

Architettura degli elaboratori

Docenti:

- **Pietro Di Gianantonio**
- Alberto Ciaffaglione: laboratorio
- Nicola Prezza: laboratorio



(Architettura degli Elaboratori)

Presentazione corso

1 / 47

Obiettivi del corso

- Spiegare come sono fatti, fisicamente, i calcolatori e come funzionano:
- quali sono i loro componenti base e come questi interagiscono tra loro.
- Nella tradizionale distinzione: hardware – software:
questo è il corso di **hardware**
Argomenti relativi all'hardware vengono trattati anche nel corso di Reti di Calcolatori.

(Architettura degli Elaboratori)

Presentazione corso

2 / 47

Organizzazione: sdoppiamento

Il corso è diversificato tra Informatica e TWM.

- **Informatica**: 12 **crediti didattici** — 72 ore di lezione + 36 ore di laboratorio.
- **TWM**: 6 crediti didattici — 48 ore di lezione.

Motivazioni:

- i due corsi di laurea hanno obiettivi differenti

(Architettura degli Elaboratori)

Presentazione corso

3 / 47

Conseguenze

- Alcuni argomenti (progettazione di circuiti, programmazione Assembly) presentati solo a Informatica. Argomenti trattati anche nel corso di laboratorio.
- Corso semestrale.
- INF: 3 lezioni a settimana + laboratorio.
TWM: 2 lezioni a settimana.

(Architettura degli Elaboratori)

Presentazione corso

4 / 47

Organizzazione

Le ore di lezione divise tra ore per presentazione della teoria ed esercitazioni.

Orario di ricevimento giovedì: 14:30 – 16:30.

Laboratorio - solo Informatica

- Solo per Informatica.
- Inizio: mercoledì 7 ottobre.
- 3 crediti: 36 ore di laboratorio (2-4 ore a settimana).
- È necessario registrarsi (ora), per poter accedere ai calcolatori
modulo scaricabile dalla pagina web:
`web.uniud.it/didattica/facolta/scienze/info_dida/laboratorio/`
da consegnare negli uffici tra Lab1 e Lab2

Contenuti

Corso in parte **nozionistico - descrittivo**:

- si descrivono le diverse parti del calcolatore e come interagiscono,
- vengono presentati principi di funzionamento e idee base.
- non si va nel dettaglio, pochi aspetti tecnici-progettuali,
- come sempre, le nozioni vanno **comprese**, non solo memorizzate; bisogna:
 - capire i meccanismi di funzionamento delle diverse componenti,
 - costruirsi una visione generale, collegare tra loro le diverse nozioni.

Contenuti

Tre aspetti più **metodologici**, progettuali:

- algebre booleane - progettazione di circuiti logici,
- rappresentazione binaria dei numeri,
- programmazione assembly del processore ARM (tablet, smartphone).

Obiettivo di queste parti del corso: insegnare a *fare*, farvi acquisire abilità.

Progettazione circuiti e programmazione assembly vengono trattati anche a laboratorio.

Esame

- scritto: alcuni esercizi e una lista di domande sulla teoria,
- **per Informatica:** prova di laboratorio,
- orale: alcune domande di teoria.

5 appelli nel corso dell'anno.

Consigli

Per superare l'esame:

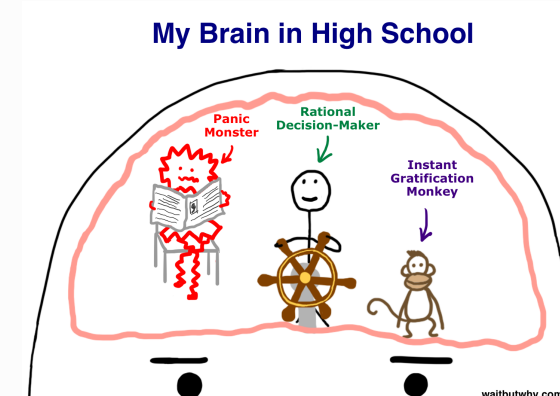
- bisogna principalmente studiare, capendo gli argomenti studiati.
- è necessario svolgere esercizi (per la preparazione agli scritti).
- Ci sono pochi numeri o sigle da ricordare a memoria, è sufficiente conoscere gli ordini di grandezza dei valori in gioco.

