



**UNIVERSITÀ DEGLI STUDI
DEL SANNIO** Benevento
DING
DIPARTIMENTO DI INGEGNERIA


CORSO DI "PROGRAMMAZIONE I"

Prof. Franco FRATTOLILLO
Dipartimento di Ingegneria
Università degli Studi del Sannio

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 1

Programmi e calcolatori

- Un programma è un testo che permette ad un calcolatore di svolgere un compito
- Il contenuto di un programma viene detto codice



```
01001010010010100
11100100110011100
00110111100001010
11001101001000110
01001010010010100
11100100110011100
00110111100001010
11001101001000110
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 2

Programmi e linguaggi

- Ogni testo è scritto in un qualche linguaggio
- I programmi sono scritti in un linguaggio specializzato detto linguaggio di programmazione

```
01001010010010100
11100100110011100
00110111100001010
11001101001000110
01001010010010100
11100100110011100
00110111100001010
11001101001000110
```

```
class Nothing {
    public static void main (String[] arg) {
    }
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 3

Breve storia dei linguaggi di programmazione...

Linguaggio	Assembly	LISP	ALGOL	Pascal	C++
Macchina					
		FORTRAN			
				C	ADA
					Smalltalk
					Java

1945 1950 1960 1967

- C'è stata un'evoluzione dei linguaggi verso:
 - Astrazione,
 - Semplificazione,
 - Similarità con il ragionamento umano

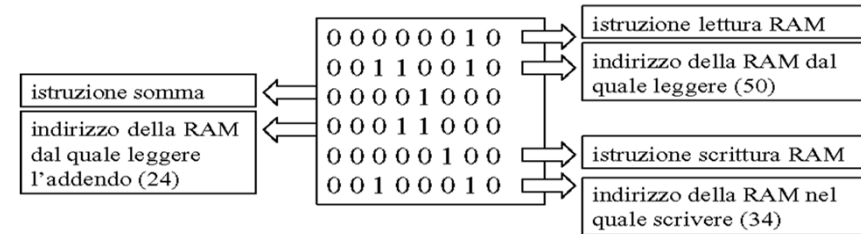
Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 4

Linguaggi di programmazione

- I linguaggi di programmazione sono classificati in tre livelli:
 - linguaggi macchina
 - linguaggi assembly
 - linguaggi ad alto livello

Il linguaggio macchina

- Il programma viene inizialmente caricato nella memoria del calcolatore, detta RAM
- L'elaborazione comincia a partire dalla prima istruzione



Assembly

- Il programmatore non deve più ricordare sequenze astruse di numeri binari, ma può usufruire di assembler che traducono automaticamente:
 - codici operativi per le istruzioni macchina
 - nomi simbolici o mnemonici per registri e per locazioni di memoria che memorizzano i dati
- Esempio:

```
load ACC, var1
add  ACC, var2
store tot, ACC
```

I problemi dei linguaggi macchina

- Sono specifici della macchina
 - Ogni macchina ha il proprio linguaggio macchina
 - Occorre conoscere l'architettura della macchina per scrivere programmi
 - I programmi non sono portabili
- I codici sono illeggibili all'uomo
- I programmatori si specializzano nel cercare efficienza su una macchina specifica, anziché concentrarsi sul problema

... e dell'assembly

- Sono comunque legati all'architettura della macchina
- I linguaggi assembly non sono sufficienti a gestire l'enorme complessità dei programmi moderni
 - TOP_DOWN o BOTTOM_UP ?
 - Il modo naturale di procedere è pensare prima alla struttura generale e poi curare i dettagli ...
 - ma questo è impossibile con l'Assembly, che è fatto SOLO da dettagli...

