

# Elementi di Base dei Linguaggi di Programmazione

Dott. Emanuele Lattanzi  
 Istituto di Scienze e Tecnologie dell'Informazione  
<http://www.sti.uniurb.it/lattanzi/>  
 Tel. 0722-4475 interno 42  
 e-mail: lattanzi@sti.uniurb.it

## Caratteristiche del corso

- |                          |        |
|--------------------------|--------|
| • Lezioni frontali       | 36 ore |
| • Esercitazioni pratiche | 33 ore |
| • Formazione a distanza  | 4 ore  |
| • Verifica apprendimento | 5 ore  |

---

**Totale** **78 ore**

## Programma

- Rappresentazione delle informazioni come configurazione di bit
- Compressione dei dati
- Algoritmi
- Linguaggi di programmazione
- Strutture dati
- Linguaggio macchina
- Programmazione orientata agli oggetti
- Linguaggio Java
- Dal programma sorgente al programma eseguibile
- Markup languages (HTML)

## Esercitazioni pratiche

- Programmazione in linguaggio *assembly*
- Programmazione procedurale (*linguaggio C*)
- Programmazione orientata agli oggetti (*linguaggio Java*)
- Programmazione sistemi wireless embedded (*Java Micro Edition*)
- Pagine Web (*linguaggio HTML*) ?

**N.B. Le esercitazioni dovranno essere strettamente individuali**

## Metodologia di valutazione

- Prova scritta sugli argomenti teorici del corso (durata 1 ora)
- Prova pratica di programmazione (durata 2 ore)

## Rappresentazione delle informazioni come configurazione di bit

Che cos'è l'informazione ?

→ **Riduzione di incertezza!**

Esempi:

- Una foto riduce l'incertezza di chi la osserva riguardo al luogo raffigurato
- Ogni parola riduce l'incertezza riguardo al significato di una frase

## Informazione oggi

- **Informatica:** disciplina che studia l'elaborazione automatica di informazioni.
- **Elaboratore:** sistema per l'elaborazione automatica delle informazioni.
- **Programmabilità:** un elaboratore e' programmabile se e' in grado di svolgere compiti diversi in base ad un programma
- **Codifica:** Ogni informazione può (a meno di un'approssimazione) essere rappresentata come una sequenza di due soli simboli (Es: 0 e 1)

## BIT (*Binary Digit*)

- Letteralmente la parola *bit* significa cifra binaria
- In generale un bit e' una unita' che puo' assumere un valore tra due possibili (normalmente si parla di 0 e 1)
- La rappresentazione fisica di un bit richiede un qualsiasi dispositivo in grado di trovarsi in uno di due possibili stati
  - Interruttore (accesso/spento)
  - Un condensatore (carico/scarico)
  - Una bandiera (alzata/abbassata)
  - Una particella magnetica (Nord/Sud)

## Codifica

- La codifica e' un'operazione che consente di utilizzare i calcolatori elettronici per elaborare dati di qualsiasi natura.
- Un bit puo' assumere solo due valori (0 e 1)
- Per rappresentare insieme costituiti da piu' di due stati/simboli si usano stringhe di bit
- Una stringa di bit e' costituita da un numero  $n$  di bit ed e' comunemente detta parola (*word*)
- Con  $n$  bit si possono formare  $2^n$  configurazioni diverse e quindi si possono rappresentare  $2^n$  informazioni diverse

## Codifica del testo

- Un testo e' una sequenza di caratteri alfabetici, separatori e caratteri speciali
  - Ad ogni carattere e' associata una diversa configurazione di bit.
  - Esempio: 21 lettere dell'alfabeto + 10 numeri + 10 punteggiatura = 41 simboli
    - $2^5 = 32$  combinazioni
    - $2^6 = 64$  combinazioni → ok
- Possiamo usare una codifica a 6 bit.

## Esempio codifica testo a 6 bit

000000 = a	Un testo e' rappresentato dalla sequenza di byte associati ai caratteri che lo compongono, nell'ordine in cui essi compaiono.
000001 = b	
000010 = c	
000011 = d	
000100 = e	
000101 = f	
000110 = g	
000111 = h	

## Standard ASCII

(*American Standard Code for Information Interchange*)

- La codifica ASCII prevede l'utilizzo di 128 caratteri diversi
- Ogni carattere e' associato ad una diversa configurazione di 7 bit
- La codifica ASCII estesa prevede 256 simboli e 8 bit (*1byte*) per ogni carattere
- Quindi un testo di 1000 caratteri richiede 1Kbyte per essere rappresentato

## Esempi di codifica ASCII estesa

- Il testo "ciao" richiede 4 byte
- "ciao a tutti" richiede 12 byte
- "perche' no" richiede 10 byte
- "perché no" richiede 9 byte

## Codifica(2)

- Una codifica esatta a  $n$  bit è possibile solo quando l'insieme delle informazioni da codificare è finito e di dimensione inferiore a  $2^n$  (*cardinalità*).
- I calcolatori sono oggetti finiti che elaborano e memorizzano un numero finito di bit
- Se l'insieme da codificare ha una cardinalità maggiore di  $2^n$  se ne può dare solo una rappresentazione approssimata o parziale
  - Operazioni di limitazione
  - Operazioni di partizionamento