Considerazioni generali

- L'università fornisce pochi stimoli a studiare durante l'anno (interrogazioni, compiti in classe).
- Si richieda una maggiore maturità/
- É facile perdersi: 30% degli iscritti al primo anno non si iscrive al secondo.

Pericolo: rimandare troppo lo studio

Un modello del cervello



Dal blog www.waitbutwait.com,
Why procrastinators procrastinate.

Psicologia pratica

Fissarsi una serie di obiettivi, a breve termine, verificabili.

Obiettivi a lungo termine, o generali sono inutili.

- Ogni settimana seguire il 90% delle lezioni.
- Ripassare ogni giorno, gli argomenti trattati a lezione.
- Ripassare a sufficienza gli argomenti trattati da seguire agevolmente le lezioni.

Psicologia pratica

Vincolarsi agli obiettivi.

- Dichiarare agli altri i propri impegni.
- Studiare con amici.

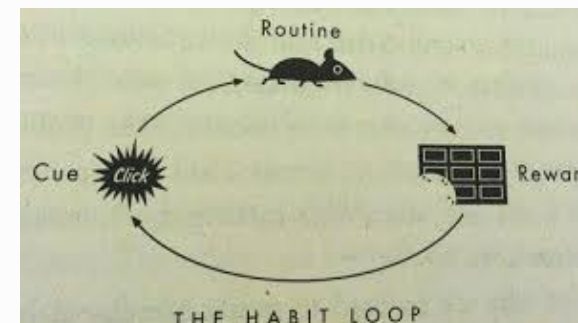
Abitudini

Non affidarsi alla propria forza di volontà, sfruttare le abitudini.

Crearsi abitudini virtuose, abbandonare quelle dannose.

Continue piccole modifiche nelle abitudini portano ad importanti risultati.

Meccanismo



Eliminare (o introdurre) il segnale di attivazione, Sostituire la risposta.

Per questo corso

- **Distribuire il carico di lavoro**, non rimandare tutto alle due settimane precedenti l'esame.
- Non perdere il contatto con ciò che viene presentato a lezione.
- Sfruttate l'orario di ricevimento e, in generale, i servizi di tutorato.
- Per Informatica: seguire le lezioni di laboratorio.

Libro di testo

Andrew S. Tanenbaum, *Architettura dei calcolatori, un approccio strutturale*. VI edizione.

Pearson, 2013,

versione inglese:

Structured computer organization, VI edition.

Pearson Prentice Hall, 2012.

È possibile utilizzare anche la **V** edizione del libro di testo, solo poche parti sono state aggiornate nella nuova edizione.

Pregi

- Un testo classico (un bestseller).
- Chiaro e piacevole da leggere.
- L'autore è un autorità nella didattica dell'informatica e nel campo dei sistemi operativi. Autore di MINIX, un precursore di LINUX.
- Il corso segue da vicino il libro di testo.

Difetti

- Non completamente aggiornato, la nuova edizione solo 5% diversa dalla versione di 7 anni prima.
- Si danno per noti una serie di concetti: input-output, memoria, transistor, analogico - digitale.
- Molto discorsivo. Vengono evitati gli argomenti difficili, tecnici, quantitativi.
- Ordine di presentazione degli argomenti poco ortodosso:
 - capitolo 2: descrizione superficiale del calcolatore, riassunto di tutti gli argomenti;
 - capitoli successivi: descrizione dettagliata.

