

Linguaggio di Programmazione

E` una notazione per descrivere gli algoritmi.

Programma:

e` la rappresentazione di un algoritmo in un particolare linguaggio di programmazione.

In generale, ogni linguaggio di programmazione dispone di un insieme di "parole chiave" (*keywords*), attraverso le quali e` possibile esprimere il flusso di azioni descritto dall'algoritmo.

Ogni linguaggio e` caratterizzato da una sintassi e da una semantica:

- **sintassi:** e' l'insieme di regole formali per la composizione di programmi nel linguaggio scelto. Le regole sintattiche dettano le modalita` di combinazione tra le parole chiave del linguaggio, per costruire correttamente istruzioni (frasi).
- **semantica:** e` l'insieme dei significati da attribuire alle frasi (sintatticamente corrette) costruite nel linguaggio scelto.

Il linguaggio macchina

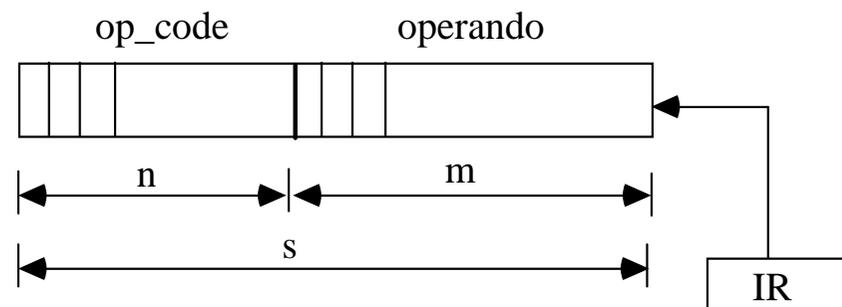
Il **linguaggio macchina** è direttamente eseguibile dall'elaboratore, senza nessuna traduzione.

- **Istruzioni:**

Si dividono in due parti: un **codice operativo** ed, eventualmente, uno o più **operandi**:

- Il **codice operativo** specifica l'operazione da compiere
- gli **operandi** individuano le celle di memoria a cui si riferiscono le operazioni.

- Se consideriamo istruzioni ad un solo operando:



Istruzioni Macchina

s , lunghezza di un'istruzione (in bit)

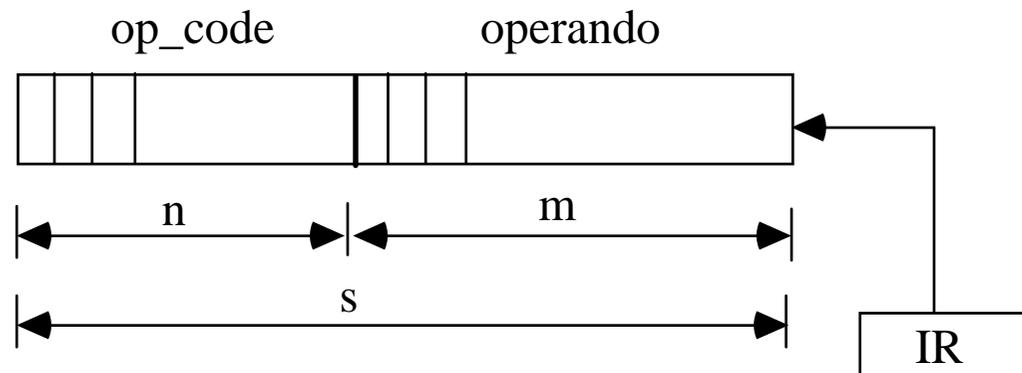
$$s = n + m$$

n , numero di bit dedicati al codice operativo

m , numero di bit dedicati all'indirizzamento degli operandi.

Insieme di istruzioni del linguaggio: al più 2^n istruzioni diverse (ciascuna ha un diverso *op_code*)

Memoria indirizzabile: al più 2^m celle di memoria diverse.



Set di Istruzioni di un elaboratore:

E` l'insieme delle istruzioni che la macchina e` in grado di eseguire direttamente.

- Ad esempio: 14 istruzioni (VAX della Digital 304!) sono sufficienti 4 bit ($2^4 > 14$).

<i>op_code</i>	<i>istruzione</i>
0000	LOADA
0001	LOADB
0010	STOREA
0011	STOREB
0100	READ
0101	WRITE
0110	ADD
0111	DIF
1000	MUL
1001	DIV
1010	JUMP
1011	JUMPZ
1100	NOP
1101	HALT

Linguaggio Macchina:Esempio

0	READ	8
1	READ	9
2	LOADA	8
3	LOADB	9
4	MUL	
5	STOREA	8
6	WRITE	8
7	HALT	
8	DATO INTERO	
9	DATO INTERO	

Rappresentazione reale (binaria):

0	0100	0000	0000	1000
1	0100	0000	0000	1001
2	0000	0000	0000	1000
3	0001	0000	0000	1001
4	1000	0000	0000	0000
5	0010	0000	0000	1000
6	0101	0000	0000	1000
7	1101	0000	0000	0000
8	0000	0000	0000	0000
9	0000	0000	0000	0000

Il linguaggio ASSEMBLER

E' difficile leggere e capire un programma scritto in forma binaria:

Linguaggi assembleri (Assembler):

- Le istruzioni corrispondono univocamente a quelle macchina, ma vengono espresse tramite nomi simbolici (parole chiave).
 - I riferimenti alle celle di memoria sono fatti mediante nomi simbolici (identificatori).
 - Identificatori che rappresentano dati (costanti o variabili) oppure istruzioni (etichette).
- Il programma prima di essere eseguito deve essere tradotto in linguaggio macchina (**assemblatore**).

Il linguaggio ASSEMBLER: esempio

READ X

READ Y

LOADA X

LOADB Y

MUL

STOREA X

WRITE X

HALT

X INT

Y INT

Linguaggi di programmazione

Linguaggio Macchina:

- necessita` di conoscere dettagliatamente le caratteristiche della macchina (registri, dimensioni dati, set di istruzioni)
- conoscenza dei metodi di rappresentazione delle informazioni utilizzati.
- conoscenza della collocazione in memoria di istruzioni e dati

Linguaggio Assembler:

- corrispondenza uno-a-uno tra istruzioni macchina e istruzioni assembler
- possibilita` di esprimere in modo simbolico istruzioni e dati

Limite:

- i programmi dipendono strettamente dalle caratteristiche architetturali del microprocessore: **NON C'E` PORTABILITA` !!**

Linguaggi di programmazione di alto livello

Con un **linguaggio di alto livello** il programmatore puo` astrarre dai dettagli legati all'architettura ed esprimere i propri algoritmi in modo simbolico.

Linguaggi di alto livello: Sono indipendenti dalla macchina:

- Capacita` di astrazione nel progetto di programmi;
- Portabilita` dei programmi.

Esecuzione di programmi scritti in linguaggi di alto livello:

- E` necessaria la **traduzione** di ogni programma nella corrispondente sequenza di istruzioni macchina (direttamente eseguibili dal processore), attraverso:
- **interpretazione** (ad es. BASIC)
- **compilazione** (ad es. C, FORTRAN, Pascal)