



FONDAMENTI DI INFORMATICA 1

<http://www.diee.unica.it/~marcialis/FI1>

A.A. 2010/2011

Docente: **Gian Luca Marcialis**

LINGUAGGIO C Introduzione

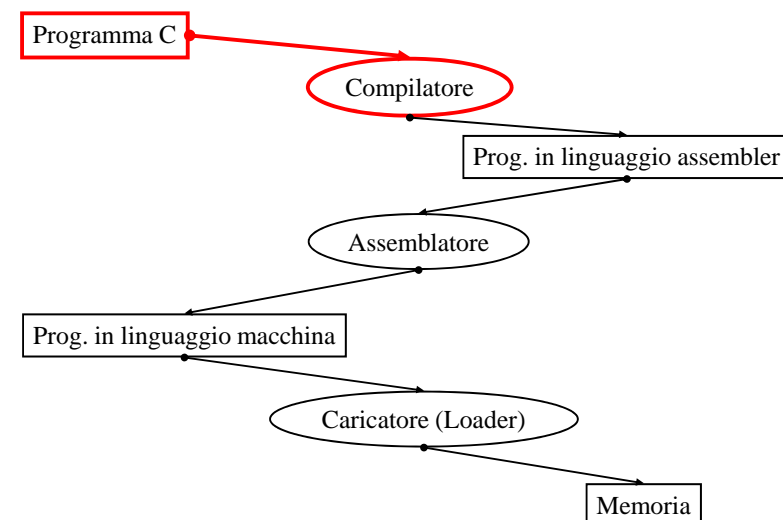
Ricapitoliamo un po'

- Ci siamo soffermati su come l'informatica possa essere lo **studio degli algoritmi**
- Un algoritmo per essere "eseguito" necessita di un modello di calcolo → la macchina di Von Neumann
 - Un algoritmo scritto in un opportuno linguaggio prende il nome di **programma**
- Tuttavia abbiamo rilevato quanto complesso sia esprimersi in linguaggio macchina
 - Nonché effettuare il controllo della stessa (v. Ciclo di esecuzione e Sist. Operativi)
- Dall'altra parte il linguaggio naturale non è ideale per scrivere algoritmi rispettandone le proprietà
- Esistono dei **linguaggi** che permettono la scrittura di programmi in una forma accessibile sia a noi che alla macchina
 - Il **LINGUAGGIO C** è uno di questi

Sommario

- Linguaggio C: Introduzione
- Tipi di dati in C
- Strutture di controllo
- Funzioni e procedure
 - Funzioni ricorsive (cenni)
- Gestione dei file
- I vettori
 - La ricerca
 - L'ordinamento
- I puntatori (cenni)

Programmi: da linguaggio di alto livello a caricamento in memoria



Il nucleo del linguaggio C

- Come tutti i linguaggi, esso è dotato di:
 - Sintassi
 - Semantica
- La sintassi è l'insieme di regole per la costruzione corretta di una "frase", ovvero di una istruzione
- La semantica è il significato che si dà alla "frase"
- Poiché il linguaggio C deve essere comprensibile al calcolatore, non ci possono essere ambiguità semantiche
 - Una "frase" in C ha sempre lo stesso significato

5

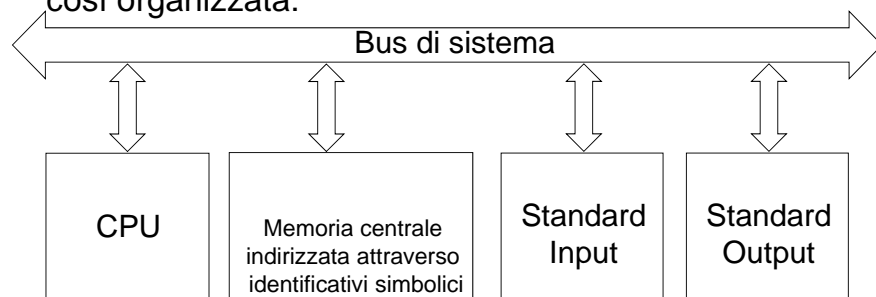
Elementi base del C

- Identificatori simbolici
 - Pubblici
 - Rappresentano aree di memoria assegnate a certi dati dal programmatore
 - Es. un certo valore
 - ...oppure a certi gruppi di istruzioni che realizzano un determinato compito (funzioni)
 - Privati (non utilizzabili dal programmatore)
 - Parole-chiave
 - Istruzioni base del linguaggio
 - Nomi di "funzioni" di libreria
 - Gruppi di istruzioni che realizzano un compito accessibili tramite interfacce
 - Es. istruzioni aritmetiche complesse, stampa a schermo
 - Direttive
 - Es. attivazione di un certo insieme di funzioni di libreria

