

Programmazione

Docente: **Eugenia Occhiuto**

e-mail: occhiuto@di.unipi.it

web: <http://www.di.unipi.it/~occhiuto>

Ricevimento al Polo : **Negli intervalli e dopo le lezioni**

Ricevimento a Pisa: **Ven.16-18, Mar.11-13**

Modalità d'esame

Scritto e orale obbligatori.

Le verifiche se superate con successo danno l'ammissione all'orale:

1. la **prima** nel periodo 2-4/11,
2. la **seconda** nel periodo 16-18/12

se si fallisce o si è assenti alla prima prova è comunque consigliato fare la seconda

Orario

La suddivisione lezione/esercitazione è orientativa

Martedì 11 – 13	Lezione	Aula A4
Martedì 14 – 16	Esercitazione	Aula A2
Mercoledì 11 – 13	Laboratorio	Lab. 1
Mercoledì 14 – 16	Lezione	Aula A3
Giovedì 14 – 16	Lezione	Aula A3

Consigli

- 1) Consultare spesso il **sito** del corsi su cui vengono mantenute le informazioni:
<http://www.di.unipi.it/~occhiuto/Programmazione.html>
- 2) seguire le lezioni e le esercitazioni anche quelle in laboratorio
- 3) fare gli esercizi assegnati e discutere la soluzione nella lezione successiva
- 4) studiare (2 ore per ogni ora di lezione) e possibilmente anticipare gli argomenti che verranno trattati nella lezione successiva.

Informazioni di riferimento

- **Dispense** scaricabili dal sito:
 - Elementi di sintassi dei linguaggi di programmazione.
 - Programmazione funzionale.
 - Elementi di semantica operativa
- **Testi** di programmazione
 - funzionale con Caml :
 - Cousin .
 - imperativa in C

Programma del corso

- 1. Introduzione e concetti generali**
- 2. Elementi di sintassi dei linguaggi di programmazione: grammatiche**
- 3. Programmazione funzionale in Caml:** espressioni, tipi primitivi e complessi, funzioni, liste, polimorfismo, ricorsione, induzione
- 4. Programmazione imperativa:** nucleo del C, semantica operativa.
- 5. Interprete del nucleo del C in Caml**

Informazione

Per fornire informazione è necessario:

1. **comprendere la richiesta.** *Es. Tra quanto passa l'espresso per Torino?*
2. **avere i dati necessari a disposizione.** *L'espresso per Torino passa alle 12. Sono le 11.30*
3. **essere in grado di elaborarli.** *Saper calcolare $12.00 - 11,30 = 0.30$*
4. **comunicare la risposta.** *Tra 30 minuti*

Informazione automatica

L'informazione automatica viene fornita da una macchina che:

1. **per comprendere la richiesta** deve avere un modo per specificare la richiesta ad es. una sequenza di tasti da pigiare.
2. **avere i dati a disposizione** scritti in un dispositivo di memorizzazione.
3. **essere in grado di elaborarli**. Saper calcolare la funzione che in base alla richiesta e ai dati memorizzati calcola la risposta.
4. **comunicare la risposta**. Avere un dispositivo su cui comunicare il risultato del calcolo.

Un esempio: la calcolatrice

- La calcolatrice
 - ha 10 tasti per specificare i dati (cifre).
 - 4 tasti per specificare le operazioni (funzioni) da calcolare
 - Una memoria su cui mantenere i dati parziali
 - Un foglio o un display su cui stampare il risultato
- Analogamente posso definire altre macchine che calcolano altre funzioni su altri dati ma ..
- ogni macchina è limitata ad un numero finito di funzioni (la calcolatrice: le 4 operazioni aritmetiche) mentre per i dati ?...

Informazione automatica

Sorgono spontanee alcune domande:

1. Quali sono le **funzioni** che si possono **calcolare**?
2. Quante (diverse) sono le **funzioni calcolabili**?
3. Ma quali sono le funzioni che una **macchina** può calcolare?
4. Posso costruire una macchina per ogni funzione che posso calcolare?
5. È possibile definire una macchina che calcoli **tutte** le **funzioni calcolabili**?

Funzioni calcolabili

Molto prima che venisse inventato il primo calcolatore, i matematici avevano definito il dominio delle funzioni calcolabili Ω .

