1

淺談 C 程式語言

本章大綱

1-1 何謂程式語言

程式設計師(programmer)利用程式語言(programming language)撰寫程式(program)，以完成某一項任務，如圖 1-1 所示。如利用 C 程式語言撰寫 C 程式，以完成 10000 筆資料由小到大的排序工作或是以 C 撰寫一您親朋好友的通訊錄，…等等。

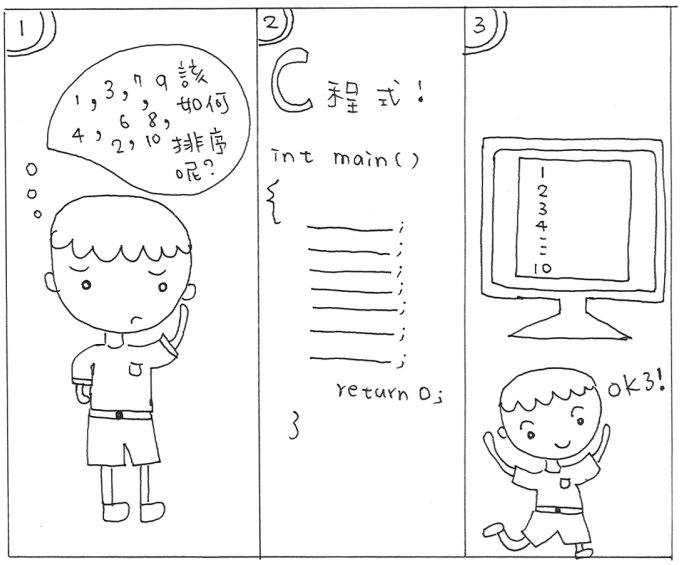

圖 1-1：小明利用 C 程式將一堆雜亂無章的資料，由小至大排序之

您撰寫的 C 程式，電腦是看不懂的，這之間需要 C 的**編譯程式(compiler)**，將 C 程式編譯為電腦看得懂的 0 與 1 所組成的機器語言(machine language)，讓電腦知道要做什麼。編譯程式可比喻是一位翻譯者(translators)，例如，有一位日本人到德國慕尼黑大學拜訪某一位教授，由於他不會講德文，所以請了一位會講德文的日本人當作翻譯者，做為與這一位教授溝通的橋樑，讓彼此了解對方的意思。如圖 1-2 所示。

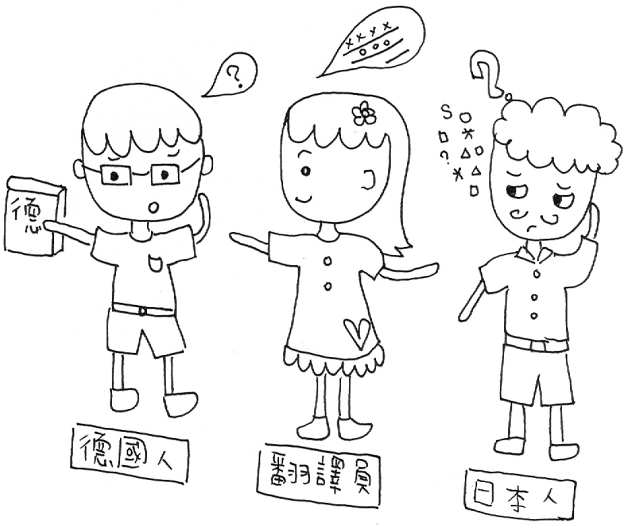

圖 1-2：中間的翻譯人員，有如編譯程式(compiler)

