

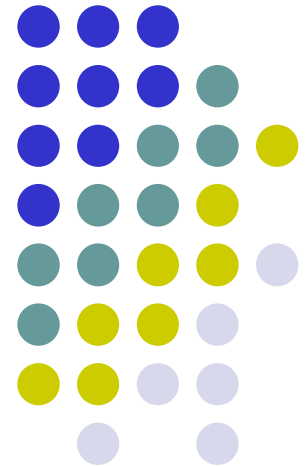
# Laboratorio di Calcolatori 1

Corso di Laurea in Fisica

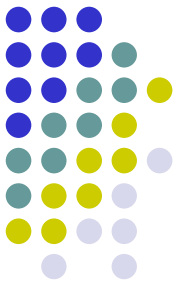
A.A. 2006/2007

Dott. Davide Di Ruscio

Dipartimento di Informatica  
Università degli Studi di L'Aquila

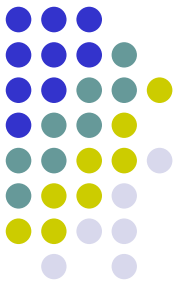


# Sommario (I parte)



- Concetti fondamentali
- Aspetti architettureali di un sistema di calcolo
  - hardware
  - software
    - software di base
    - software applicativo
- Codifica dell'informazione
  - numeri naturali, interi, reali
  - caratteri
  - immagini
- Macchina di Von Neumann
  - CPU (UC, ALU, registri, clock)
  - memoria centrale
  - bus di sistema
  - periferiche
- Linguaggio macchina
- Linguaggio assembler
- Sistema operativo
- Ambiente di programmazione

# Sommario (II parte)



## Il Linguaggio C

- Caratteristiche generali
- Un linguaggio C semplificato ed esempi di semplici programmi
- Struttura di un programma C
- Direttive del pre-processore
- Parte dichiarativa:
  - tipi
  - definizioni di tipi
  - definizioni di variabili
- Parte esecutiva
  - istruzione di assegnamento
  - istruzioni (funzioni) di input-output
  - istruzioni di selezione
  - istruzioni iterative
- Vettori mono e multidimensionali
- Funzioni e procedure
- File
- Allocazione dinamica di memoria
- Suddivisione dei programmi in piu' file e compilazione separata

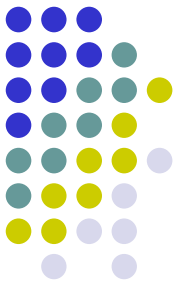
- Algoritmi elementari
  - ricerca sequenziale e binaria
  - ordinamento di un vettore: per selezione, per inserimento, per fusione e a bolle
- Aspetti avanzati di programmazione
  - ricorsione
  - strutture dati dinamiche

## RIFERIMENTI

Ceri, Mandrioli, Sbattella  
[Informatica arte e mestiere](#)  
McGraw-Hill

N.B. (copyright): alcuni di questi lucidi sono tratti dal materiale distribuito dalla McGraw-Hill

# Alcuni testi consigliati



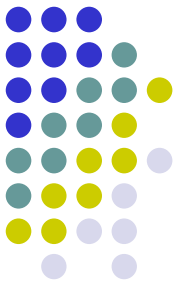
Ceri, Mandrioli, Sbattella  
Informatica Arte e Mestiere  
McGraw Hill

Deitel e Deitel  
C corso completo di programmazione  
Apogeo

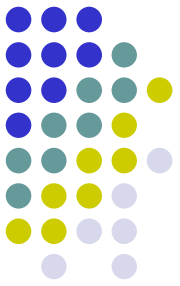
Kernighan e Ritchie  
Il linguaggio C  
Pearson Education Italia

Kelly e Pohl  
C didattica e programmazione  
Pearson Education Italia

# Lucidi delle lezioni



I lucidi del corso sono basati su del materiale fornito dal Prof. Flammini Michele



# Contatti

Sito del corso:

[www.di.univaq.it/diruscio/labcalc1](http://www.di.univaq.it/diruscio/labcalc1)

