

---

# Cenni all'architettura dei calcolatori

---

Walter Didimo

# Hardware e Software

---

La prima decomposizione di un calcolatore è relativa ai seguenti macro-componenti

- hardware

- Insieme delle sue componenti elettroniche e meccaniche

- software

- l'insieme dei programmi che consentono all'hardware di svolgere dei compiti utili
- il software si divide in *software di base* (tra cui il sistema operativo) e il *software applicativo*

# Macchina di Von Neumann

---

L'architettura dell'hardware di un calcolatore reale è molto complessa

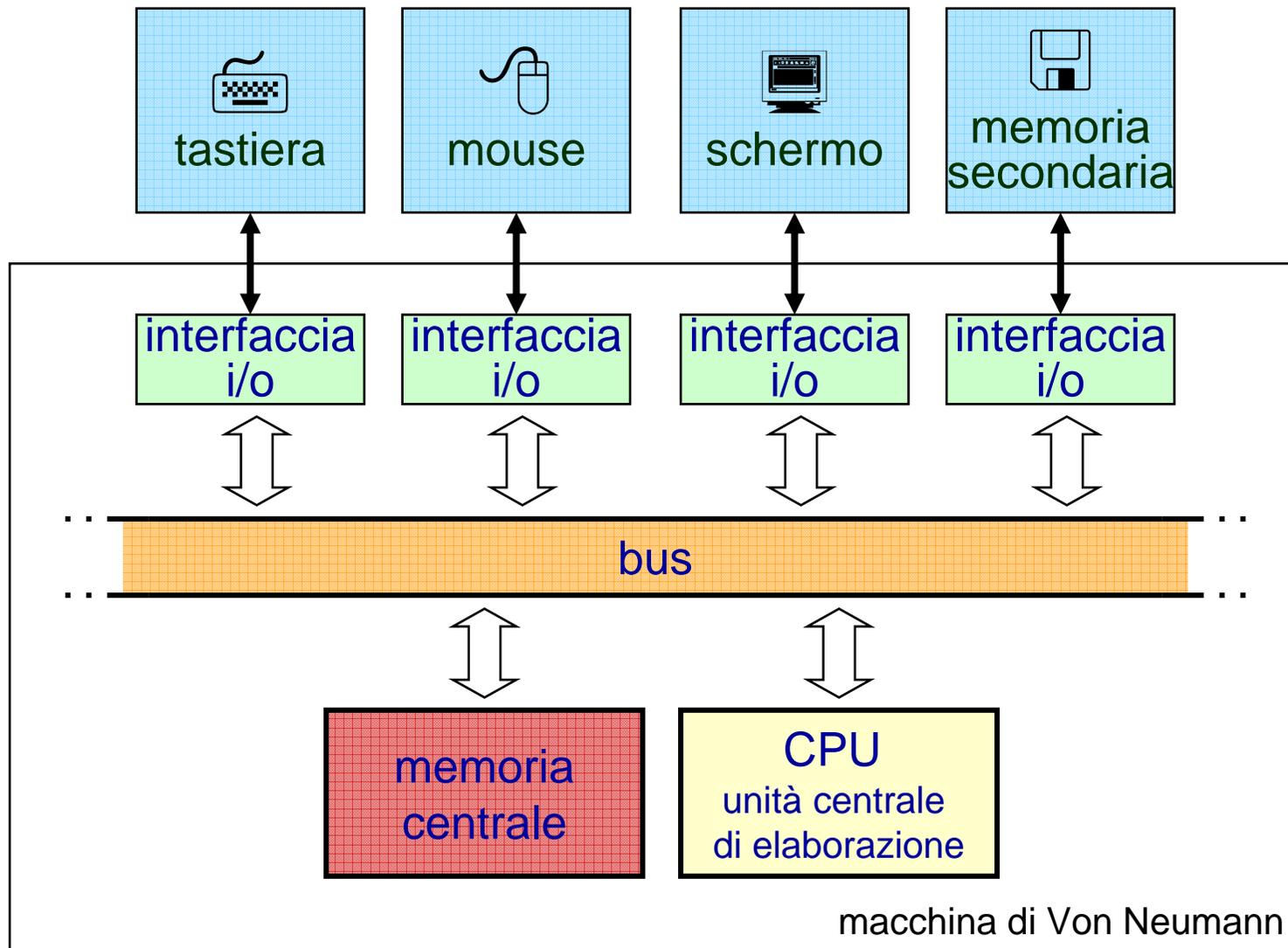
- Introdurremo la **macchina di Von Neumann**, che è un modello semplificato dei calcolatori moderni
- Von Neumann è stato il progettista (intorno al 1950) del primo calcolatore in cui i programmi potevano essere memorizzati anziché codificati mediante cavi e interruttori

