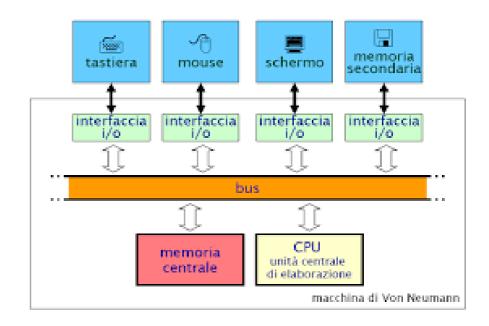
# **Architettura**

#### Francesco Isgrò

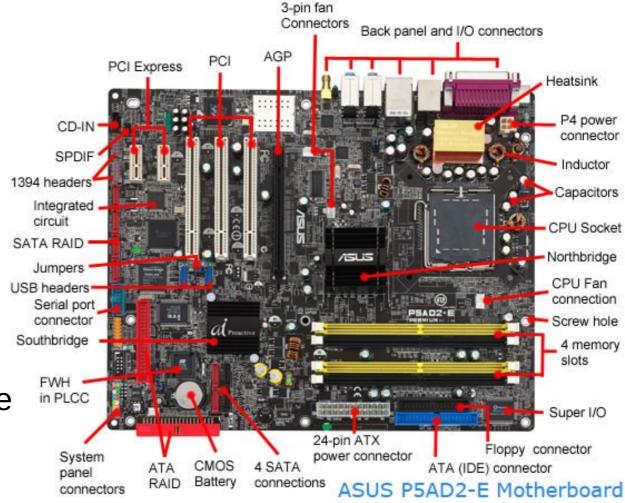
Contiene materiale dalle lezioni di Paolo Valente

#### Schema di un calcolatore

- Schema semplificato di un calcolatore
- Consideriamo solo
  - CPU
  - Memoria
  - Dispositivi I/O



#### Motherboard



La scheda madre connette fisicamente fra loro i vari blocchi.

ComputerHope.com

#### La CPU



- CPU: Central Processing Unit
- Tutte le operazioni di elaborazione delle informazioni effettuate da un calcolatore sono
  - Svolte direttamente dalla CPU
  - Svolte da altri componenti dietro comando del processore

- Un processore è in grado di compiere solo operazioni molto semplici:
  - lettura/scrittura/copia di una o più celle di memoria
  - somma/sottrazione/moltiplicazione/divisione del contenuto di una o più celle di memoria
  - lettura/scrittura in zone di memoria 'speciali' per pilotare dispositivi di ingresso/uscita (ad esempio schede video)
  - altre semplici operazioni sulle celle di memoria

- Tipicamente un processore riesce a lavorare su un certo numero di celle contigue alla volta.
- Tale sequenza di celle è detta parola di macchina (machine word)
- Si dice che un processore ha una architettura a 16, 32 oppure 64 bit se lavora su parole da 2, 4 oppure 8 byte

### Linguaggio macchina

- Ogni processore è caratterizzato da un proprio insieme di istruzioni, tramite le quali è possibile fargli svolgere le precedenti operazioni
- L'insieme delle istruzioni di un processore viene chiamato linguaggio macchina di quel processore
- Ogni istruzione è identificata da una certa configurazione di bit

## Linguaggio Assembly

- Il linguaggio macchina è una sequnza di bit
- Difficile da capire.
- L'assembly: linguaggio molto simile al linguaggio macchina
- Consente di ignorare il formato binario del linguaggio macchina
- Ogni codice operativo del linguaggio macchina è sostituito da una sequenza di caratteri che lo rappresenta in forma mnemonica
- Assembler traduce in linguaggio macchina

# Esempio in assembler

	000000 000000		Dato da elaborare (una WORD) Risultato (un BYTE)
***			
40B1A203	MOV MOV CMP JE SHL JC JMP ADD JMP	\$0, %AL (3F0D0100), %DX %DX, \$0 40B1A232 %DX 40B1A225 40B1A207 \$1, %AL 40B1A207	Azzera il contatore AL Copia in DX il dato da elaborare Confronta il contenuto di DX con 0 Salta se uguale Trasla a sinistra il contenuto di DX Salta se CF è settato all'indirizzo 225 Salta incondizionatamente a 207 Incrementa il contatore AL Salta incondizionatamente
40B1A232	MOV HLT	%AL, (3F0D0102)	Memorizza il risultato ALT

