Architetture di calcolo e sistemi operativi

Paolo Trunfio
DIMES
Università della Calabria

paolo.trunfio@unical.it

Informazioni sul corso

- Corso di Laurea:
  - Medicina e Chirurgia - TD (secondo anno – a scelta)
  - Ingegneria Informatica - Bioingegneria (primo anno - obbligatorio)
- SSD: ING-INF/05 – Sistemi di Elaborazione delle Informazioni
- CFU: 6
  - 78 ore di lezione
  - 72 ore di studio individuale
- Docente:
  - Paolo Trunfio
  - Email: paolo.trunfio@unical.it
  - Studio: DIMES, Cubo 41C quinto piano
- Ricevimento:
  - Presso studio docente oppure online, previo appuntamento via email

Corso su Microsoft Teams

- Link: https://teams.microsoft.com/l/team/19%3ajgUa30uamgDcuYPYyIeJfiH-or_-sDS1aZs8Nurl_F81%40thread.tacv2/conversations?groupId=7d2bd593-1598-4c46-86a0-bfa2a79fb62c&tenantId=7519d0cd-2106-47d9-adcb-320023abff57

- Codice di iscrizione: 2jhy54r
- Slide delle lezioni:
  - Canale: Lezioni Prof. Paolo Trunfio
    - Cartella: File

Obiettivi formativi

- Conoscere i principi di funzionamento dei sistemi digitali e la struttura dei sistemi di calcolo; comprendere il funzionamento della memoria, dei meccanismi di input/output e di comunicazione; conoscere le tecniche d'interazione di un computer con dispositivi digitali ad esso connessi.
- Conoscere l'organizzazione dei sistemi operativi; comprendere le tecniche di gestione di esecuzione dei programmi, della memoria principale e della memoria di massa, nonché le caratteristiche dei sistemi operativi per dispositivi utilizzati in ambito medico.

Contenuti

- In accordo agli obiettivi formativi, i contenuti del corso saranno strutturati in due parti principali:
  - **Architetture di calcolo**: si concentra sui principi di funzionamento dei sistemi digitali e sulla struttura dei sistemi di calcolo, sul funzionamento della memoria e sui meccanismi di input/output e di comunicazione.
  - **Sistemi operativi**: si concentra sui principi di progettazione e sulla struttura dei sistemi operativi, sulla gestione dei processi, della memoria principale e di massa, sul file system e sui sistemi operativi per dispositivi medici.

Modalità d'esame

- La valutazione si baserà su due prove.
  - La prima prova, da svolgersi in forma scritta e della durata di due ore, consiste nella risoluzione di esercizi riguardanti architetture di calcolo e sistemi operativi.
  - In caso di superamento della prima prova, lo studente potrà accedere alla seconda prova, da svolgersi oralmente e della durata di circa 30 minuti, costituita da domande sugli argomenti teorici e pratici affrontati durante il corso.

Attribuzione del voto finale

- La prova scritta è valutata in trentesimi. La prova scritta è superata se il voto è maggiore o uguale a 15. In caso di superamento della prova scritta lo studente è ammesso alla prova orale.
- Se la prova orale è giudicata almeno sufficiente, la prova è superata con attribuzione di un voto finale in trentesimi.
- Il voto finale è assegnato tenendo conto in egual misura del voto riportato nella parte scritta dell'esame e della valutazione sulla parte orale dell'esame.

Testi di riferimento

- Libro di testo 1:

  Andrew S. Tanenbaum, Todd Austin, "**Architettura dei calcolatori: un approccio strutturale**" 6a edizione, Pearson Education, 2013

- Libro di testo 2:

  Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "**Sistemi operativi: Concetti ed esempi**", 10a edizione, Pearson Education, 2019.

Concetti introduttivi

Computer e programmi

- **Computer**: una macchina che può risolvere problemi eseguendo le istruzioni che le vengono assegnate.
- **Programma**: sequenza di istruzioni che descrive come portare a termine un dato compito.
- I circuiti elettronici di un computer possono riconoscere ed eseguire direttamente soltanto un *insieme limitato di istruzioni elementari* in cui tutti i programmi devono essere convertiti prima di poter essere eseguiti.

Linguaggio macchina

- Esempi di istruzioni elementari:
  - sommare due numeri
  - controllare se un numero vale zero
  - copiare una porzione di dati da una parte all'altra della memoria.
- L'insieme di queste istruzioni elementari forma il cosiddetto **linguaggio macchina**, attraverso il quale è possibile comunicare con il computer.

