國立屏東大學 資訊工程學系 程式設計(一)

4. Hello World! 您的第一隻c語言程式

- C語言程式開發流程
- Console模式程式架構
- 註解(comment)
- 敘述(Statement)
- printf()函式
- 函式標頭檔(Header File)

許多的程式語言的書籍都會以一個"Hello World!"的範例，做為第一個程式範例。我們也將透過此一"Hello World!"的程式，帶領您瞭解基本的C語言程式架構與語法規則，以及程式設計的基本流程。

4.1 程式開發流程

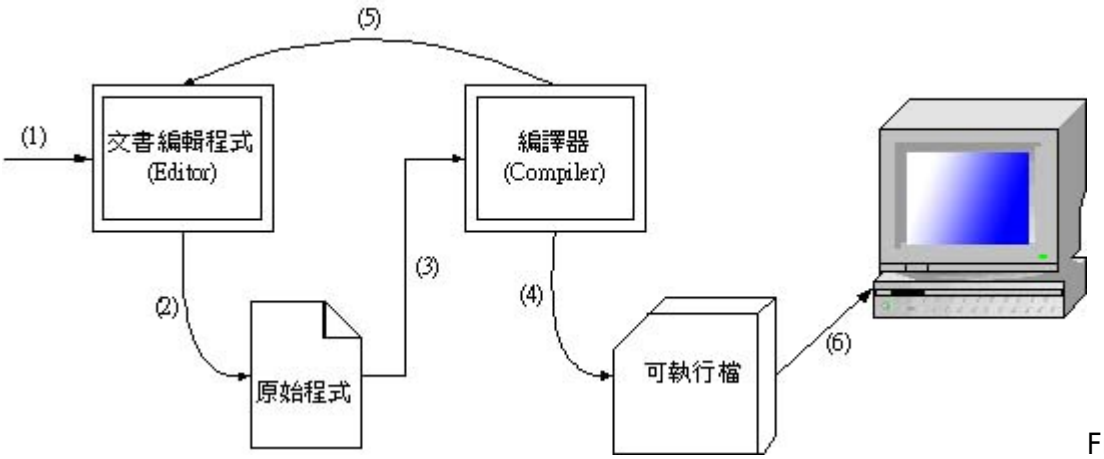

Fig. 1: 典型的程式開發流程

使用C語言開發程式的流程十分簡單，可概分為撰寫原始程式碼(source code)、編譯(compile)原始程式與執行(run/execute)。請參考figure 1，其步驟包含：

1. 以文字檔編輯軟體撰寫原始程式。
2. 完成後，產生副檔名.c的原始程式檔。
3. 將原始程式交由編譯器(Compiler)編譯。
4. 編譯成功後，會產生符合作業平台的可執行檔(例如Windows平台下的EXE檔)。
5. 若是編譯時發生錯誤，則重新回到步驟1進行除錯。
6. 最後就可在作業平台上執行程式。

我們可使用任何一套文字檔編輯軟體(text editor)來進行程式的撰寫，再進行後續的編譯與執行。或者您也可以使用如Dev-C++、Code::Blocks或是Microsoft Visual Studio這一類的IDE(integrated development environment 整合式開發環境)，來進行程式的撰寫、編譯、執行甚至除錯等工作。您可以自行尋找適合或喜好的開發方式，以下我們將以Linux系統為例，說明如何進行程式的開發，請依照下面的說明，來完成您的第一隻C語言程式 – Hello World!

