

程式設計

Ch01. Introduction to Programming

Chuan-Chi Lai 賴傳淇

Department of Communications Engineering
National Chung Cheng University

Spring Semester, 2024

Outline

- ① Programming Language 程式語言
 - Machine Language 機器語言
 - Assembly Language 組合語言
 - High-Level Language 高階語言
- ② C 語言
- ③ 第一支程式－ Hello World
- ④ 基本輸出 printf 函式
- ⑤ 附錄：建立 C 語言開發環境

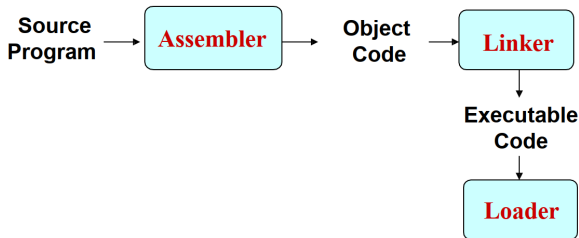
- ① 一種被標準化的，能用來向電腦發出命令的指令。
- ② 讓程式設計師準確地定義電腦所需要使用資料的電腦語言，並精確地定義在不同情況下所應當採取的行動。
- ③ 每一種程式設計語言可以被看作是一套包含語法、詞彙和含義的正式規範，這些規範通常包括：
 - ① 資料和資料結構
 - ② 指令及流程控制
 - ③ 參照機制和重用
 - ④ 設計哲學
- ④ 大多數被廣泛使用或經久不衰的程式語言，擁有負責標準化的組織，經常開會來創造及發布該程式語言的正式定義，並討論擴充或貫徹現有的定義。

Machine Language 機器語言

- 可由 CPU 直接解讀與執行的指令，由二進位數字 (0 與 1) 所構成。
- 優點：
 - 靈活、直接執行和速度快
- 缺點：
 - 人類難以編寫與閱讀。
 - 編寫程式花費的時間往往是實際執行時間的幾十倍或幾百倍。
 - 不同種類的電腦其機器語言是不相容的，按某種電腦的機器指令編制的程式不能在另一種電腦上執行。

Assembly Language 組合語言

- 使用英文縮寫來表示基本運算元。
- 使用組譯器 (Assembler) 將組合語言轉為機器語言。



組合語言執行流程圖

High-Level Language 高階語言

- 通常只需一個敘述 (Statement) 就能達成組合語言的很多行指令。
- 使用許多英文單字和數學符號來撰寫指令。
- 可讀性高且容易學習與撰寫。



高階語言分類

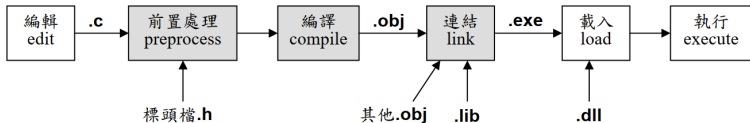
- ① 編譯式語言 (Compiled language) :
 - 使用編譯器 (compiler) 將高階語言轉換為機器語言才可以執行。
 - C, C++, Visual Basic...
- ② 直譯式語言 (Interpreted language) :
 - 使用直譯器 (interpreter) 逐行將高階語言直譯成機器語言並執行。
 - JavaScript, Python, PHP...

C 語言簡介

- 於 1969 年至 1973 年間，爲了移植與開發 UNIX 作業系統，由 Dennis Ritchie 與 Kenneth Lane Thompson，以 B 語言爲基礎，在貝爾實驗室設計、開發出來。
- 半世紀以來，已廣泛用於各種系統軟體與應用軟體的開發。
- 高效、靈活、功能豐富、表達力強和較高的可移植性。
- 影響了眾多後來的程式語言，例如：C++、C#、Java 等。

C 語言編譯

- C 語言須經由編譯器 (Compiler) 與連結器 (Linker) 將程式碼轉換成執行檔 (.exe) 來執行。
 - 前置處理器 (Preprocessor)：在開始編譯前自動執行，依據程式碼中 # 所標示的指示 (Preprocessor directives)，進行代換或插入等動作。
 - 編譯器：把原始碼 (Source code) 變成機械碼 (Machine code) 的目的檔 (Object file)。
 - 連結器：將一個或多個目的檔 (.obj) 與靜態程式庫檔 (.lib) 連結，產生可執行檔 (.exe)。
 - 載入器 (Loader)：將可執行檔載入記憶體，並與動態程式庫檔 (.dll) 連結。動態程式庫可減少執行檔所佔的硬碟和記憶體空間。