# Elementi della macchina di Von Neumann

---

- unità centrale di elaborazione (CPU)
  - esegue istruzioni per l'elaborazione dei dati
  - controlla e coordina altre componenti funzionali
- memoria centrale
  - memorizza e fornisce l'accesso a dati e programmi
- interfacce di ingresso e uscita
  - permettono di collegare il calcolatore a componenti periferiche (considerate esterne al calcolatore), quali monitor, stampante, mouse, ecc.
- bus
  - permette il trasferimento dei dati e di informazioni di controllo tra le varie componenti funzionali

# Architettura della macchina di Von Neumann



# Codifica dei dati

---

In un calcolatore, i dati (e le istruzioni) sono codificati in forma binaria, cioè come sequenze finite di cifre 0 e 1

- il bit è la più piccola unità di informazione in un calcolatore – può avere valore 0 oppure 1
  - la parola “bit” è una forma contratta per binary digit (cifra binaria)
- ciascun bit è memorizzato in una cella elementare di memoria, realizzata come dispositivo elettronico in cui sono chiaramente distinguibili due stati (associati a 0 e 1)
- i dati vengono codificati sotto forma di *sequenze di bit*

# Byte

---

Un'altra importante unità di informazione è il byte

- un byte è una sequenza di 8 bit

Le possibili combinazioni degli 8 bit in un byte sono  $2^8 = 256$

- un byte può essere utilizzato per rappresentare un valore tra 256 diversi possibili valori (es. un piccolo intero, un carattere in un alfabeto che contiene non più di 256 caratteri, ...)
- quindi, l'interpretazione di ciascun byte (ovvero, sequenza di 8 bit) deve avvenire opportunamente, in funzione del dato che esso rappresenta

# Linguaggio Macchina

---

Una CPU sa eseguire istruzioni in un linguaggio noto come linguaggio macchina. Esso prevede operazioni elementari di manipolazione dei dati, tra le quali:

- *operazioni aritmetiche* (somma, prodotto, sottrazione e divisione)
- *operazioni relazionali* (confronto tra dati)
- *operazioni di accesso e memorizzazione dati*

Un calcolatore svolge dunque poche tipologie di operazioni, ma in modo molto efficiente

- può eseguire centinaia di milioni di istruzioni del linguaggio macchina al secondo

# Elaborazione e Controllo

---

La CPU ha due sottocomponenti:

- L'unità aritmetico-logica (ALU): elabora i dati
- L'unità di controllo: coordina tutte le componenti

L'attività di controllo svolta dall'unità di controllo

- avviene in modo sincrono rispetto alla scansione temporale imposta dall'*orologio di sistema (clock)*
- è un coordinamento dell'esecuzione temporale delle operazioni elementari che debbono essere svolte, sia nella CPU sia dalle altre componenti