Le risposte:

1. Le funzioni calcolabili sono quelle per le quali esiste un modo effettivo per calcolarle (**algoritmo**).
2. (Le funzioni calcolabili sono) **Infinite** (numerabili).
3. Sì! posso costruire una macchina per ogni funzione calcolabile.
4. Ma la vera cosa interessante è che posso costruire una macchina che calcoli tutte le (infinite) **funzioni calcolabili**: questa macchina è il moderno **calcolatore**.

Algoritmo

- Sequenza anche infinita di azioni, che descrive un modo per calcolare la funzione (calcolabile).
- Le azioni sono ordinate
- Le azioni e la loro composizione sono espresse in un linguaggio (non necessariamente di programmazione)

Importanti proprietà Ω

- Ω contiene tutte le funzioni i cui valori possono essere calcolati in modo effettivo: non solo concepiti (Es. razionali vs. reali).
- Ogni funzione calcolabile ha una (?) **descrizione** finita D. Es. la funzione fattoriale ha la seguente descrizione:

D(fattoriale) sia

$$\text{fatt}(0) = 1$$

$$\text{fatt}(n+1) = (n+1) * \text{fatt}(n)$$

Funzione Universale U_{Ω}

- Esiste una funzione universale U_{Ω} che calcola tutte le funzioni calcolabili così definita:

$$\begin{aligned} &\forall \text{ funzione } f \in \Omega, \\ &\forall \text{ descrizione } D(f) \\ &\forall \text{ valore } n, \\ &U_{\Omega}(D(f), n) = f(n) \end{aligned}$$

- U_{Ω} appartiene ad Ω cioè U_{Ω} è calcolabile
- Il calcolatore è una realizzazione di U_{Ω} .cioè calcola la funzione universale U_{Ω} prendendo come dati la descrizione della funzione ($D(f)$) e i dati (n) su cui calcolare f .

Definizione di funzione

Dati 2 insiemi A e B una funzione è un sottoinsieme del prodotto cartesiano $A \times B$ (A è il **dominio** e B è **codominio** della funzione), tale che $\forall x \in A$ esiste uno e uno solo elemento $y \in B$ tale che $(x, y) \in f$.

Si scrive $f: A \rightarrow B$. Con $f(a)$ si indica l'applicazione di f ad a che calcola l'elemento $b \in B$

Come si descrivono le funzioni? (in Matematica)

Es. $\{(0,1), (1,1), (2,2), (3,6), (4,24), (5,120), \dots\}$

che funzione è?

Quale è il dominio?

e il codominio?

Descrizione insiemistica o estensionale

Descrizione intenzionale

$$f(x) = \begin{cases} 1 & \text{se } x=0 \\ x \cdot f(x-1) & \text{se } x>0 \end{cases}$$

Come si descrivono le funzioni?

- Linguaggi per descrivere funzioni:
 - Detti linguaggi di **programmazione**
 - Sono **formalismi**, con **sintassi** e **semantica** formalmente definite.
 - Esempi: Caml, Haskel, Prolog, C, JAVA, Pascal, linguaggi macchina
- Le descrizioni delle funzioni da calcolare sono frasi in un linguaggio di programmazione e sono detti **programmi**

La programmazione

- Una delle attività principali di un informatico (forse la principale) è quella di scrivere **programmi** che il calcolatore esegue.
- Lo scopo del corso di Programmazione è duplice:
 - insegnare a programmare,
 - insegnare i concetti di base dei **linguaggi di programmazione**, che ne permettono l'implementazione.
- Durante il corso si utilizzeranno due linguaggi di programmazione: **CamL**, linguaggio funzionale e **C**, linguaggio imperativo.

Linguaggi di programmazione

- I linguaggi di programmazione sono tutti **equivalenti**,:
 - ho $D_{L_1}(f)$ (programma nel linguaggio L1) e $D_{L_2}(f)$ (programma nel linguaggio L2 che calcola la stessa funzione).

$$D_{L_1}(f) \xrightarrow{\text{traduzione}} D_{L_2}(f)$$

- La **traduzione** è una funzione calcolabile e quindi **descrivibile** con un **programma** (cioè può essere fatta dal calcolatore).

Il linguaggio macchina

- Il linguaggio direttamente eseguibile dal calcolatore (**linguaggio macchina**) è un linguaggio di programmazione poco comprensibile agli umani (programmatori) che devono scrivere il programma perché:
 - le operazioni sono molto semplici,
 - sono specificate in binario (sequenze di 0 e 1) quindi
 - specificare un'operazione complessa richiede la scrittura di un lungo programma, incomprensibile.