C 語言編譯流程

C 語言架構

- C 語言的架構是由一個個模組所組成，包含：
 - ① 前端處理程式
 - ② 全域變數
 - ③ 主程式模組
 - ④ 函數式模組
- C 語言的概念是以函數 (Function) 為主，連主程式 `main()` 都是一個函數。

C 語言架構

前端處理程式

```
#include  標頭檔  //引進所需要之庫存函數
:
#define    識別字或常數  //增加可讀性
:
全域變數宣告
:
```

主程式模組,
只能有一個

```
main( ) //程式進入點,由此開始執行
{
    主程式區域變數宣告
    敘述...
}
```

函數模組,
可以有很多個

```
functions( )
參數的宣告
{
    函數區域變數宣告
    敘述...
}
```

第一支程式－ Hello World

```
1  /**
2   * @file HelloWorld.c
3   * @author Trent Lai
4   */
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main(void)  //主程式
9  {
10     printf("Hello, World!\n");  //印出文字
11     return 0;    //結束程式
12 }
```

Hello World 範例程式碼

第一支程式－ Hello World

註解 (Comment)：不會被電腦/編譯器解讀與執行，通常用於說明程式碼，使程式更好閱讀。在開頭也可以寫註解來註明檔名和此檔案用途。

- 使用//作為單行註解的開頭。
- 多行註解使用/* 作為開頭，*/作為結尾。

```
1  /**
2   * @file HelloWorld.c
3   * @author Trent Lai
4   */
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main(void) //主程式
9  {
10     printf("Hello, World!\n"); //印出文字
11     return 0; //結束程式
12 }
```

第一支程式－ Hello World

#include 指令：用於引入函式庫標頭檔。在未來的課程中，我們會認識許多函式庫。

- `stdio.h` 叫做「標準輸入輸出標頭檔 (standard input/output header)」。
- `stdlib.h` 則是「通用函式庫」的標頭檔。

```
1  /**
2   * @file HelloWorld.c
3   * @author Trent Lai
4   */
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main(void) //主程式
9  {
10     printf("Hello, World!\n"); //印出文字
11     return 0; //結束程式
12 }
```

第一支程式－ Hello World

主程式 (main function)：控制這個程式的開始與結束，程式以 main 函式做為運行的起點，並由 main 函式結束程式。

```
1  /**
2   * @file HelloWorld.c
3   * @author Trent Lai
4   */
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main(void) //主程式
9  {
10     printf("Hello, World!\n"); //印出文字
11     return 0; //結束程式
12 }
```

第一支程式－ Hello World

回傳值 (return value)：在函式結束時，必須回傳一個值，回傳一個值也同時代表該函式的運行結束。函式的回傳值型態會寫在函式名稱前面，在函式回傳時須回傳與「回傳值型態」相同型態的數值。

- main 函式的回傳值型態只可以是 int(整數) 並且回傳 0 是固定用法，若沒有特殊需求，一律回傳 0。
- 在 main 函式中，若省略回傳敘述 (return)，程式運行到 main 的右大括號時會自動回傳 0。

```
1  /**
2   * @file HelloWorld.c
3   * @author Trent Lai
4   */
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main(void) //主程式
9  {
10     printf("Hello, World!\n"); //印出文字
11     return 0; //結束程式
12 }
```


第一支程式－ Hello World

區塊 (Block)：以大括號 (brace) 包住的程式碼稱為一個區塊。

- 爲了美觀與方便辨識程式結構，區塊中的任何敘述式都必須往內縮一個縮排。
- 一個縮排 = 1 個 tab 鍵或 4 個空白鍵

```
1  /**
2   * @file HelloWorld.c
3   * @author Trent Lai
4   */
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main(void) //主程式
9  {
10     printf("Hello, World!\n"); //印出文字
11     return 0; //結束程式
12 }
```