Linguaggio macchina

- Chi progetta un computer deve decidere quali istruzioni costituiranno il suo linguaggio macchina.
- Per ridurre la complessità e il costo dei dispositivi elettronici, si cerca di progettare le più semplici istruzioni primitive che siano compatibili con il tipo di utilizzo e i requisiti prestazionali del computer.
- Poiché quasi tutti i linguaggi macchina sono semplici ed elementari, risulta difficile e noioso utilizzarli.

Approccio strutturale

- Per gestire più agevolmente la complessità dei computer e progettarli in modo sistematico e organizzato, i computer sono strutturati come una serie di *livelli di astrazione*, ciascuno dei quali è costruito sulla base di quello sottostante.

- Questo modo di concepire l'architettura dei computer è detto **approccio strutturale.**

Linguaggi, livelli e macchine virtuali

- **Problema**: Gli utenti vogliono fare *X*, ma i computer possono fare soltanto *Y*
- **Soluzione**: Definizione di un *nuovo insieme d'istruzioni* che sia più comodo da utilizzare rispetto a quanto non lo siano le istruzioni predefinite del linguaggio macchina.
- Anche questo nuovo insieme d'istruzioni forma un linguaggio (che chiamiamo **L1**), allo stesso modo in cui le istruzioni macchina formavano a loro volta un linguaggio (che chiamiamo **L0**).

Linguaggi, livelli e macchine virtuali

- Come eseguire un programma scritto in L1?
  - **Traduzione**: Sostituire, in una fase iniziale, ogni istruzione del programma scritto in L1 con un'equivalente sequenza di istruzioni in L0. Il programma che ne risulta è costituito interamente da istruzioni di L0 e può essere eseguito dal computer al posto del programma L1 originale.
  - **Interpretazione**: Scrivere un programma in L0 (detto *interprete*) che accetta come dati d'ingresso programmi in L1. Tale *interprete* esegue un programma in L1 esaminando un'istruzione alla volta e sostituendola direttamente con l'equivalente sequenza di istruzioni L0.

Linguaggi, livelli e macchine virtuali

- Si può immaginare l'esistenza di un ipotetico computer, o **macchina virtuale**, il cui linguaggio macchina sia L1. Chiamiamo questa macchina virtuale M1, e chiamiamo M0 la macchina virtuale corrispondente al linguaggio L0.

- Per rendere la traduzione o l'interpretazione utilizzabili in pratica, i linguaggi L0 e L1 non devono essere troppo diversi fra loro. Per la maggior parte delle applicazioni questo vincolo fa sì che L1, pur essendo migliore di L0, sia spesso ancora lontano dal linguaggio ideale.