I linguaggi di programmazione

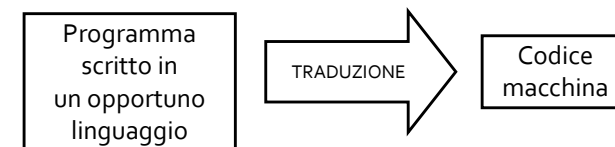
- I linguaggi di programmazione sono stati introdotti per facilitare la scrittura dei programmi
- Sono linguaggi simbolici e in continua evoluzione
- Sono definiti da un insieme di regole formali, le regole grammaticali o sintassi

Sintassi e semantica

- Le regole di sintassi definiscono come si devono comporre i simboli e le parole per formare istruzioni corrette
- La semantica di un'istruzione definisce il significato della stessa
- Un programma sintatticamente corretto non è necessariamente semanticamente corretto
 - I programmi fanno quello che prescriviamo che facciano e non quello che vorremmo che facessero

L'idea della traduzione

- IDEA: scrivo le mie istruzioni usando un linguaggio a me più comprensibile e poi le traduco in linguaggio macchina



Alto e basso livello

- Nell'ambito dei linguaggi di programmazione:
 - Se ci si avvicina al linguaggio umano, si parla di linguaggi di Alto livello
 - Se ci si avvicina al linguaggio macchina, si parla di linguaggi di Basso livello

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 13

Diversi livelli di espressività ...

- In un linguaggio ad alto livello:

`tot = var1 + var2;`
- In un linguaggio assembly:

`load ACC, var1
add ACC, var2
store tot, ACC`

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 14

... Diversi livelli di espressività

- In un linguaggio ad alto livello:

`se (a==b) allora c=0
altrimenti c=a+b;`
- In un linguaggio assembly:

`load R1, a
load R2, b
sub R1, R2
jzero R1, fine
load R1, a
add R1, R2
fine: store c, R1`

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 15

Traduzione dei linguaggi

- Il concetto di traduzione dei linguaggi ha permesso l'evoluzione verso sistemi simbolici più espressivi e più facilmente manipolabili dai programmatori
 - Il programmatore scrive un programma in un linguaggio ad alto livello senza preoccuparsi della macchina che esegue il programma

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 16

Linguaggi e traduzione

linguaggio di programmazione di alto livello

```
class Nothing {
    public static void main (String[] arg) {
    }
}
```

