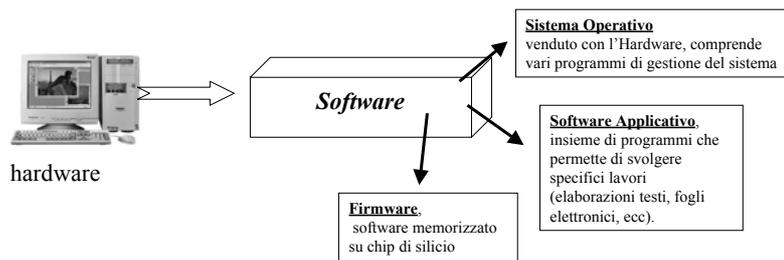
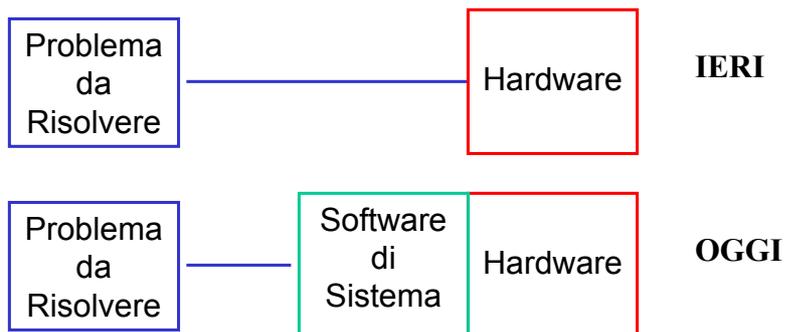


Il calcolatore oggi : UN SISTEMA DI ELABORAZIONE



Il Computer: generalità



Macchina in grado di risolvere PROBLEMI eseguendo opportunamente le istruzioni che gli sono fornite.
Un insieme di tali istruzioni e' detto **programma**.

Programma e Linguaggio

- Programma:
 - Descrizione della *procedura di elaborazione*
 - Sequenza di **istruzioni** espresse in un *linguaggio* comprensibile dalla macchina
- L'insieme delle istruzioni primitive di un elaboratore definiscono un *linguaggio* con cui è possibile comunicare con esso

Distanza notevole
fra il linguaggio di descrizione dell'uomo
e il linguaggio comprensibile dalla macchina

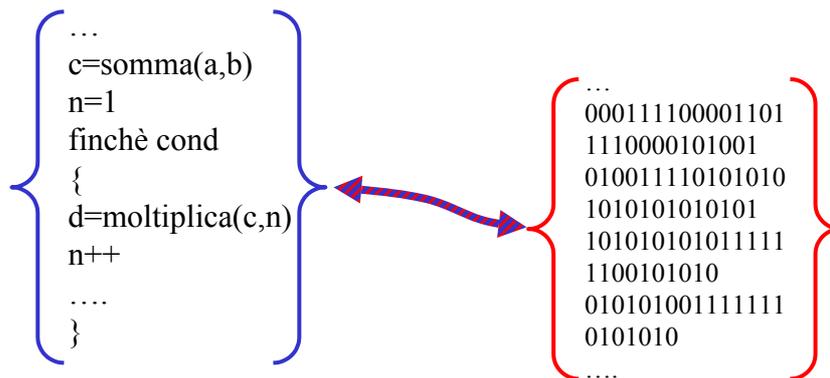
- Necessità di introdurre un insieme di istruzioni (nuovo linguaggio) più semplici da utilizzare rispetto alle istruzioni primitive
- Introduzione di una **macchina virtuale** sul livello di macchina reale (HW)

L'uomo risolve i problemi ad un
livello alto di ASTRAZIONE
utilizza istruzioni più vicine al problema

Solo Hardware :-(
la distanza deve essere percorsa interamente
dall'uomo.