Altri riferimenti

- le pagine web del corso:

`www.dimi.uniud.it/pietro/architetturaInf`

`www.dimi.uniud.it/pietro/architetturaTWM`

raggiungibili dalla mia home page:

`www.dimi.uniud.it/pietro`

- istruzioni sul corso
- lista delle pagine di testo svolte
- testi di esame
- lucidi presentati a lezione, **non sostituiscono il libro di testo.**

Appunti ed esercizi.

- Non è indispensabile prendere appunti (si trova quasi tutto sui testi).
- Fonti di esercizi: esercitazioni, laboratorio, pagina web del corso.

Altri riferimenti

- Documenti relativi

- alla programmazione assembly,
- alla progettazione di circuiti

sono scaricabili dalla pagina web del corso.

- Morris M. Mano *Computer system architecture*, 3rd edition Prentice Hall, 1993.
nel primo capitolo: descrizione dei circuiti sequenziali.

Per i più curiosi

- **Wikipedia** (preferibilmente nella versione inglese)
per approfondimenti su singoli argomenti.
- Per chi cercasse una trattazione alternativa degli argomenti svolti:
J. Hennessy, D. Patterson *Architettura degli elaboratori*, Apogeo.
un altro testo classico; più tecnico, quantitativo e progettuale.

Corso di azzeramento

Un calcolatore: macchina capace di eseguire una sequenza di istruzioni semplici, **istruzioni macchina**. Queste sequenze di istruzioni sono chiamate **programmi**.

Il calcolatore mediante l'esecuzione, molto veloce, di programmi di grosse dimensioni, può avere dei comportamenti molto sofisticati.

Programmi, e i dati, su cui il programma lavora sono organizzati in **file** e contenuti nella memoria del calcolatore.

Memoria

La memoria del calcolatore è di due tipi:

- permanente, **memoria di massa, memoria secondaria, disco rigido** contiene in maniera permanente i dati del calcolatore
- operativa, **memoria principale, memoria RAM** contiene i dati al momento della loro elaborazione.

Memoria

La capacità delle memorie si misura in **byte**, o in suoi multipli:

- **KB** (chilo byte $2^{10} = 1024 \sim 1.000$ byte),
- **MB** (mega byte $2^{20} \sim 1.000.000$ byte)
- **GB** (giga byte $2^{30} \sim 1.000.000.000$ byte).

Il byte è uno spazio di memoria sufficiente a memorizzare un carattere, o un numero naturale tra 0 e 255.

Processore

Il cuore del calcolatore è il **processore (CPU)** il circuito che materialmente esegue le istruzioni. Processore, e memoria principale, vengono realizzati mediante particolari circuiti chiamati **circuiti integrati** o **chip**, un chip ha la dimensione di pochi centimetri e può contenere miliardi di transistor, un solo chip è sufficiente per realizzare un processore.

Bus

Il processore scambia i dati con l'esterno attraverso una serie di **periferiche**: schermo, tastiera, stampante, modem, scanner, lettore CD, . . . , Processore, memoria, periferiche collegati tra loro, attraverso degli insiemi di fili **bus**, formano il calcolatore.

Sistema Operativo

L'interazione tra utente e calcolatore avviene tramite un particolare programma, il **sistema operativo**. Il sistema operativo è permanentemente in **esecuzione**.

Fornisce funzionalità agli altri programmi.

Attraverso il sistema operativo si può mettere in esecuzione, far svolgere **programmi applicativi**, es. browser, word-processor.

Unico corso su hardware.

L'informatica si occupa principalmente del software:

- infatti: possiamo interagire con un calcolatore senza sapere come questo è fatto fisicamente
- similmente a come possiamo guidare un'auto senza conoscere meccanica e termodinamica.

D'altra parte ...

- ci si aspetta che un meccanico o un ingegnere abbiano una conoscenza approfondita della meccanica,
- ma la stessa padronanza non è indispensabile ad un programmatore - progettista.