Traduzione del programma

- Tutti i linguaggi (così detti) ad **alto livello** devono essere tradotti in **linguaggio macchina** per poter essere eseguiti.
- Fortunatamente la traduzione (come abbiamo già detto) può essere fatta dalla macchina stessa.
- I programmi che effettuano la traduzione si chiamano **compilatori**.

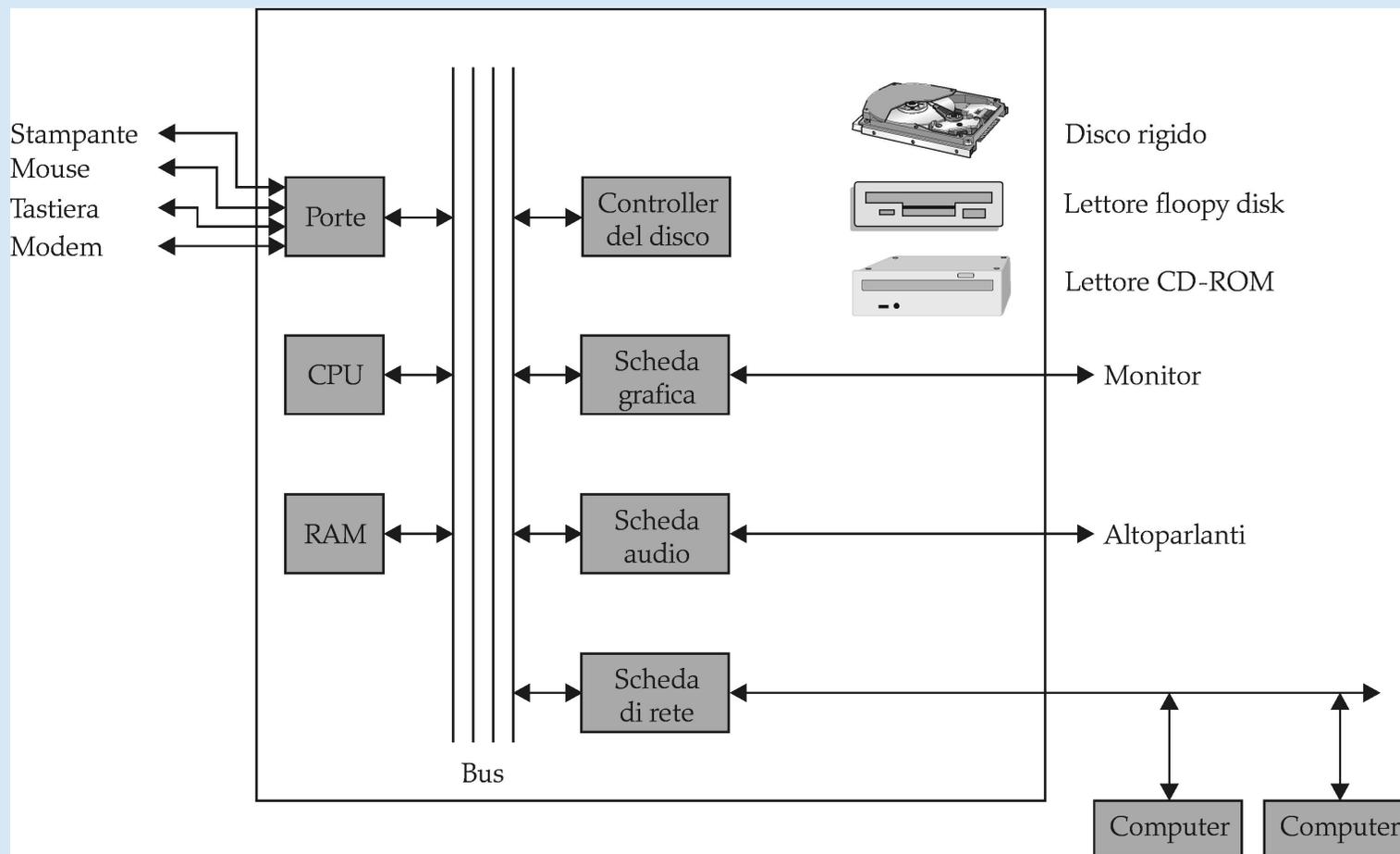
Implementazione dei linguaggi

- Traduzione del programma in linguaggio macchina: **compilazione**.
- Esecuzione del programma da parte di un altro programma: **interpretazione**
- Vedremo un esempio di **interprete**.