# Memorizzazione

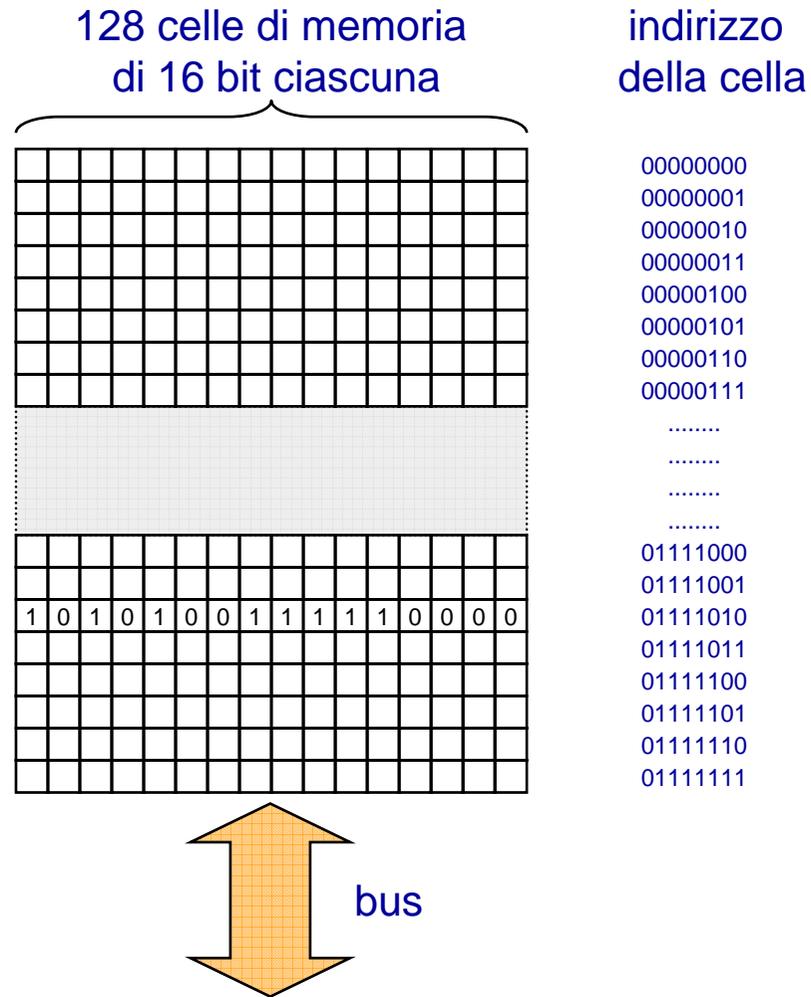
---

La memoria centrale è la componente del calcolatore in cui vengono immagazzinati e da cui vengono acceduti i dati e i programmi

- può essere acceduta direttamente dalla CPU
- si compone di celle (o locazioni)
- ogni cella di memoria può memorizzare una parola (ovvero, un sequenza di bit di lunghezza fissata)
- ogni cella è caratterizzata da
  - un *indirizzo*, che è un numero che identifica la cella e ne consente l'accesso
  - un *valore*, che è la sequenza di bit memorizzata dalla cella

# Struttura di una memoria centrale

---



# Trasferimento

---

Il bus è il componente del calcolatore dedicato al trasferimento dei dati e di informazioni di controllo tra le varie componenti

- il bus ha un certo numero di piste su cui “viaggiano” i dati

## *Vantaggi del bus*

- ci sono due modalità per permettere la comunicazione tra i componenti di un calcolatore
  - collegare ciascun componente con ogni altro componente
  - collegare tutti i componenti a un unico insieme di linee (il bus, appunto)
- l'uso del bus favorisce la modularità e l'espandibilità del calcolatore

# Periferiche e interfacce di ingresso-uscita

---

Un calcolatore può essere collegato a diversi dispositivi di ingresso e/o uscita (chiamati periferiche)

- ad esempio, la tastiera, il mouse, lo schermo, le stampanti, il modem, gli hard-disk ed i lettori CD/DVD;
- ogni periferica viene controllata mediante un'opportuna interfaccia
- una interfaccia deve tradurre i segnali interni del calcolatore in un formato comprensibile alla periferica stessa, e viceversa