È ammessa una parziale ignoranza

- il calcolatore è un oggetto troppo complesso per poterlo conoscere completamente,
- un sistema di calcolo (calcolatore e relativo software) è strutturato in maniera tale si possa interagire con esso conoscendo solo alcune funzionalità, e ignorando come queste sono implementate,
- lo studio, la progettazione di un sistema di calcolo può essere affrontato a vari livelli (hardware, software).

Macchine virtuali

Uno dei concetti base dell'informatica.

Necessario per gestire l'enorme complessità di un sistema di calcolo.

Il calcolatore può essere visto come una **macchina a livelli**:

un insieme stratificato di **macchine virtuali**.

Concetti illustrati tramite un esempio.

Esempio

- **Un utente** naviga in rete: interagisce col browser, gli è sufficiente conoscere i comandi del browser
- **Il browser**: un programma applicativo, creato da programmatori mediante un linguaggio di programmazione come C, Java, Pascal.
- **Linguaggio di programmazione**: implementato da altri programmatori, si basa sull'uso del sistema operativo e delle istruzioni macchina,

Esempio

- **Sistema operativo**: insieme di funzioni base per interagire con il calcolatore, realizzato da altri programmatori, si basa sulle istruzioni macchina.
- **Istruzioni macchina**: i comandi base del calcolatore (processore), progettato da ingegneri a partire dalle porte logiche
- **Porte logiche**: eseguono le operazioni elementari, realizzate mediante transistor.

Macchine virtuali

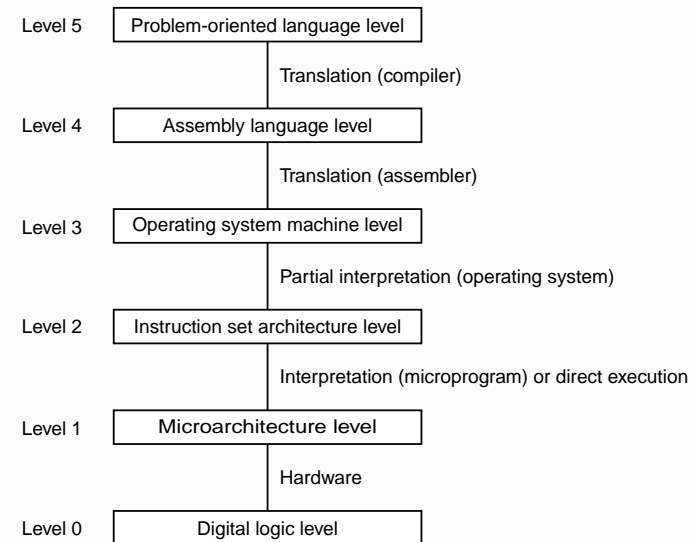
Sistema di calcolo: serie di macchine virtuali.

Ciascuna fornisce una serie di funzionalità e viene realizzata a partire da una macchina virtuale sottostante.

I progettisti possono occuparsi solo di una parte del sistema, macchina virtuale.

Si “fattorizza” il lavoro: una stessa macchina virtuale viene utilizzata per implementare diverse macchine virtuali sovrastanti. (Tanti sistemi operativi sullo stesso hardware, tanti linguaggi di programmazione per un stesso sistema operativo, ...)

Esempio di schema a livelli



Motivi per studiare l'hardware

- **Completezza culturale:** avere una conoscenza complessiva dei sistemi di calcolo.
- **Utilità:**
 - valutare, scegliere o gestire l'hardware,
 - conoscere i fattori che determinano le prestazioni,
 - gestire i malfunzionamenti.

Ordine di presentazione degli argomenti

La struttura a livelli è il filo conduttore del corso: In questo corso vengono presentati i 3 livelli più bassi: l'hardware

L'ordine di presentazione è dal basso verso l'alto.

Il prossimo anno, nel corso di Sistemi Operativi, studierete il quarto livello.

In altri corsi (Linguaggi di programmazione, basi di dati) i livelli superiori.