第一支程式－ Hello World

敘述式 (Statement)：指使電腦執行 1 個動作 (action)。

- 敘述式以分號「;」 (semicolon) 作為結束符號 (statement terminator)。

```
1  /**
2   * @file HelloWorld.c
3   * @author Trent Lai
4   */
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main(void) //主程式
9  {
10     printf("Hello, World!\n"); //印出文字
11     return 0; //結束程式
12 }
```

基本輸出 printf 函式

用來做格式化輸出 (print format)

printf 函式

```
1 printf("Hello World!\n"); //印出文字
```

- printf 是 `<stdio.h>` 所提供的函式，使用時必須引入 `<stdio.h>` (`#include <stdio.h>`)。
- 函式名稱後方的小括號內的東西稱作引數 (argument)，此函式的引數為 1 個字串 (string) “Hello World!\n”，此敘述表示將 “Hello World\n” 輸出印到螢幕上。

基本輸出 printf 函式

```
1 "This is a string."  
2 'a' // This is a character
```

- 使用雙引號包住的文字叫做字串 (string) 或稱字元陣列 (character array)。
- 字串由多個字元 (character) 組成，若要表示字元，則使用單引號包住。

跳脫序列 (Escape Sequences)

- 跳脫序列用來表示一些特殊的字元，如換行、tab 等字元。
- 如範例 `printf("Hello World\n")`；中的 `\n`，在執行的時候 `\n` 並沒有顯示出來，那是因為它是個表示換行的跳脫序列。
- 跳脫序列的反斜線 (`\`) 稱為跳脫字元 (escape character)。當出現跳脫字元時，編譯器會尋找下一個字元，與跳脫字元組成跳脫序列。

常見的跳脫序列

//重要的

'\n' //換行
'\t' //水平 tab 鍵
'\"' //雙引號
'\"' //單引號
'\\' //反斜線

//補充的

'\a' //嗶聲
'\b' //退格鍵
'\r' //回行首
'\v' //垂直 tab 鍵
'\0' //字串結尾 (ASCII 碼 0)

非跳脫序列的特殊字元：「%」

- 「%」(百分比符號) 跟格式指定字 (format string) 的符號重複，在 printf 要表示「%」時須使用%%。
- 例如下方敘述句只會印出 1 個「%」。

```
printf("本門課的程式作業佔比為30%%\n");
```


附錄：建立 C 語言開發環境

使用 VSCode 與 GCC 作為開發環境

Visual Studio Code

- 一款由微軟開發且跨平台的免費原始碼編輯器。
- 下載網址：<https://code.visualstudio.com/download>

GCC (GNU C Compiler)

- C/C++ 的編譯器
- 許多作業系統，包括許多類 UNIX 系統，如 Linux 及 BSD 家族都採用 GCC 作為標準編譯器
- Windows 建議使用 MinGW (Minimalist GNU for Windows，包含 GCC 在 Windows OS 下的版本)

安裝 MinGW64

- 下載網址：<https://sourceforge.net/projects/mingw-w64/files/mingw-w64/>
- 選擇最新版的 x86_64-posix-sjlj，並下載、解壓縮至固定的位置 (例如 C:\)
- 位置路徑中建議不要有中文與空格。

