

Assemblatori e compilatori

Stefano Lodi

Insegnamento di Informatica

Alma Mater Studiorum Università di Bologna

`stefano.lodi@unibo.it`

Linguaggio assoluto

Linguaggio assoluto

Struttura delle istruzioni assolute

Tabella dei simboli
Programma in linguaggio assoluto

Le funzioni dell'assemblatore
Assemblatori e compilatori

Livelli di astrazione
Linguaggi ad alto livello

- Il programma per il prodotto di interi espresso in linguaggio assembly non potrebbe essere eseguito da una CPU reale, che esegue solo programmi in linguaggio assoluto, pertanto scritti come sequenze di numeri binari
- Il programma dovrebbe quindi essere tradotto, seguendo la corrispondenza tra istruzioni in linguaggio assembly e linguaggio assoluto, stabilita al momento della definizione del linguaggio assembly, durante il progetto della CPU
- Per il nostro linguaggio supporremo la seguente corrispondenza

LOAD in R1	0000 nnnn
LOAD in R2	0001 nnnn
STORE R1 in	0010 nnnn
STORE R2 in	0011 nnnn
ADD	0100
SUB	0101
JUMP to	0111 nnnn
READ	1000 nnnn
WRITE	1001 nnnn
STOP	0110

Struttura delle istruzioni assolute

Linguaggio assoluto
Struttura delle
istruzioni assolute
Tabella dei simboli
Programma in
linguaggio assoluto
Le funzioni
dell'assemblatore
Assemblatori e
compilatori
Livelli di astrazione
Linguaggi ad alto
livello

- Le istruzioni sono composte da due byte
 - ◆ Il primo byte distingue le istruzioni in base al tipo di funzione eseguita ed è detto *codice operativo* (opcode)
 - ◆ Il secondo byte, se significativo, è interpretato dalla CPU come il numero binario associato a una cella di memoria, o *indirizzo* della cella
- Tutte le istruzioni assemblative che richiedono un identificatore di cella di memoria (LOAD, STORE, JUMP, READ, WRITE) corrispondono, in linguaggio assoluto, ad istruzioni in cui il secondo byte è significativo
- Si noti che la CPU da noi ipotizzata è un puro esecutore, e non effettua alcun controllo sugli indirizzi specificati nel secondo byte; in particolare, non controlla che l'indirizzo di una istruzione STORE o WRITE sia una delle celle utilizzate per memorizzare il programma

Tabella dei simboli

Linguaggio assoluto
Struttura delle
istruzioni assolute
Tabella dei simboli
Programma in
linguaggio assoluto
Le funzioni
dell'assemblatore
Assemblatori e
compilatori
Livelli di astrazione
Linguaggi ad alto
livello

- Per tradurre correttamente un programma assembler nel corrispondente programma in linguaggio assoluto, occorre stabilire una corrispondenza che ad ogni simbolo associi uno e un solo indirizzo
 - ◆ Tutte le istruzioni in cui è indicato un simbolo A sono tradotte in istruzioni assolute in cui il secondo byte è l'indirizzo associato ad A
- Si costituisce una *tabella dei simboli*

Nome simbolico operando	indirizzo
PROD	00110
W	10010
Y	10011
ZERO	10100
Z	10101
U	10110
UNO	10111