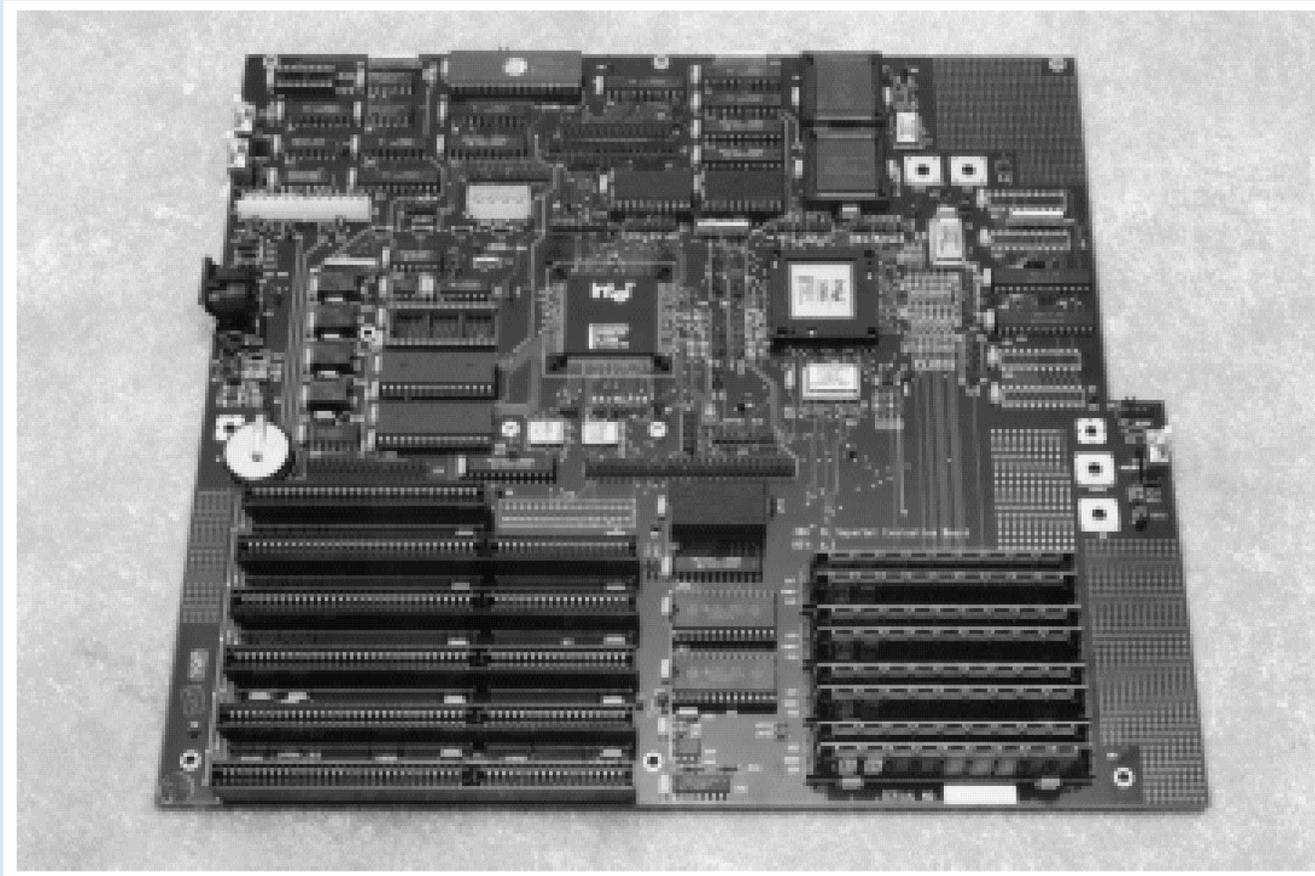
Il calcolatore vista esterna



Schema di struttura o struttura logica



Struttura fisica



Struttura del calcolatore

- CPU
- Memoria:
 - RAM, memoria volatile fissa (molto veloce)
 - Hard disk, memoria permanente fissa (non molto veloce)
 - CD, Floppy disk, ecc memoria permanente rimovibile
- Dispositivi di Input/Output

