

Introduzione al Linguaggio C

Marco Coïsson

29 ottobre 2006

A tutti i lavoratori precari

Indice

1	Introduzione	7
2	Panoramica sul C	9
2.1	Il C come linguaggio di “basso livello”	9
2.2	Il C come linguaggio “strutturato”	10
2.3	Il C come linguaggio “strongly-typed”	11
2.4	Vantaggi e svantaggi del C	12
2.5	Varianti del C	12
3	Anatomia di un programma	15
3.1	File, funzioni, blocchi e variabili	15
3.2	File di header e file di codice	18
3.3	Librerie	20
3.4	<i>Scope</i>	22
4	Tipi, operatori ed espressioni	25
4.1	Tipi di variabile	25
4.2	Costanti	32
4.3	Operatori	36
4.4	Espressioni	42
5	Blocchi	43
5.1	Blocchi senza nome	43
5.2	Blocchi if	47
5.3	Blocchi switch	53
5.4	Blocchi while	57
5.5	Blocchi do-while	59
5.6	Blocchi for	61
5.7	Uscire dai cicli	63

6	Funzioni	65
6.1	La funzione main	65
6.2	Le altre funzioni	66
6.3	Le chiamate a funzioni standard del sistema operativo come funzioni	72
6.4	Variabili statiche	73
6.5	Ricorsività	76
7	Puntatori ed Array	81
7.1	Allocazione statica e dinamica di variabili	81
7.2	Puntatori	82
7.3	Array	87
7.4	Allocazione dinamica della memoria	92
7.5	Argomenti di invocazione	98
7.6	Puntatori a funzioni	100
8	Strutture, Liste e Unioni	107
8.1	Strutture	107
8.2	Unioni	119
9	Librerie standard	121
9.1	stdio.h	122
9.1.1	printf()	122
9.1.2	scanf()	123
9.1.3	Altre funzioni di stdio.h	123
9.2	stdlib.h	124
9.3	math.h	126
9.4	float.h e limits.h	126
9.5	ctype.h e string.h	126
9.6	time.h	128
10	I/O su file	129
10.1	I/O su file noti	129
10.2	I/O su file non noti	132
10.3	Aspetti generali dell'accesso a file	135
10.4	I parametri di fopen()	136
11	File multipli di codice	139
11.1	I file sorgenti	140
11.2	Compilare Master Mind	146

<i>INDICE</i>	5
11.3 Giocare a Master Mind	148
11.4 Come funziona Master Mind	148
12 Bibliografia	151

Capitolo 1

Introduzione

Questa *Introduzione al Linguaggio C* è in realtà una seconda edizione. La prima è un tutorial, un manualetto disponibile gratuitamente in formato pdf sulla mia homepage (http://homepage.mac.com/marco_coisson) nato come prima puntata di una “trilogia” dedicata alla programmazione su MacOS X. Tale trilogia, che prosegue con due volumi dedicati all’Objective-C e a Cocoa rispettivamente, aveva (ed ha) l’ambizione di introdurre il neofita all’affascinante mondo della programmazione di applicazioni native per MacOS X, dotate di interfaccia grafica, localizzazione in più lingue, preferenze e quant’altro.

Il linguaggio C, però, è un linguaggio universale, specie se si fa riferimento alla sua versione standard (ANSI C), disponibile ormai da tanti anni virtualmente su qualunque sistema operativo. L’esigenza di rivedere la prima edizione di questo manuale, pertanto, si è fatta sempre più pressante, a mano a mano che è risultato chiaro che diventava necessario superare due limiti insiti nella prima stesura:

- occorre proporre un manuale di base sul C che non fosse esplicitamente vincolato ad una sola piattaforma di sviluppo o ad un solo sistema operativo, dal momento che il C standard è universale e chiunque può usarlo con profitto sul proprio computer col sistema operativo che preferisce;
- occorre ampliare la trattazione del linguaggio C comprendendo alcuni argomenti in precedenza omessi, come l’allocazione dinamica della memoria, l’input/output su file, la gestione di file sorgenti multipli, più tante piccole cose che andavano riviste o approfondite.

Così nasce questa seconda edizione, largamente rivista ed ampliata, della

mia *Introduzione al Linguaggio C*. Un manuale accessibile a chiunque abbia voglia o necessità di imparare questo linguaggio, indipendentemente dall'ambiente di sviluppo o dal sistema operativo utilizzato. Per scelta, il manuale mostrerà come compilare i programmi proposti ad esempio usando `gcc` da una shell Unix (come quelle di MacOS X o Linux), ma solo perché non sarebbe possibile coprire in un solo testo tutti gli aspetti relativi agli ambienti di sviluppo integrati (IDE) disponibili sulle principali piattaforme (MacOS X, Linux, altre varianti *nix e Windows). Tutti i programmi proposti in questo libro, però, seguiranno sempre le convenzioni dell'ANSI C, pertanto saranno tutti funzionanti su qualunque sistema operativo decediate di provarli. Intenzionalmente, non faremo mai riferimento a librerie specifiche di un sistema operativo o di un ambiente di sviluppo, o ad alcunché non faccia parte dell'installazione standard del C, proprio per consentire a chiunque di apprendere al massimo da questo testo senza doversi preoccupare dei dettagli di implementazione di componenti non standard che potrebbero differire (o addirittura mancare) da un sistema operativo ad un altro.

I programmi che vedremo nel corso del testo sono tutti brevi (fatta eccezione per l'ultimo, proposto nel capitolo 11) ed autocontenuti; in genere non svolgono compiti utili, ma sono finalizzati al mettere in evidenza un aspetto specifico del linguaggio, quello di cui ci si sta occupando in quel momento. Il fatto che ogni programma sia autocontenuto, consente di scriverlo sul proprio computer, compilarlo ed eseguirlo, avendo così sempre un riscontro immediato di quello che si sta apprendendo dalla lettura del libro. È un punto a mio avviso importante, che spero vi tornerà utile.

Il testo si articola in un paio di capitoli iniziali più "filosofici", in cui il linguaggio C viene esaminato in alcuni suoi aspetti generali. Questi capitoli non sono strettamente necessari per la comprensione e l'apprendimento dei dettagli del linguaggio, ma possono costituire un utile bagaglio culturale per inquadrare sommariamente il ruolo del C nel panorama vastissimo dei linguaggi di programmazione oggi disponibili. Tutti i capitoli a seguire sono orientati allo studio di aspetti specifici del linguaggio C, partendo dai rudimenti fino ad alcuni dei temi più complessi. Nulla è dato per scontato, e nulla si richiede al lettore di sapere, se non l'aver una conoscenza di base del funzionamento e dell'uso di un computer e un'installazione (funzionante) di un compilatore C e/o di un ambiente integrato di sviluppo. La speranza è che, terminata la lettura di questo manuale, abbiate le conoscenze necessarie per affrontare autonomamente e serenamente i primi problemi di programmazione in C, ed eventualmente lo studio di manuali più avanzati dedicati all'approfondimento degli aspetti e delle tecniche di questo splendido linguaggio.