- L'assembly è un linguaggio molto semplice
- Si possono scrivere programmi complessi
  - Sistema operativo
  - Interfaccia grafica
- Lavoro molto complesso
- Problema principale: istruzioni di basso livello

#### **Astrazione**

- Necessarie istruzioni più vicine al modo di ragionare umano
- Si astrae dalla macchina sottostante
- Linguaggi di alto livello
  - Più vicini al linguaggio umano che al linguaggio macchina
  - Generale: non dipende dal processore

### Vari livelli di astrazione

- <u>Linguaggio macchina</u>: conoscenza dei metodi di rappresentazione delle informazioni
- Assembly: conoscenza delle caratteristiche della macchina (registri, set di istruzioni, dimensione dei dati)
- <u>Linguaggio di alto livello</u>: si astrae dai dettagli legati alla macchina e si esprimono gli algoritmi in modo simbolico

# Algoritmo e programma

- Algoritmo
  - Sequenza di passi elementari per la risoluzione di un problema
  - Sequenza finita
  - Non ambigua
  - Deterministica
  - Generale
- Programma: formalizzazione di un algoritmo in un linguaggio di programmazione

# Linguaggi di programmazione

- Rendono facile la scrittura di programmi
- Un programma non può essere direttamente eseguito
- La macchina capisce solamente il linguaggio macchina
- Deve essere tradotto in linguaggio macchina
- Compilatore

# Memoria principale

 Il contenitore in cui sono memorizzati i dati su cui lavora il prcessore



- La memoria principale si può schematizzare come una sequenza di celle contigue
- Le celle si chiamano locazioni di memoria
- Ciascuna cella costituisce l'unità minima di memorizzazione
- Tipicamente 1 byte (8 bit)
- Ciascuna cella è identificata da un indirizzo

- I bit contenuti in una cella possono essere usati per rappresentare un numero intero
- Si usa la notazione binaria
- Facciamo qualche esempio

#### **Notazione binaria**

- Sistema numerico posizionale in base 2
- Solo due simboli invece dei 10 usati nella notazione decimale
- Quanti numeri si possono rappresentare con un byte

(numero di simboli)  $(numero di posizioni) = 2^8 = 256$ 

Interi positivi

```
- 00000000 0
```

- 00000001 1

- 0000010 2

- 00000011 3

- 00000100 4

- 00000101 5

• 8 bit bastano per [0 .. 255]

- Interi negativi
  - Si utilizza una parte dell'informazione per codificare il segno
  - 1 bit
  - 7 bit per i numeri
  - Si rappresentano  $2^7 = 128$  numeri
  - L'intervallo [-127 .. 127]

# Tipi di dato

- La stessa quantità di memoria può rappresentare 2 cose diverse
  - Numero naturale
  - Numero intero
- Ma anche
  - Numero reale
  - Carattere
- Si astrae dalle singole celle di memoria
- Si ragiona sulla natura dei dati

- Questa astrazione è raggiunta tramite i tipi di dato
- Si scrive il programma direttamente in termini di cosa sono i dati
- Si lavora ad alto livello, senza preoccuparsi di come e dove saranno realmente memorizzati e manipolati tali numeri a basso livello

- In C si può scrivere int variabile = 5;
- Non ci preoccupiamo di come memorizzare i valori in una sequenza di celle
- Non ci preoccupiamo di dove memorizzare esattamente in memoria tale sequenza di celle
- Usiamo il tipo int nel suo significato astratto di contenitore di numeri interi

- Se il valore del dato int è maggiore di 255 non si può memorizzare in una singola cella di memoria
- L'astrazione dei tipi di dato gestisce per noi in maniera trasparente l'interpretazione del contenuto di più celle di memoria contigue come un singolo dato

- Ogni linguaggio di programmazione definisce diversi tipi di dato
- Ad esempio in C

