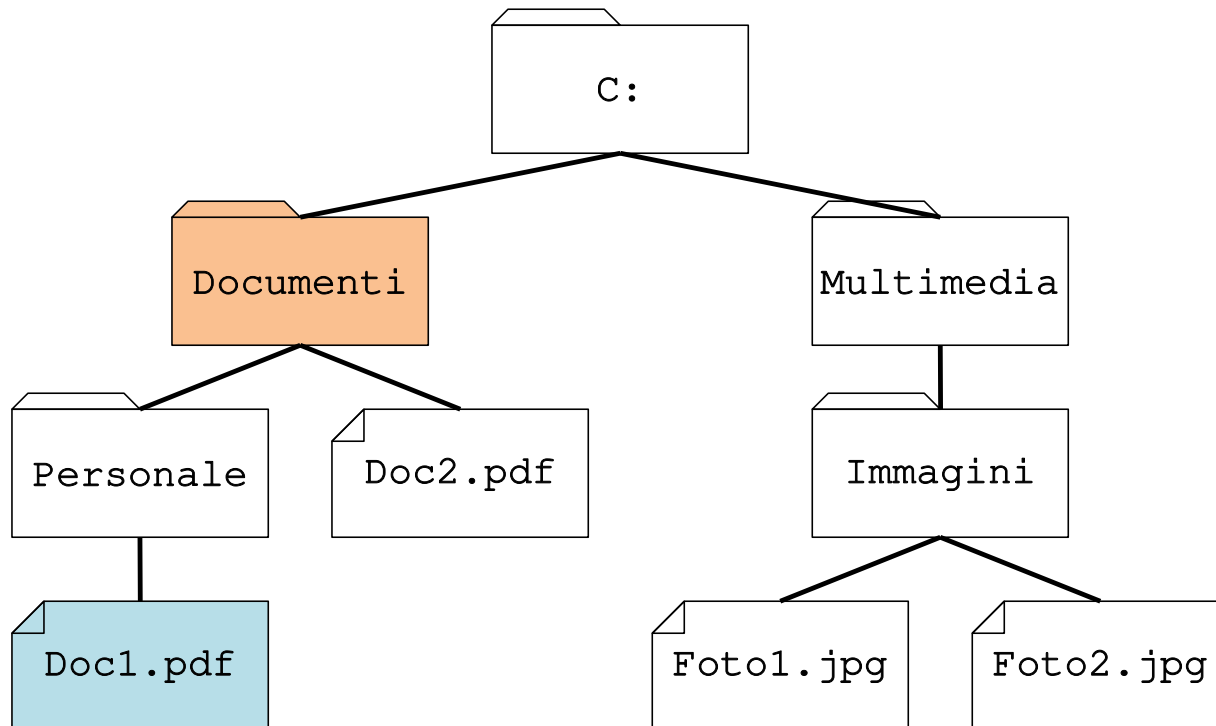


`/Documenti/Personale/Doc1.pdf`

Percorsi

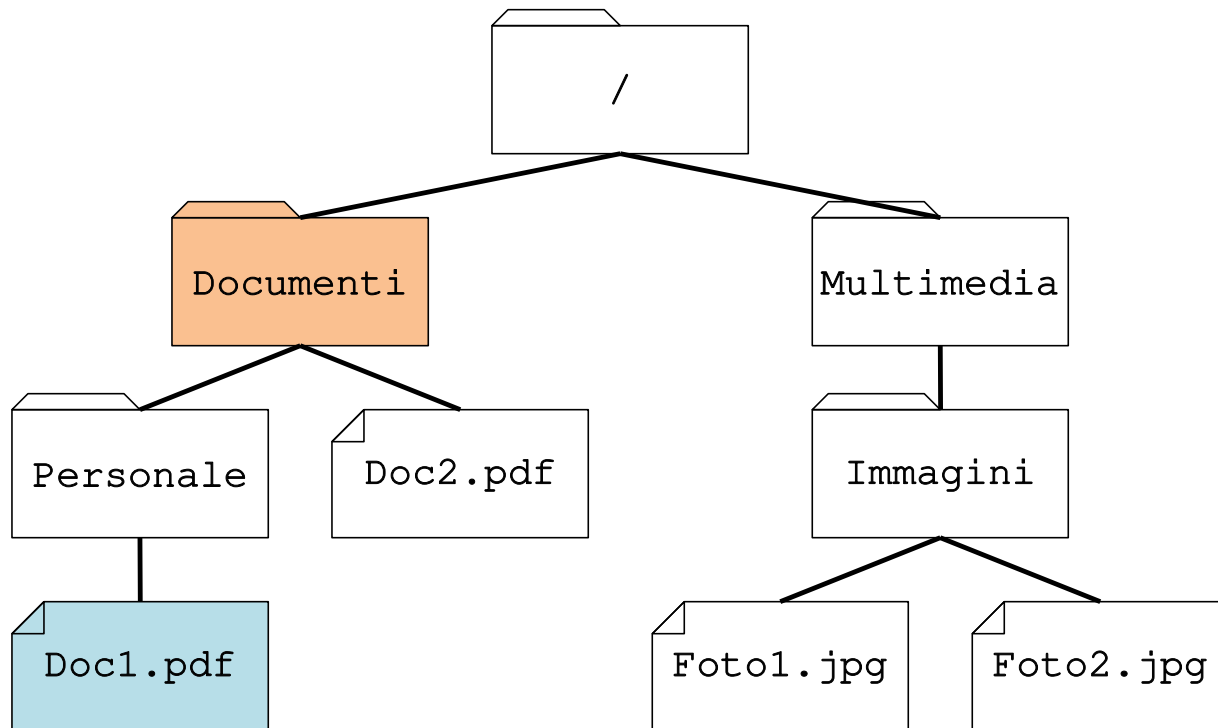
- Il percorso di un file può essere specificato anche relativamente ad una cartella di riferimento
 - di solito la cartella in cui si sta lavorando
- In questo caso si parla di percorso relativo
- Il percorso va specificato indicando la sequenza di cartelle che bisogna visitare per andare dalla cartella di riferimento al file di interesse