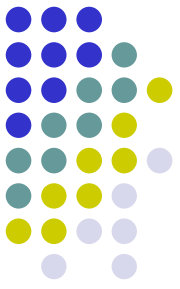
Email:

[diruscio@di.univaq.it](mailto:diruscio@di.univaq.it)

Ricevimento:

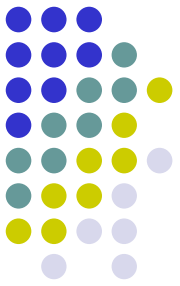
Mercoledì 11:30 – 12:30, stanza A1/2, Coppito 2

# Concetti di base (1/6)



- **INFORMATICA:** scienza della rappresentazione e dell'elaborazione dell'informazione.
- **ELABORATORE:** non è l'oggetto fondamentale di studio nell'informatica, ma è l'attore principale, poiché permette l'elaborazione di grandi quantità di dati in poco tempo, rendendo quindi realizzabili nella pratica i principi e le tecniche informatiche.

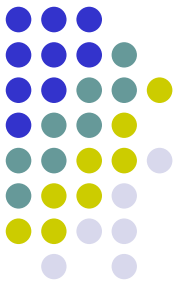
# Concetti di base (2/6)



- **ALGORITMO:** procedura passo passo o sequenza di passi per la risoluzione di un determinato compito
- **PASSO:**
  - compiuto da un ipotetico esecutore
  - elementare: corrisponde ad un'azione semplice del suo esecutore
  - limitato: un lavoro "illimitato" può essere eseguito soltanto con un numero "illimitato" di passi
  - comprensibile dall'esecutore
  - non ambiguo: non deve dare adito ad alcuna libera interpretazione da parte dell'esecutore
- **NB:** l'esecutore per eccellenza è il calcolatore, ma possiamo essere anche noi esecutori quando effettuiamo un prelievo da uno sportello del bancomat, quando montiamo apparecchiature o assembliamo componenti, ...

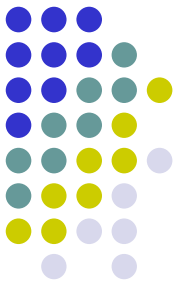


# Concetti di base (3/6)



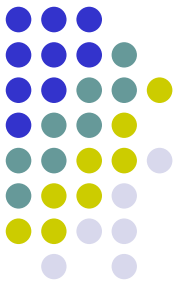
- PROPRIETÀ DI UN ALGORITMO:
  - **correttezza**: deve effettivamente risolvere il compito a cui è preposto, senza difettare di alcun passo fondamentale
  - **efficienza**: deve pervenire alla risoluzione del compito utilizzando il minor numero possibile di risorse, come ad esempio il tempo (numero di passi eseguiti) e la memoria
  - **robustezza**: anche in presenza di input errati, situazioni non previste o anomalie deve mantenere la correttezza
  - ...

# Concetti di base (4/6)



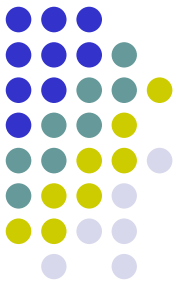
- **PROGRAMMA:** algoritmo descritto in un linguaggio comprensibile dall'esecutore
- **ISTRUZIONE**  $\approx$  passo
- **LINGUAGGIO DI PROGRAMMAZIONE:** linguaggio usato per descrivere l'algoritmo
- **CODIFICA:** processo di descrizione o trascrizione di un algoritmo in un linguaggio di programmazione

# Concetti di base (5/6)



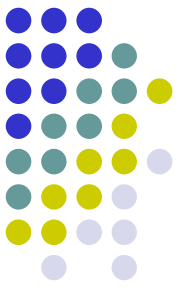
- **LINGUAGGIO MACCHINA:**
  - l'unico comprensibile dall'elaboratore
  - istruzioni date da sequenze di bit (es. 011001100000).
- **LINGUAGGIO ASSEMBLER:**
  - utilizza nomi simbolici per indicare le istruzioni del linguaggio macchina ed i dati su cui operano, come ad esempio READ X invece di 011001100000
  - ha la stessa potenza del linguaggio macchina, poiché esiste una corrispondenza uno-a-uno tra le rispettive istruzioni e quindi non aggiunge nuove potenzialità
- **ASSEMBLATORE:** traduce un programma in linguaggio assembler in uno equivalente in linguaggio macchina che può essere eseguito dall'elaboratore