i circuiti logici di una macchina sono in grado di
riconoscere ed eseguire un numero limitato di
istruzioni (operazioni elementari)
LINGUAGGIO MACCHINA

Linguaggio ad alto livello vs. linguaggio macchina



?PROBLEMA: ridurre la distanza tra questi due livelli

!Soluzione: organizzare il calcolatore come una serie di macchine astratte (structured computer organization)

Livelli di Astrazione

Complessità di un sistema di elaborazione:

Scomposizione in livelli di astrazione

I sistemi di elaborazione sono **sistemi gerarchici**
Ogni livello costituisce un sottosistema a se

Esiste una **relazione** fra i singoli livelli con il livello sovrastante e quello sottostante

Ogni livello fornisce **servizi al livello superiore**

Ogni livello **utilizza il livello inferiore**

Per ogni livello è quello immediatamente sottostante che ne rende possibile il funzionamento

Il livello Hardware → il funzionamento di tutti i livelli sovrastanti

Macchina virtuale

M0 Hardware L0
Macchina reale

M1 Software L1 **Macchina virtuale**

M0 Hardware L0
Macchina reale

?

- Il set di istruzioni comprensibili dalla macchina virtuale costituisce un nuovo linguaggio (L1) orientato alla rappresentazione più efficace del problema

- Realizzare una **macchina reale** capace di comprendere L1 sarebbe troppo costoso e/o poco efficiente.

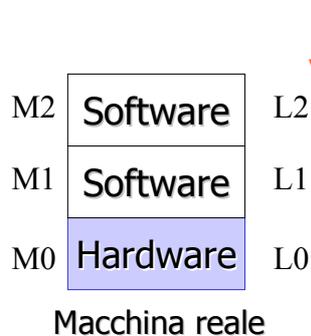
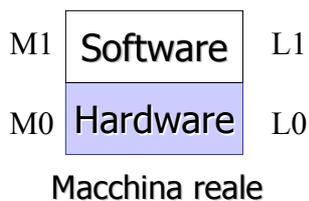
Relazione tra MV e linguaggio

- Ad ogni macchina virtuale corrisponde un linguaggio costituito da tutte le istruzioni che quella macchina può eseguire
- Ogni istruzione in un linguaggio L_1 è sostituita (tradotta) da una sequenza di istruzioni in linguaggio L_0

Problema:

- per rendere comoda e efficiente la traduzione, la distanza tra L_1 e L_0 (e quindi tra M_1 e M_0) non può essere elevata;
- non è detto che M_1 sia la soluzione desiderata (potrebbe essere ancora troppo distante dal livello *umano*).

Macchina virtuale



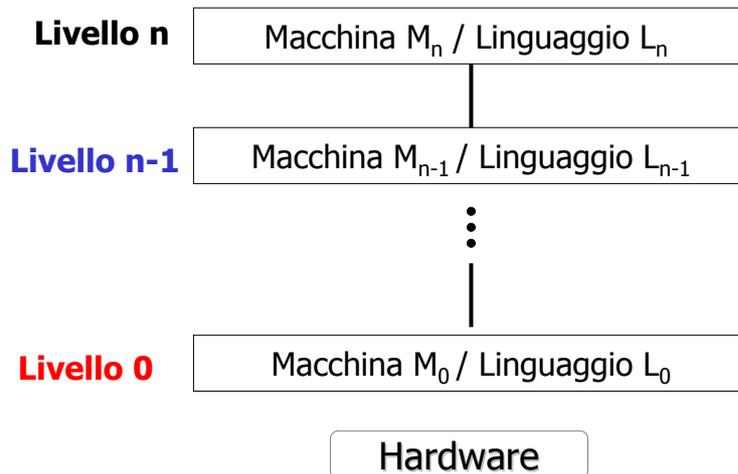
Macchina virtuale M2

- Per la traduzione dei programmi scritti in L_2 esistono due possibilità:

- tradurli in L_1 ed eseguirli con M_1
- tradurli in L_0 ed eseguirli con M_0

- Iterando si possono definire ulteriori macchine virtuali M_3, M_4, \dots, M_n fino a raggiungere il livello di usabilità desiderato.