Più in dettaglio

- porte logiche,
- circuiti logici, memorie,
- processore,
- programmazione assembly,
- dispositivi periferici,
- bus, collegamenti tra dispositivi,
- calcolatori paralleli, sistemi multiprocessore.

Storia dei sistemi di calcolo

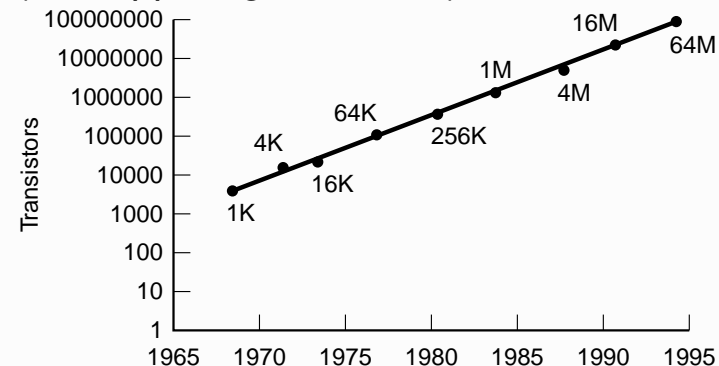
- 1642 Pascal — calcolatrice meccanica per somma e sottrazione
- 1670 Leibniz — calcolatrice meccanica per prodotto e divisione
- 1834 Babbage — Analytical engine — meccanica programmabile
- 1930 Zuse — Macchina calcolatrice a relè.
- 1944 Aiken — Mark I — macchina programmabile a relè.

Storia dei sistemi di calcolo

- 1946 Ecker-Mauchley — Eniak — primo calcolatore a valvole
- 1951 Von Neumann — IAS — con architettura simile agli attuali computer
- 1954 IBM — primi calcolatori a transistor
- 1958 — Calcolatori a circuiti integrati.

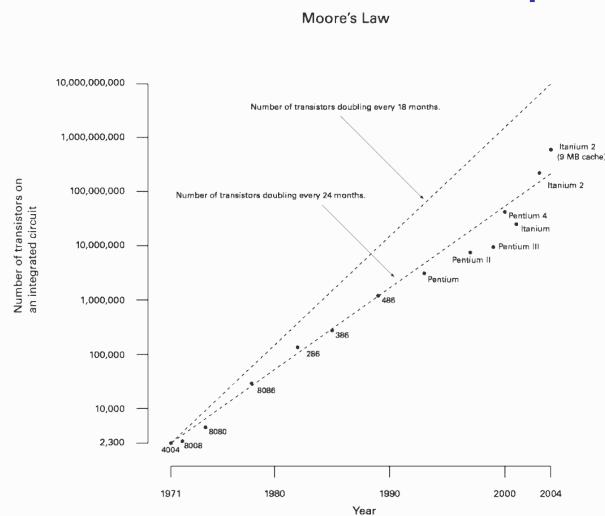
Progressi della tecnologia

Legge di Moore (Gordon Moore, 1965): il numero dei transistor all'interno di una (circuito integrato) chip di memoria quadruplica ogni tre anni, (raddoppia ogni 18 mesi).



Numero di transistor nei di memorie

Num. di transistor per processore



(Architettura degli Elaboratori)

Presentazione corso

45 / 47

Progressi della tecnologia

Progressi analoghi anche per le altre componenti.

- capacità dei dischi magnetici, memorie flash.
- velocità dei bus e reti di interconnessione,
- risoluzione delle fotocamere e stampanti.

(Architettura degli Elaboratori)

Presentazione corso

46 / 47

Pervasive computing

Ci si muove verso una frammentazione, distribuzione dei sistemi di calcolo:

- Mainframe.
- Minicomputer, server.
- Workstation – Personal Computer.
- Tablet – Smartphone.
- Embedded Computer.

(Architettura degli Elaboratori)

Presentazione corso

47 / 47