程式語言有上百，上千種，如：FORTRAN，COBOL，Pascal ，C，C++，Java，C#，Visual Basic，Ada，Objective C，Delphi，等等不勝枚舉，試問是不是每一種都要加以學習呢？其實大可不必，因為程式語言的架構幾乎大同小異，如每一種程式語言都有選擇敘述，迴圈敘述，只是以不同的關鍵字來表示罷了。了解大架構後，再去探討每一種語言都有其獨有的特性及功能。如：FORTRAN(發表於 1957 年)它的計算精確度高，適合於工程。COBOL(發表於 1960 年)適用於資料量大，但計算少的情況，而且製造美觀的報表，如目前的自來水公司，台電所印出的繳費單都是以 COBOL 程式語言寫的。這兩種我在大學時都有學過。後來 Pascal 興起，因為它有指標(pointer)，可以用來實作資料結構(data structures)的一些主題，如：鏈結串列(linked list)，二元搜尋樹(binary search tree)等等。

目前的 FORTRAN 已有 FORTRAN 2003(發表於 2003 年)，但慢慢的被 C(發表於 1970 年)取代了，只有少數如數學系，物理系還有在使用，因為他們有些特殊的應用軟體是以 FORTRAN 撰寫的，所以不得不要學它。COBOL 沒有新的版本，也逐漸的被另一種程式語言取而代之。

1-2 C 語言的興起

Pascal 發表於 1971，紅於 1980 年代，但卻好景不常，因為 C 語言也在這時候來湊熱鬧，由於 C 有威力更強的指標，而且又是 UNIX 或 Linux 作業系統所使用的程式語言，所以取而代之的流行程式語言就是 C。

C 是從 B 語言所延伸過來的程式語言，而 B 語言的前身是 BCPL 語言，同時多多少少也受 ALGOL 68(發表於 1968 年)的影響。從圖 1-3 可看出一些常用高階程式語言的發展史。目前大家使用的 C 語言是 1989 年的美國國家標準局所公佈標準的 C 語言，簡稱 ANSI C (American National Standard Institute, ANSI C)。

圖 1-3 是程式語言的發展史[1]。此圖顯示這麼多的程式語言，是不是都要老師教後，您才會呢？不對，以我來說，大學時只學過 FORTRAN 和 COBOL，上研究所自己讀 Pascal，用它來實作程式，將執行的結果用來證明論文所闡述的理論，到行政院工作時，因為要使用 C 和 C++來實作系統，所以也自己 K，這樣一路走來，還是走得很不錯，這要歸於大學時，將 FORTRAN 和 COBOL 這兩種程式語言弄得很熟的關係，所以再去看其它的程式語言是很容易的。因為程式語言的架構大都是相同的，只要再花一些時間了解此程式語言特有的功能即可。以目前這麼多的程式語言來說，建議大家以 C 為出發點，因為它淺顯易懂，可以很容易地使用它來撰寫程式，從而了解程式設計到底是什麼。