# Concetti di base (6/6)



- **LINGUAGGIO AD ALTO LIVELLO:** istruzioni più sintetiche, più potenti, più vicine al modo di ragionare umano
- **COMPILATORE:** traduce un programma in un linguaggio ad alto livello in uno equivalente in linguaggio macchina per permettere la sua esecuzione da parte dell'elaboratore
- **INTERPRETE:** simile al compilatore, ma esegue la traduzione in fase d'esecuzione, istruzione per istruzione

# Evoluzione dei linguaggi ad alto livello (1/3)



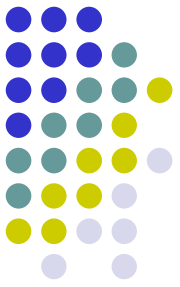
Tra i primi:

- **FORTRAN** (FORmula TRANslation): adatto ad applicazioni scientifiche e di calcolo
- **COBOL** (Common Business Oriented Language): per applicazioni gestionali
- **BASIC**: semplice, interpretato
- **LISP**: funzionale

In seguito:

- **ALGOL 60**: ispirato (come i seguenti) ai principi di programmazione ed in particolare alla programmazione strutturata
- **PASCAL**: adatto per uso didattico, perché è molto semplice e pulito
- **C**: permette di agire anche a basso livello ed è quindi adatto anche per la scrittura di moduli o parti del software di base
- ...

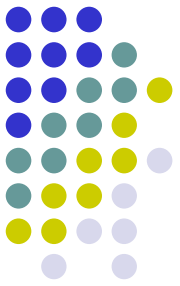
# Evoluzione dei linguaggi ad alto livello (2/3)