4.2 編寫程式(coding)

由於C語言的原始程式檔案格式為純文字，所以您可以使用任一套文字檔編輯軟體(texteditor)來編寫程式。在Linux/Unix系統下，我們推薦joe或emacs□在Windows系統下，我們推薦使用UltraEdit□PSPad□Crimon Editor□Notepad++或MadEdit等軟體。首先，請使用您偏好的文字檔編輯軟體來編寫一個名為hello.c的檔案，並鍵入以下內容:

```
// <code c 1|h hello.c>
/* This is my first C program */
#include <stdio.h>

int main()
{
     printf("Hello World!\n");
}
// </code>
```

4.3 編譯與執行程式(Compile and Run)

C語言的原始程式檔案只是一般的純文字檔(text file)□而非可執行檔(executable file)□如果沒有經過編譯(compile)是無法執行的。簡單來說，文字檔是使用人類看的懂的編碼方式；執行檔則必須是電腦認識的機器指令，通常是二進位檔(binary file)。所以還需要有一個編譯程式(Compiler)將原始程式檔轉換成電腦認識的二進位檔。要如何編譯C語言的原始程式碼呢？我們可以使用下面的指令來進行編譯：

```
[1:18 user@ws example] cc hello.c
```

若沒有任何的錯誤訊息產生，那麼您已經順利地完成了編譯的動作。如果您看到錯誤訊息，那麼請再仔細檢查一下是否原始程式有輸入錯誤的地方，修改後請再次編譯，直到沒有問題為止□cc與gcc預設會將所欲compile的程式，經編譯無誤後產生一個名為a.out的檔案。檢查您的目錄下是否多了一個名為a.out的檔案？請用以下指令執行這個可執行檔：

```
[1:18 user@ws example] ./a.out
```

您應該可以看到以下的輸出的結果

```
[1:18 user@ws example] ./a.out
Hello World□
[1:18 user@ws example]
```

其實cc是C compiler的縮寫，是內建於在Linux/Unix系統上的C語言編輯器。另外一個常見的編譯器是gcc(全名為GNU operating system C compiler)。在GNU/Linux系統上□cc與gcc其實是相同一個檔案。使用以下指令再次進行編譯與執行，您應該會得到完全一樣的結果：

```
[1:18 user@ws example] gcc hello.c
[1:18 user@ws example] ./a.out
Hello World!
[1:18 user@ws example]
```

您還可以使用編譯器-o參數，來指定所輸出的可執行檔檔名，例如以下的例子，將hello.c編譯成為hello並加以執行：

```
[1:18 user@ws example] gcc hello.c -o hello
[1:18 user@ws example] ./hello
Hello World!
[1:18 user@ws example]
```

4.4 程式碼說明

本節說明前述的hello.c程式碼的內容。

4.4.1 Console Framework與程式進入點

首先這種在console模式下執行的程式，大多具備以下的架構，我們將其稱為Console Framework

h Console Framework

```
int main()
{
 程式碼
}
```

Console模式的程式執行時，電腦系統會將其載入到主記憶體，並從程式當中特定的位置開始一行一行、逐行地依序執行程式碼。這個所謂的特定位置，也就是指程式首先被執行的地方，我們稱之為Entry Point(程式進入點)。事實上，絕大多數的程式語言都有類似的機制以啟動程式的執行，其中Console模式的C語言程式，其entry point是程式中的main函式(function以後會詳細說明)。在前面的Console Framework當中int main()就是在定義這個entry point在int main()的後面緊接著一對大括號{}，電腦會從左大括號開始，一行一行地加以執行，直到遇到右大括號為止。因此，一個簡單的C語言程式，就是將欲由電腦執行的功能，以符合C語言語法規則的方式寫成程式碼，依序 寫在int main()的大括號內。

<note>

Console終端機模式

所謂的console指的是作業系統中一個用以操作的管道，它只支援標準輸入與輸出，也就是只接受來自鍵盤的輸入以及回應在螢幕上的文字輸出；我們可以透過console來操作作業系統，包含下達指令與執行程

式等操作。在Windows系統中的命令列提示字元（Putty）以及在Linux/Unix/MacOS中的terminal都是這一類型的操作環境。

</note>

前述的hello.c程式，就是一個Console模式下的程式，也具備如同Console Framework一樣的程式碼架構。在後續的小節，我們將繼續說明hello.c這個程式的細節。

4.4.2 註解(comment)