Livelli di Macchina Virtuale



Relazione tra macchina e relativo linguaggio macchina

- **La macchina definisce il linguaggio**
una macchina (reale o virtuale) permette di definire il **linguaggio macchina** ad essa associato come l'insieme di tutte le istruzioni che la macchina stessa è in grado di eseguire.
- **Il linguaggio definisce la macchina**
un linguaggio permette di definire la macchina (reale o virtuale) ad esso associata come l'esecutore capace di comprendere tutti i programmi scritti in quel linguaggio.

Esecutori e Linguaggi

- **Esecutore** (definito in base a 3 elementi):
 - insieme delle **operazioni** che può compiere
 - insieme delle **istruzioni** che capisce (**sintassi**)
 - quali **operazioni** associa ad ogni **istruzione** che riconosce (**semantica**)
- **Il calcolatore capisce le istruzioni che fanno parte del linguaggio macchina (L0)**

Obiettivo è definire L1 più facile da utilizzare rispetto a L0:

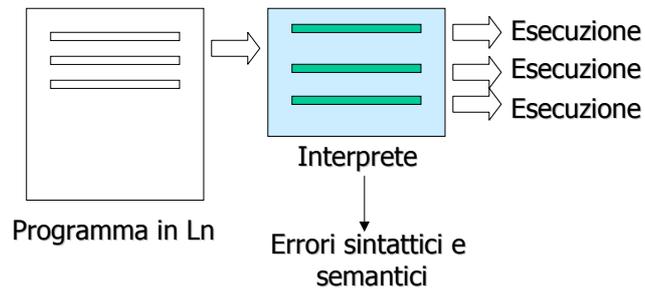
 - utilizzare le istruzioni di L0 come operazioni di L1;
 - definire l'insieme delle istruzioni che fanno parte di L1;
 - definire quali operazioni vengono associate a quali istruzioni.

Traduzione

- traduzione delle istruzioni espresse in linguaggio **L_n**
in istruzioni espresse in linguaggio
linguaggio **L_{n-1}**
- Approcci:
 - *Compilazione*
 - *Interpretazione*
 - *Combinazione*

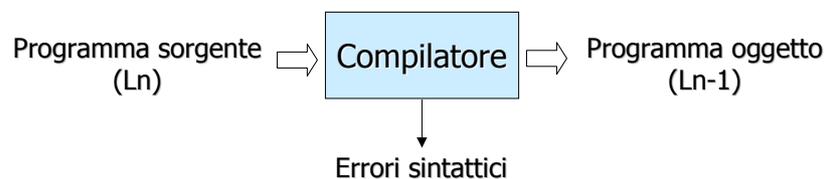
Interpretazione

- ogni istruzione in linguaggio L_n viene esaminata, decodificata e poi **eseguita** immediatamente in una sequenza di istruzioni in linguaggio L_{n-1}



Compilazione

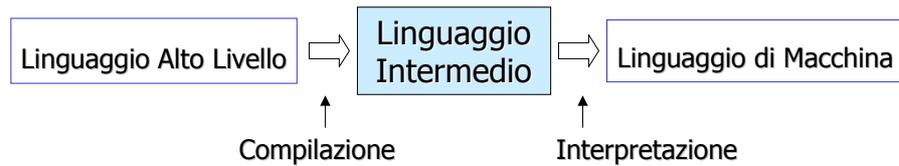
- Un **Compilatore** è un programma che **traduce** un programma scritto in linguaggio L_n in un programma scritto in linguaggio L_{n-1} che viene quindi eseguito



