**1° lezione di Programmazione 1**

**Il calcolatore**

Permette di eseguire svariate funzioni come ad esempio memorizzare dati elaborare dati, trasferire i dati dall'esterno o verso l'esterno.

Un calcolatore permette di eseguire del lavoro in modo automatico(computazione) che viene specificato in forma di algoritmo che va specificato in un linguaggio comprensibile al calcolatore.

**Modelli teorici**

Tra i modelli teorici vi sono una macchina di Touring e la macchina di Von Neumann.

**Macchina di Touring**

**Alan Turing** ha sviluppato una macchina teorica (automa) che era in grado di risolvere qualsiasi algoritmo. È stato il primo modello di calcolo. E’ una formalizzazione del concetto di algoritmo, che è, infatti, una particolare macchina di Turing.

La macchina di Turing è formata da:

* **Un nastro di lunghezza infinita** che è fatto da un numero di caselle infinito.
* **Le caselle** indicano simboli appartenenti all’insieme finito (binario 0 e 1 ad esempio).
* **La testina** è in grado di muoversi a destra e sinistra, leggere e cambiare il contenuto di ogni singola cella.

Con questa modello semplice si può modellare qualsiasi algoritmo basta specificare il comportamento di questo automa (essere umano) che esegue un programma in base ad una specifica che aderisca a questo modello (algoritmo come muoviti ad dx se leggi 0). Si può specificare il comportamento di questo automa in funzione dell’input e quindi anche dal set di simboli utilizzato. (**Input** sono le informazioni che prendo dall’esterno attraverso i dispositivi di input e l’**Output** è quello che restituisce l’elaboratore). L’automa prende delle decisioni in base al suo stato interno cioè quello che ha letto prima (ad esempio quando si entra in auto questa è spenta, ci sarà lo stato interno che porterà ad accendere l’auto). Lo stato interno dell’ automa dipende dall’osservazione dell’ambiente circostante (input). **Il comportamento della macchina di Turing può essere programmato come un insieme di regole o quintuple che è una tupla (insiemi di elementi di tipo eterogeno) composta da cinque elementi**.

**In simboli: (S, α, S∗, β, {<, >})**

* **S**: indica lo stato interno corrente
* **S\***: indica il prossimo stato interno
* **α** : è il simbolo letto nella cella corrente (input)
* **β**: è il simbolo da scrivere nella cella (output)
* **{}**: si va a specificare un elemento da un insieme
* **<** : sposta la testina a sx
* **>**: sposta la testina a dx

**Per descrivere un algoritmo con la MDT si devono avere un insieme di quintuple come specificato sopra che descrivono le azioni di questo automa in funzione…. (44:44) info1**

L’insieme degli stati interni deve includere **lo stato inziale** e quello **finale** ed il risultato della computazione è il contenuto del nastro quando la MDT ha raggiuto lo stato finale.

**Un algoritmo che non termina non è un algoritmo infatti questo viene considerato tale quando termina in un tempo finito.** La MTD è un dispositivo ideale cioè che non è stata realizzata. Il concetto di **MDT universale** è una MTD che è in grado di calcolare tutte le possibili funzioni calcolabili da ogni singola MDT.

**Modello di Von Neumann**

con questo modello si vanno a memorizzare le istruzioni (insiemi di dati) in memoria pertanto si deduce che si ha un calcolatore interamente programmabile cioè si da in pasto a un calcolatore la descrizione di un algoritmo o di una macchina di Turing ci darà in output un risultato. È formato da:

* **RAM(Random Access Memory)**
* **CPU(Central Processing Unit)**

Una macchina di Von Neumann è definita da una terna:

1. **N:** insieme dei numeri naturali che rappresentano l’alfabeto della macchina
2. **IS(Instruction Set):** set delle istruzioni che dipendo dalla CPU
3. **P**: sequenza di istruzioni finite che operano su dati specifici

**Immagine che contiene testo, schermata, Rettangolo, diagramma

Descrizione generata automaticamente**

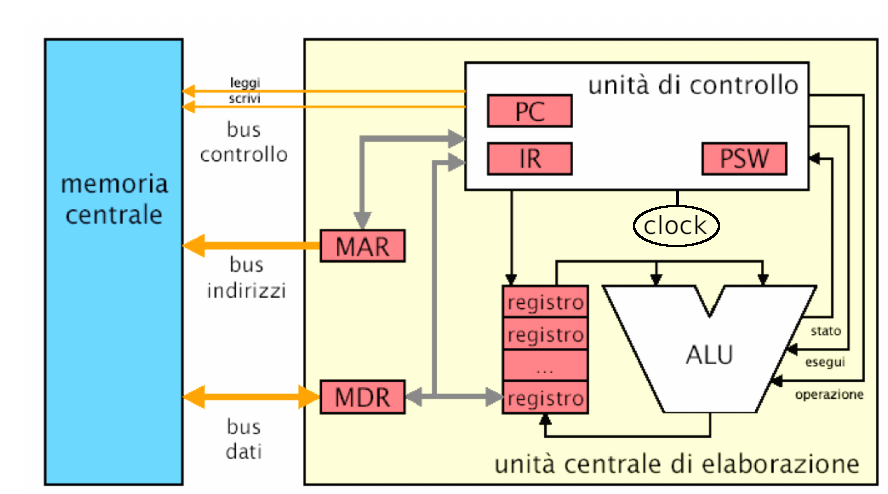
**La CPU** è formata:

* **CU(Control Unit):** unita di controllo
* **ALU(Aritmetic Logic Unit):** unità aritmetico logica (fa i calcoli)
* **Memory Unit(RAM):** è una memoria volatile ciò vuol dire che quando si arresta il PC il contenuto di questa viene perso.
* **Input devices(dispositivi di input):** come tastiera mouse
* **Output devices(dispositivi di Output):** come lo schermo

Una parte delle istruzioni del Software(programma) che si sta utilizzando va registrato in memoria RAM dopo di che le istruzioni vengono eseguite secondo l’ordine specificato dal programma e la sua esecuzione viene arrestata quando si riceve un istruzione di arresto[…].

La RAM è composta da celle di memoria che sono identificate da un indirizzo che per comodità è scritto in esadecimale (base 16) ma in realtà il PC capisce solo il binario o linguaggio macchina (base 2) […].

Le istruzioni che compongono il programma sono inserite nella prima parte di memoria, invece, i dati di questo sono inserite nelle locazioni successive.



Tra i componenti più importanti troviamo:

* **PC(Program Counter)**: una cella in memoria che contiene l’indirizzo della prossima istruzione da eseguire;
* **IR(Instruction Registrer):** locazione di memoria contenente l’istruzione da esguire;
* **ALU(Aritmetic Logic Unit):** unità aritmetico logica (fa i calcoli)
* **CU(unita di controllo):** sistema che si occupa dell’esecuzione dell’istruzione, controllando le altre componenti del processore;

**Algoritmi**

è una sequenza di passi concepita per essere eseguita automaticamente da una macchina in modo da risolvere il problema.

Per la risoluzione di un problema dobbiamo:

* **Prendere dei dati in input quindi provenienti dall’esterno**
* **Concepire e codificare l’algoritmo in modo che sia interpretabile da un calcolatore**
* **Eseguire il programma**
* **Vedere cosa ci da in output (il risultato dell’elaborazione)**

**L’uomo concepisce l’algoritmo l’elaboratore è solo un esecutore**

**Algoritmo**

**Sequenza ordinata e finita di passi (azioni o istruzioni) che producono un ben determinato risultato in un tempo finito.**

Un algoritmo per essere considerato tale deve avere diverse caratteristiche come:

* **Azioni eseguibili e non ambigue**: si deve rispettare l’ IS (Istruction Set)
* **Determinismo**: ogni passo deve essere ben determinato con alternative ammesse ma la scelta deve essere univoca
* **Numero finito di passi**
* **Terminazione cioè il programma deve finire**
* **Deve produrre lo stesso risultato con le stesse condizioni** come lo stesso input e valori iniziali delle variabili.

Il calcolatore deve poter comprendere i passi dell’algoritmo ciò vuol dire attribuire la corretta **semantica** alle istruzioni.

Il programma è un algoritmo quindi un insieme di passi per la risoluzione di un problema codificato in un linguaggio ad alto livello.

* **I linguaggi a basso livello**: si avvinano di più al linguaggio macchina (binario)
* **I linguaggi ad alt livello:** si avvicinano di più al linguaggio parlato (nostro)

**Come ci approcciamo alla risoluzione di un problema?**

Un problema grande va suddiviso in sotto problemi o sottoprogrammi. Prima si ha una visione generale del problema(nella sua interezza) e pian pino si va a decomporre finche non si ottengono problemi sempre più piccoli.

Ci sono due approcci:

1. **L’approccio bottom-up o dal basso verso l’alto implica** la progettazione di un algoritmo partendo dal livello di base e costruendolo verso il livello complesso. Nell’approccio i moduli sono progettati in modo indipendente e vengono quindi integrati insieme per formare un progetto algoritmico completo.
2. **Nella programmazione, un approccio dall’alto verso il basso in inglese top-down,** denominato anche progettazione graduale, un algoritmo complesso viene suddiviso in unità più piccole, denominate moduli., l’approccio top-down riduce le complicazioni che di solito sorgono durante la progettazione degli algoritmi.
3. Si utilizza anche un approccio ibrido