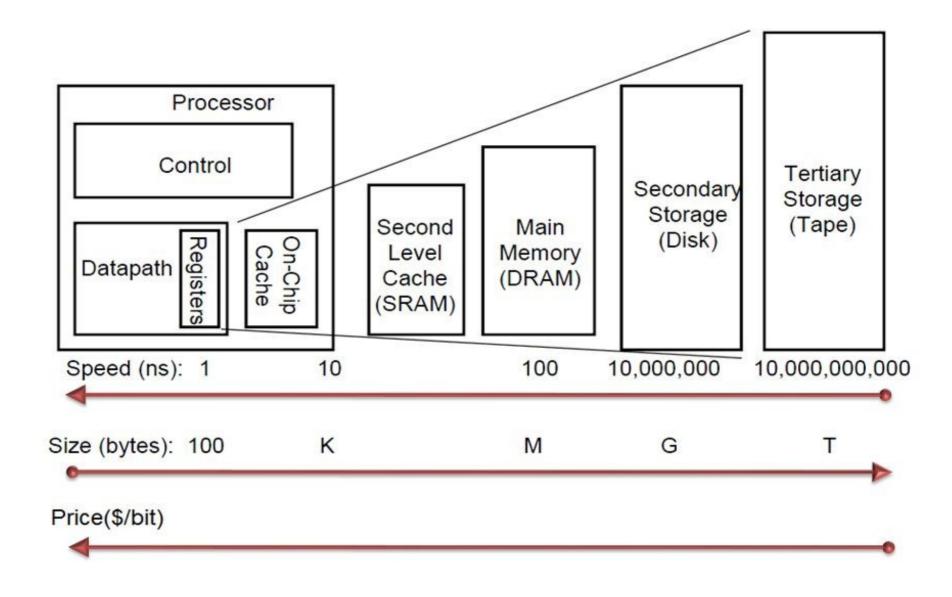
bit

- int numeri interi 32 bit
- float numeri reali 32 bit
- double numeri reali 64 bit

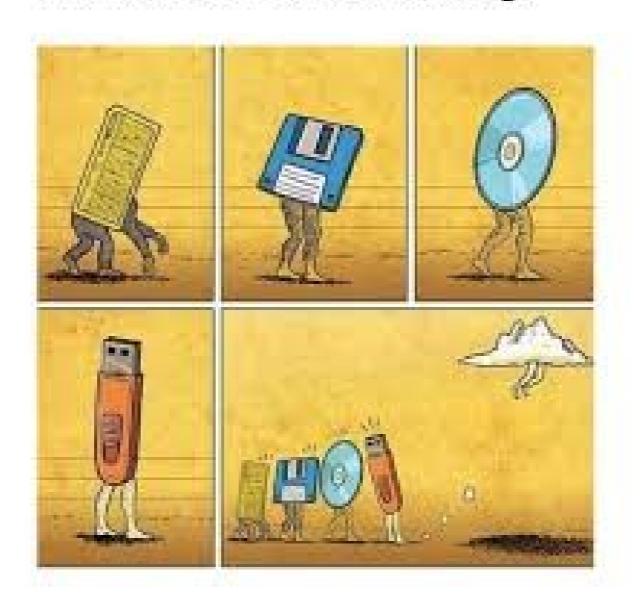
#### Memoria

- La memoria principale può contenere svariati programmi e dati ma
  - E' lenta
  - E' volatile
- Per ovviare alla lentezza si costruiscono delle gerarchie
- Le più veloci sono costose, quindi sono piccole
- Volatilità
  - Memorie di massa

# Livelli della gerarchia di memoria



#### The Evolution of Data Storage



#### In un calcolatore moderno



# Esecuzione di un programma

- Per fare eseguire un programma da una CPU
  - memorizzare da qualche parte nella memoria la sequenza di configurazioni di bit relativa alle istruzioni da eseguire
  - dire al processore a che indirizzo si trova la prima di tali istruzioni

# **Avvio del Sistema Operativo**

- Il primo programma eseguito all'accensione del computer
- Il sistema operativo dice al processore dove si trova la prima istruzione del programma di cui l'utente chiede l'esecuzione
- Ma chi dice al processore dove si trova la prima istruzione del sistema operativo?