建立 C 語言開發環境：安裝 MinGW64

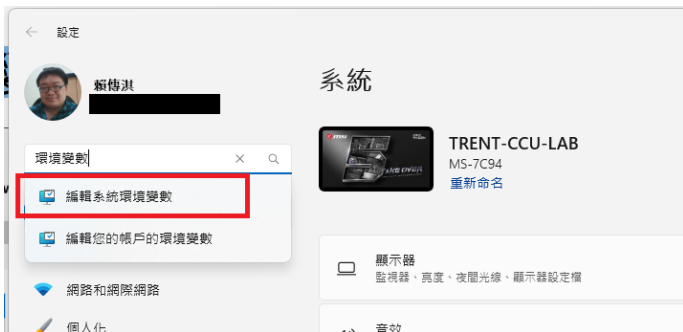
在安裝 MinGW 時選的路徑底下的 bin 資料夾裡可以找到

- gcc.exe：編譯 C 語言檔案用的編譯器。
- g++.exe：C++ 的編譯器。
- gdb.exe：C/C++ 的 debugger。



建立 C 語言開發環境：安裝 MinGW64

在 windows 設定視窗的搜尋欄中，搜尋“環境變數”，並選擇“編輯系統環境變數”。



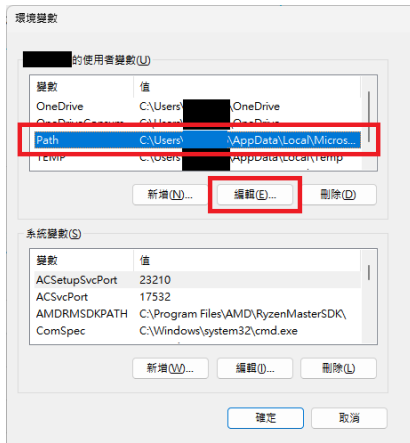
建立 C 語言開發環境：安裝 MinGW64

選擇“環境變數(N)”。



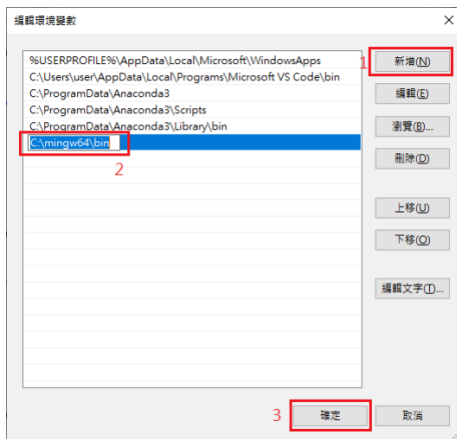
建立 C 語言開發環境：安裝 MinGW64

在使用者變數清單選擇“Path”後，點選“編輯”。



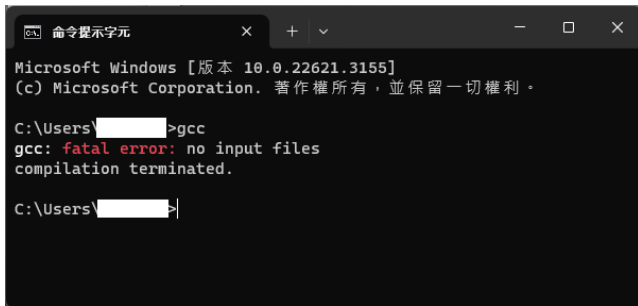
建立 C 語言開發環境：安裝 MinGW64

- 1 點選“新增”
- 2 在新欄位輸入 gcc.exe 的所在路徑
- 3 然後按下“確定”



建立 C 語言開發環境：安裝 MinGW64

開啓命令提示字元，輸入 gcc，若沒有出現「不是內部或外部命令、可執行的程式或批次檔。」就代表安裝成功。



```
Microsoft Windows [版本 10.0.22621.3155]  
(c) Microsoft Corporation. 著作權所有，並保留一切權利。  
  
C:\Users\>gcc  
gcc: fatal error: no input files  
compilation terminated.  
  
C:\Users\>
```