C 語言除了淺顯易懂外，同時也是 UNIX 與 Linux 作業系統的核心語言，更是受大家歡迎的程式語言，請參閱表 1-1。

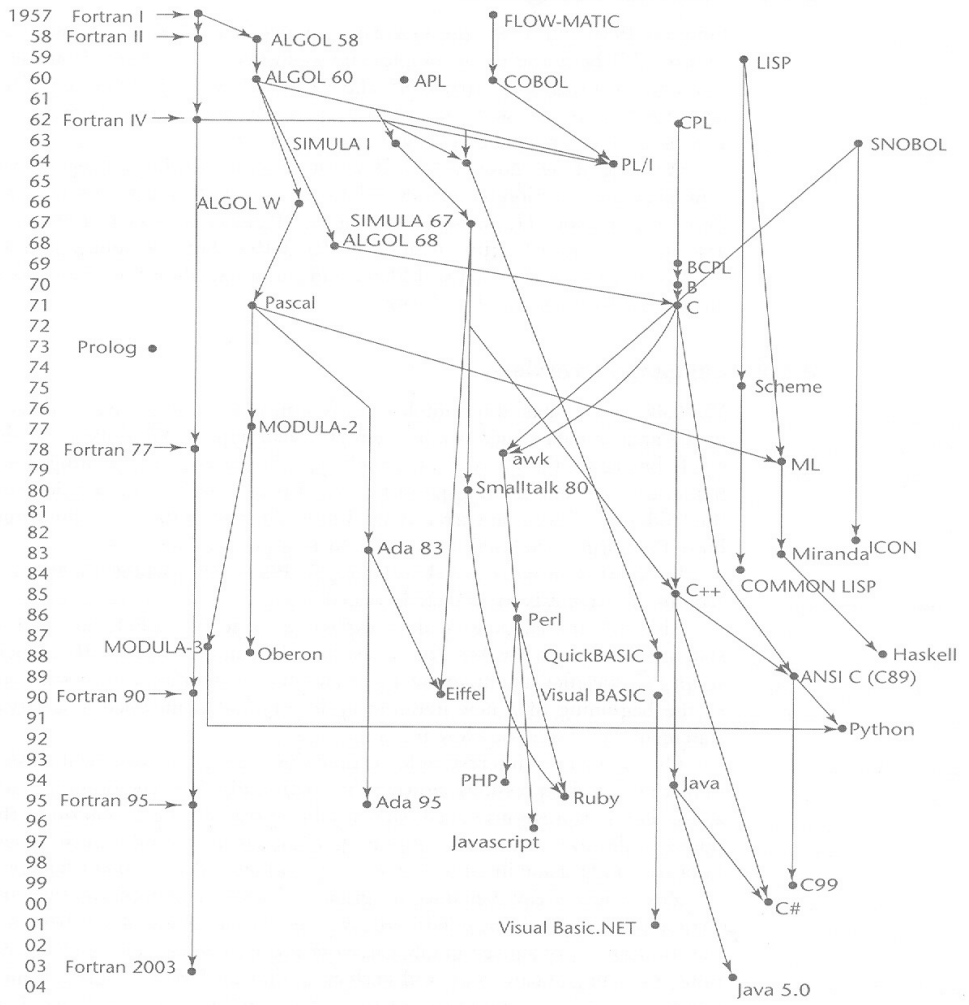

圖 1-3：程式語言的發展史[1]

根據 TIOBE programming community index for December 2008[2]，C 和 C++是 2008 很受歡迎的程式語言，而且其使用率僅次於 Java，請參閱表 1-1。當您熟悉 C 程式語言後，再學 C++就可以很容易進入其核心，即所謂的物件導向程式設計的特性。

表 1-1：目前常用的程式語言使用率排行榜，C 是很受歡迎的程式語言

2008 年 12 月 排行榜	2007 年 12 月 排行榜	程式語言	2008 年 12 月 使用百分比
1	1	Java	19.367%
2	2	C	16.163%
3	5	C++	10.893%
4	4	PHP	9.479%
5	3	(Visual) Basic	9.478%
6	8	C#	4.643%
7	6	Python	4.567%
8	7	Perl	3.603%
9	10	JavaScript	3.062%
10	11	Delphi	3.055%
11	9	Ruby	2.308%
12	12	D	1.185%
13	13	PL/SQL	1.140%
14	14	SAS	0.843%
15	19	Pascal	0.689%
16	15	COBOL	0.631%
17	16	ABAP	0.603%

如果您精通 C 語言之後，相信您可以很容易看懂其它的程式語言。這也正是筆者寫這一系列有關 C 語言書籍的最大原因。透過這一系列的書，希望您可以深入了解 C 語言，而不是一知半解，往後要看懂其它的程式語言就很簡單了，各位加油。

1-3 如何學好程式設計

已在這領域打滾 20 幾年的我，告訴您學習程式語言的不二法門是，多做，多看，多親自除錯(debug)，所謂的三多。多做一些題目，多看一些相關的書籍，並且能夠做中學，有錯誤(bugs)能親自除錯，不要有 bugs 就找人替您 debug，這是不好的習慣，因為久而久之，他的功力愈來愈好，而您的功力愈來愈差。千萬要記住自己去 debug。記住這些，您才有辦法「有朝一日，出人頭地」。

1-4 如何編譯程式

利用 C 程式語言的語法寫出的程式，稱之為 C 原始程式(C source code)，它的延伸檔名是 .c。如何撰寫 C 程式，之後將它變為可執行檔，請看以下的步驟：

