Linguaggi di programmazione

- Un calcolatore basato sul modello di von Neumann permette l'esecuzione di un *programma*, cioè di una sequenza di istruzioni descritte nel linguaggio interpretabile dal calcolatore che realizzano un particolare algoritmo, ma …

Linguaggio macchina

… ma quali sono le caratteristiche di tale linguaggio ?

- è codificato tramite sequenze di bit
- accede ai dati tramite gli indirizzi di memoria o i registri interni della CPU
- ogni istruzione può compiere solo azioni molto semplici
- non gestisce direttamente i tipi di dati di interesse
- è strettamente legato alla particolare macchina su cui è definito

Non a caso viene definito *linguaggio macchina*

Scrivere un programma

- Se si volesse implementare un dato algoritmo attraverso la scrittura di un programma sarebbe quindi necessario:
  - conoscere dettagliatamente tutti i codici operativi e la loro codifica
  - decidere in quali registri (di memoria o interni alla CPU) vadano memorizzati i dati
  - determinare, per ogni singola operazione richiesta dall'algoritmo, la sequenza di istruzioni in linguaggio macchina che la realizzano
  - definire un'opportuna tecnica di codifica per ogni tipo di dati considerato
  - limitarsi a utilizzare solo i calcolatori per cui esista una tale competenza, tenendo comunque presente che il programma scritto per un certo calcolatore non è eseguibile su altre macchine

Impresa difficile, ma non impossibile

Il gap semantico

Esecutore umano
- linguaggio naturale
- gestione completa dei tipi
- istruzioni semanticamente ricche

Calcolatore
- linguaggio rigido e complicato
- gestione dei tipi quasi nulla
- istruzioni estremamente semplici

Linguaggio di programmazione

Esecutore umano
- linguaggio naturale
- gestione completa dei tipi
- istruzioni semanticamente ricche

Orientato al problema

Linguaggio di programmazione
- linguaggio formale, con costrutti precisi per la definizione dei dati e delle operazioni
- gestione completa dei tipi fondamentali; possibilità di definire tipi strutturati
- istruzioni che realizzano le principali azioni elaborative richieste

Orientato alla macchina

Calcolatore
- linguaggio rigido e complicato
- gestione dei tipi quasi nulla
- istruzioni estremamente semplici