Percorsi: esempi



Personale\Doc1.pdf

Percorsi: esempi

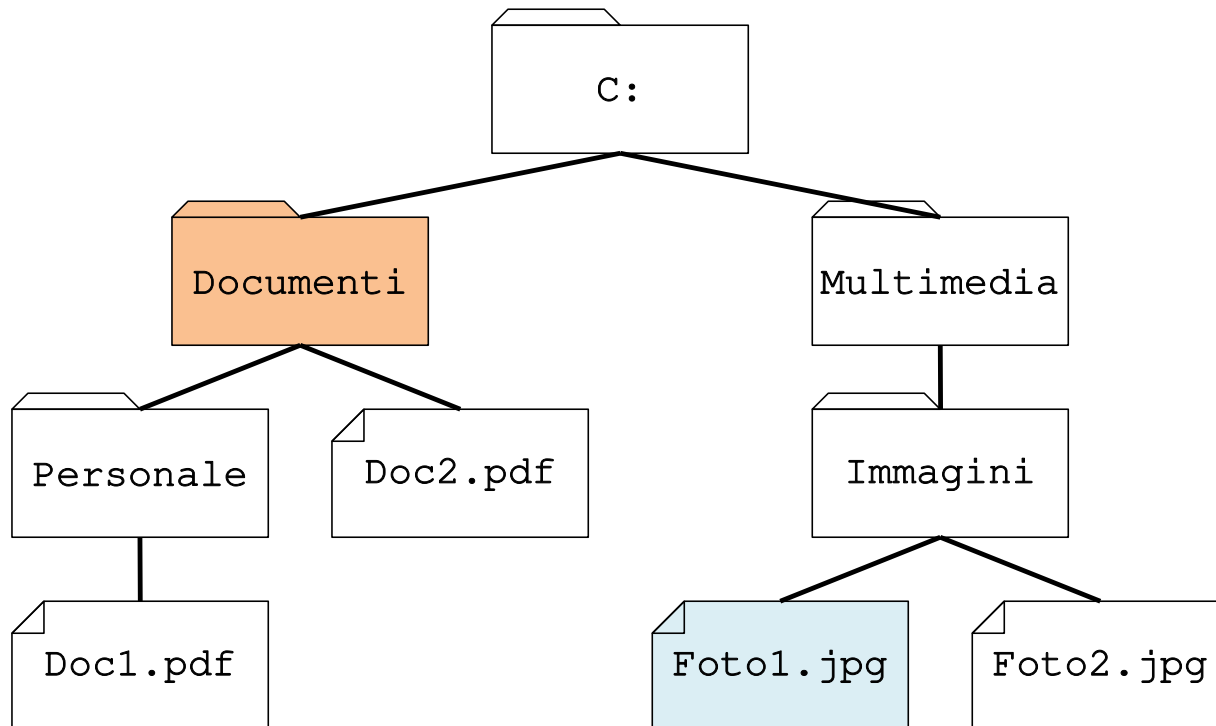


Personale/Doc1.pdf

Percorsi

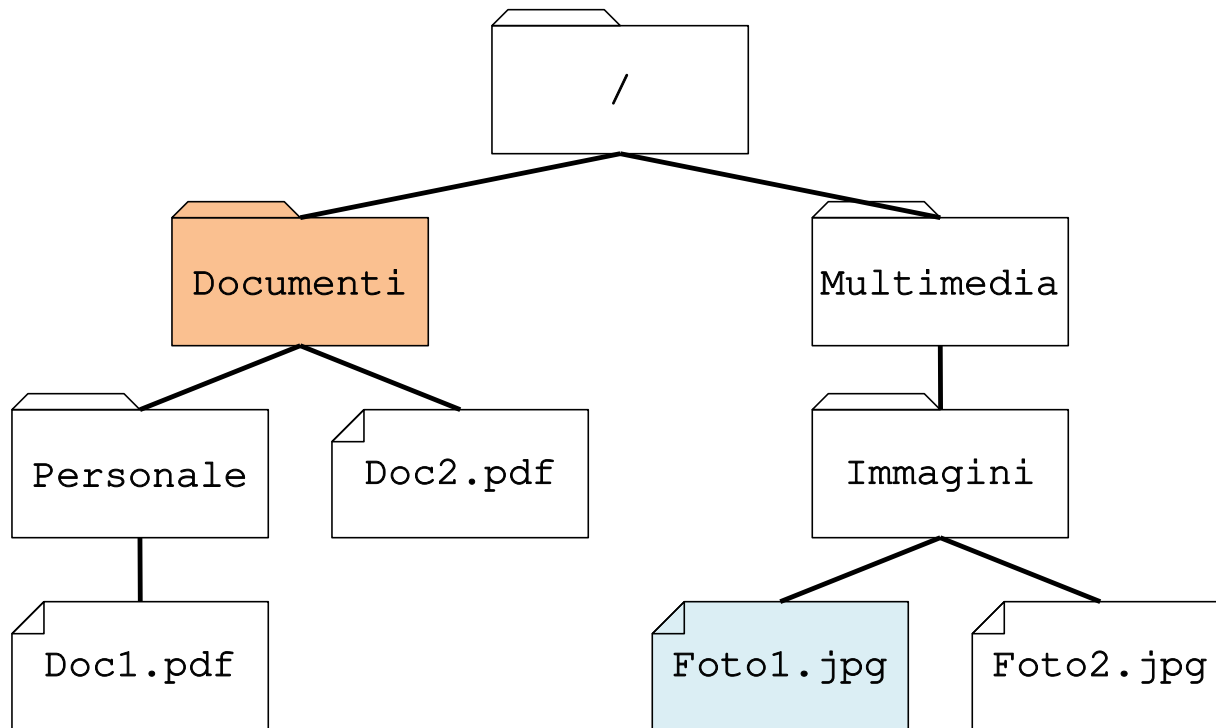
- Nello specificare un percorso si possono usare anche due caratteri speciali:
 - il punto (.) che indica la directory corrente
 - il doppio punto (..) che indica la directory superiore
- Oltre i file anche le directory possono essere identificate tramite percorsi assoluti o relativi

Percorsi: esempi



..\Multimedia\Immagini\Foto1.jpg

Percorsi: esempi



../Multimedia/Immagini/Foto1.jpg

L'interfaccia utente

- L'interfaccia utente è il componente che consente all'utente di interagire con il SO e, di conseguenza, con il calcolatore
- Tramite essa l'utente può inviare comandi al SO ed ottenere da esso informazioni di vario tipo
- L'utente può, ad esempio:
 - esplorare il file system
 - avviare i programmi applicativi
 - configurare i dispositivi hardware
 - modificare le impostazioni del SO
 - ...