L'elaboratore esegue il programma in L_{n-1} rilevandone gli errori semantici

Approccio Misto

- Definizione di un **linguaggio intermedio**

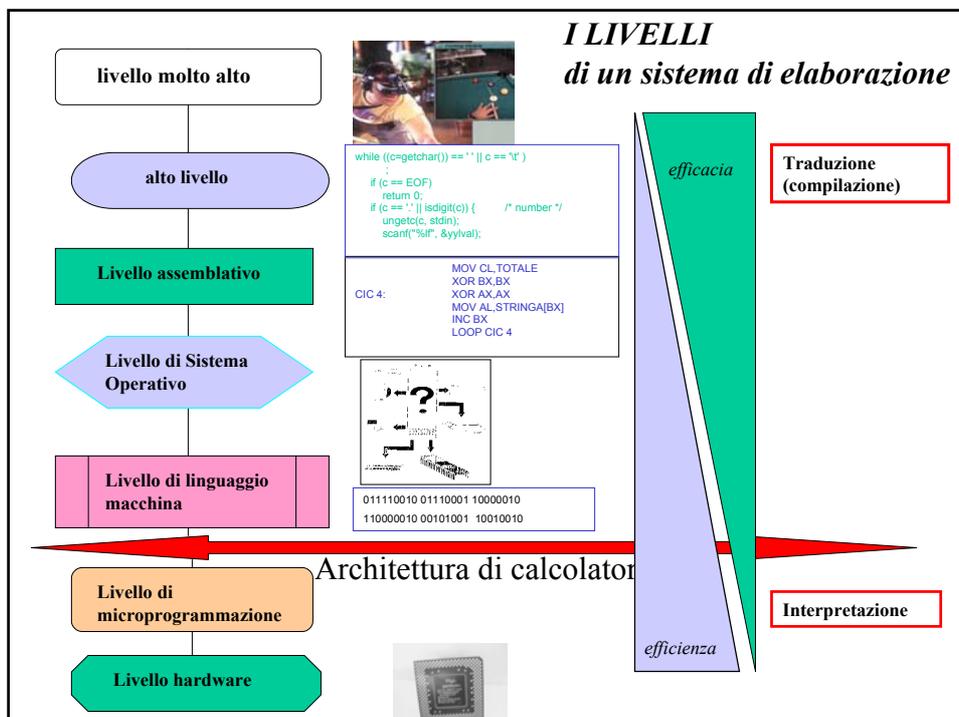


Confronto fra Interpretazione e Compilazione

- **Controllo:**
 - interprete: sempre
 - compilatore: solo nella I fase (trad da PL_n a PL_{n-1})
- **Efficienza e velocità di esecuzione:**
 - Bassa per i linguaggi interpretati
 - Alta per i linguaggi compilati
- **Facilità di messa a punto dei programmi:**
 - Alta per i linguaggi interpretati
 - Bassa per i linguaggi compilati

Compilazione

- Tranne che per la pura interpretazione, un programma su calcolatore subisce almeno 2 fasi:
 - fase di compilazione (**compile-time**)
 - fase di esecuzione (**run-time**)
- **compile-time:**
 - è interessata alle **proprietà statiche** del programma e viene effettuata **una volta**
- **run-time:**
 - è interessata alle **proprietà dinamiche** del programma e viene effettuata **+ volte**



Calcolatore a n-livelli

- **Per scrivere programmi ad alto livello**

non è necessario conoscere come viene effettuata la traduzione e quindi l'esecuzione. Ln non dipende dalla macchina reale (piattaforma-hardware)M0:

- a) diversi linguaggi disponibili sulla stessa piattaforma;
- b) stesso linguaggio disponibile su piattaforme diverse.

- **I programmi possono essere**

- eseguiti direttamente dalla macchina reale
- tradotti direttamente nel linguaggio L0
- interpretati da un interprete che viene a sua volta interpretato da un altro interprete....
- ...

- **La conoscenza dei livelli intermedi è importante per chi voglia capire:**

- come **funziona** un calcolatore
- come si **progetta** una macchina virtuale