## Codifica dei numeri

- *Notazione posizionale*: simboli che assumono valori differenti secondo la posizione occupata dal *grafema*.
- *Codifica binaria*: fornisce una diretta trasposizione in un alfabeto a due valori della notazione decimale comunemente utilizzata
  - $c_n c_{n-1} \dots c_1 c_0 . c_{-1} c_{-2} \dots c_{-m}$
- Il valore ( $v$ ) del numero rappresentato è la somma pesata delle cifre, dove il peso di ogni cifra dipende dalla base di numerazione ( $B$ ) e dalla posizione della cifra:
  - $v = c_n B^n + c_{n-1} B^{n-1} + \dots + c_1 B + c_0 + c_{-1} B^{-1} + c_{-2} B^{-2} + \dots + c_{-m} B^{-m}$

## Numeri interi

- I numeri interi sono un insieme discreto illimitato.
- Per poter essere codificati devono essere limitati.
- Sottoinsieme simmetrico rispetto allo 0.
- Si usa  $1$  bit per rappresentare il segno e i restanti  $a = n-1$  per rappresentare il modulo.
- Il massimo numero rappresentabile è (in modulo)  $2^{a-1}$ .
- Se il risultato di un'operazione eccede il modulo  $2^a-1$  non può essere codificato e il calcolatore restituisce un messaggio di overflow.

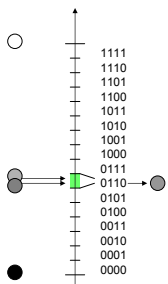
## Numeri reali

- I numeri reali sono un insieme continuo e illimitato. Per poterli rappresentare occorre limitarli (in modo simmetrico rispetto allo 0) e partizionarli.
- *Rappresentazione in virgola fissa*: Degli  $n$  bit della parola,  $1$  rappresenta il segno,  $a$  rappresentano le cifre prima della virgola e  $b$  le cifre dopo la virgola.
  - Il massimo numero rappresentabile è  $(2^{n-1}-1)/2^b$
  - L'accuratezza assoluta è  $2^b$
- *Rappresentazione in virgola mobile*: (Floating point) espressa nella forma  $\rightarrow s.O.M B^{sE}$

## Rappresentazione delle immagini

- Le immagini sono informazioni continue in tre dimensioni: due spaziali ed una colorimetrica.
- Per codificarle occorre operare tre discretizzazioni.
  - Due discretizzazioni spaziali riducono l'immagine ad una matrice di punti colorati, detti pixel.
  - La terza discretizzazione limita l'insieme di colori che ogni pixel può assumere.

### Esempio: Livelli di grigio



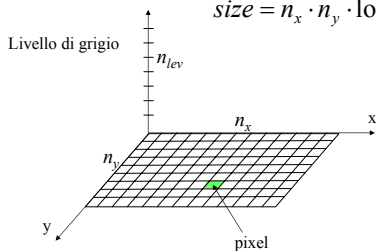
La codifica associa un unico codice ad un intervallo di livelli di grigio

Tutti i livelli di grigio all'interno dell'intervallo vengono codificati allo stesso modo comportando una perdita di informazione

Il livello di grigio originale non può essere ricostruito in maniera esatta dal codice binario

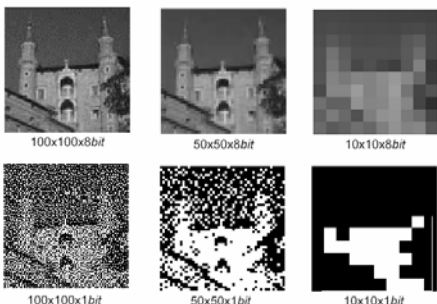
### Esempio: Dimensione

$$size = n_x \cdot n_y \cdot \log_2 n_{lev}$$



Un'immagine di 100x100 pixel a 256 colori richiede 10000 byte (10 Kb) per essere rappresentata.

### Esempio immagini bitmap



### Immagini vettoriali

- La grafica vettoriale scompone in gruppi logici di componenti (linee, cerchi, rettangoli, ecc. )
- Le forme vengono memorizzate in termini di coordinate e colori dei vari elementi geometrici che le compongono
- Durante la visualizzazione, coordinate e colori vengono utilizzati per ricreare l'immagine
- La grafica vettoriale e' comunemente usata nei disegni, disegni animati e nella grafica lineare in generale

### Esempio: Oggetti lineari

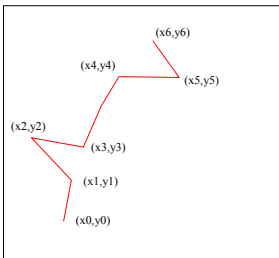
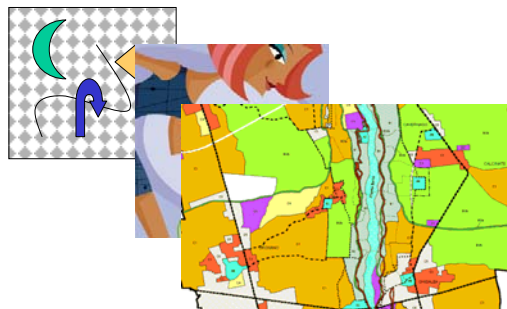


