Elementi di Informatica

Prof. Alberto Broggi
Dip. di Ingegneria dell'Informazione
Università di Parma

Organizzazione del corso

- Teoria:
  - architettura del calcolatore,
  - elementi di informatica,
  - algoritmi,
  - linguaggi,
  - sistemi operativi
- Laboratorio:
  - linguaggio Fortran,
  - Applicazioni di *office automation* (Word, Excel)

Crediti

- Corso formato da 6 crediti (52 ore)
  - lezioni (36 ore)
  - esercitazioni (8 ore)
  - laboratorio (8 ore)

Orario lezioni ed esami

- Lezioni:
  - Venerdì 11:00 - 13:00
  - Venerdì 14:00 - 16:00)
- Esami:
  - in corso di definizione

Ricevimento studenti

- Dopo le lezioni:
  - Venerdì, ore 16:00 - 17:00

Dispense e lucidi

- I lucidi presentati a lezione sono disponibili in Internet all'indirizzo:

  http://www.ce.unipr.it/broggi/elinfo

- Dispense: A.Broggi, "Sintesi dei principali argomenti di Fondamenti di Informatica", Ed. Spiegel, ISBN 88-7660-147-3

Modalità di esame

- Due prove scritte
  - una a metà corso
  - una al termine
- Il voto finale è calcolato come media delle due prove (entrambe devono essere sufficienti)
- È possibile *modificarlo* con un esame orale

Presenze

- E' necessario almeno il 70% delle presenze a lezione
- Foglio delle presenze (firme)

DATA ORA AULA Docente: argomenti [file scaricabile da internet]

05/10/2001 11-13	A1	AB:	Introduzione [file 1]
05/10/2001 14-16	A1	AB:	Sistemi di numerazione [file 1]
12/10/2001 11-13	A1	AB:	Codifiche [file 1]
12/10/2001 14-16	A1	AB:	Aritmetica binaria [file 2]
19/10/2001 11-13	A1	DV:	Word e Excel (in aula)
19/10/2001 14-16	C1	DV:	Word/Excel (in laboratorio, gruppo 1)
26/10/2001 11-13	A1	AB:	Unita' di I/O, controllo [file 3]
26/10/2001 14-16	A1	AB:	Mem, CPU [file 3]
02/11/2001 11-13	A1	AB:	periferici IO grafico [file 3]
02/11/2001 14-16	A1	AB:	algoritmi/flowchart [file 6]
09/11/2001 11-13	A1	AB:	macro, procedure [file 7]
09/11/2001 14-16	A1	AB:	A disposizione per chiarimenti
16/11/2001 11-13	C1	DV:	Word/Excel (in laboratorio, gruppo 2)
16/11/2001 14-16	C1	DV:	Word/Excel (in laboratorio, gruppo 3)
23/11/2001 11-13	A1		Sospensione lezioni
23/11/2001 14-16	A1		Sospensione lezioni
30/11/2001 11-13	A1		Sospensione lezioni
30/11/2001 14-16	A1	AB:	compito (I parte)
07/12/2001 11-13	A1/C3	DV:	Fortran
07/12/2001 14-16	A1/C3	DV:	Fortran
14/12/2001 11-13	A1/C3	DV:	Fortran
14/12/2001 14-16	A1/C3	DV:	Fortran
21/12/2001 11-13	C3		Fortran (esercitazioni in laboratorio)
21/12/2001 14-16	C3		Fortran (esercitazioni in laboratorio)
11/01/2002 11-13	A1	AB:	algoritmi di ordinamento (1) [file 7]
11/01/2002 14-16	A1	AB:	algoritmi di ordinamento (2) [file 7]
18/01/2002 11-13	A1	AB:	linguaggi (formalismo, grammatiche) [file 9]
18/01/2002 14-16	A1	AB:	compilatori/interpreti [file A]
25/01/2002 11-13	A1	AB:	sistemi operativi [file A]
25/01/2002 14-16	A1	AB:	gestione memoria [file B]
01/02/2002 11-13	A1	AB:	compito (II parte)