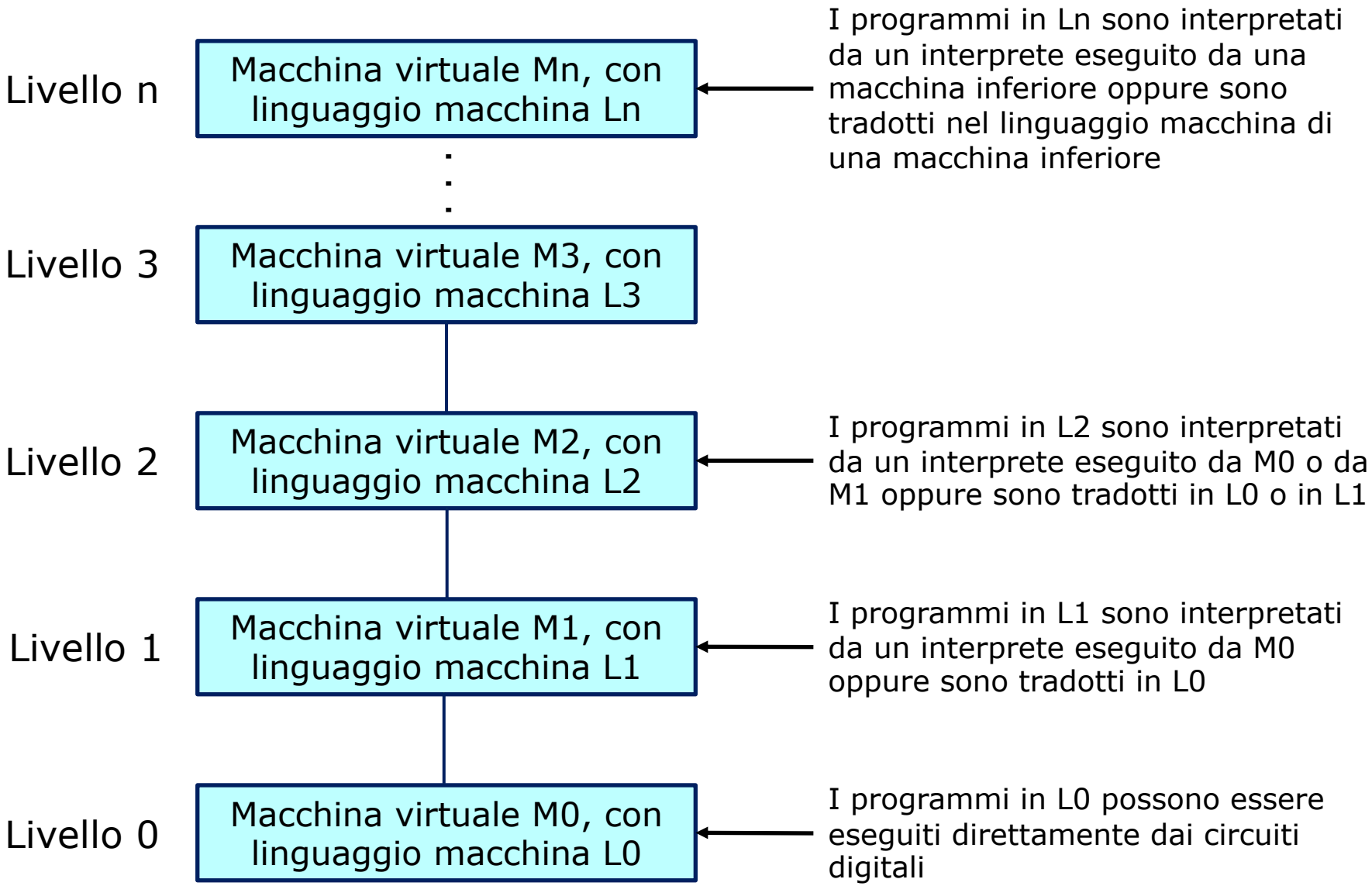
Linguaggi, livelli e macchine virtuali

- Per risolvere questo problema, si può definire un nuovo insieme d'istruzioni che sia, rispetto a L1, maggiormente orientato agli utenti piuttosto che alle macchine.
- Anche questo terzo insieme forma a sua volta un linguaggio, che chiamiamo L2 (e la cui corrispondente macchina virtuale sarà M2).
- Si possono scrivere direttamente programmi in L2 come se esistesse realmente una macchina virtuale dotata di tale linguaggio macchina.
- Questi programmi possono essere tradotti in L1 oppure eseguiti da un interprete scritto in L1.