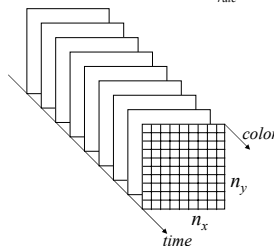
Immagine = x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6,rosso

### Esempio immagini vettoriali



### Codifica di un filmato video

$$size = s_{rate} \cdot T \cdot s_{size} = s_{rate} \cdot T \cdot \log_2 n_{col} \cdot n_x \cdot n_y$$



- $s_{rate}$  = frame rate
- $n_{col}$  = numero di colori
- $n_x n_y$  = dimensione del frame

Filmato di 10 minuti a 25 frame al secondo, con risoluzione di 100x100 pixel a 256 colori: dimensione complessiva di 600x25x100x100x8 = 1.2Gbit.

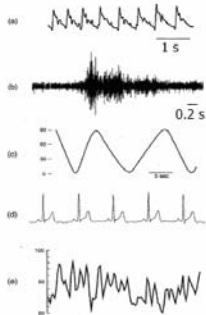
### Codifica di segnali analogici

- **Segnale:** Quantita' fisica che varia nel tempo
  - *Analogico:* tempo-continuo, valore-continuo
  - *Digitale:* tempo-discreto, valore-discreto
- La codifica digitale di un segnale continuo comporta:
  - *Campionamento*(discretizzazione nel tempo)
  - *Quantizzazione* (discretizzazione nel valore)

$$size = s_{rate} \cdot T \cdot s_{size}$$

frequenza di campionamento      Durata      Dimensione del campione

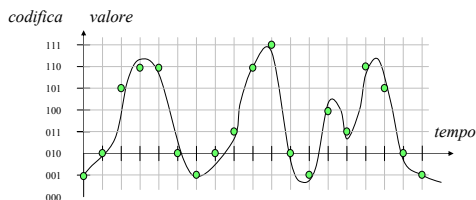
### Esempi di segnali analogici



- a) Velocità del flusso sanguigno nell'arteria cerebrale di un soggetto umano
- b) EMG (contrazione e rilassamento della lingua)
- c) Angolo di rotazione del ginocchio
- d) ECG
- e) Frequenza cardiaca istantanea in battiti al minuto (100 battiti)

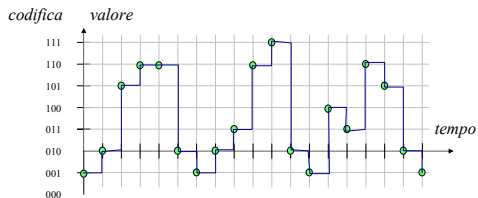
35

### Campionamento e quantizzazione



File finale = 001-010-101-110-110-010-001-010-011-110-111-010-001-100-011-110-101-010-001

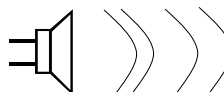
### Campionamento e quantizzazione



File finale = 001-010-101-110-110-010-001-010-011-110-111-010-001-100-011-110-101-010-001

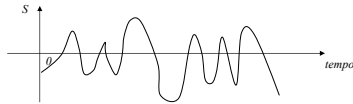
### Codifica del suono

- Il suono è un segnale analogico tempo-continuo
- Un'onda sonora non è altro che una successione di rarefazioni e compressioni di piccole porzioni d'aria



## Una descrizione del suono

- Il suono prodotto da un altoparlante e' prodotto dalla vibrazione di una membrana.
- Descrivendo la posizione della membrana nel tempo (e quindi il suo spostamento) a tutti gli effetti descriviamo il suono.



Il suono percepibile dall'orecchio umano viene riprodotto fedelmente se la frequenza di campionamento (il numero di campioni in un secondo) è non inferiore a 30KHz.

Lo standard telefonico prevede un campionamento a 8KHz ed una quantizzazione a 256 livelli (codificati con 8 bit). Quindi per ogni secondo di conversazione servono 64Kbit

Questo tipo di codifica e' comunemente utilizzato nel formato WAVE

## Formato MIDI (Musical Instrument Digital Interface)

- Il formato MIDI ha, tra i formati audio, lo stesso ruolo che, tra i formati grafici, ha il formato vettoriale.
- Un file MIDI non è una registrazione sonora
- Contiene comandi che, quando eseguiti, producono dei suoni (mediante un sintetizzatore hardware o software).
- I file MIDI, per questo motivo, sono molto piccoli rispetto ai file audio, ma riproducono un suono "sintetico", non registrato.
- Un file MIDI ha una forma del tipo:
  - Suona la nota  $x$  per un tempo  $t$  con lo strumento  $y$