Testi consigliati

- P.Demichelis, E.Piccolo "Introduzione all'Informatica in C", McGraw-Hill
- C.Batini, L.C.Aiello, M.Lenzerini, A.Marchetti Spaccamela, A.Miola "Fondamenti di Programmazione dei Calcolatori Elettronici", Franco Angeli
- Paolo Tosoratti, "Introduzione all'Informatica", seconda edizione, Casa Editrice Ambrosiana
- Peter Bishop, "L'Informatica", Jackson
- Manuali di Fortran (o dispense)

Rappresentazione dell'informazione

- Problema che coinvolge aspetti filosofici
- Interessa soprattutto *distinguere* informazioni *diverse*
- Con un solo simbolo è impossibile
- Pertanto l'insieme minimo è costituito da **2 simboli** (alfabeto binario)

Rappresentazione dell'informazione

- Le informazioni vengono rappresentate mediante sequenze di simboli
- Nel caso dei simboli binari, le informazioni (numeri, oggetti, parole) sono rappresentate da sequenze dei due simboli
- Servono regole di manipolazione dei simboli

Sistemi numerici

- Per determinare un sistema numerico serve:
  - un insieme limitato di simboli (le *cifre*), che rappresentano quantità prestabilite (1, 2, V, X, M)
  - le regole per costruire i numeri:
    - sistemi numerici posizionali
    - sistemi numerici non posizionali

Sistemi numerici

- Sistemi numerici non posizionali:
  - valore delle cifre è indipendente dalla posizione
- Sistemi numerici posizionali:
  - il valore delle cifre dipende dalla loro posizione all'interno del numero (ogni posizione ha un peso)

Sistemi numerici posizionali

- Esempio:

$N = d_1\, d_2\, d_3\, d_4$; $V(N) = d_1*p_1 + d_2*p_2 + d_3*p_3 + d_4*p_4$

- Sistemi a base fissa:
  - $p_i = r^i$ dove:
    - r è la base del sistema
    - d_i rappresentano le cifre

Sistema decimale

- Il sistema decimale utilizza:
  - r = 10
  - d = 0,1,2,3,4,5,6,7,8,9
- È importante notare che qualsiasi sistema posizionale a base fissa è **irridondante**

Sistema binario

- Il sistema binario utilizza:
  - r = 2
  - d = 0,1
- Ogni cifra è detta **bit** (da **BI**nary digi**T**)

Altri sistemi utilizzati

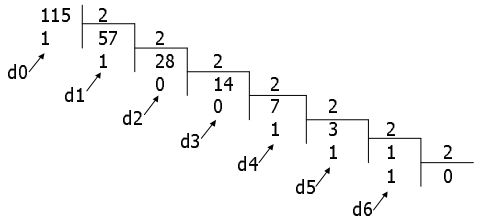
- Sistema ottale:
  - r = 8
  - d = 0,1,2,3,4,5,6,7
- Sistema esadecimale:
  - r = 16
  - d = 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Conversioni di base

- Utilizzando la definizione:
  - $1010_2 = (1*8 + 0*4 + 1*2 + 0*1)_{10} =$
    $= (8+2)_{10} = 10_{10}$
- Oppure si può utilizzare il seguente formato:
  - $N = ((d_{n-1}*r + d_{n-2})*r + d_{n-3}) \ldots)*r + d_0$

Conversioni di base

- Esempio: $115_{10} = 1110011_2$:



Numeri frazionari

- E' possibile anche rappresentare numeri frazionari:
  - le potenze variano anche nel campo negativo

$$N = \sum_i a_i \, b^i$$

Numero di cifre necessario

- Le macchine hanno vincoli spaziali:
  - è necessario conoscere il massimo valore rappresentabile:
  - con *n* bit si può rappresentare al massimo il numero 2^n-1
  - è facile determinare che

    $n = INT(\log_2 (X+1))$

Operazioni artimetiche

- Per effettuare operazioni è necessario conoscere la definizione del comportamento per ogni coppia di simboli
- Per ogni operazione esiste una tabella

Somma binaria

- La tabella di definizione è:
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 0 con riporto di 1
  - 1 + 1 + 1 = 1 con riporto di 1
- Esempi

Sottrazione binaria

- La tabella di definizione è:
  - 0 - 0 = 0
  - 1 - 0 = 1
  - 1 - 1 = 0
  - 0 - 1 = 1 con prestito di 1 dal bit di peso superiore
- Esempi

