

Università degli Studi di Milano
Laurea Specialistica in Genomica Funzionale e Bioinformatica
Corso di Linguaggi di Programmazione per la Bioinformatica

Introduzione ai linguaggi di programmazione

Giorgio Valentini

e –mail: valentini@dsi.unimi.it

DSI – Dipartimento di Scienze dell' Informazione
Università degli Studi di Milano

1

Caratteristiche dei linguaggi di programmazione

- Sono analoghi ai linguaggi naturali, con la differenza che vengono usati per comunicare con una macchina

Come i linguaggi naturali sono caratterizzati dalle seguenti componenti:

- Insieme di simboli (*alfabeto*) e di parole (*dizionario*) che possono essere usati per formare le frasi del linguaggio
- Insieme delle regole grammaticali (*sintassi*) per definire le frasi corrette composte dalle parole del linguaggio
- Significato (*semantica*) delle frasi del linguaggio
- Per utilizzare correttamente un linguaggio è necessario conoscerne la *pragmatica* (ad es: quali frasi è opportuno usare a seconda del contesto).

• I linguaggi di programmazione, a differenza di linguaggi naturali, non devono essere ambigui e devono essere formalizzati (definiti in maniera non equivocabile).

2

Linguaggi di programmazione:
strumenti per comunicare ad una macchina come
risolvere un problema.

- Strumenti per la comunicazione uomo-
macchina
- Permettono di esprimere e rappresentare
programmi = algoritmi + strutture dati
comprensibili ed eseguibili da una macchina

3

Linguaggi macchina

- Linguaggi immediatamente comprensibili
per una macchina:
 - Istruzioni e dati sono sequenze di numeri binari
 - Le istruzioni operano direttamente sull'
hardware (registri, locazioni di memoria, unità
fisiche di I/O del calcolatore)
 - Sono specifici per un determinato processore o
famiglia di processori
 - Assumono il modello computazionale di Von
Neumann

4

Esempio:

Calcolo della somma S di due numeri A e B

Linguaggio macchina

00000010101111001010

00000010111111001000

00000011001110101000

Linguaggio assembly

LOAD A

ADD B

STORE S

Un *linguaggio assembly* è la forma simbolica di un linguaggio macchina: si usano nomi al posto dei codici binari per le operazioni e locazioni di memoria delle macchine.

5

Linguaggi a basso ed alto livello

- Linguaggi assembly e macchina sono *linguaggi a basso livello*
- *I linguaggi ad alto livello* permettono di scrivere programmi con un linguaggio piu' vicino a quello naturale

Esempio:

“Stampa sullo schermo la somma fra C ed il prodotto di A e B”:

Linguaggio ad alto livello (C++):

```
cout << A * B + C;
```

Linguaggio Assembly:

```
mov eax,A  
mul B  
add eax,C  
call WriteInt
```

Linguaggio macchina:

```
A1 00000000  
F7 25 00000004  
03 05 00000008  
E8 00500000
```

6

Esempio: funzione per il calcolo della media in C ed in linguaggio assembly

Linguaggio C (alto livello)

```
double mean (double* x, unsigned n)
{
    double m = 0;
    int i;
    for (i=0; i<n; i++)
        m += x[i];
    m /= n;
    return m;
}
```

