Università degli Studi dell'Insubria
Dipartimento di Scienze Teoriche e Applicate

Astrazione e Macchine Virtuali

Il calcolatore

- Un *sistema digitale* che esegue una sequenza di *istruzioni*, chiamate *programma*, operando su dati, per ottenere risultati.

- L'utente adatta il comportamento del calcolatore alle sue esigenze, modificando:
  - il programma
  - i dati

- Un calcolatore è caratterizzato da una *complessa* organizzazione interna

Il concetto di astrazione

- Astrazione = ignorare (nascondere) dettagli (anche rilevanti) per semplificare il problema
- Utile (necessario) per affrontare problemi complessi
- Approccio tipico dell'informatica e dell'ingegneria
- Esempio:
  - DBMS: sistema in grado di gestire dati persistenti,
    - il livello fisico è nascosto, il progettista di una base di dati opera a livello concettuale e logico, l'analista può definire delle viste

L'astrazione e il calcolatore

- Il calcolatore può essere descritto e costruito come una gerarchia di macchine astratte (o virtuali)
  - Ogni livello maschera i dettagli dei livelli sottostanti
- Utile per la descrizione e la costruzione:
  - focus sul funzionamento del livello in esame, non dei sottostanti
  - costruire una 'macchina powerpoint' direttamente con transistor sarebbe di complessità proibitiva…

1. Costruiamo una prima macchina (l'elaboratore) in grado di eseguire istruzioni elementari.
2. Utilizziamo queste istruzioni elementari per scrivere un programma (il Sistema Operativo) in grado di eseguire comandi più complessi
   - Il programmatore del S.O. non conosce i dettagli relativi alla tecnologia realizzativa del microprocessore
3. Scriviamo programmi che si basano sulle funzioni del S.O.
   - il programmatore di powerpoint non conosce i dettagli costruttivi del S.O.

Istruzioni e operazioni

- Un esecutore è definito in base a tre elementi:
  - l'insieme delle **operazioni elementari** che è capace di compiere;
  - l'insieme delle **istruzioni** che capisce (**sintassi**);
  - quali **operazioni** associa ad ogni **istruzione** che riconosce (**semantica**).

- Un elaboratore è in grado di eseguire un insieme di istruzioni
- Per ogni **istruzione** esegue diverse **operazioni**:
  1. Preleva da memoria l'istruzione da eseguire
  2. Esamina l'istruzione e capisce cosa deve fare
  3. Esegue l'istruzione (cioè esegue le operazioni elementari corrispondenti al significato dell'istruzione).

Operazioni fisse (le stesse per ogni istruzione)

Operazioni diverse per ogni istruzione

Differenza tra istruzione e operazioni

- Istruzione (comando): *“Vammi a prendere un caffé”*

- Operazioni:
  - *Mi alzo*
  - *Mi dirigo verso la porta*
  - *Apro la porta*
  - *Raggiungo la macchina del caffé*
  - *…*

- Il comando va convertito nella opportuna serie di operazioni.

Il linguaggio macchina

- Il linguaggio è l'insieme delle istruzioni (comandi) che un esecutore è in grado di comprendere ed eseguire.

- Il linguaggio macchina è il linguaggio compreso dal processore
  - Ovviamente esistono tanti linguaggi macchina quanti sono i processori: Zilog Z80, DEC Alpha, Intel 8080, 80386, Pentium IV, ecc.
- Caratteristiche del linguaggio macchina:
  - Binario: le istruzioni sono delle sequenze di 0 e 1
  - Di basso livello: ogni istruzione ha un effetto molto elementare. Ad es.:
    - Copia il contenuto di un registro in un altro registro
    - Trasferisci il contenuto di una data cella di memoria in un registro
    - Somma i contenuti di due registri e metti il risultato in un terzo registro
    - …
  - Scrivere programmi complessi con istruzioni di questo genere è lungo, tedioso e passibile di errori anche banali.

Il dilemma

- La macchina “semplice” costruita con i transistor mi fornisce un linguaggio inadeguato alla programmazione di programmi complessi e sofisticati.
- Costruire direttamente con i transistor una macchina in grado di supportare la programmazione di alto livello è un’impresa di costo e complessità enormi.

- Che fare?

☞ Teniamo la nostra macchina di basso livello, ma ci “costruiamo sopra” una macchina di più alto livello (cioè più potente e facile da programmare)

Esecutori e linguaggi

- Il calcolatore “capisce” le istruzioni che fanno parte del linguaggio macchina (che indichiamo con **L0**).
- L’obiettivo è definire una macchina che capisca un linguaggio **L1** più potente e facile da utilizzare rispetto al linguaggio macchina **L0**:
  - definire l’insieme delle istruzioni che fanno parte di **L1**;
  - utilizzare le **istruzioni** di **L0** come **operazioni** di **L1**;
  - definire quali operazioni vengono associate a quali istruzioni.

Definizione di una macchina di più alto livello

- La macchina M_0 capisce il linguaggio L_0, che comprende le istruzioni I_0, I_1, I_2
  - $L_0 = \{I_0, I_1, I_2\}$
- M_0 è data e non ci interessa come funziona internamente.
- Definiamo una macchina M_1 che capisce un linguaggio L_1, che comprende le istruzioni J_0, J_1, J_2, J_3
  - $L_1 = \{J_0, J_1, J_2, J_3\}$
  - M_1 è una macchina di più alto livello rispetto a M_0, cioè le sue istruzioni sono più potenti e facili da usare.
  - Tutte le istruzioni di L_1 saranno definite in termini di istruzioni di L_0.
  - Cioè ogni istruzione J_k corrisponde a una sequenza di istruzioni della macchina M_0 ad es.
    - $J_0 = I_0; I_1; I_1; I_2.$
    - $J_1 = I_1; I_2.$
    - $J_2 = I_2; I_0; I_1.$
    - $J_3 = I_2; I_2; I_0$

$J_2; J_0; J_1 \rightarrow ???$