6

Una "macchina virtuale" per il C

- Per l'esecuzione di un programma C, assumeremo che la nostra macchina di Von Neumann (il nostro PC) sia così organizzata:



7

Il mio primo programma C

- Si scriva in linguaggio C un programma che stampi su Standard Output (es. video) il seguente messaggio:

Ciao, mondo!

8

Il mio primo programma in C

```
/*Il mio primo programma in C*/

#include <stdio.h>
int main()
{
    printf("\nCiao, mondo!\n");
    return 0;
}
```

9

Il mio primo programma in C

```
/*Il mio primo programma in C*/ ←
#include <stdio.h>
int main()
{
    printf("\nCiao, mondo!\n");
    return 0;
}
```

Questo è un **commento**: non contiene istruzioni dell'algoritmo ma in genere serve per spiegare cosa fa una certa parte di programma (codice)

Un **commento** viene inserito fra i simboli:

/* → inizio commento

*/ → fine commento

10

Il mio primo programma in C

```
/*Il mio primo programma in C*/

#include <stdio.h> ←
int main()
{
    printf("\nCiao, mondo!\n");
    return 0;
}
```

È una **direttiva**: serve in questo caso ad attivare un insieme di funzioni già pronte per comunicare con i periferici di nome "stdio.h"

Una direttiva è un identificatore simbolico preceduto dal carattere #

Il mio primo programma in C

```
/*Il mio primo programma in C*/

#include <stdio.h>
int main() ←
{
    printf("\nCiao, mondo!\n");
    return 0;
} ←
```

È l' "intestazione" del programma.

Il termine privato "main" indica appunto che la sequenza di istruzioni compresa entro le parentesi graffe è l'algoritmo

L'identificatore **int** significa che al termine del programma è attesa un'istruzione che segnali la terminazione attraverso un valore numerico (come "halt")

Il mio primo programma in C

```
/*Il mio primo programma in C*/  
  
#include <stdio.h>  
  
int main()  
{  
    printf("\nCiao, mondo!\n");  
    return 0;  
}
```

E' un'istruzione composta da una **funzione** presente nella libreria "stdio.h".

L'identificatore "printf" è il nome della funzione (codice operativo), e prevede che si stampi su SO la sequenza di caratteri fra virgolette.

13

Il mio primo programma in C

```
/*Il mio primo programma in C*/  
  
#include <stdio.h>  
  
int main()  
{  
    printf("\nCiao, mondo!\n");  
    return 0;  
}
```

E' un'istruzione **base** (key-word) del linguaggio e corrisponde all'istruzione "halt".

Significa che l'algoritmo ha termine restituendo al processore un valore intero atteso (in questo caso, 0)

14

Esecuzione del programma

- Scriviamo su un editor (es. blocco note) il programma visto e diamogli come nome `ciao.c`
- Compiliamolo
 - Es. usando Linux od Cygwin su ambiente Windows:
- gcc ciao.c -o ciao
 - Avremo nella stessa dir un eseguibile `ciao` oppure `ciao.exe` (su Win)
- Scrivendo a questo punto `ciao` e premendo il tasto "invio", avremo su schermo la seguente stampa:
Ciao, mondo!

15

Riassumendo

- Un programma C può essere caratterizzato da una o più righe di **commento**
- Seguono opportune **direttive** per l'attivazione di **funzioni di libreria** che possono essere utili per la scrittura del programma
- L'algoritmo, scritto in linguaggio C, viene inserito **tra due parentesi graffe** precedute dall'**intestazione di funzione principale** ("main")
- L'algoritmo termina con un'istruzione di "ritorno" (return)

16

L'istruzione di assegnamento

➤ Sia dato il seguente problema:

Scrivere un programma in C che riceva da tastiera due valori interi e stampi su video la loro somma

17

Soluzione

```
/*Programma per la stampa a video della somma di due
numeri*/

#include <stdio.h>
int main()
{
    int a, b, somma;
    printf("Inserire due valori da sommare\n");
    scanf("%d%d",&a,&b);
    somma = a + b;
    printf("%d + %d = %d\n",a,b,somma);
    return 0;
}
```