## Paradigma Imperativo

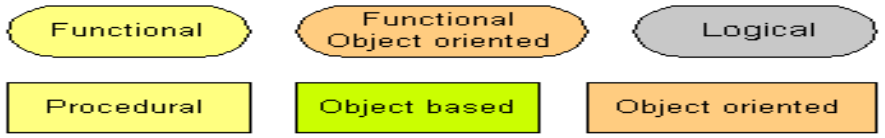
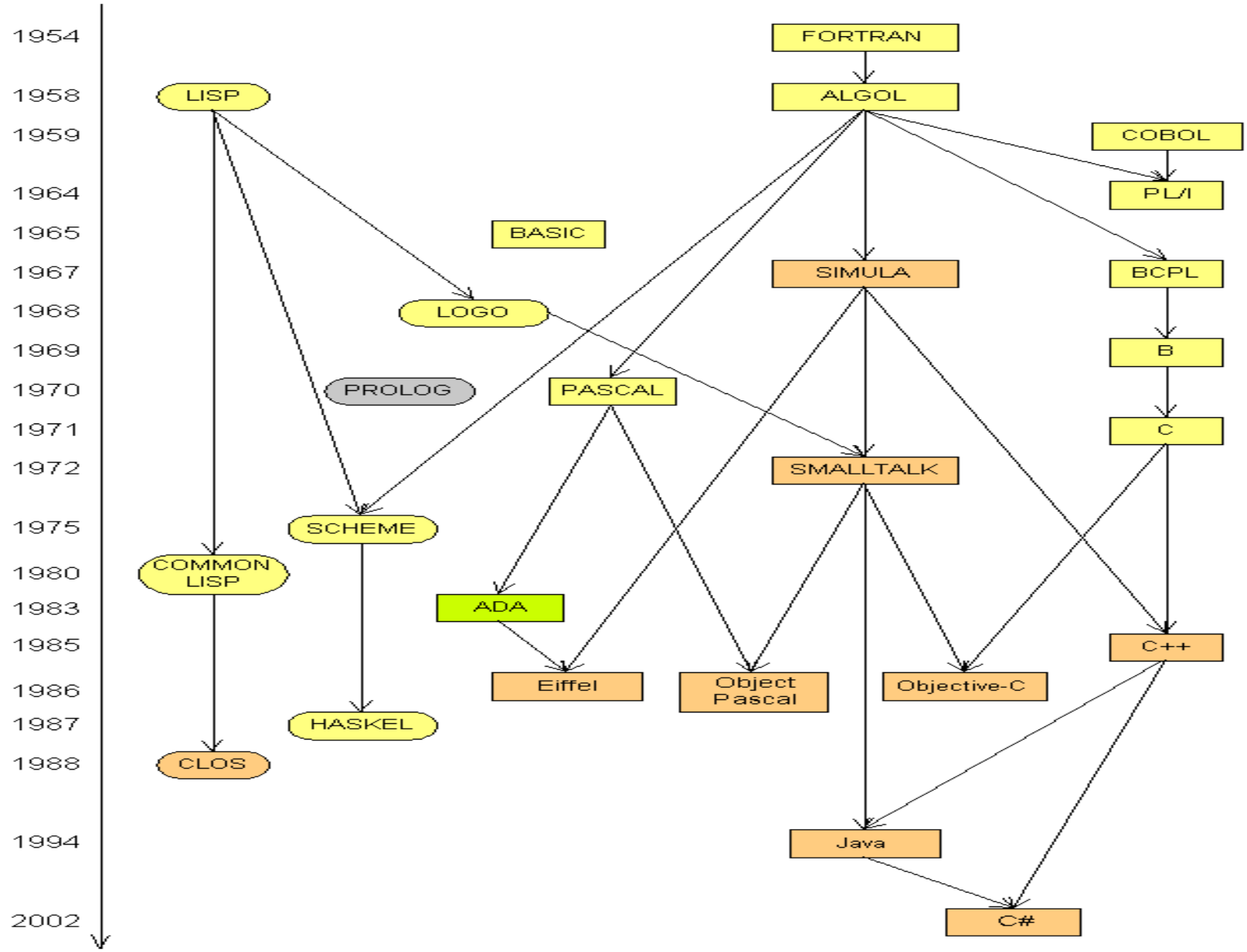
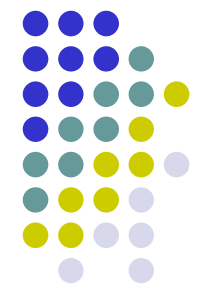
- Le istruzioni corrispondono ad ordini impartiti all'esecutore, ossia all'elaboratore
- Tutti i linguaggi ad alto livello citati precedentemente, ad eccezione del LISP, fanno parte di questo paradigma
- Per esigenze descrittive, per avvicinarsi di più al linguaggio ed ai concetti matematici ed infine per facilitare i processi di sviluppo di software sono stati proposti altri paradigmi

# Evoluzione dei linguaggi ad alto livello (3/3)



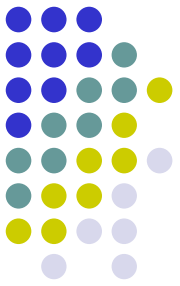
## Altri paradigmi:

- **Paradigma funzionale:**
  - si basa sul concetto di funzione matematica
  - le istruzioni corrispondono al calcolo di funzioni
  - ne fanno parte i linguaggi LISP, SCHEME, HUSKEL, ML...
- **Paradigma logico:**
  - si basa sulla logica matematica e sul concetto di relazione matematica
  - ne fanno parte i linguaggi PROLOG, ...
- **Paradigma orientato ad oggetti (OO):**
  - nasce come un'evoluzione del paradigma imperativo in cui si raggruppano i tipi di dati e le operazioni e procedure che possono essere eseguite su di essi
  - ne fanno parte i linguaggi SMALLTALK, SIMULA 67, C++ (estensione del C), JAVA (adatto alle applicazioni web), Eiffel, ...
- Un diagramma dell'evoluzione dei linguaggi di programmazione ad alto livello è il seguente (per procedural si intende imperativo):