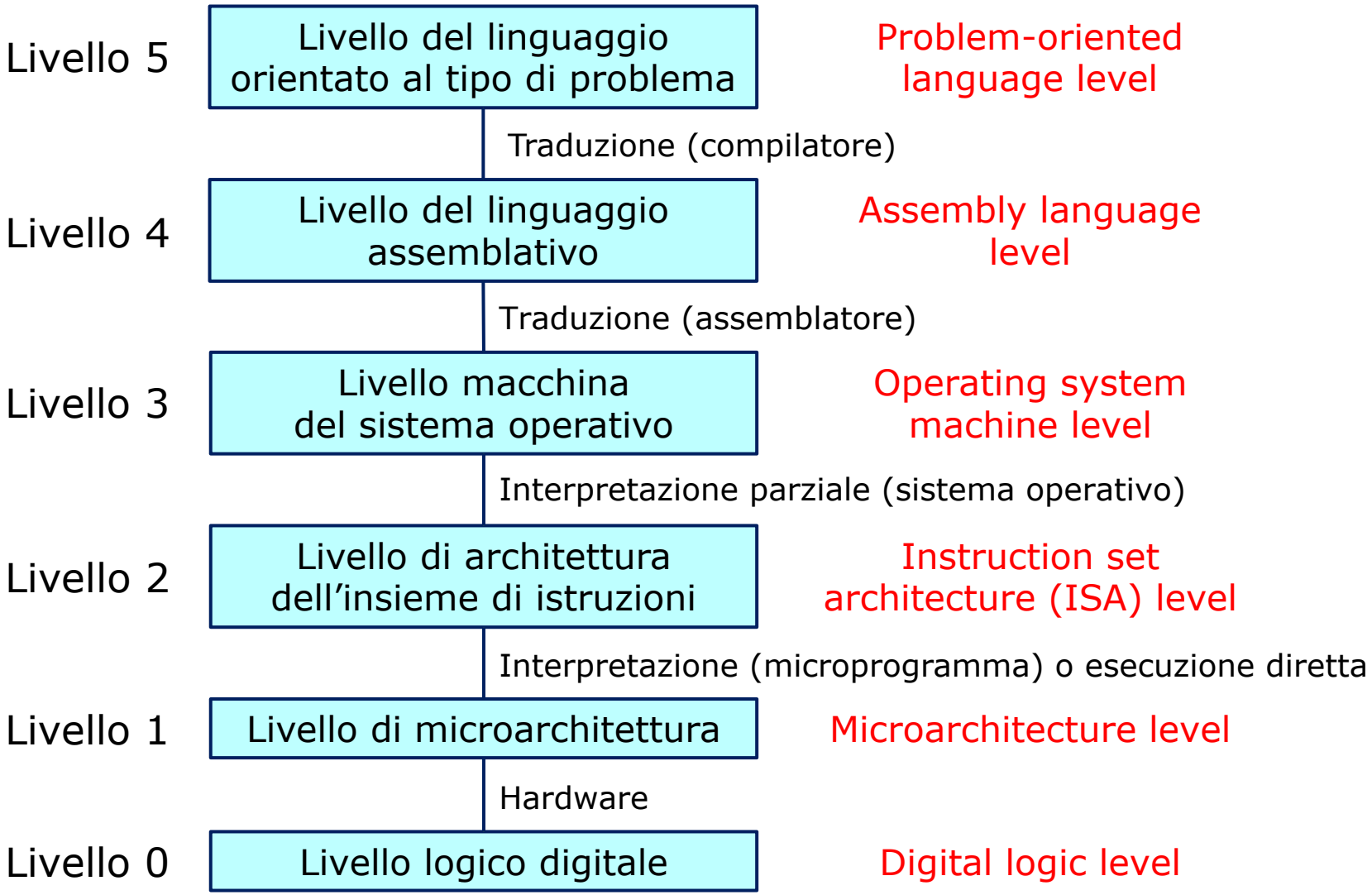
Linguaggi, livelli e macchine virtuali

- La definizione di una serie di linguaggi, ciascuno dei quali più pratico da utilizzare rispetto al precedente, può continuare indefinitamente finché non se ne ottenga uno sufficientemente adeguato.
- Ciascun linguaggio utilizza il precedente come base; un computer che usa questa tecnica può quindi essere immaginato come una serie di **strati** o **livelli** disposti l'uno sopra l'altro.
- Il livello, o linguaggio, che si trova più in basso è il più semplice, mentre quello più in alto è il più sofisticato.

Linguaggi, livelli e macchine virtuali



Modello di macchina a 6 livelli



Modello di macchina a 6 livelli

- **Livello logico digitale** (**livello 0**): rappresenta l'hardware della macchina, i cui circuiti eseguono i programmi scritti nel linguaggio macchina del livello 1.
- Gli elementi di base del livello 0 sono le **porte** (**gate**).
- Ciascuna porta è dotata di uno o più input digitali (segnali corrispondenti ai valori 0 o 1) e calcola in output una semplice funzione dei valori d'ingresso, per esempio AND od OR.
- E possibile combinare un piccolo numero di porte per formare una memoria a 1 bit, che può memorizzare i valori 0 e 1.
- Combinando le memorie a 1 bit in gruppi, per esempio di 16, 32 o 64, è possibile formare i cosiddetti **registri**.
- Ogni registro può contenere un numero il cui valore può variare fino a un certo limite.

Modello di macchina a 6 livelli

- **Livello di microarchitettura** (**livello 1**)**:** contiene una memoria locale, formata da un gruppo di registri (in genere da 8 a 32), e un circuito chiamato ALU (*Arithmetic Logic Unit,* unità aritmetico-logica), capace di effettuare semplici operazioni aritmetiche e logiche.

- I registri sono connessi alla ALU per formare un **percorso dati** (**data path**) lungo il quale si spostano i dati. L'operazione base del percorso dati consiste nel selezionare uno o due registri, permettere alla ALU di operare su di loro (per esempio sommandoli) e memorizzare infine il risultato in uno dei registri.