18

La soluzione presentata

- Il programma presenta in generale le stesse caratteristiche del precedente
- Cambia ovviamente la sequenza di istruzioni all'interno delle graffe
 - In particolare, viene chiamata la funzione `scanf` che serve a leggere da Standard Input (la tastiera) dei valori
 - Tali valori sono identificati dalla coppia simboli `%d` che identificano ciascuno un valore intero
 - **I valori letti da tastiera vengono assegnati alle variabili `a`, `b`**
 - Infine la somma di `a` con `b` viene **assegnata** ad un'altra variabile `somma`...
 - ... che viene stampata a video tramite la funzione `printf`

19

Variabili in C

- Cos'è una **variabile**?
 - E' un contenitore di valori (numeri o caratteri)
 - Ovvero, una locazione di memoria assegnata ed indirizzata mediante il suo identificatore simbolico
 - L'indirizzo di memoria specifico si ottiene antepoendo l'operatore `&` al nome della variabile
 - Es. `&a ==` indirizzo di `a` (mentre `a` indica il suo generico contenuto)
- Una variabile necessita di essere **dichiarata** perché essa trovi posto in memoria centrale, ovvero sia **allocata**
 - `int a, b, somma;`
- Ad essa può quindi essere assegnato un valore con apposita istruzione
 - `somma = a + b;`

20

Assegnamento ed operazioni aritmetiche e booleane elementari

- L'istruzione di assegnamento prevede la seguente sintassi:
 - `Variabile Destinazione = F(V1, V2, ..., VN);`
 - Le variabili V1, ..., VN sono elaborate attraverso operatori:
 - Somma, sottrazione, prodotto, divisione: +, -, *, /
 - Uguaglianza e disuguaglianza: ==, !=
 - Se i valori interi (**int**) vengono espressi come valori booleani (0, 1):
 - AND, OR, NOT: &&, ||, !
 - Esempi:
 - `area_quadrato = lato*lato;`
 - `segue_informatica = (biomedico|elettrico)&&matricola;`

21

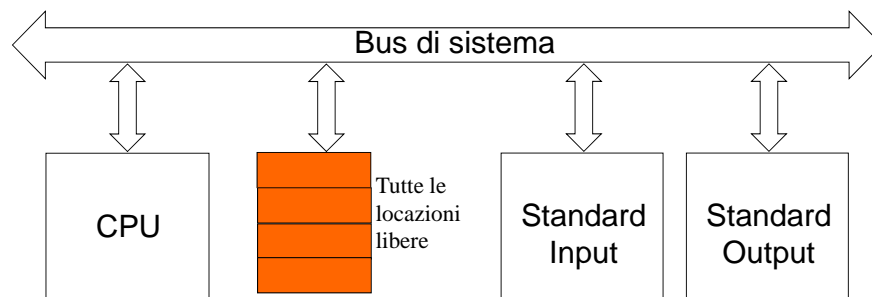
Esercizio

- Siano date due variabili a, b, di tipo intero
- Si supponga a=15, b=-20
- Dire il valore della seguente variabile c in funzione delle espressioni date:
 - `c = a-b;`
 - `c = a!=b;`
 - `c = !a`
 - `c = (a==b)||((a+b)==-5)`

22

L'istruzione di assegnamento

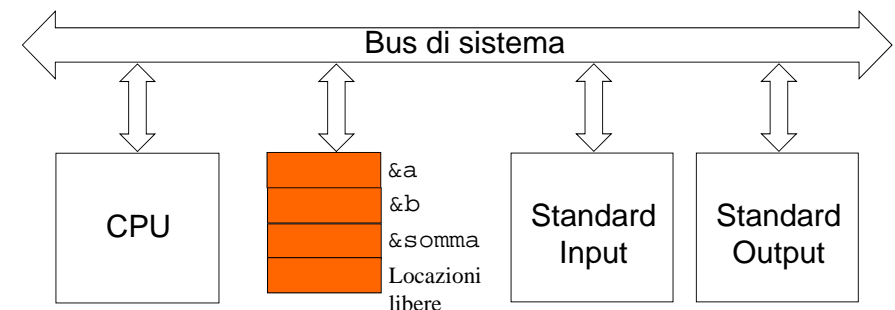
- Lo stato della "macchina virtuale C" prima della dichiarazione di a, b, somma