linguaggio di programmazione di basso livello

```
01001010010010100
11100100110011100
00110111100001010
11001101001000110
01001010010010100
11100100110011100
00110111100001010
11001101001000110
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 17

Il meccanismo della traduzione

- Tradurre è estremamente complesso. Si preferisce fare una catena di traduzioni tra linguaggi leggermente differenti andando sempre "verso" la macchina

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 18

Traduttori

- Servono a generare software
 - Generano codice in linguaggio macchina a partire da codice scritto in un linguaggio di programmazione ad alto livello (ad es. C, C++, Java ...).
- Si distinguono in:
 - interpreti
 - compilatori

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 19

Compilatori

- Un compilatore è un programma che riceve in input un codice sorgente e lo traduce, fornendo in output un codice oggetto
 - Per eseguire un programma sorgente P , scritto in un linguaggio di programmazione L :
 - P viene tradotto in un programma Q equivalente scritto in linguaggio macchina;
 - il programma Q viene eseguito
- Esempi di linguaggi compilati:
 - C, C++, Pascal, Cobol, Fortran, ...
- Il compilatore è legato all'architettura della macchina

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 20

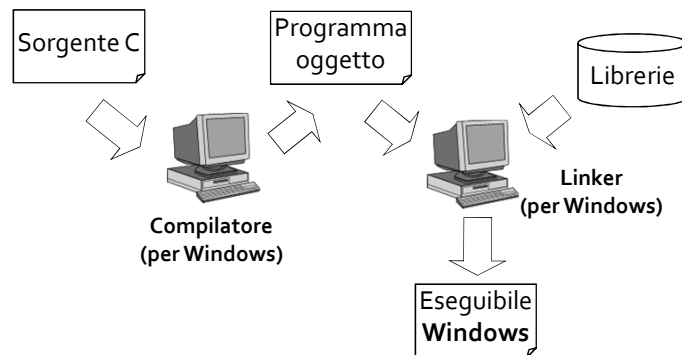
Interpreti

- Un interprete è un programma che riceve in input un codice sorgente e, passo dopo passo, traduce ed esegue ogni singola istruzione
 - La traduzione avviene dunque simultaneamente all'esecuzione
 - Per ogni istruzione del programma sorgente *P*:
 - Viene tradotta la singola istruzione generando il corrispondente insieme di istruzioni in linguaggio macchina;
 - Si esegue il codice in linguaggio macchina e si passa all'istruzione sorgente successiva
- Esempi di linguaggi interpretati:
 - VBasic, Lisp, Prolog, Java (inizialmente)

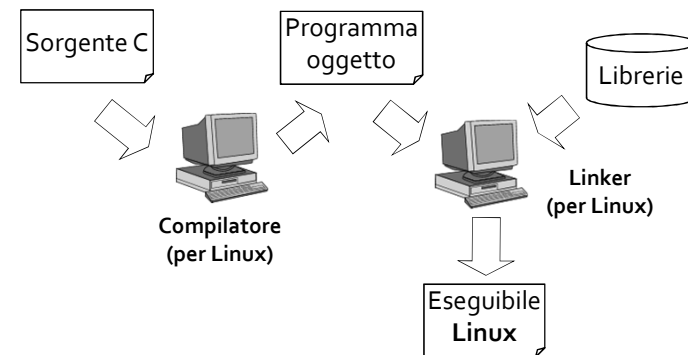
Software necessario

- L'impiego dei linguaggi simbolici, come il C, è possibile grazie alla disponibilità del software di base:
- Editor: produzione del programma origine
- Compilatore: produzione del programma oggetto, mediante la traduzione del programma origine da linguaggio simbolico a linguaggio macchina
 - ... in alternativa si può usare un interprete ...
- Collegatore: produzione del programma eseguibile mediante il collegamento del programma oggetto con componenti di librerie
- Caricatore: caricamento del programma eseguibile in memoria al fine della sua esecuzione

Traduzione C



Traduzione C



Cosa offre un linguaggio di alto livello

- **Modularizzazione:**
 - È possibile creare sottoprogrammi il più possibile indipendenti tra loro ed isolare in tali “moduli” le operazioni più semplici
 - costruire da moduli semplici moduli via via più complessi...
- **Astrazione:**
 - È possibile slegare il programmatore dal modello della macchina e avvicinarlo al modello del problema da risolvere
- **Strutturazione:**
 - È possibile rendere i salti nell'esecuzione del programma espliciti, visibili e chiari

Una nuova generazione di linguaggi

- C, Pascal, MODULA, FORTRAN77
- Supportano la programmazione strutturata:
 - funzioni che isolano i sottoprogrammi;
 - controllo della gestione della memoria mediante variabili “tipizzate”
 - ...

Evoluzione delle tecniche di programmazione



I paradigmi di programmazione

- Forniscono la filosofia con cui si scrivono i programmi e stabiliscono :
 - la metodologia con cui si scrivono i programmi
 - il concetto di computazione
- I linguaggi devono consentire, ma soprattutto spingere, all'adozione di un particolare paradigma
 - Funzionale
 - Logica
 - Imperativa
 - Modulare
 - Orientata agli oggetti

Paradigma procedurale

- Enfasi sulla soluzione dei problemi mediante modifica progressiva dei dati
 - Esecuzione sequenziale di istruzioni
 - Stato della memoria
 - Cambiamento di stato tramite esecuzione di istruzioni
- Aderente al modello della macchina di von Neumann
- Molto efficiente
- Ha mostrato limiti nello sviluppo e manutenzione di software complessi
- Pascal, C ...

Influenza del modello di macchina

- Concetto di istruzione
- Concetto di sequenzialità e iterazione
 - Il programma assolve al compito eseguendo le istruzioni in sequenza
- Concetto di variabile e di assegnamento
 - Le celle di memoria hanno un indirizzo e contengono i dati da manipolare
 - Le variabili hanno un nome e un valore
 - L'assegnamento di un valore ad una variabile equivale al trasferimento di un dato in una cella

Compilare ed eseguire