# Caratteristiche fisiche delle componenti

# Microprocessore

---

L'unità centrale di elaborazione è fisicamente realizzata sotto forma di microprocessore

- dispositivi elettronici estremamente complessi, formati da milioni di circuiti in spazi molti ridotti

# Caratteristiche dei microprocessori

---

- *repertorio di istruzioni*: le istruzioni del linguaggio macchina
- *velocità* (misurata come frequenza del clock)
  - la frequenza del clock misura la durata del ciclo macchina (l'unità di tempo all'interno del processore)
  - l'esecuzione di una istruzione può richiedere più cicli macchina
- *ampiezza del bus*: numero di bit (piste) nel bus di sistema
- *co-processore*: i moderni processori sono integrati a co-processori specializzati (ad esempio, il co-processore matematico)
- *cache*: una memoria veloce locale al microprocessore, che consente un'accelerazione nell'esecuzione dei programmi

# Caratteristiche delle memorie

---

## ■ *capacità*

- il numero di bit che possono essere memorizzati, misurati in byte (e multipli del byte)
- 1Kbyte indica  $2^{10}$  byte = 1024 byte
- 1Mbyte indica  $2^{20}$  byte = 1024 Kbyte = 1 048 576 byte
- 1Gbyte indica  $2^{30}$  byte = 1024 Mbyte = 1 073 741 824 byte

## ■ *velocità di accesso*

- velocità di esecuzione delle operazioni di lettura/scrittura

## ■ *volatilità*

- le memorie RAM (random access memory) possono essere sia lette che scritte, ma i dati memorizzati vengono persi allo spegnimento del calcolatore
- le memorie ROM (read only memory) possono essere solo lette, e i dati sono memorizzati in modo permanente

# Memorie secondarie

---

Una memoria secondaria ha alcune caratteristiche fondamentali che la differenziano dalla memoria centrale

- *non volatilità*

- i dati memorizzati non vengono persi allo spegnimento del calcolatore (perché memorizzati in forma magnetica o ottica e non elettronica)

- *grande capacità*

- una unità di memoria secondaria ha capacità maggiore (anche di diversi ordini di grandezza) rispetto alla memoria centrale

- *bassi costi*

- il costo per bit di una memoria secondaria è minore (di diversi ordini di grandezza) rispetto alla memoria centrale

# Memorie secondarie

---

Altre caratteristiche delle memorie secondarie rispetto a quelle centrali:

- *minore velocità di accesso*: i tempi di accesso a una memoria secondaria sono maggiori (di qualche ordine di grandezza) rispetto alla memoria principale
- *dati non direttamente accessibili*: i dati di una memoria secondaria per essere acceduti dal microprocessore devono comunque transitare nella memoria centrale

Nella pratica

- la memoria secondaria memorizza tutti i programmi e i dati del calcolatore
- la memoria centrale memorizza i programmi in esecuzione e i dati necessari per la loro esecuzione