Linguaggio Assembly

```
.file "qq.c"
.text
.globl mean
.type mean,@function

mean:
    pushl   %ebp
    movl   %esp, %ebp
    subl   $24, %esp
    movl   $0, -8(%ebp)
    movl   $0, -4(%ebp)
    movl   $0, -12(%ebp)

.L2:
    movl   -12(%ebp), %eax
    cmpl   12(%ebp), %eax
    jb     .L5
    jmp    .L3

.L5:
    movl   -12(%ebp), %eax
    leal   0(,%eax,8), %edx
    movl   8(%ebp), %eax
    fldl   -8(%ebp)
    faddl  (%eax,%edx)
    fstpl  -8(%ebp)
    leal   -12(%ebp), %eax
    incl   (%eax)
    jmp    .L2

.L3:
    movl   12(%ebp), %eax
    movl   $0, %edx
    pushl  %edx
    pushl  %eax
    fldll  (%esp)
    leal   8(%esp), %esp
    fldl   -8(%ebp)
    fdivp  %st, %st(1)
    fstpl  -8(%ebp)
    movl   -8(%ebp), %eax
    movl   -4(%ebp), %edx
    movl   %eax, -24(%ebp)
    movl   %edx, -20(%ebp)
    fldl   -24(%ebp)
    leave
    ret

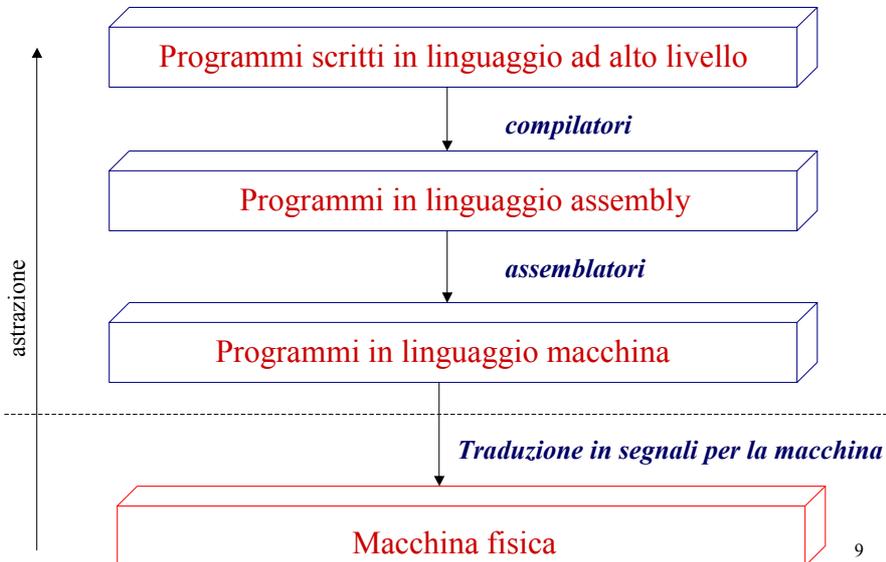
.Lfe1:
.size mean,.Lfe1-mean
.ident "GCC: (GNU) 3.2.2"
7
```

Linguaggi ad alto livello

- Sono definiti *astrando* rispetto alla macchina fisica
- Realizzano una *macchina virtuale* soprastante alla macchina fisica e visibile al programmatore
- Richiedono di essere implementati su un particolare sistema di calcolo tramite strumenti opportuni (*compilatori* o *interpreti*)

Esempi: *fortran*, *C*, *lisp*, *java*, *R*

Livelli di rappresentazione e macchine astratte



Linguaggi compilati e linguaggi interpretati

- I *programmi compilati* vengono tradotti completamente dalla prima all'ultima istruzione nel linguaggio macchina del sistema sottostante (netta distinzione fra compile-time e run-time).
Es: programmi in *C*, *fortran*, *C++*
- I *programmi interpretati* vengono tradotti ed eseguiti immediatamente riga per riga (l'interprete simula una macchina astratta, no distinzione netta fra compile-time e run-time).
Es: programmi in *R*
- Esistono casi "intermedi": es: *java*.
- I *compilatori* e gli *interpreti* sono i programmi che effettuano la traduzione dal linguaggio ad alto livello al linguaggio macchina

10

Paradigmi di programmazione

Paradigma	Modello computaz.	In cosa consiste un programma	Esempi
Imperativo	Stato=astrazione della memoria	Eeguire comandi	Pascal, C
Funzionale	Funzioni (definizione ed applicazione)	Valutare espressioni	Lisp
Ad oggetti	Universo di oggetti	Mandare messaggi ad oggetti	Java, C++
Logico	Predicati e deduzione logica	Soddisfare un goal	Prolog

11

Linguaggi di programmazione e produzione del software

Modello tradizionale “a cascata” per la produzione del sw:

1. Analisi e specificazione dei requisiti
2. Progetto (design) del sistema
3. **Implementazione: Produzione del codice nel linguaggio prescelto**
4. Verifica e validazione
5. Manutenzione

In realtà il processo di produzione è ciclico.

12

Linguaggi di programmazione e ambienti di sviluppo

Ogni fase dello sviluppo del sw può essere supportato da *ambienti di sviluppo*.

Ambienti di sviluppo per l'implementazione del sw:

- Text editor
- Compilatori
- Linker
- Librerie
- Debugger
- ...

13

Linguaggi di programmazione per la bioinformatica

- In linea di principio qualsiasi linguaggio ad alto livello può essere utilizzato.
- Esistono comunque *linguaggi con librerie e package specifici* specializzati per la bioinformatica:

Progetti
Open
Source

- Perl e *BioPerl*: <http://bioperl.org>
- Python e *Biopython*: <http://biopython.org>
- Java e *BioJava*: <http://biojava.org>
- R e *Bioconductor*: <http://www.bioconductor.org>
- Matlab e toolbox per la bioinformatica

14