Programma in linguaggio assoluto

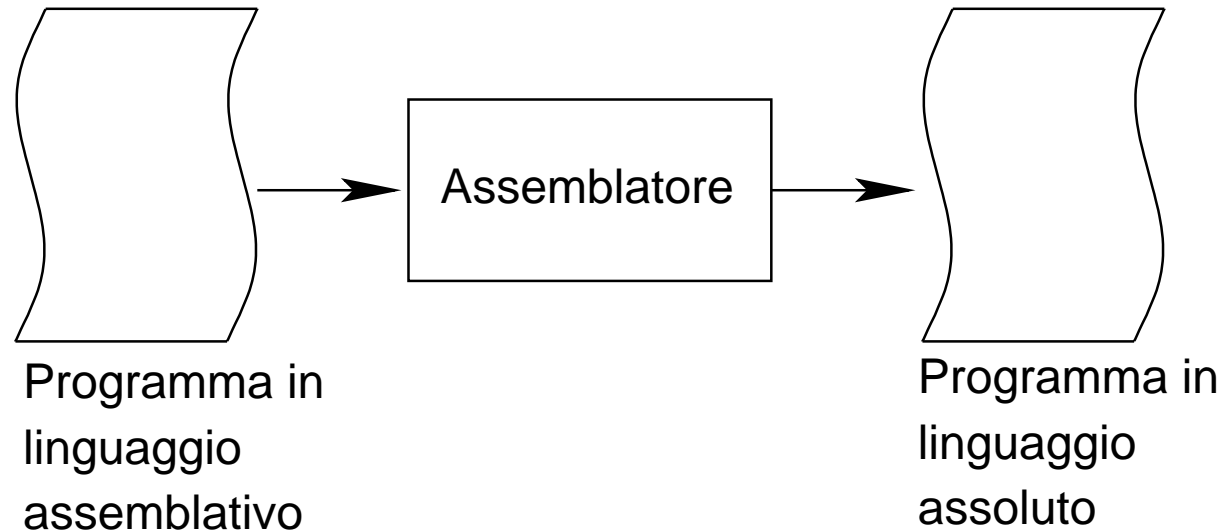
Linguaggio assoluto
Struttura delle
istruzioni assolute
Tabella dei simboli
Programma in
linguaggio assoluto
Le funzioni
dell'assemblatore
Assemblatori e
compilatori
Livelli di astrazione
Linguaggi ad alto
livello

Ind. cella	cod. op.	ind.	Ind. cella	cod. op.	ind.
00000	1000	10010	10000	1001	10101
00001	1000	10011	10001	0110	
00010	0000	10100	10010		
00011	0010	10101	10011		
00100	0000	10011	10100		
00101	0010	10110	10101		
00110	0000	10101	10110		
00111	0001	10010	10111		
01000	0100		11000		
01001	0010	10101	11001		
01010	0000	10110	11010		
01011	0001	10111	11011		
01100	0101		11100		
01101	0010	10110	11101		
01110	0000	10110	11110		
01111	0111	00110	11111		

Le funzioni dell'assemblatore

Linguaggio assoluto
Struttura delle istruzioni assolute
Tabella dei simboli
Programma in linguaggio assoluto
Le funzioni dell'assemblatore
Assemblatori e compilatori
Livelli di astrazione
Linguaggi ad alto livello

- L'assemblatore per una CPU è un programma che assume in ingresso un programma scritto in linguaggio assembly per la CPU e produce l'equivalente programma in linguaggio assoluto eseguibile dalla CPU