# Dischi magnetici

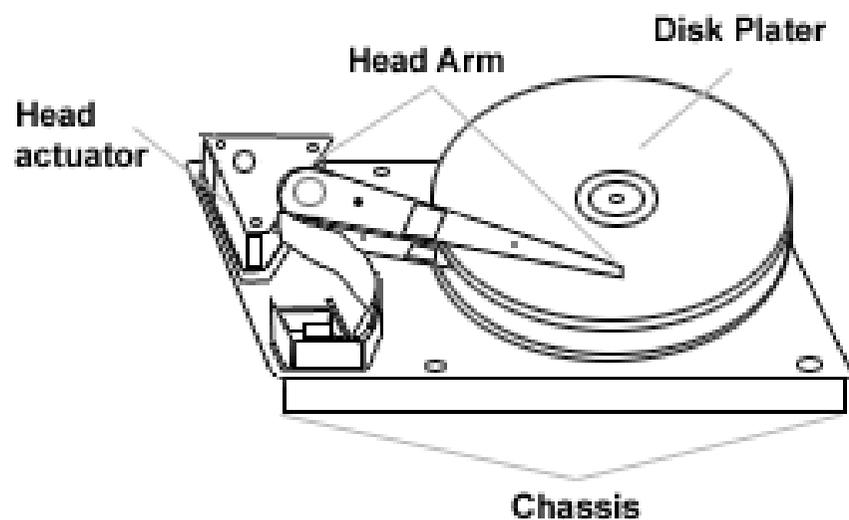
---

Un disco magnetico (disco fisso o hard disk) è composto da uno o più piatti di alluminio rotanti ricoperti di materiale magnetico e da testine

- una *testina* permette la scrittura e lettura di bit su un disco, memorizzati sotto forma di stati di polarizzazione (positiva e negativa)
- ciascun piatto è composto da due *superfici* (*facce*)
- ciascuna faccia è suddivisa in *tracce* (circolari) e *settori* (a spicchio)
  - un *cilindro* è l'insieme delle tracce in una data posizione radiale
- i settori sono le unità logiche di memorizzazione

# Dischi magnetici

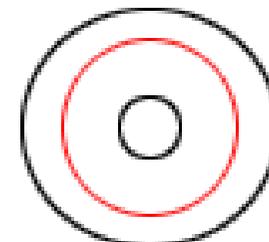
---



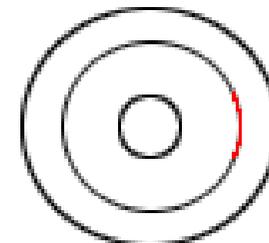
<http://www.computerhope.com>



Platter



Tracks



Sector



Cylinder

<http://www.computerhope.com>

# Il Sistema Operativo

# Il sistema operativo

---

Il sistema operativo è il software di base responsabile della gestione delle risorse del calcolatore

- ad esempio, l'utente può richiedere al sistema operativo di eseguire un programma, facendo un "doppio click" sulla corrispondente icona
- per avviare l'esecuzione del programma, il sistema operativo svolge le seguenti attività
  - individua il codice eseguibile del programma (residente sul disco)
  - alloca al programma le risorse necessarie per la sua esecuzione (ad esempio, una certa quantità di memoria centrale)
  - carica il codice eseguibile del programma in memoria centrale
  - avvia il programma