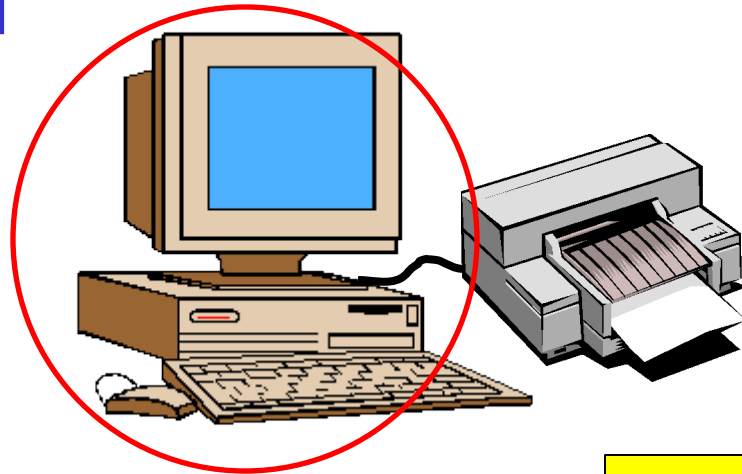
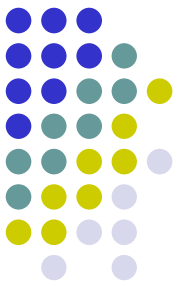


# Architettura di un sistema informatico



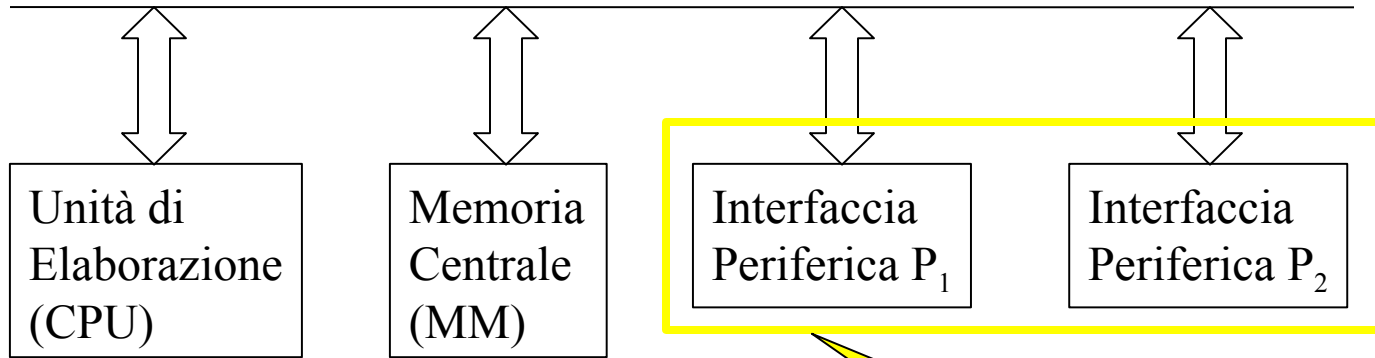
- Parti che compongono un sistema informatico e loro funzionalità
- Si suddivide in
  - **Hardware**: insieme delle componenti fisiche ed elettroniche che compongono l'elaboratore
  - **Software**: insieme dei programmi eseguiti dall'hardware
- La linea di demarcazione tra hardware e software non è netta
  - **Firmware**: strato intermedio formato dall'insieme dei programmi realizzati direttamente a livello hardware tramite particolari dispositivi (es. memorie EPROM)

# Hardware: la macchina di Von Neumann



Collegamento

Bus di sistema



Unità di Elaborazione (CPU)

Memoria Centrale (MM)

Interfaccia Periferica P<sub>1</sub>

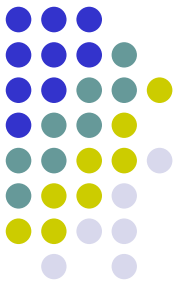
Interfaccia Periferica P<sub>2</sub>

Esecuzione istruzioni

Memoria di lavoro

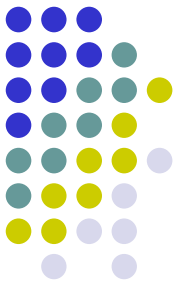
Memoria di massa, stampante, terminale...

