**20221004安裝並使用DEV-C++來設計檔案IO程式**

1. 下載[Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup.exe](http://lfwiki.kmvs.km.edu.tw/tree/Dev-Cpp%205.11%20TDM-GCC%204.9.2%20Setup.exe)(約49MB，**如系統防護軟體啟用中可能無法下載**)並執行，以裝妥DEV-C++ C語言編譯開發環境(SDK)。

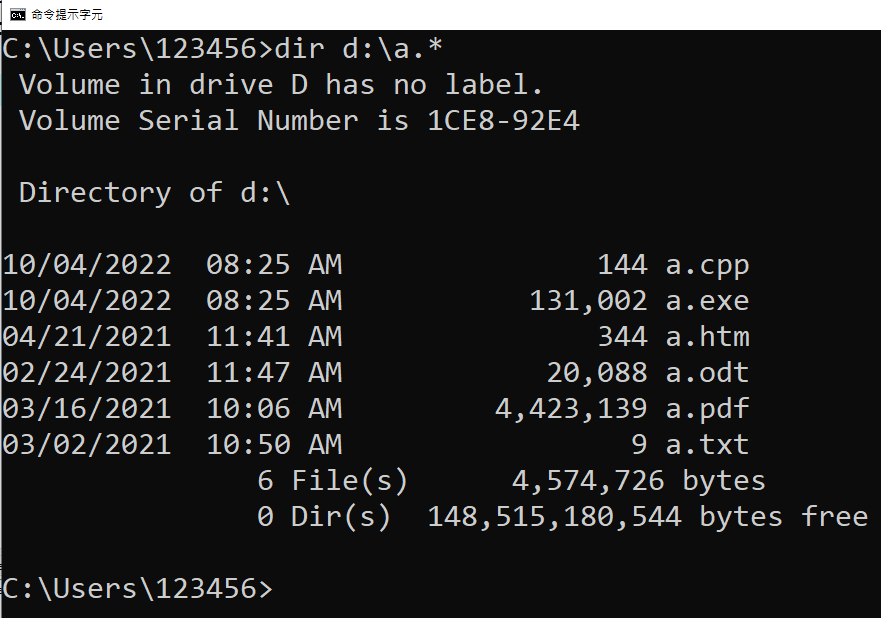
(1a)執行DEV-C++並按CTRL+N 編寫如下程式碼並存為在**d:\a.cpp**

|  |
| --- |
| #include <stdio.h>  int main(void) {  int a=11,b=22; //宣告二個int整數變數a和b，其初值分別為11和22  int c=a+b; //宣告1個int整數變數c，其初值為a+b  printf("a=%d b=%d c=%d\n",a,b,c); //依給定的格式字串在給定的位置輸出給定的變數值  return (0); //傳回0值表示正常結束  } |

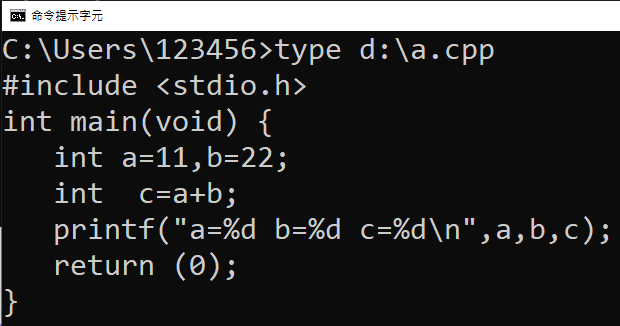
(1b)按F11 編譯產生執行檔d:\a.exe並執行之，會於新視窗上輸出結果如下(按WIN+SHIFT+S可截圖選定的區域，CTRL+V可貼入截圖)

|  |
| --- |
|  |

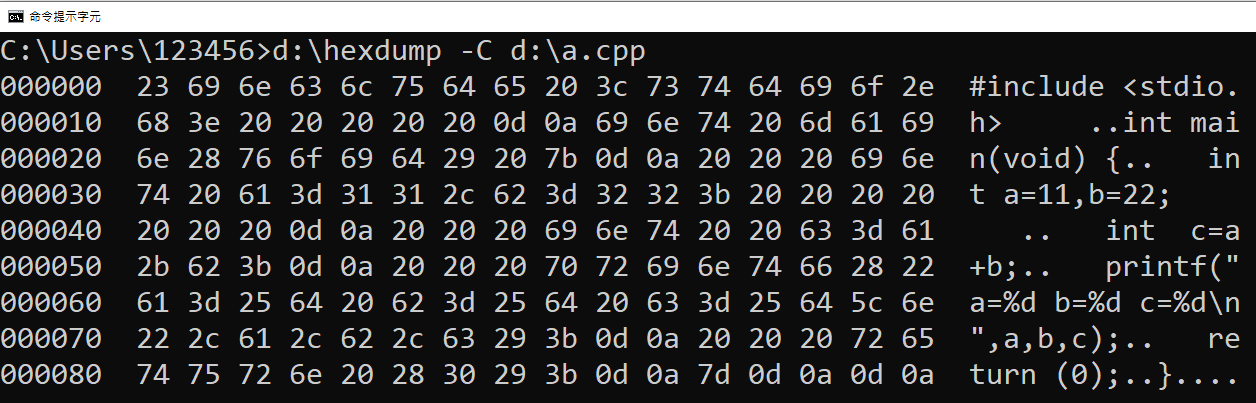
(1c)按WIN鍵/執行cmd(命令提示字元) 並輸入dir d:\a.\* 指令，查得如上新產生的二個檔案如下：