# Architettura di un sistema operativo

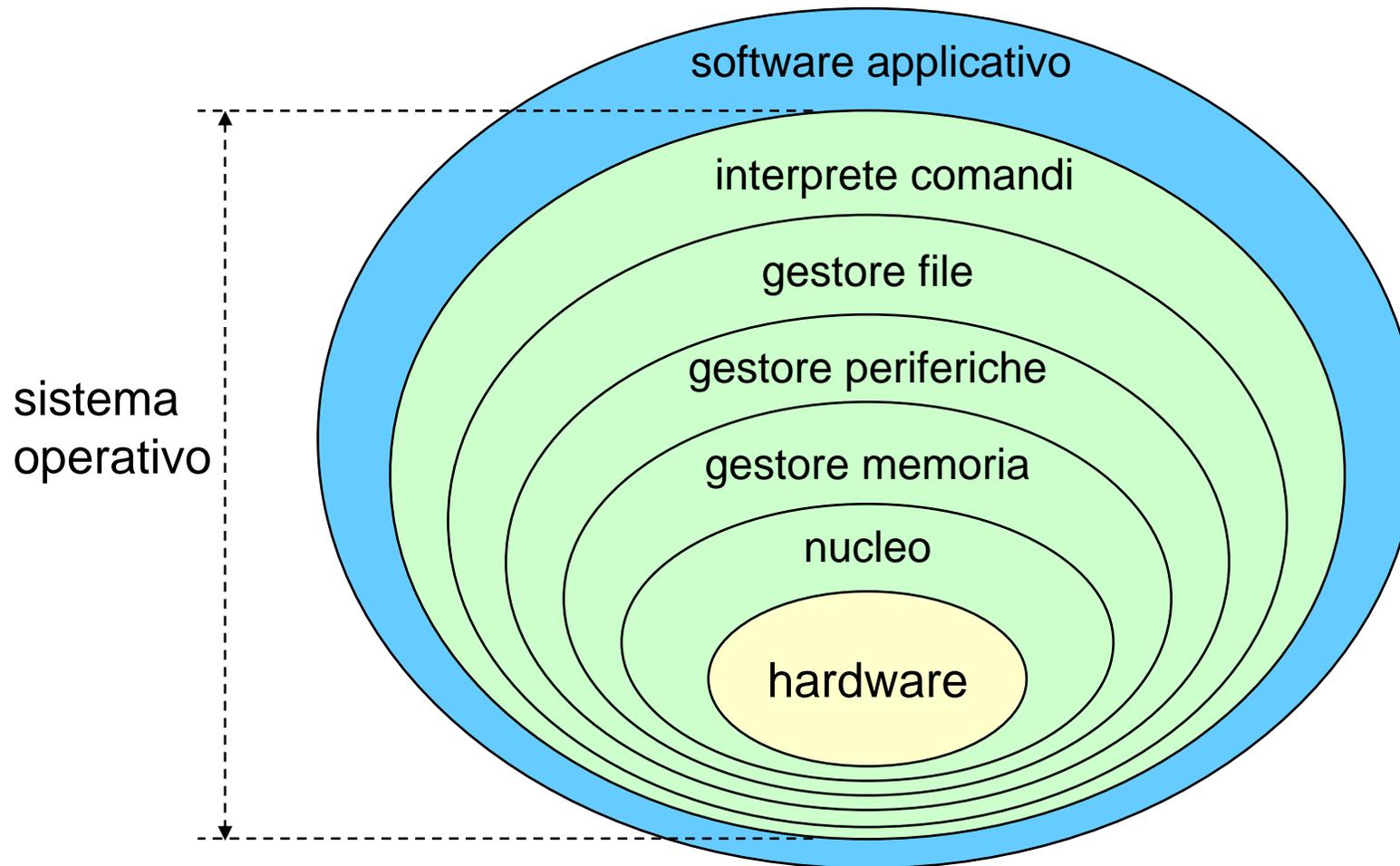
---

I sistemi operativi hanno una struttura complessa, che può essere descritta come una *gerarchia di livelli*

- ciascun livello gestisce una diversa tipologia di risorse
  - le risorse gestite sono i processori, la memoria, le periferiche, le memorie secondarie, l'interfaccia utente, ecc.
- l'utente del calcolatore interagisce solo con *l'interprete comandi*, che gestisce l'interfaccia del sistema operativo

# I livelli di un sistema operativo

---



# Nucleo

---

Il nucleo del sistema operativo è responsabile della gestione della risorsa microprocessore

- sotto il nucleo esiste fisicamente una sola unità di elaborazione, quindi il calcolatore potrebbe svolgere un solo programma alla volta
- il nucleo definisce tante unità di elaborazione virtuali, una per ciascun programma che si vuole eseguire
  - la capacità dell'unica unità di elaborazione reale è ripartita tra le tante unità di elaborazione virtuali
- il nucleo gestisce anche l'eventuale presenza di più unità di elaborazione (architetture parallele)

# Gestore della memoria

---

Il gestore della memoria consente l'allocazione dinamica della memoria centrale ai programmi in esecuzione