```

Terminal — bash — 80x24
Last login: Sun Mar 5 22:38:47 on ttty1
Welcome to Darwin!
host139-94:~ GC$ cd Documents/workspace/
host139-94:~/Documents/workspace GC$ cd pr1/
host139-94:~/Documents/workspace/pr1 GC$ javac c1.java
host139-94:~/Documents/workspace/pr1 GC$ java c1
Hello, World!
host139-94:~/Documents/workspace/pr1 GC$
  
```

Compilare ed eseguire

```

Lista 2.c - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Plugins DoxyBlocks Settings Help

1
2
3
4
5
6
7
8
9
10
11
12
13
14 node *newNode(char *) {
15     void print();
16
17     int main() {
18         char buffer[SIZE];
19         int n;
20
21         printf("Inserisci il numero di stringhe da leggere: ");
22         scanf("%d", &n);
23
24         for (; n>0; n--) {
25             scanf("%s", buffer);
26             addStr(buffer);
27         }
28
29         print();
30         return 0;
31     }
32
33     char *halfStr(char *stringa) {
34         int i, length = strlen(stringa);
35         int halfLength = ceil(length / 2.0);
36         char *temp = calloc(halfLength + 1, sizeof(char));
37         for(i=0; i< halfLength; i++) temp[i] = stringa[i];
38     }
  
```


Il primo programma C

```
#include <stdio.h>
void main( )
{
    printf("Hello, world.\n");
}
```

- `printf("Hello, world.\n");`
 - è un'azione elaborativa non semplice, decomponibile in altre azioni elaborative che sono dettagliate nella libreria `stdio.h`

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 33

Librerie

- Una libreria contiene componenti scritte da altri programmatori che possono essere riusate nel nostro programma
 - le informazioni sulle componenti disponibili nella libreria sono fornite in un header file (con estensione '.h') da includere nel programma
- `#include <stdio.h>`
 - l'estensione .h indica un header file, così come l'estensione .c indica un file contenente un programma C
 - la libreria descritta dall'header file `stdio.h` è predefinita

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 34

Istruzione di uscita

- Nella libreria `stdio.h` l'uscita standard viene normalmente inviata sullo schermo

```
printf ("Hello, world.\n");
```

- In C ogni istruzione semplice termina con un ";"

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 35

Dati di uscita

- Nell'esempio precedente lo stream di uscita è dato dalla sequenza di caratteri (stringa):

Hello, world.
- In C una stringa di caratteri viene racchiusa tra doppi apici
- Sullo schermo comparirà il messaggio:

Hello, world.
 —

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 36

Struttura generale dei Programmi

File: hello.c

Questo programma visualizza il messaggio
"Hello, world." sullo schermo.

```
#include <stdio.h>
void main( )
{
    printf("Hello, world.\n");
}
```

Apertura e chiusura di un commento

Inclusione di libreria

Programma principale

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 37

Commenti

- I commenti sono utilizzati per fornire al programmatore descrizioni informali del programma
 - non vengono eseguiti dalla macchina
 - facilitano la comprensione del programmatore
 - possono trovarsi su una o più righe e sono delimitati da `/*` e `*/`, oppure sono preceduti da `//` se si trovano su una sola riga
- possono essere inseriti sia in testa al programma (per dare una descrizione complessiva della sua funzionalità) o vicino a particolari parti di codice

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 38

Istruzioni di I/O

- In C non sono definite istruzioni o operatori specifici volti a realizzare le classiche operazioni di ingresso/uscita
- Queste funzioni sono delegate ad apposite librerie esterne predefinite
 - ... di cui bisogna includere l'header file nel programma
 - `stdio.h` per il C

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 39