(1d)在命令提示字元下輸 type d:\a.cpp指令，會顯示(1a)的程式碼文字



(1e)在CMD下輸入 d:\hexdump -C d:\a.cpp指令，會把給定檔案的每一位元組以16進位數傾印並列出對應的可印字元。



**20221012C語言檔案IO程式(續)**

(2a)執行DEV-C++並按CTRL+N 編寫如下程式碼並存為在**d:\b.cpp**

|  |
| --- |
| #include <stdio.h>  int main(void) {  int a,b,c; //宣告3個int整數變數a,b,c  printf("a=");  scanf("%d",&a); //自鍵盤取得1個整數的輸入值給a  printf("b=");  scanf("%d",&b); //自鍵盤取得1個整數的輸入值給b  c=a+b;  printf("a=%d b=%d c=%d\n",a,b,c); //依給定的格式字串在給定的位置輸出給定的變數值  return (0); //傳回0值表示正常結束  } |

(2b)按F11 會編譯d:\b.EXE並執行之，  
 結果如下

|  |
| --- |
|  |

(2c)在cmd下輸入dir d:\b.\* 指令結果會列出剛產生的2個檔案，b.cpp佔204 BYTE及b.EXE 佔131176BYTE。

**20221019C語言檔案IO程式(續)**

(3a)在DEV-C++ 編寫如下程式碼並存為**d:\fwrite.cpp**

|  |
| --- |
| **#include <stdio.h>**  **int main( ) {**  **//宣告一檔案指標變數outfile來存放以二元寫入模式打開的檔案a.bin的檔案指標**  **FILE \*outfile = fopen("a.bin", "wb");**  **int a=11,b=22; //宣告二個int整數變數a和b，其初值分別為11和22**  **int c=a+b; //宣告1個int整數變數c，其初值為a+b**  **fwrite(&a, sizeof(int), 1, outfile); //寫入a位址所指的1個整數的內容到outfile**  **fwrite(&b, sizeof(int), 1, outfile);**  **fwrite(&c, sizeof(int), 1, outfile);**  **fclose(outfile); //關閉檔案 outfile**  **printf("write a=%3d b=%3d c=%3d to a.bin ok\n",a,b,c);**  **//依給定的格式字串在給定的位置輸出給定的變數值**  **return (0); //傳回0值表示正常結束**  **}** |

(3b)按F11 編譯並執行fwrite .cpp，會於新視窗上輸出結果如下

|  |
| --- |
|  |

**20221025C語言檔案IO程式(續)**

(3c)在cmd(命令提示字元)輸入如下指令d:\**hexdump.exe -C** [**d:\a.bin**](file:///C:\D:\a.bin)**用16進制數顯示a.bin**

|  |
| --- |
| C:\Users\123456>d:\hexdump.exe -C d:\a.bin  000000 0b 00 00 00 16 00 00 00 21 00 00 00 ........!... |

註1：一個int型態的整數變數佔用4個位元組的空間。

註2：**在命令提示字元視窗選住一段文字後按下ENTER鍵即可複製選取的文字。**

註3：將一個多位數的低位放在較小的位址處，高位放在較大的位址處，則稱小端序(little-endian)； 反之則稱大端序(big-endian),因如上首四個位元組「0b 00 00 00」是表示16進位數(00 00 00 0b)，故a.bin內的整數是採小端序。

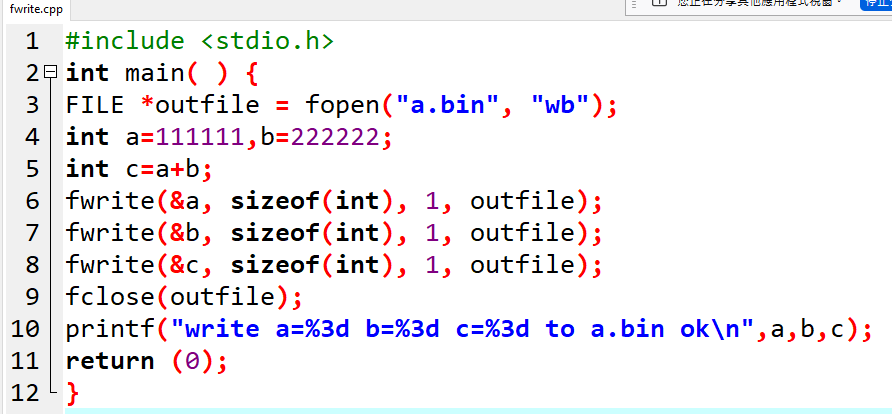
(4a)在DEV-C++ 編寫如下程式並存為**d:\fread.cpp**

|  |
| --- |
| **#include <stdio.h>**  **int main(void) {**  **//宣告一檔案指標變數infile來存放以二元讀取模式打開的檔案a.bin的檔案指標**  **FILE \*infile = fopen("a.bin", "rb");**  **int a,b,c;**  **fread(&a, sizeof( int ), 1, infile); //自infile讀取1個整數給a**  **fread(&b, sizeof( int ), 1, infile);**  **fread(&c, sizeof( int ), 1, infile);**  **printf("Get a=%d b=%d c=%d from a.bin ok\n",a,b,c);**  **printf("size of a.bin = %d bytes.\n\n",ftell (infile)) ; //以ftell函數傳回目前指標位址**  **fclose(infile);**  **return (0);**  **}** |

(4b)按F11 編譯fread.cpp產生執行檔fread.exe並執行，結果如下

|  |
| --- |
|  |

(5a) 編修fwrite.cpp 來寫入111111，222222，333333到a.bin成功





(5b) 同(4b)編譯fread.cpp並執行，有順利讀出(5a)所寫入的3個數字。