23

L'istruzione di assegnamento

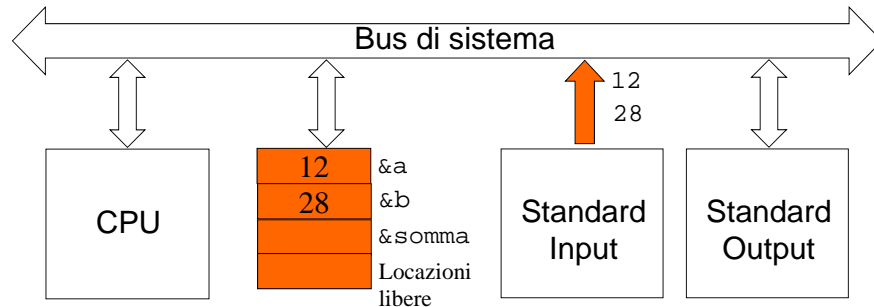
- Lo stato della "macchina virtuale C" dopo la dichiarazione di a, b, somma



24

L'istruzione di assegnamento

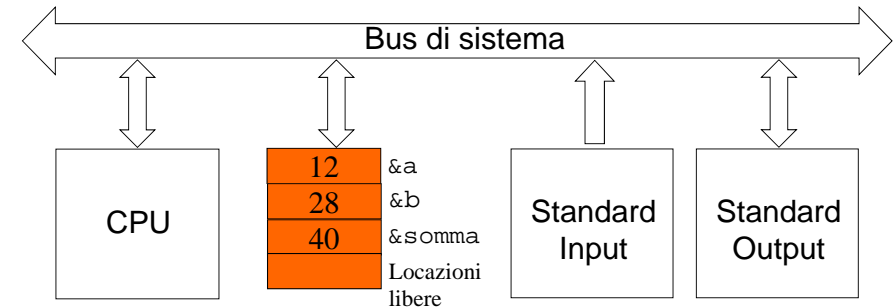
- Lo stato della “macchina virtuale C” dopo la lettura di due valori numerici dalla tastiera



25

L'istruzione di assegnamento

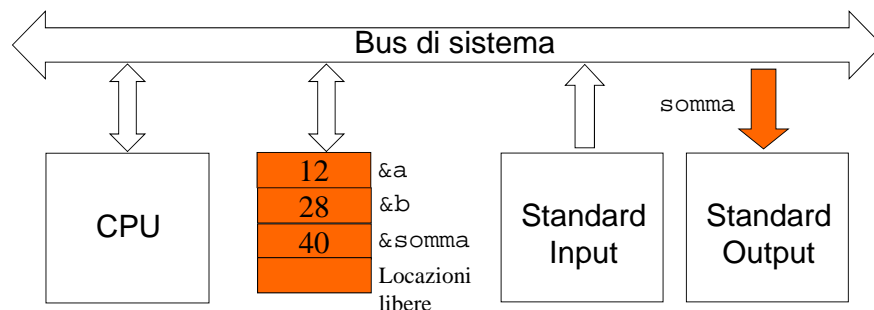
- Lo stato della “macchina virtuale C” dopo l'assegnamento della variabile `somma`



26

L'istruzione di assegnamento

- Lo stato della “macchina virtuale C” dopo la stampa di `somma` a video



27

Struttura generale di un programma C

- Commenti
- Direttive (anche dichiarazione di costanti)
- `int main()`
- {
 - PARTE DI DICHIARAZIONE DELLE VARIABILI;
 - ALGORITMO VERO E PROPRIO;
 - `return 0;`
- }

28

Esercizi

- Scrivere un programma C che riceva da tastiera un valore (intero) del lato di un quadrato e ne stampi a video l'area ed il perimetro
- Scrivere un programma C che riceva da tastiera tre valori booleani (0, 1) indicanti appunto i valori di verità di `iscritto_in_biomedica`, `iscritto_in_ambientale`, `primo_anno`, e stampi a video il valore di verità `segue_informatica`

Per saperne di più

- Ceri, Mandriola, Sbattella, *Informatica – arte e mestiere*, Capp.3-4, McGraw-Hill
- Kernighan, Ritchie, *Il linguaggio C*, Cap. 1, Pearson-Prentice Hall