- a ciascun programma viene allocata un'area di memoria virtuale, sufficiente per la sua esecuzione
- il gestore della memoria gestisce la corrispondenza tra le memorie virtuali e l'unica memoria reale
- la dimensione della memoria virtuale può essere maggiore di quella reale; i dati possono essere parcheggiati temporaneamente nella memoria secondaria

# Gestore delle periferiche

---

Il gestore delle periferiche adatta la modalità d'uso delle singole periferiche (che possono essere estremamente diverse) a quello di poche tipologie di periferiche virtuali

- ad esempio, un programma può usare una stampante senza conoscerne i dettagli di funzionamento
- un driver è un software che adatta i dati dal formato virtuale (generico) usato sopra al gestore delle periferiche a quello reale della periferica effettivamente in uso

# Gestore dei file

---

Il gestore dei file (o file system) si occupa di gestire le memorie secondarie (dischi)

- Organizza logicamente la memoria secondaria in *volumi*, *direttori (cartelle)* e *file*
  - un file è una sequenza di dati di dimensioni variabili, con accesso sequenziale oppure casuale
  - Un direttorio è il contenitore di un insieme di file e sottodirettori
  - un volume è un contenitore di file e direttori, solitamente corrispondente ad una unità a disco
- Il gestore dei file mostra ai livelli superiori anche le periferiche come dei direttori, da cui è possibile leggere e/o su cui è possibile scrivere file

# Interprete comandi

---

L'interprete comandi gestisce l'interfaccia utente del sistema operativo

- definisce i **comandi** che l'utente può utilizzare
- può offrire una interfaccia
  - testuale – *shell dei comandi*
  - grafica — come nei sistemi operativi Windows e negli ambienti X di Unix/Linux

# Programmi di utilità

---

Il sistema operativo è corredato da un insieme di programmi di utilità, per la configurazione, l'ottimizzazione e la gestione ordinaria del sistema

- *gestione degli utenti*
- *gestione delle periferiche*
- *gestione delle applicazioni*
- *altri accessori*

# Rappresentazione binaria di numeri

# Rappresentazione posizionale di numeri naturali

---

La rappresentazione posizionale di un numero naturale  $N$ , consente di esprimere  $N$  in una qualunque base  $b$

$$N = c_n b^n + \dots + c_4 b^4 + c_3 b^3 + c_2 b^2 + c_1 b^1 + c_0 b^0$$

Dove ogni  $c_i \in \{0, \dots, b-1\}$  è detta cifra. Si scrive  $N_b = c_n c_{n-1} \dots c_2 c_0$

Esempio. Se  $b = 10$  (notazione decimale classica) ed  $N_{10} = 120$

$$120 = 1 \cdot 100 + 2 \cdot 10 + 0 \cdot 1$$

# Rappresentazione binaria di numeri naturali

---

In accordo con la rappresentazione posizionale, possiamo rappresentare un numero naturale in notazione binaria, cioè con base  **$b = 2$**

Esempio. Se  $b = 2$  (notazione binaria) ed  $N_{10} = 120$

$$\begin{aligned} N_{10} = 120 &= 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 \\ &= 1\ 1\ 1\ 1\ 0\ 0\ 0 = N_2 \end{aligned}$$

Esempio. Se  $b = 2$  (notazione binaria) ed  $N_{10} = 33$

$$N_{10} = 33 = 1 \cdot 32 + 0 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 1\ 0\ 0\ 0\ 0\ 1 = N_2$$

# Rappresentazione di numeri in base $b$

---

Per rappresentare un numero naturale  $N_{10}$  in base  $b$  si effettuano ripetutamente divisioni di  $N_{10}$  per  $b$ , e si pongono le cifre (da destra verso sinistra) uguali ai resti