Moltiplicazione e divisione

- Si utilizzano le stesse procedure:
  - per la moltiplicazione: somma e scorrimento
  - per la divisione: differenza e scorrimento
- Esempi

Overflow e Underflow

- Sono condizioni in cui si ha un errore nella rappresentazione del risultato
- Generalmente la rappresentazione è formata da un numero finito di bit: se si supera tale limite si ha errore

Rappresentazione dei numeri nei calcolatori

- Esiste un limite al numero di bit impiegati per rappresentare un numero
- Tale limite dipende da:
  - intervallo di variabilità
  - occupazione di memoria

Numeri positivi

- La rappresentazione di numeri positivi non crea problemi
- Si può avere *overflow* se il risultato delle operazioni richiede un numero maggiore di bit di quanto disponibile
- Esempio: somma modulo 16

Numeri negativi

- Esistono diverse possibilità:
  - modulo e segno:
    - bit più significativo: positivo (0) e negativo (1)
    - esistono due rappresentazioni per lo '0'
  - complemento a 2:
    - per definizione il complemento a 2 di X è 2^n -X
    - unica rappresentazione dello '0'
    - Esempio: -1 <=> 11111111

Uso dei numeri negativi

- Modulo e segno:
  - la somma algebrica di numeri positivi e negativi può generare problemi
  - servono sistemi hardware specifici per la gestione corretta del formato
- Complemento a due:
  - la somma algebrica non genera problemi

Complemento a 2

- Motivazione:
  - Sia dato un numero di bit *n*
  - i numeri che si possono rappresentare sono nel range [0 - 2^n-1]
  - si vuole calcolare A-B
  - si sostituisce -B con (2^n-B)
  - si ottiene A+(2^n-B)
  - La sottrazione si esegue mediante una somma!

Rappresentazione numeri reali

- I numeri reali sono nel range $[-\infty \div +\infty]$
- Talvolta è necessaria una rappresentazione estesa sulla retta dei reali
  - con 3 simboli [+/-], X, Y, $Z \in \{0,1,...9\}$ è possibile rappresentare -999 ÷ +999
  - oppure $9 * 10^{[+/-]\,99}$
  - oppure $[+/-]\,9 * 10^{[+/-]\,99}$

Virgola mobile

- E' la risposta alla necessitá di manipolare numeri di ordini di grandezza diversi
- Numeri espressi nella forma:
  $X.YYY * 10^{WW}$
  - X: parte intera
  - Y: parte frazionaria
  - W: esponente

Virgola mobile

- Nomenclatura:
  $A = M * B^E$
  - M: mantissa
  - B: base
  - E: esponente
- Necessita di un segno per la mantissa e uno per l'esponente

Virgola mobile

- Forma normalizzata:
  - si sceglie di avere la seguente relazione:
    $0 \leq M < 1$
  - l'esponente è espresso in complemento a B (talvolta in *eccesso 127*)
  - la mantissa è espressa in modulo e segno

Virgola mobile

- Esempi usando: B=10, 2 cifre all'esponente e 8 alla mantissa:
  - +1 0 01 10000000
  - -63517,8 1 05 63517800
  - -0,00063517,8 1 97 63517800
  - $-8{,}75 * 10^{-13}$ 1 88 87500000

Virgola mobile

- Moltiplicazione e divisione:
  - si moltiplica o si dividono le mantisse in modo consueto
  - si sommano o si sottraggono gli esponenti
  - si normalizza
  - Esempio: 10,4 * 200 =
    0 02 10400000 * 0 03 20000000 =
    0 05 02080000 = 0 04 20800000

Virgola mobile

- Somma e sottrazione:
  - si uguagliano gli esponenti
  - le mantisse vengono sommate
  - aggiustamento in caso di traboccamento
  - Esempio: 10,4 + 2 =
    0 02 10400000 + 0 01 20000000 =
    0 02 10400000 + 0 02 02000000 =
    0 02 12400000 = 12,4

Virgola mobile

- Approssimazioni:
  - 34,56 + 0,005 =
    0 02 3456 + 0 98 5000 =
    0 02 3456 + 0 02 0000 =
    0 02 3456 = 34,56
  - La precisione è data dal numero di cifre della mantissa:
    - Doppia precisione: doppia lunghezza della mantissa (range invariato, precisione raddoppiata)