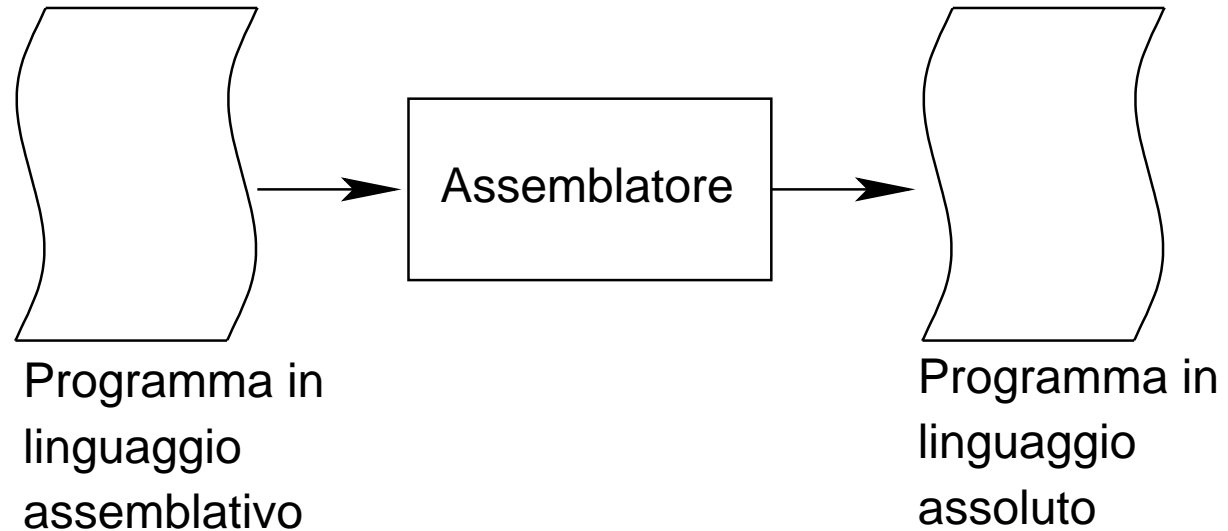


- L'assemblatore si occupa di
 - ◆ Creare la tabella dei simboli: Assegnare un indirizzo univoco a ciascun simbolo (operando, etichetta di istruzione)
 - ◆ Tradurre ogni istruzione con il relativo codice operativo e, se l'istruzione include un simbolo, scrivere nel secondo byte dell'istruzione l'indirizzo corrispondente al simbolo nella tabella dei simboli

Assemblatori e compilatori

Linguaggio assoluto
Struttura delle istruzioni assolute
Tabella dei simboli
Programma in linguaggio assoluto
Le funzioni dell'assemblatore
Assemblatori e compilatori
Livelli di astrazione
Linguaggi ad alto livello

- La scrittura in linguaggio assoluto è difficile
 - ◆ per programmi non banali è impensabile
- La scrittura in linguaggio assembly e l'uso di un assemblatore eliminano parte della difficoltà, liberando il programmatore dalla gestione dei codici operativi e degli indirizzi
- Oggi gran parte dei programmi sono scritti in *linguaggi ad alto livello* e trasformati in linguaggio assoluto utilizzando *compilatori*



Livelli di astrazione

Linguaggio assoluto
Struttura delle istruzioni assolute
Tabella dei simboli
Programma in linguaggio assoluto
Le funzioni dell'assemblatore
Assemblatori e compilatori
Livelli di astrazione
Linguaggi ad alto livello

- Riassumiamo le possibili rappresentazioni di un problema e della sua soluzione algoritmica in ordine discendente di astrazione

definizione del problema e algoritmo	formalismi astratti, linguaggio naturale
rappresentazione bidimensionale o strutturata (costrutti di ripetizione)	linguaggi ad alto livello
rappresentazione linearizzata	linguaggi con salto condizionato
rappresentazione con le istruzioni assemblative di una specifica CPU, con codici operativi simbolici, operandi e etichette	linguaggi assemblativi
scrittura con codici operativi e indirizzi binari	linguaggi assoluti

Linguaggi ad alto livello

Linguaggio assoluto
Struttura delle
istruzioni assolute
Tabella dei simboli
Programma in
linguaggio assoluto
Le funzioni
dell'assemblatore
Assemblatori e
compilatori
Livelli di astrazione
Linguaggi ad alto
livello

- I linguaggi ad alto livello si differenziano dai linguaggi assemblativi
 - ◆ maggiore potenza espressiva
 - una istruzione ad alto livello corrisponde a molte istruzioni in linguaggio assemblativo
 - permettono la ripetizione di sequenze di istruzioni senza utilizzare un'istruzione di salto condizionato
 - ◆ indipendenti dalla CPU
- Alcuni linguaggi ad alto livello distinti per campo di applicazione
 - ◆ scientifico: FORTRAN, APL
 - ◆ commerciali: COBOL, RPG
 - ◆ uso universale (**general purpose**): C, JAVA, PASCAL, BASIC, PL/1
 - ◆ speciali: SQL, APEX
- Alcuni linguaggi ad alto livello distinti per caratteristiche
 - ◆ imperativi: C, JAVA, PASCAL, BASIC, FORTRAN, COBOL, ...
 - ◆ orientati a oggetti: JAVA, C++, OBJECT PASCAL, SMALLTALK
 - ◆ logici: PROLOG
 - ◆ funzionali: LISP