Esempio. Se  $b = 2$  (notazione binaria) ed  $N_{10} = 33$

1 divisione:  $33 / 2 = 16$  con resto 1

2 divisione:  $16 / 2 = 8$  con resto 0

3 divisione:  $8 / 2 = 4$  con resto 0

4 divisione:  $4 / 2 = 2$  con resto 0

5 divisione:  $2 / 2 = 1$  con resto 0

6 divisione:  $1 / 2 = 0$  con resto 1

$$\textcircled{1} \cdot 32 + \textcircled{0} \cdot 16 + \textcircled{0} \cdot 8 + \textcircled{0} \cdot 4 + \textcircled{0} \cdot 2 + \textcircled{1} \cdot 1 = 100001$$

# Somma di numeri binari

---

La somma tra due numeri binari si fa come per i numeri decimali, considerando che:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 1 = 0 \text{ con il riporto di } 1$$

$$\text{Esempio: } 1011 + 0010 = 1101 \quad (\text{in binario})$$

$$11 + 2 = 13 \quad (\text{in decimale})$$

$$\text{Esempio: } 1111 + 0001 = 10000 \quad (\text{in binario})$$

$$15 + 1 = 16 \quad (\text{in decimale})$$

# Rappresentazione binaria di numeri negativi

---

Esistono diverse convenzioni adottate per rappresentare un numero naturale negativo in notazione binaria. La più diffusa è la così detta notazione in complemento a due

Si usa la cifra più a sinistra (la più rappresentativa) in negativo. Se cioè la cifra più a sinistra è un 1 allora il numero si intende negativo ed il suo valore assoluto è la differenza di questa potenza con le altre.

Esempio:  $N_{10} = -33$

$$-33 = 1 \cdot 64 + 0 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 1011111$$

$$-33 = -64 + 31$$

# Rappresentazione binaria su k cifre

I calcolatori usano rappresentare i numeri di un certo dominio con un fissato numero di cifre. Con  $k$  cifre si possono rappresentare  $2^k$  numeri differenti

Esempio: Con 2 cifre posso rappresentare 4 numeri: 00, 01, 10, 11

- ◆ Se vogliamo gestire solo numeri naturali su  $k$  cifre, il calcolatore userà le  $k$  cifre per rappresentare i numeri nell'intervallo  $[0, 2^k - 1]$
- ◆ Se vogliamo gestire numeri interi relativi su  $k$  cifre, il calcolatore userà le  $k$  cifre per rappresentare i numeri nell'intervallo  $[-2^{k-1}, 2^{k-1} - 1]$ 
  - *Per rappresentare i numeri negativi si usa la notazione del complemento a due (la potenza più a sinistra di tutte va intesa con segno negativo)*

# Complemento a due

---

Per snellire i calcoli, ecco un procedimento per rappresentare in complemento a due un numero negativo  $N_{10}$  su  $k > 1$  cifre.

- ◆ Si rappresenta  $|N_{10}|$  in notazione binaria sulle prime  $k-1$  cifre più a destra (la  $k$ -esima cifra più a sinistra si mette a 0)
- ◆ Si sostituiscono gli 1 con gli 0 e viceversa (complemento ad uno)
- ◆ Si somma 1 al numero ottenuto

Esempio: se  $k=8$  ed  $N_{10} = -33$

- ◆  $33 = 00100001$
- ◆ complemento ad uno =  $11011110$
- ◆ sommando 1 al complemento ad uno si ottiene:  $11011111$

# Overflow

---

Se ho un numero fissato di cifre per rappresentare i miei numeri, allora le operazioni di somma possono dar luogo a problemi di overflow (traboccamento), fornendo risultati sbagliati

- ciò avviene tutte le volte che la somma fornisce un numero al di fuori dell'intervallo che può essere rappresentato

Esempio: Se dispongo di 4 cifre e voglio rappresentare solo numeri naturali, allora posso rappresentare i numeri nell'intervallo [0, 15]

La somma tra 9 e 8, ad esempio, darà luogo ad overflow:

$$9 = 1001 +$$

$$8 = \underline{1000}$$

$$1\text{ } \textcircled{0001}$$

traboccamento

Risultato fornito  
(sbagliato)