步驟一： 選擇一個適當的編譯程式，開始編輯(edit) C 程式，並取原始碼的延伸檔名為.c。

步驟二： 利用此編譯程式將此原始碼，編譯成目的程式(object code)，其延伸檔名是.obj。

步驟三： 利用連結程式(linker)，將程式用到庫存函數之目的程式，加以連結為可執行程式(execution code)，其延伸檔名為 .exe。

如我們撰寫了一支用來排序 10000 筆資料的 C 程式，名為 myfirst.c，經過編譯程式將它編譯成 myfirst.obj 的目的程式，再經由連結程式加以連結程式中所需的函數庫程式碼，最後成為 myfirst.exe 的可執行程式碼。其編譯的過程，如圖 1-4 所示。

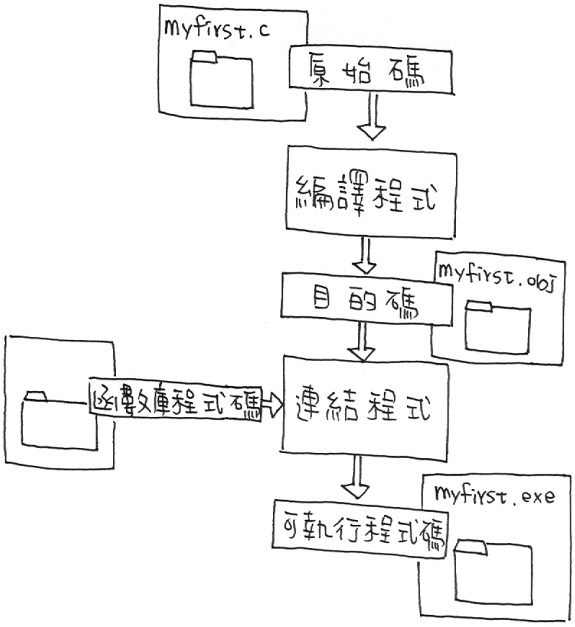

圖 1-4：原始程式碼(myfirst.c)，經由編譯程式編譯成目的碼(myfirst.obj)，再由連結程式變成可執行程式碼(myfirst.exe)

目前 C 的編譯程式有許多，其中有一免費而且好用的 C 編譯程式，那就是 Dev-C++，它除了可以編譯 C 程式(延伸檔名為 .c)以外，也可以編譯 C++程式(延伸檔名為 .cpp)。除此 Dev-C++之外，還有其它的編譯程式，如 Microsoft Visual Studio 2008。在這一系列叢書中，有一書名為「**精彩的指標：燦爛的星星(wonderful pointer: beautiful star)**」，可能有些程式需要使用它來編譯。

這些編譯程式使用的都是整合性的界面環境，熟悉一種之後，另一種很快就可以上手了。有關 Dev-C++編譯程式使用手冊，請參閱附錄 A。

1-5 關鍵字

- 程式設計師(programmer)
- 程式語言(programming language)
- 程式(program)
- 機器語言(machine language)
- 編譯程式(compiler)
- 翻譯者(translator)
- 錯誤(bugs)
- 除錯(debug)
- 原始程式(source code)
- 目的程式(object code)
- 連結程式(linker)
- 指標(pointer)
- 資料結構(data structures)
- 二元搜尋樹(binary search tree)

1-6 上機實習

請選擇一種編譯程式，將下列的程式逐字鍵入，不必管程式內部的細節，看看此程式的輸出結果為何？此處旨在讓大家熟悉一下，往後要用的編譯程式之操作。

```
/* first.c */
#include <stdio.h>
#include <stdlib.h>
int main()
{
    printf("大家一起來學 C 程式語言");
```

```
    printf("\n");
    system("PAUSE");
    return 0;
}
```

1-7 參考文獻

1. Concept of programming language, 8 edition, Robert W. Sebesta, Addison Wesley.
2. http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html