- In alcune macchine, le operazioni del percorso dati sono controllate da un programma chiamato **microprogramma,** mentre su altre il percorso dati è controllato direttamente dall'hardware.

Modello di macchina a 6 livelli

- **Livello di architettura dell'insieme d'istruzioni** (**livello 2**): Più noto con l'acronimo inglese di **livello ISA** *(Instruction Set Architecture level)* si può definire come l'aspetto che il computer assume agli occhi di un programmatore in linguaggio macchina.
- Ogni diversa CPU ha un proprio ISA e quindi istruzioni diverse spesso non compatibili tra loro.
- Il termine **assembly** si riferisce ad un linguaggio costituito da codici mnemonici corrispondenti alle istruzioni ISA.
- Al fine di produrre codice di livello ISA, si deve conoscere il modello di memoria, quali registri ci sono, quali sono i tipi di dati e di istruzioni disponibili, e così via; l'insieme di tutte queste informazioni definisce il livello ISA.

Modello di macchina a 6 livelli

- Il **livello macchina del sistema operativo (livello 3)** è generalmente un livello *ibrido*. La maggior parte delle istruzioni nel suo linguaggio fa parte anche del livello ISA.
- Inoltre, vi è un insieme di nuove istruzioni, una diversa organizzazione della memoria e la capacità di eseguire programmi in modo concorrente, oltre a varie altre funzionalità.
- I nuovi servizi aggiunti nel livello 3 sono realizzati da un interprete eseguito al livello 2, storicamente chiamato *sistema operativo*. Le istruzioni del livello 3, identiche a quelle del livello 2, sono eseguite direttamente dal microprogramma (o dai circuiti elettronici) e non dal sistema operativo.
- In altre parole, alcune delle istruzioni del livello 3 sono interpretate dal sistema operativo, mentre altre sono interpretate in modo diretto dal microprogramma; per questo motivo tale livello viene detto ibrido.

Modello di macchina a 6 livelli

- Tra i livelli 3 e 4 vi è una spaccatura fondamentale. I tre livelli inferiori sono concepiti principalmente per eseguire interpreti e traduttori necessari come supporto per i livelli più alti.
- Questi interpreti e traduttori sono scritti dai cosiddetti **programmatori di sistema,** specializzati nella progettazione e nell'implementazione di nuove macchine virtuali.
- Il livello 4 e quelli superiori sono invece pensati per i programmatori che devono risolvere problemi applicativi.
- I linguaggi macchina dei livelli 1, 2 e 3 sono numerici; i loro programmi consistono quindi in lunghe serie di numeri, certamente più adatte alle macchine che agli esseri umani.
- A partire dal livello 4 i linguaggi contengono invece parole e abbreviazioni umanamente comprensibili.

Modello di macchina a 6 livelli

- **Livello del linguaggio assemblativo** (**livello 4**): fornisce ai programmatori un modo per scrivere programmi per i livelli 1, 2 e 3 in una forma meno difficile rispetto a quella dei linguaggi delle rispettive macchine virtuali.
- I programmi in linguaggio assemblativo sono inizialmente tradotti nei linguaggi dei livelli 1, 2 o 3 e successivamente interpretati dall'appropriata macchina virtuale o reale.
- Il programma che esegue la traduzione è chiamato **assemblatore.**

Modello di macchina a 6 livelli

- **Livello del linguaggio orientato al tipo di problema** (**livello 5**): consiste generalmente di linguaggi, spesso chiamati **linguaggi ad alto livello,** definiti per essere utilizzati dai programmatori di applicazioni.
- Ne esistono letteralmente a centinaia; alcuni fra i più conosciuti sono C, C++, Python e Java.
- I programmi scritti in questi linguaggi sono generalmente tradotti al livello 3 o al livello 4 da un traduttore detto **compilatore;** in casi meno frequenti è anche possibile che essi siano interpretati.
- I programmi in Java, per esempio, di solito sono tradotti inizialmente in un linguaggio di tipo ISA chiamato *Java byte code,* che viene successivamente interpretato.

Conclusioni

- Il concetto chiave è che i computer sono progettati come una serie di livelli, ciascuno costruito su quelli che lo precedono.
- Ciascun livello rappresenta una diversa astrazione, caratterizzata dalla presenza di oggetti e operazioni diversi.
- Progettando e analizzando i computer in questo modo è possibile tralasciare temporaneamente i dettagli irrilevanti, riducendo di conseguenza un soggetto complesso in qualcosa di più facile comprensione.