

【課外補充章】

# 函式指標與視窗設計

講師: 李根逸 (Ken-Yi Lee), E-mail: [feis.tw@gmail.com](mailto:feis.tw@gmail.com)



# 課程大綱

---

## ■ 使用內建的排序函式

- ▶ 函式指標
- ▶ 使用 C 標準函式庫的 `qsort` 函式排序
- ▶ 使用 C++ STL 的 `sort` 函式排序

## ■ **Windows** 視窗程式設計

---

# 使用內建的排序函式

---

# 函式指標 (1)

---

- 之前我們提過，陣列名稱本身可以轉成一個指向陣列開頭元素的指標。除此之外，函式名稱本身也是一個指向函式位址的指標
  - ▶ 函式要在電腦上執行必須先載入記憶體，因此也會有記憶體位址
- 函式指標變數宣告：
  - ▶ 回傳值型態 (\*變數名稱)(參數型態 參數, ...);
    - `int (*f)(int, int);`
    - 表示 `f` 是一個函式指標變數，存放一個有兩個 `int` 參數並回傳 `int` 的函式位址
- 呼叫函式指標變數指向的函式
  - ▶ 變數名稱 (參數, ...);

例如： `f(3, 4);`

# 函式指標 (2)

---

```
#include <stdio.h>
#include <stdlib.h>
int add(int a, int b) {
    return a + b;
}

int main() {
    int (*f)(int, int) = add;
    printf("%d", f(3, 4));
    system("pause");
    return 0;
}
```

# 函式指標 (3)

---

## ■ 常見用途：

- ▶ 將函式作為參數傳入函式 (參照 `acc.cpp`)
- ▶ 將函式放在陣列中 (參照 `calc.cpp`)

# 使用 C 標準函式庫的排序

- C 在 `<stdlib.h>` 內建了 `qsort` 函式來排序：

```
void qsort (  
    void * base,  
    size_t num,  
    size_t size,  
    int ( * comparator ) (const void *,const void *)  
);
```

**base:** 陣列開頭位址  
**num:** 元素個數  
**size:** 每個元素的大小 (bytes)  
**comparator:** 自訂比較函式

例如：

```
#include <stdlib.h>  
int cmp(const void *a, const void *b) {  
    return *(int *)a - *(int *)b;  
}  
int main() {  
    int data[10];  
    qsort(data, 10, sizeof(int), cmp);  
    return 0;  
}
```

當 **a** 要出現在 **b** 之前回傳負數  
當 **a** 要出現在 **b** 之後回傳正數  
當 **a** 跟 **b** 一樣時回傳 0

# 《練習》 數據統計

---

- 試寫一程式 (**stat**)，讓使用者輸入七個數字後，顯示最大值、最小值與中位數
  - ▶ 請使用內建的排序函式
- 試寫一程式 (**reverse**)，讓使用者輸入七個數字後，由大到小顯示
- 試寫一程式 (**split**)，讓使用者輸入七個數字後，先由小到大顯示奇數，再由大到小顯示偶數

範例輸入： 1 2 3 4 5 6 7

範例輸出： 1 3 5 7 6 4 2



# 使用 C++ STL 函式庫的排序

- **STL (Standard Template Library)** 是 C++ 內建的函式庫。有別於 C 標準函式庫，**STL** 使用 C++ 的語法特色，可以更深度簡化程式碼。
- `std::sort` 是 **STL** 內建在 `<algorithm>` 內的排序用函式

▶ 函式呼叫：

■ `sort`(第一個元素位址, 最後一個元素結束位址, 自訂比較函式)

例如：

```
#include <algorithm>
bool cmp(int a, int b) { return a <= b; }
int main() {
    int data[10];
    std::sort(&data[0], &data[10], cmp);
    return 0;
}
```

與 C 的不同？

---

# WINDOWS 視窗程式設計

---

# Win32 API

---

- **Microsoft Windows** 提供了 **Win32** 的 **API** 讓程式設計師可以快速撰寫視窗程式
  - ▶ `#include <windows.h>`
- 資料查詢: **MSDN**
  - ▶ <http://goo.gl/BT4FB>
  - ▶ 點擊 [Reference > Functions in Alphabetical Order]
- **WinMain** 函式:
  - ▶ Win32 API 內定的程式進入點 (取代原本的 main)

# 我們會用到的函式 (1)

---

## ■ 函式清單：（請試著用 **MSDN** 查詢細節）

### ▶ WinMain: 自訂的主函式

- <http://msdn.microsoft.com/en-us/library/ms633559%28v=vs.85%29.aspx>

### ▶ MessageBox: 對話框

- <http://msdn.microsoft.com/en-us/library/ms645505%28v=vs.85%29.aspx>

### ▶ RegisterClass: 註冊視窗類別

- <http://msdn.microsoft.com/en-us/library/ms633586%28v=vs.85%29>

### ▶ CreateWindow: 產生視窗

- <http://msdn.microsoft.com/en-us/library/ms632679%28v=vs.85%29>

# 《範例》 **Hello world**

---

- 試寫一程式 (**helloworld**)，產生一對話框 (**MessageBox**) 來顯示 “**Hello world!**”

# 讓 Dev C++ 不要出現主控台

---

- **Dev C++** 預設程式會出現主控台視窗（命令列模式），要關閉的話需要改變下列設定：
  - ▶ 「Tools > Compiler Options > Settings > Linker > Do not create a console window > Yes」

# 《範例》 產生視窗

---

## ■ 試寫一程式顯示一視窗 (**window**)

- ▶ 設定 WNDCLASS 結構
- ▶ 設定 WINTPROC 自訂函式
- ▶ 呼叫 RegisterClass 註冊 WNDCLASS 結構為自訂類別
- ▶ 呼叫 CreateWindow() 產生自訂類別的視窗
- ▶ 呼叫 ShowWindow() 顯示視窗畫面
- ▶ 呼叫 UpdateWindow() 更新視窗畫面
- ▶ 不停呼叫 GetMessage() 取得指令訊息，執行每個指定訊息，直到收到 WM\_QUIT 結束

# 《範例》 加法器

---

- 試寫一程式 (**calculator**)，讓使用者輸入兩個整數後，顯示兩個整數和

- ▶ 內建的視窗類別：

- "static" (文字)
- "button" (按鈕)
- "edit" (輸入框)

HIWORD (wParam) : 取得動作  
LOWORD (wParam) : 取得對象

- ▶ 視窗文字：

- GetWindowText: 取得視窗文字
- SetWindowText: 設定視窗文字

- ▶ 標準函式庫中文字處理函式：

- sprintf: 印出成字串
- atoi: 將字串轉為整數