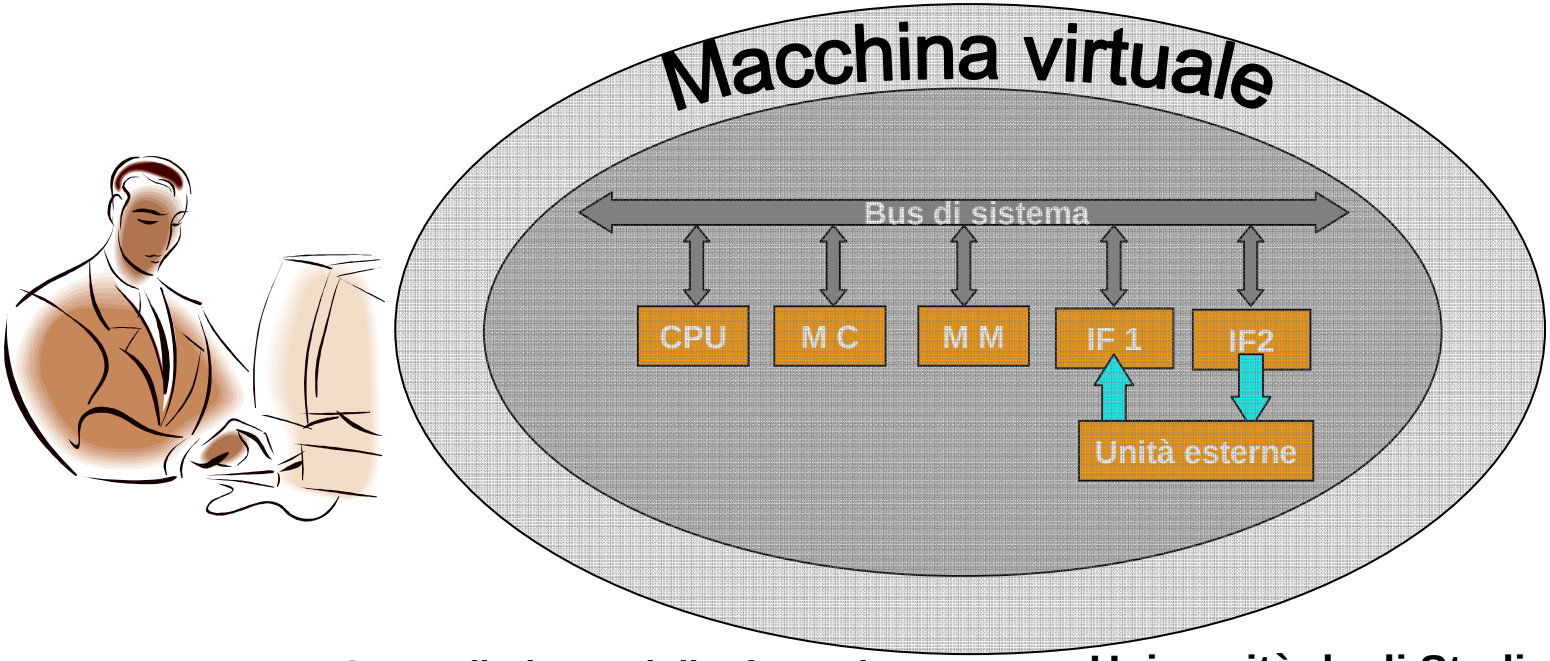
Vantaggi

L'uso di un linguaggio di programmazione permette di :

- realizzare un programma che implementa l'algoritmo in maniera precisa ed in un linguaggio "ad alto livello"
- trascurare tutti i dettagli relativi alla rappresentazione dei dati nei registri
- definire un programma che non dipende dal particolare calcolatore su cui è stato realizzato

Linguaggio=macchina virtuale

In effetti, l'utente non deve interagire con la macchina reale e le sue limitazioni, ma "vede" una macchina virtuale che nasconde le particolarità della macchina reale e con la quale è molto più agevole interagire.



Dal linguaggio ad alto livello al linguaggio macchina

- I linguaggi di programmazione ad alto livello sono linguaggi *formali*, che rispettano delle regole rigide, riguardanti sia il lessico che la sintassi del linguaggio.
- In tal modo viene eliminata l'ambiguità e le ridondanze tipiche del linguaggio naturale ed <u>è possibile realizzare in modo automatico l'analisi di un programma scritto in un linguaggio ad alto livello e la sua traduzione in linguaggio macchina</u>.

Programmi traduttori

- I programmi che svolgono il compito di tradurre un programma in linguaggio ad alto livello in un programma in linguaggio macchina sono detti *traduttori* e si dividono in due categorie:
  - **compilatori**
  - **interpreti**

Compilatori

- I compilatori eseguono la traduzione sull'intero programma in linguaggio ad alto livello.
- La traduzione viene realizzata tramite alcune fasi di analisi durante le quali si possono riscontrare degli errori, cioè delle non rispondenze alle regole formali del linguaggio.
- In caso di errori la traduzione non viene completata ed il programmatore viene informato sulla natura degli errori riscontrati e sulla loro posizione nel programma.
- Se non riscontra errori, il compilatore produce il programma in linguaggio macchina, direttamente eseguibile dal calcolatore

Interpreti

- Anche gli interpreti eseguono una fase di analisi e produzione di linguaggio macchina.
- La differenza rispetto ai compilatori è che l'interprete prende in esame un'istruzione alla volta, realizzandone la traduzione e l'esecuzione.
- Quindi, per ogni istruzione del programma in linguaggio ad alto livello, l'interprete
  - analizza l'istruzione e verifica la presenza di errori
  - in caso di errori, la traduzione viene arrestata; altrimenti, si producono e si eseguono le istruzioni in linguaggio macchina corrispondenti.