gcc 編譯 C 語言檔案的 CMD 指令

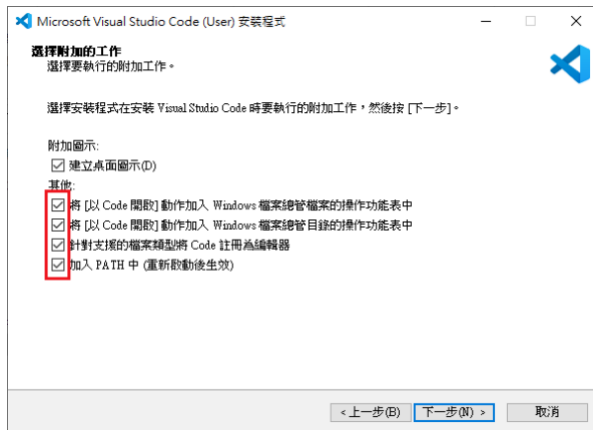
gcc 原始碼檔名.c -o 欲輸出的執行檔名.exe

安裝 VSCode

下載網址：<https://code.visualstudio.com/download>

建立 C 語言開發環境：安裝 VSCode

依照指示安裝，建議下圖選項全部打勾。



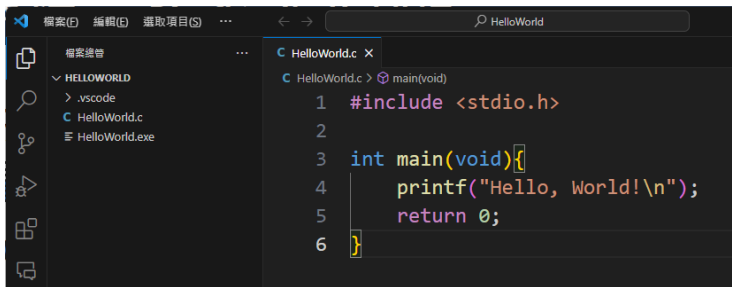
建立 C 語言開發環境：安裝 VSCode

於延伸模組搜尋 C/C++ Extension Pack 並安裝。



建立 C 語言開發環境：測試 VSCode

- 1 於上方工具列選單選擇“檔案” > “開啓資料夾”，選擇一個資料夾作為之後編寫程式的位置。
- 2 於左欄檔案總管右鍵選擇新增檔案，取名為“HelloWorld.c”
- 3 在檔案中編寫以下程式碼並存檔。

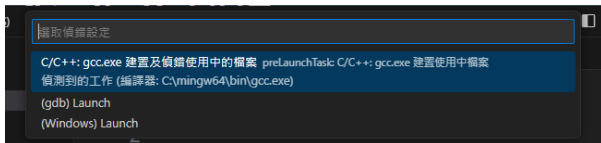
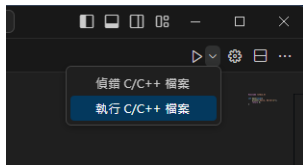


The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a folder named 'HELLOWORLD' containing files '.vscode', 'HelloWorld.c', and 'HelloWorld.exe'. The main editor area displays the 'HelloWorld.c' file with the following C code:

```
1 #include <stdio.h>
2
3 int main(void){
4     printf("Hello, World!\n");
5     return 0;
6 }
```

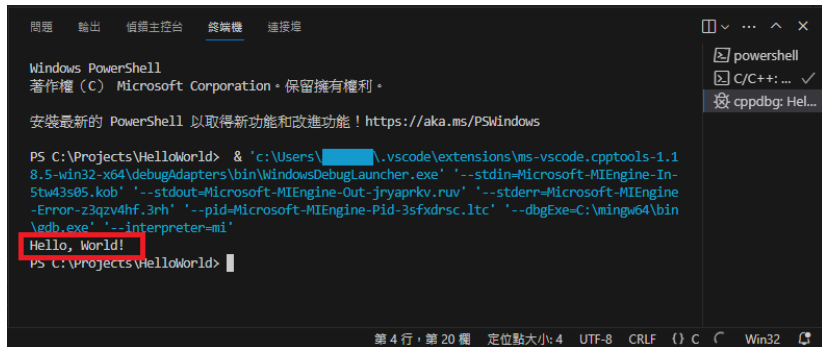
建立 C 語言開發環境：測試 VSCode

點選右上方的執行選項，選擇執行 C/C++ 檔案後，
偵錯設定選取 mingw64 的選項。



建立 C 語言開發環境：測試 VSCode

如果在終端機 (terminal) 看到 Hello, World! 即代表編譯並執行成功。



The screenshot shows a VS Code terminal window with the following content:

```
Windows PowerShell
著作權 (C) Microsoft Corporation。保留擁有權利。

安裝最新的 PowerShell 以取得新功能和改進功能！https://aka.ms/PSWindows

PS C:\Projects\HelloWorld> & 'c:\Users\██████████\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-5tw43s05.kob' '--stdout=Microsoft-MIEngine-Out-jryaprkv.ruv' '--stderr=Microsoft-MIEngine-Error-z3qzv4hf.3rh' '--pid=Microsoft-MIEngine-Pid-3sfxdrsc.ltc' '--dbgExe=C:\mingw64\bin\gdb.exe' '--interpreter=mi'
Hello, World!
PS C:\Projects\HelloWorld> |
```

The terminal window has tabs for '問題', '輸出', '偵錯主控台', '終端機', and '連接埠'. The '終端機' tab is active. On the right side, there is a list of debuggers: 'powershell', 'C/C++: ...', and 'cppdbg: Hel...'. The status bar at the bottom shows '第 4 行, 第 20 欄', '定位點大小: 4', 'UTF-8', 'CRLF', '{ } C', 'Win32', and a search icon.