Programmazione - lezione 3

Appunti di Davide Vella 2024/2025

Alessandro Mazzei
alessandro.mazzei@unito.it
Link al moodle :
informatica.i-learn.unito.it/course/view.php?id=2982

26/09/2024

Lezione 3

Il linguaggio C

Nasce nel 1969 da Denis M. Ritchie. Nasce per la necessità di non scrivere un OS in assembly. È stato standardizzato nel 1980 da ANSI e ISO per poterlo usare in hardware diversi da pdp-7 e pdp-11, computer per cui è stato inizialmente scrittor.
Il linguaggio C è :

- **Compilato**, un compilatore traduci i sorgenti c nel linguaggio macchina del caso.
- **Imperativo**, il programma è un insieme di istruzioni pensate come ordini per il compilatore.
- **Strutturato**, il codice sorgente è organizzato in blocchi racchiusi da delimitatori.
- **Fortemente** tipizzato, il programmatore deve specificare il tipo di ogni variabile.

I commenti nel codice vengono totalmente ignorati dal compilatore, servono solo al programmatore ed è molto importante inserirli per far capire meglio il codice, oltre a nominare in modo adeguato le variabili e le funzioni.

Il main

Un programma C è composto sempre da un certo numero di funzioni, fra cui almeno la funzione main, punto di ingresso al programma. Per ora tutto il nostro codice sarà contenuto nella funzione
main, quindi programma e funzione main saranno sinonimi. La funzione main accetta parametri da command line, ma noi non li utilizzeremo per ora (richiedono uso degli array). La funzione main ritorna al SO un codice di successo/errore, ma noi non lo utilizzeremo (mancanza di return esplicito) poiché viene sempre inserito di default.

Il body

il codice della funzione main deve essere chiuso fra una coppia di parentesi graffe. Il codice chiuso tra due graffe è detto blocco, il quale può essere annidato all'interno di un altro blocco.

Buona norma del codice :

Esistono varie buone regole per programmare in modo intuitivo :

- Commentare ogni funzione e darle un nome appropriato
- Indentare il codice, i blocchi annidati devo avere sempre dello spazio (tab o space che sia) prima del codice rispetto al blocco esterno.
- Lasciare sempre qualche riga vuota e carattere di spaziatura per rendere il codice più leggibile ed intuitivo.

La funzione printf

Serve a stampare a terminale output alfanumerico, per ora solo stringhe di testo racchiuse fra doppi apici (").

Example

```
printf("Hello World!");
```

La sequenza '\n' serve per andare a capo di una riga.

#Segnaposto

Nel printf si possono inserire del "segnaposto" con il simbolo '%'. Questi segnaposto servono per inserire variabili all'interno della stringa da stampare. Ogni segnaposto è seguito da uno specificatore che cambia in base al tipo della variabile che si vuole inserire :

Format Specifier	Description	Grandezza (di solito, byte)
%c	Per i caratteri	1
%d	Per gli gli interi (int) assegnati (signed)	4
%e o %E	Per la notazione scientifica dei float	4
%hd o %hi	Per gli short	2
%f	Per i float	4
%g or %G	Per i float con la precisione corrente	4
%i	signed integer	4
%ld o %li	Long	8
%lf	Double	8
%Lf	Long double	
%lu	Unsigned int o unsigned long	
%lli o %lld	Long long	
%llu	Unsigned long long	
%o	Rappresentazione ottale	
%p	Pointer	
%s	String	
%u	Unsigned int	
%x o %X	Rappresentazione esadecimale	
%n	Stampa niente	
%'% (senza ')	Stampa il carattere %	

Include

Le righe che iniziano per '#' sono direttive per il processore. La direttiva '#include' include un dato file in un altro (tipo le librerie). I files .h di intestazioni (header) contengono la dichiarazione di funzioni. L'header file stdio.h in dichiara funzioni come printf() e scanf(). Tutti gli esempi

seguenti includeranno l'header stdio.h anche quando questo è omesso riportato per ragioni di spazio.

Compilare

Per compilare un file c, dobbiamo andare da terminale nella working directory (con il comando cd "nome cartella" o cd .. per tornare indietro) e poi eseguire il comando "gcc nomeFile.c", in questo caso verrà generato un file nominato "a.exe". Si può anche rinominare il file mentro lo si compila con inserendo -o nomeFile.exe (.exe va solo su Windows), "gcc nomeFile.c -o nomeFile.exe". Per fare tutto ciò si deve prima installare il compilatore, potete trovare qui[1] una guida. Qui si usa gcc (GNU C Compiler), perché gratis, standard-compliant.

> **Tip**
>
> È consigliato usare -g tra gcc e "nomeFile.c" per includere i simboli di debug nell'eseguibile.

> **Tip**
>
> Inserire -Wall dopo gcc (o dopo -g) permette di stampare su terminale i warning (non errori) al momento della compilazione

Tipi di dati

Un tipo di dato è caratterizzato dalla rappresentazione in memoria dei dati di quel particolare tipo (ad esempio, quanti bit usa in memoria) e dall'insieme di operazioni ammissibili su dati di quel tipo. Questi permettono di effettuare controlli statici (cioè al momento della compilazione e non in fase di esecuzione/runtime) sulla correttezza sintattica del programma e quindi prevenire errori di programmazione. Potete trovare dei segnaposto nella tabella sopra, la quale contiente molti tipi di dati e i loro specificatori.

Note :
1 : Documentazione originale del GNU Project.