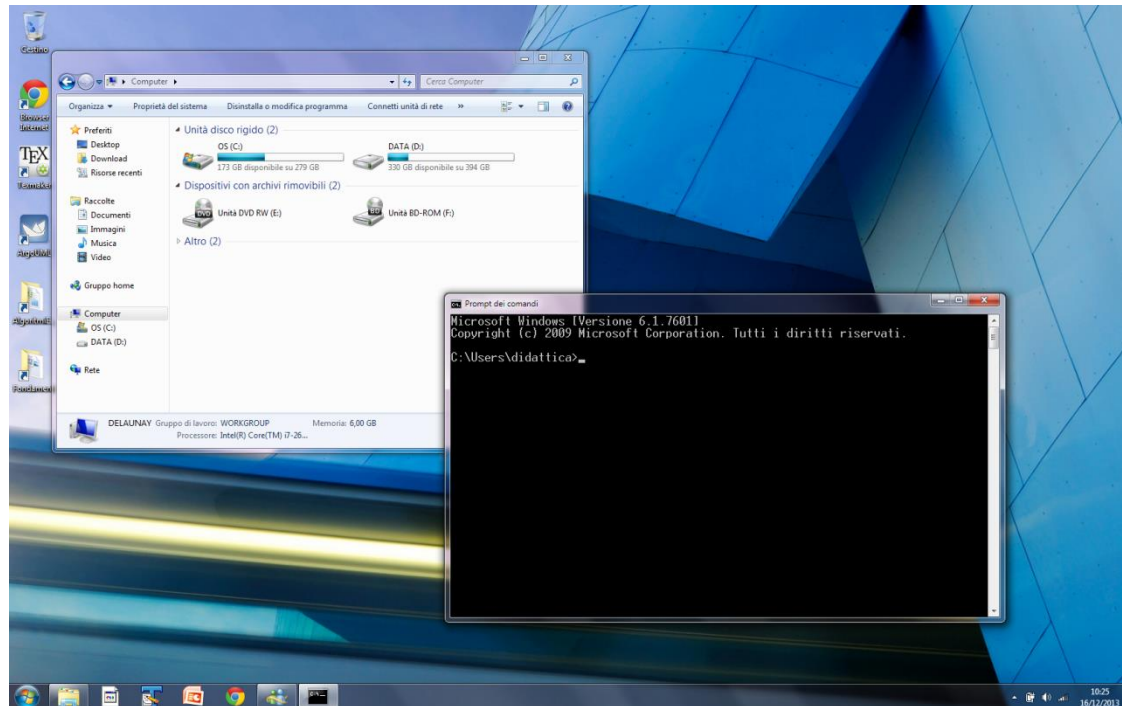
Due tipi di interfacce

- Interfaccia da linea di comando:
 - l'utente digita dei comandi in forma testuale
 - ottiene un output in forma testuale

```
digiacomo@mozart:/var/lib$ ls
apparmor  initramfs-tools  mysql          screen-profiles  update-notifier
apt       initscripts      mysql-cluster  security          urandom
aptitude  landscape        ntp            sendmail         vim
belocs    libuuid          pam            sgml-base       x11
binfmts   libxml-sax-perl  php5          tomcat6         xml-core
dbus      locales          postgresql    ucf
defoma    logrotate        pycentral     ufw
dhcp3     misc             python-support update-manager
dpkg      mlocate         samba         update-motd
digiacomo@mozart:/var/lib$ _
```

Due tipi di interfacce

- Interfaccia grafica:
 - basata sulla metafora della scrivania (desktop)
 - l'utente interagisce con gli elementi grafici tramite il mouse



Due tipi di interfacce

- Anche nei sistemi basati su interfacce grafiche è possibile un'interazione testuale tramite emulatori dei terminali testuali:
 - ad es. il prompt dei comandi di Windows