Differenze tra compilatori e interpreti

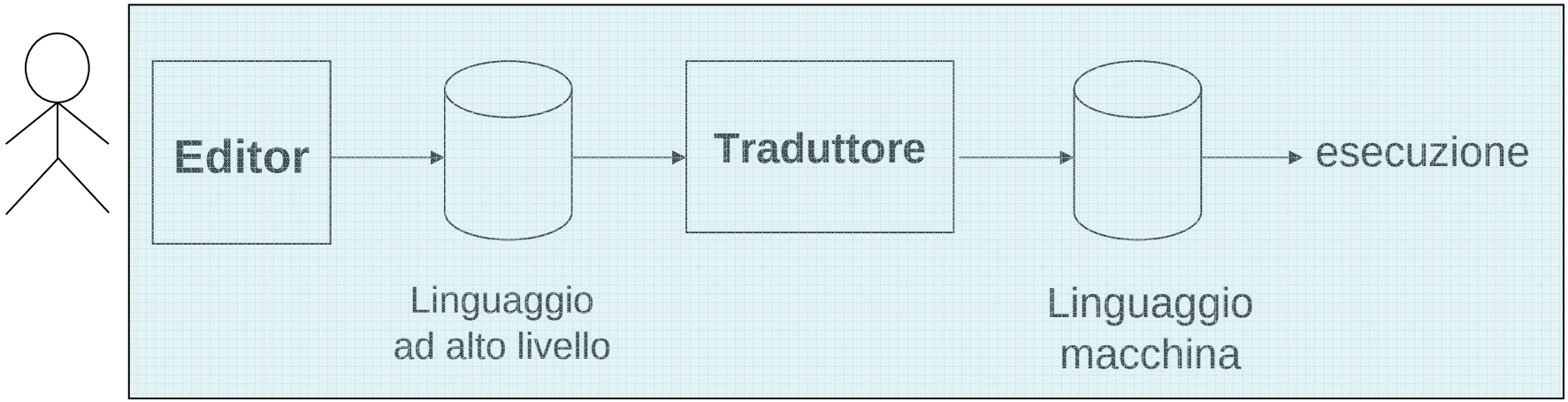
- Dal punto di vista dell' efficienza, l'esecuzione di un programma compilato è molto più veloce di quella di un programma interpretato.
- L'uso di un interprete potrebbe essere di aiuto nelle prime fasi di sviluppo di un programma, in quanto permette un immediata verifica della funzionalità del codice realizzato.

Fasi di produzione di un programma

- Le fasi per la produzione di un programma che possa essere eseguito da un calcolatore sono quindi:
  - definizione dell'algoritmo e dei dati
  - implementazione dell'algoritmo tramite un programma in un linguaggio ad alto livello
  - traduzione in linguaggio macchina
  - esecuzione del programma in linguaggio macchina

L'ambiente di sviluppo

- Nelle varie fasi c'è l'ausilio di opportuni programmi per:
  - la scrittura del programma in linguaggio ad alto livello (text editors)
  - la traduzione (compilatori o interpreti)
  - la verifica in esecuzione (debuggers)



Editor

- E' un programma che permette la scrittura e la memorizzazione su file delle istruzioni in linguaggio ad alto livello.
- Il file prodotto è un file di *tipo testo*: contiene, cioè, i caratteri scritti codificati in codice ASCII. In questo modo, è modificabile da altri programmi di *text editing* (es. NotePad) e trasferibile tra macchine diverse.
- Alcuni editors sono *orientati al linguaggio*, nel senso che organizzano il testo del programma in modo che sia resa più efficace la visualizzazione del codice. Tipiche caratteristiche di tali editors sono:
  - Indentazione
  - Syntax highlighting

Debugger

- E' un programma che permette l'esecuzione *passo-passo* del programma eseguibile, visualizzando lo stato corrente dei dati del programma.
- In questo modo è possibile verificare la rispondenza del codice prodotto all'algoritmo da implementare.