# C.P.U.



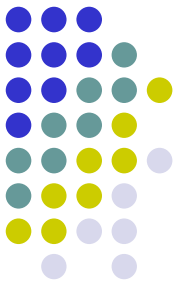
- Central Processing Unit o Unità di Elaborazione Centrale è il cervello dell'elaboratore, in quanto coordina e gestisce tutti i vari dispositivi hardware per *acquisire, interpretare ed eseguire* le istruzioni dei programmi
- Al suo interno è composta da:
  - **C.U.** (Unità di Controllo) : si occupa dell'interpretazione e della esecuzione delle istruzioni
  - **A.L.U.** (Unità Logico-Aritmetica): svolge le operazioni logiche ed aritmetiche
  - **REGISTRI**: dispositivi elettronici capaci di memorizzare sequenze di bit fungendo da piccole memorie interne alla C.P.U.
  - **CLOCK**:
    - scandisce gli intervalli di tempo in cui agiscono in modo sincrono i dispositivi interni alla C.P.U.
    - determina la velocità della C.P.U., espressa come frequenza o numero di intervalli scanditi nell'unità di tempo (es., 512MHz, 1GHz, ...)
- Alcuni esempi di CPU: Intel Centrino, AMD Turion, etc...

# Memoria Centrale



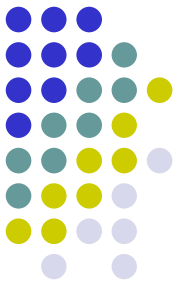
- Comunemente nota anche come RAM (Random Access Memory)
- In essa transitano le istruzioni (in linguaggio macchina) che devono essere eseguite ed i dati su cui operano
- Ha la caratteristica di essere
  - **volatile**: il suo contenuto viene perso quando viene spento l'elaboratore
  - **veloce** (ordine dei nanosecondi, ossia  $10^{-9}$  secondi)
  - **costosa**
  - di **dimensioni medio-piccole**, tipicamente centinaia di megabyte (es. 512Mb) o qualche gigabyte

# Memoria di Massa



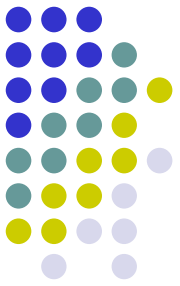
- Anche se è una componente fondamentale, non fa parte della macchina di Von Neumann in senso stretto
- È costituita dai dischi rigidi, nastri, CD e DVD ROM, ...
- Rispetto alla memoria centrale ha la caratteristica di essere
  - non volatile
  - lenta (per hard disk ordine dei millisecondi, ossia  $10^{-3}$  secondi)
  - economica
  - di grandi dimensioni (per hard disk centinaia di gigabyte)

# Unità di I/O



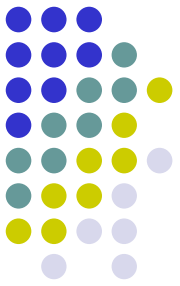
- Consentono la comunicazione dell'elaboratore con l'esterno ed in particolare la lettura di dati in input e la restituzione dei risultati delle elaborazioni in output.
- Ne fanno parte terminali (tastiera e schermo), mouse, stampanti, scanner, ...

# Bus di sistema



- Consente la comunicazione tra le varie componenti
- Si suddivide in
  - **Bus dati**: per la trasmissione di dati
  - **Bus indirizzi**: per la trasmissione di indirizzi di memoria centrale
  - **Bus di controllo**: per la trasmissione di comandi alle varie unità e di informazioni di controllo

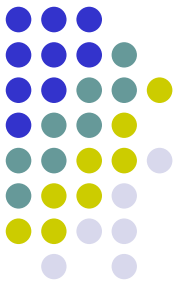
# Estensioni della macchina hardware base



- Per aumentare le prestazioni e l'affidabilità.
- Due dimensioni di crescita fondamentali:
  1. potenziamento e replicazione componenti (sistemi paralleli)
  2. collegamento in rete (sistemi distribuiti)