在hello.c的開頭處，有以下這一行：

```
/* This is my first C program */
```

其實這一行對於程式的執行沒有任何作用，只是所謂的註解(comment)（在程式語言中，凡是註解的部份，編譯器在進行編譯時都會略過此一部份，因此註解並不會對程式的執行造成任何的影響，其目的只是為您所撰寫的程式做一些說明。通常註解的內容不外乎版本、版權宣告、作者資訊、撰寫日期及程式碼的說明等。在C語言中註解使用的方式有以下兩種方式:

- 以//開頭到行尾的部份，皆視為註解。例如：
  - //註解的使用可以放在每一行的開頭，也可以放在行中其它位置。特別要說明的是，註解有時不一定只是做程式碼的說明，還可以暫時把可能有問題的程式碼加以註解，以便進行程式的偵錯，這在程式設計上是常用的一種除錯的方式。

```
//這是C的註解
int i; //宣告變數i
// int maybe wrong here;
```

- 以/*開頭到*/結尾的部份，皆視為註解。這種方式可將多行的內容都視為註解。例如：
  - 註解可用於單行的註解，如本例中第1行; 也可以用於多行的註解。

```
/*這也是C的註解,可用於單行的註解*/
/*這種方式也可以應用在多行的註解
  對於需要較多說明的時候
  可以使用這種方式*/
```

4.4.3 印出字串

最後要說明的是在hello.c的main函式內，唯一的一行程式碼(code)：

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

像這樣的一行程式碼，我們稱為敘述(statement)，也就是要告訴電腦要幫我們執行的工作為何。在main function中，可以有一行以上的statement，每一行都必須以分號(;)結尾，println(“Hello World!\n”);是一個標準的statement，其目的為輸出“Hello World!字串到螢幕上。

其實printf也是一個function，請思考以下的例子：

```
f(x)=2x+5
```

這是一個數學的函數，f of x，其函數名稱為f，輸入一個參數(parameter)x給它，它會計算2x+5的值並加以傳回，例如f(3)的值為11，f(5)的值為15。對於C語言，printf是一個函式，我們可以把它思考為以下的型式：

```
printf(s)=將s輸出在螢幕上
```

printf在取得輸入的s字串後，就會將其輸出在螢幕上。所謂的字串(string)是字元(character)的組合，並由雙引號標註。例如：

“Hello World\n”	是字串
Hello World\n	缺少雙引號標註，不是字串
“Hello World\n	缺少右方的雙引號標註，不是字串

那麼”Hello World!\n”當中的\n又是什麼呢? 其實很簡單，請您自己修改一下hello.c的程式，將\n拿掉，再編譯與執行看看有何差異，您就可以得到答案了！

4.4.4 函式標頭檔

在C語言中，已預先定義好許多有用的函式(function)，我們可以視需要在程式中選擇使用這些function來完成特定的工作。這些function依功能或性質，被分類存放在不同的函式庫(library)中。C語言也提供許多函式標頭檔(function header file)，其副檔名為.h，用以定義性質或功能相關的function，在程式內使用特定的function前，您必須明確地告訴C語言的compiler，將含有該function定義的header file載入，這個動作就是透過#include這個前置處理器命令(preprocessor directive)，將特定的function header file加以載入。

<note>

Preprocessor前置處理器

所謂的preprocessor是一個電腦程式，在compiler進行編譯前執行。所有以#開頭的程式碼，都會被視為交付給preprocessor的命令(directive)。

</note>

在hello.c中的：

```
#include <stdio.h>
```

就是要求載入stdio.h檔案，其中定義了與標準輸入輸出(standard input/output)相關的函式定義。例如printf()函式就包含在stdio.h的定義當中，所以我們必須先將stdio.h載入，後續compiler才能正確地處

理printf()函式。

課堂練習2

From:
https://junwu.nptu.edu.tw/dokuwiki/ - **Jun Wu的教學網頁**
國立屏東大學資訊工程學系
CSIE, NPTU
Total: 194514

Permanent link:
https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=c:helloworld

Last update: **2021/03/28 11:40**