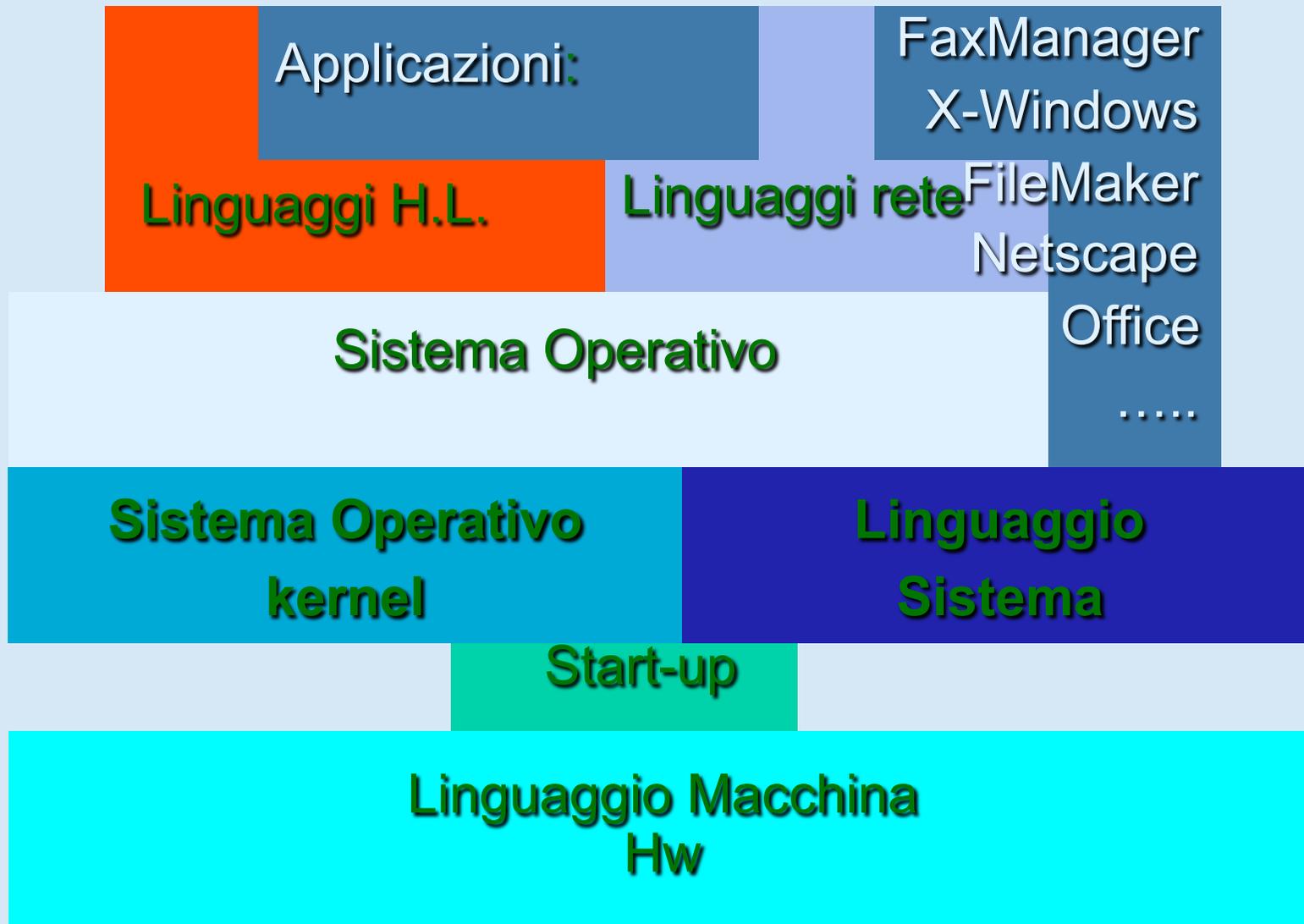
Sistema operativo

- La gestione della memoria e dei dispositivi di I/O tutto deve essere specificato nella descrizione del programma da eseguire. (schede perforate)
- Ovviamente il programma sarebbe sempre complicatissimo ed ogni volta dovrei riscrivere parti molto simili (ad es. che mandano in stampa dei dati ecc)
- Esistono delle funzioni la cui descrizione (programma) è definita una volta per tutte nella macchina e che vengono invocate quando servono. (Sistema operativo)

Interazione

- **Sistemi Operativi**
 - Resettings
 - File system
 - gestione I/O
- **Applicazioni:**
 - Explorer , Netscape
 - Word (office)
 - CinemaDvD
 - Photoshop
- **Linguaggi**
 - ocaml, cc, javac, java
 - Nuove applicazioni