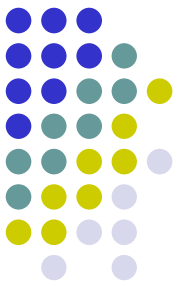
# Potenziamento e replicazione componenti (1/3)



Rispetto alla CPU:

- **Pipeline:** estensione della struttura interna della CPU in modo da eseguire in parallelo le varie fasi connesse *all'acquisizione, interpretazione ed esecuzione* delle istruzioni. Ciascuna di queste fasi è affidata ad un dispositivo specifico
- **Co-processor:** processori dedicati cui delegare particolari funzionalità della CPU da svolgere efficientemente ed in parallelo, quali
  - input/output (canali indipendenti)
  - esecuzione calcoli numerici
  - elaborazioni grafiche
- **Parallelismo:** replicazione CPU per poter eseguire più istruzioni in parallelo

# Potenziamento e replicazione componenti (2/3)



## Rispetto alla memoria centrale

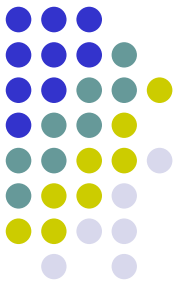
- **Gerarchie di memorie:** memorie cache multilivello, con elevata velocità di lettura e scrittura, dove vengono trasferite temporaneamente le informazioni di uso più comune o più recente da parte dell'unità centrale

## Rispetto alla memoria di massa

- **Replicazione hard disk** (tecnologia RAID-Redundant Array of Inexpensive Disks) per
  - RAID-0: i dati vengono suddivisi su più dischi (striping) per aumentare la velocità di accesso
  - RAID-1: i dati vengono scritti contemporaneamente su più dischi per aumentare l'affidabilità o la tolleranza ai guasti
  - RAID-5: per velocità e affidabilità (striping e mirroring)
  - ...

.....

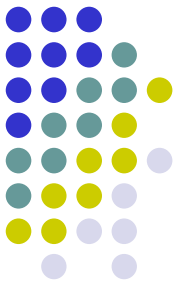
# Potenziamento e replicazione componenti (3/3)



Rispetto a questa dimensione di crescita troviamo in ordine di prestazioni

- **Personal Computer:** dedicati ad un singolo utente
- **Workstation:** con capacità di calcolo e grafiche potenziate per permettere un utilizzo professionale
- **Mini Calcolatori:** capaci di servire contemporaneamente anche decine di utenti collegati tramite terminale
- **MAIN-FRAME:**
  - Capaci di servire centinaia di utenti
  - Con molte CPU e grandi memorie di massa
  - adatti ad applicazioni gestionali
- **Super Calcolatori**
  - altamente paralleli (anche migliaia di CPU)
  - adatti per applicazioni a bassa sequenzialità, tipo numeriche e vettoriali

# Collegamento elaboratori in rete (1/2)

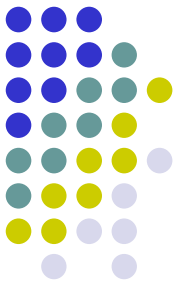


Rete: insieme di canali di interconnessione che permettono ad elaboratori diversi di comunicare tra loro

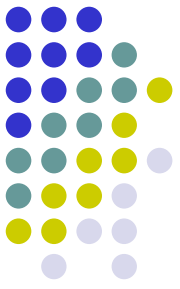
Si suddividono principalmente in due categorie:

- **locali (Local Area Network):**
  - di piccola estensione (tipicamente un edificio)
  - veloci
  - permettono agli utenti di condividere dati ed usufruire di servizi (stampanti, server, accesso a reti esterne, ...)
  - tra le più note: ethernet, ...
- **geografiche (Wide Area Network):**
  - molto estese (anche tutto il globo)
  - più lente
  - tra le più note: Internet