- Il calcolatore carica in memoria, all'avvio, il contenuto di un blocco predefinito
- Master Boot Record (MBR)
  - Ospitato in uno dei dispositivi di memorizzazione di massa disponibili
  - Configurabile (possono esistere più sistemi operativi installati sul disco, e si può decidere una specifica sequenza di avvio)

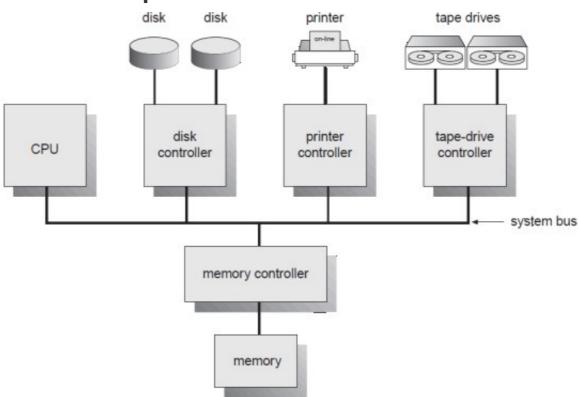
- Il codice presente nel MBR viene tipicamente chiamato bootloader
- Il suo compito è quello di caricare il sistema operativo installato nella macchina
- Consiste in una sequenza di byte, che si assume quindi essere un programma in linguaggio macchina

## Sistema operativo

- Alza il livello di astrazione della macchina
- Più facile da programmare dell'hardware sottostante

Semplificazione nell'implementazione dei

programmi



## Funzioni principali del SO

- Gestione del processore
- Gestione dei processi (programmi in esecuzione)
  - Creazione e terminazione
  - Schedulazione
  - Sospensione e riattivazione
  - Gestione di casi di deadlock
  - Comunicazione tra processi

- Gestione della memoria centrale
  - Gestione della memoria dedicata ai processi
  - Caricare e rimuovere i processi dalla memoria centrale
  - Protezione della memoria centrale

- Gestione delle periferiche
  - Configurazione e inizializzazione
  - Gestione dei dispositivi di I/O
  - Buffering
  - Caching

- Gestione del file system
  - Nasconde all'utente l'organizzazione della memoria attraverso il concetto di file
  - File e directory
    - Creazione e cancellazione
    - Lettura e scrittura
    - Copiatura
    - Ricerca
    - Salvataggio e ripristino
    - Protezione e sicurezza (diritti di accesso)

- Gestione interfaccia utente
  - Interfacce a riga di comando (interprete dei comandi)
  - Interfacce grafiche

- Rilevare errori
  - Esempi
    - Divisione per 0
    - Fine della carta
    - Guasto di una connessione di rete
  - Per ogni errore è necessario eseguire una appropriata azione di recovery

#### Servizi del SO

- Il SO fornisce servizi ai programmi e agli utenti dei programmi
- Elaborazione: il sistema deve potere caricare un programma in memoria e eseguirlo
- Operazioni di I/O: in generale gli utenti non possono controllare direttamente i dispositivi di I/O; è il SO che deve fornire i mezzi per compiere le operazioni di I/O
- Manipolazione del file system: i programmi devono poter leggere, scrivere, creare e cancellare i file

- Comunicazione fra processi sullo stesso computer o su computer differenti
  - Le comunicazioni possono avvenire
    - memoria condivisa
    - attraverso scambio di messaggi
- Rilevamento degli errori
  - nella CPU
  - nell'hardware della memoria,
  - nei dispositivi di I/O e nei programmi dell'utente

Esistono servizi aggiuntivi per una gestione/ monitoraggio efficiente del sistema

- Allocazione delle risorse assegnazione delle risorse a più utenti o più processi in funzione contemporaneamente
- Accounting mantiene traccia di quali utenti usano le risorse e di quale tipo e genere di risorse si tratta per stilare statistiche d'uso o
- Protezione e sicurezza implica la garanzia che tutti gli accessi alle risorse del sistema siano controllati

#### Installazione di un SO

- Tipicamente lo stesso SO può essere eseguito su più macchine
- Il sistema deve essere quindi configurato
  - CPU
  - Memoria
  - Formattazione del disco di avvio (partizioni, dimensioni, contenuto)
  - Dispositivi disponibili
  - Opzioni (e.g. algoritmo di schedulazione, max. num. processi )

## Implementazione di un SO

- Tradizionalmente scritti in linguaggio assembly, oggi in linguaggi di alto livello
  - un codice scritto in un linguaggio ad alto livello:
    - può essere scritto più velocemente
    - è più compatto
    - è più facile da capire e da mettere a punto
- Un SO è molto più flessibile e manutenibile se è scritto in unlinguaggio ad alto livello