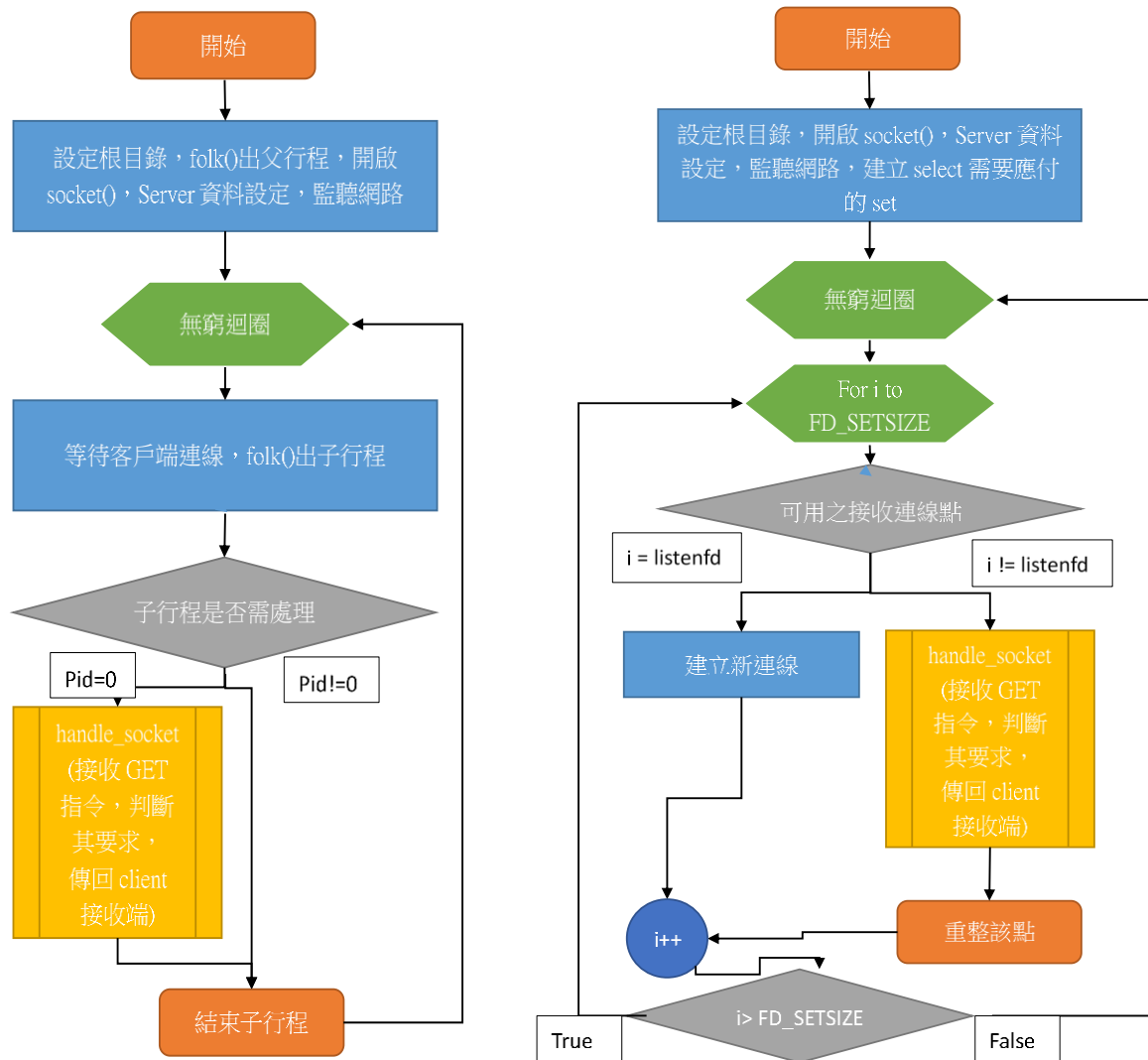


網路程式設計作業一 簡易 Web Server

數學三 405210011 黃伯丞

(1) 程式流程圖



(a) Folk 版本的 web server (左圖)

(b) Select 版本的 web server (右圖)

(2) 實作技巧描述

本次作業中需要撰寫 web server，並且可以藉由瀏覽器 (client 端) 傳送指令執行開啟網頁的動作。首先介紹兩組程式碼的共通部分，一開始設置根目錄做為取檔案的位置，然後開啟網路 socket，設置 sever 位址等資訊 (因為沒有 port 80 的權限所以設為 8000)，接下來監聽網路。現在是這兩支程式不同的地方，folk 的部分要避免 zombie 的情況發生，所以先建立出一個父 (parent) 行

程，這個行程會持續運作直到 `timeout` 或直接終止，一旦接到 `client` 的連線要求就 `fork` 出子行程來，進入判讀指令的步驟，再來介紹 `select` 的部分，開啟一個 `set` 用來存放多個 `sockets` 讓 `select` 來檢查，在收聽的目標 `socket` 與 `client` 端建立連線，之後 `select` 巡視一番，進入判讀指令的步驟，完成後重置目標 `socket`。

接下來是函式 `handle_socket` 的部分，讀取瀏覽器 (`client` 端)所傳送之指令 `GET`，將檔名擷取出來，如果有檔案在資料夾中就寫入傳送回去，沒有的話寫入 "Failed to open file"。

(3) 重要的資料結構

1. `sockaddr_in` (存放 `address`)

```
struct sockaddr_in{
    short sin_family;
    unsigned short sin_port;
    struct in_addr sin_addr;
    char sin_zero[8];
};
```

參數

`sin_family` 位址 (必須是 `AF_INET`)。

`sin_port` IP 通訊埠。

`sin_addr` IP 位址。

`sin_zero` 使結構做為具有相同大小的邊框距離 `SOCKADDR`。

2. `fd_set` (存放多個 `sockets`)

```
typedef struct
{
#ifdef __USE_XOPEN
    __fd_mask fds_bits[__FD_SETSIZE / __NFDBITS];
#define __FDS_BITS(set)((set)->fds_bits)
#else
    __fd_mask __fds_bits[__FD_SETSIZE / __NFDBITS];
#define __FDS_BITS(set)((set)->__fds_bits)
#endif
} fd_set;
```

(4) 額外功能

在瀏覽器輸入網址時，後面所接的檔案不只可以讀 `.html` 檔，還可以將圖片檔、文字檔讓 `client` 讀取。