# Collegamento elaboratori in rete (2/2)



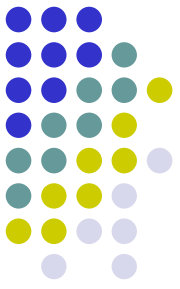
- **Sistema distribuito:** insieme di elaboratori che cooperano per lo svolgimento un determinato compito scambiando dati e risultati di elaborazioni attraverso una rete di interconnessione
- Si contrappongono ai **sistemi paralleli** in quanto elaboratori diversi eseguono contemporaneamente parti diverse di uno stesso lavoro, mentre nei sistemi paralleli uno stesso elaboratore esegue contemporaneamente istruzioni diverse sfruttando CPU diverse
- Chiaramente un singolo elaboratore di un sistema distribuito può anche essere parallelo



# Software (1/3)

- È formato dall'insieme dei programmi che vengono eseguiti sulla macchina hardware
- Si suddivide in
  - **software di base**: per permettere l'utilizzo dell'elaboratore
  - **software applicativo**: per specifiche esigenze da parte degli utenti

# Software (2/3)



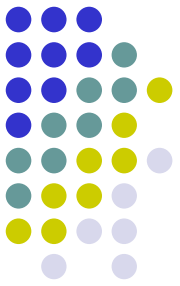
Il software di base comprende

- il sistema operativo:
  - è il cuore del software di base
  - consente un utilizzo facile ed efficiente delle risorse dell'elaboratore
- protocolli: insieme dei programmi che gestiscono la comunicazione nelle reti
- ....

Anche se generalmente si usa suddividere il software solamente in due categorie (di base ed applicativo), esiste tuttavia uno strato intermedio che ha le caratteristiche dell'una e dell'altra

# Software (3/3)

## Software applicativo



---

### Strumenti di produttività

- fogli elettronici
- basi di dati
- ...

### Software personale

- videoscrittura
- agende elettroniche
- ipertesti
- posta elettronica
- ...

### Ambiente di programmazione

- editor
- compilatore
- collegatore (linker)
- debugger

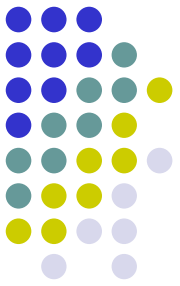
---

### Software di base

- sistema operativo
- protocolli
- ...

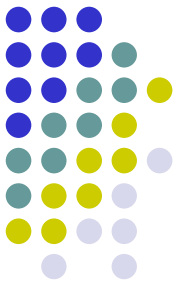


# Applicazioni informatiche

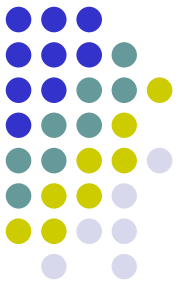


- **Numeriche:**
  - richiedono grandi quantità di calcoli
  - es.: studi statistici, traiettorie satellitari, previsioni meteorologiche, studio andamento fenomeni fisici, ...
- **Gestionali:**
  - riguardano la gestione di aziende ed attività
  - es.: gestione produzione, fornitori, clienti, dipendenti, ...
- **Telematiche:**
  - sfruttano le reti per offrire servizi e permettere transazioni a distanza
  - es.: prenotazioni, bancomat, archivi pubblici, applicazioni web, ...
- **Automazione industriale:**
  - per la gestione automatica di impianti, catene di montaggio in aziende manifatturiere, ...
  - includono applicazioni di robotica, software progettuali, ...
- **Dedicate:**
  - per contesti e funzionalità molto specifiche e delicate che richiedono soluzioni ad-hoc
  - es: controllo reattori nucleari, dispositivi ad alto rischio, strumentazione di bordo di aerei, ...
- ...

# Are discipline dell'informatica



- Algoritmi e strutture dati
- Linguaggi di programmazione
- Architetture degli elaboratori
- Sistemi operativi
- Ingegneria del software
- Basi di dati e sistemi per il reperimento dell'informazione
- Intelligenza artificiale
- Grafica computerizzata
- Interazione uomo-macchina
- Visione e robotica
- ...



# Punti Chiave

- Concetti fondamentali
  - Informatica
  - Algoritmo
  - Programma
- Linguaggi di programmazione
- Aspetti architettureali di un sistema di calcolo
- Software

*[Ref: Cap. 1, 2, 12 del libro di testo]*