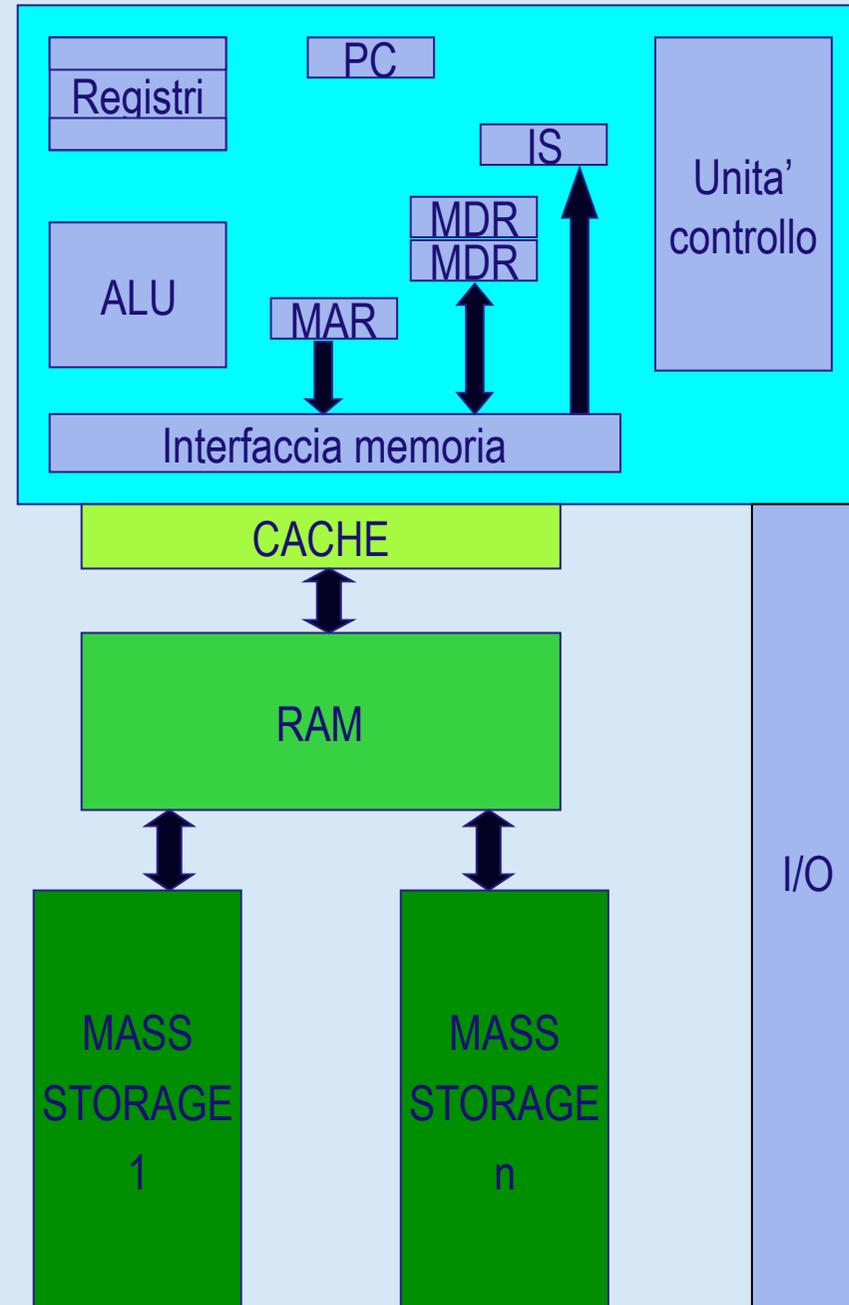
Relazione tra Hw e Sw



Cosa fa un calcolatore?

- In ogni momento il calcolatore esegue un programma, scritto in linguaggio macchina, in questo modo esegue:
 - operazioni del S.O.,
 - applicazioni disponibili,
 - compila programmi,
 - esegue programmi compilati dall'utente

Algoritmo Fondamentale



Programmazione

Dal problema alla sua soluzione

- Concettualizzazione:
 - Cosa si vuole fare
- Rappresentazione dati coinvolti
 - Come si astraggono i dati del problema
- Definizione procedimento di soluzione
 - Come si opera: sequenza dei passi
(**algoritmo**)
- Formulazione in un linguaggio di programmazione
 - Descrizione della funzione calcolabile
(**programma**)