

# 網路程式設計 作業一報告

403410102 劉怡廷

## 一、作業目錄：

```
.  
|--lib  
|--hw1                //主要的作業資料夾。  
|   |-- Makefile      //產生執行檔的指令集。  
|   |-- sigchldwaitpid.c //修正 fork 版本會產生 zombie 的程式，並能夠正確  
|   |                  //處理 child process 的中止訊號。  
|   |-- tcpservfork.c  //fork 版本的 tcp server  
|   |-- tcpservselect.c //select 版本的 tcp server  
|--config.h  
|--Make.defines  
|--libunp.a
```

## 二、實作技巧描述：

程式碼中的 `response[]` 存放網頁的相關訊息，`servaddr` 變數設定 socket address 結構，之後進行 Bind 和 Listen，接下來程式碼隨版本不同而有所差異：

### 1. Fork 版本：

捕捉訊號 SIGCHLD，當該訊號發生執行 `sig_chld` 函數(`sigchldwaitpid.c` 中)使 child 不變為 zombie 程式，接下來 listening socket，一旦 accept 後，fork 出一個子程序，子程序關閉 listening socket 並將 `response[]` 中網頁訊息寫入 `connfd`，寫入後用 `shutdown` 函數告知 `connfd` server 不再寫入與讀取，最後離開；父程序關閉 `connfd` 並繼續監聽客戶程序的連接。

檢查 zombie 的方法 -- terminal 中輸入以下指令

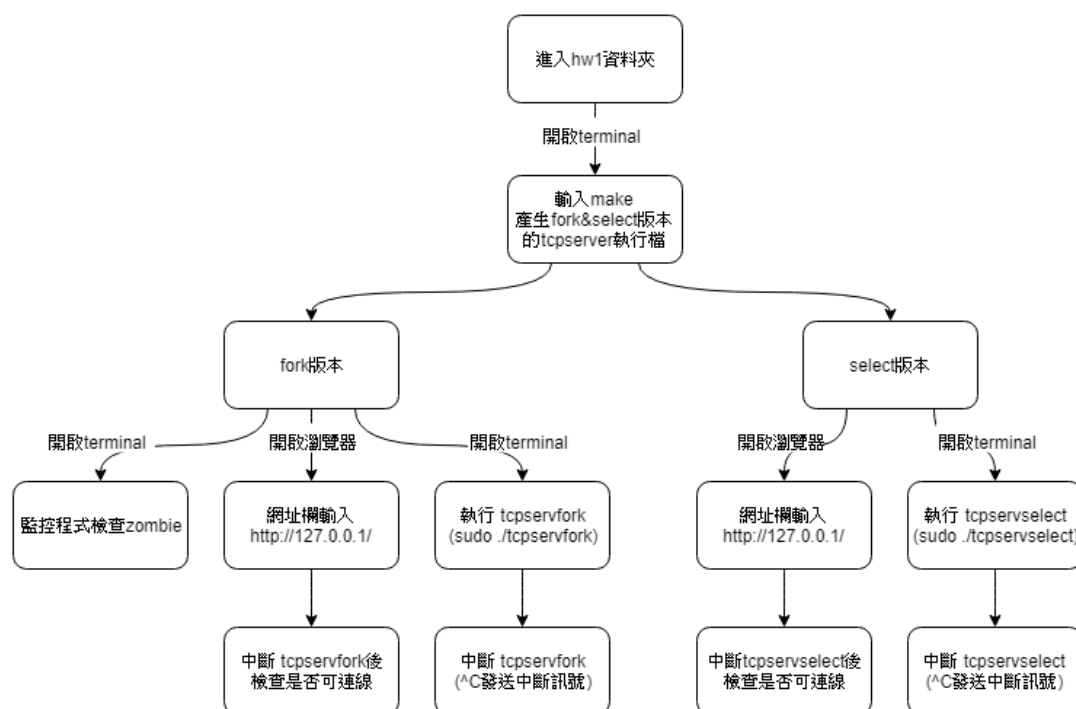
```
ps -ef  
watch -n 0.1 ps -t pts/4 -o pid,ppid,tty,stat,args,wchan
```

### 2. Select 版本：

等待戶建立連接或者是收到資料、FIN、RST 訊號，`rset` 為準備好讀的 set，並將 `listenfd` 放入這個 set 當中，如果 `listenfd` 實際上變為可讀，便呼叫 `Accept` 建立連線並更新 `client[]` 資料結構。

之後輪流 check 每個現有 client 連接並發送 `response[]` 中的網頁訊息，發送後用 `shutdown` 函數告知 `connfd` server 不再寫入與讀取，並更新 `client[]=-1` 表示離開連線

### 三、執行流程圖：



### 四、重要的資料結構

#### A. sockaddr\_in

長度	AF_INET
16 位 port number	
32 位 IPv4 address	
(未使用)	

#### B. tcpserverselect 中重要的資料結構：

1. client[]：client[]紀錄 client 的 fd 號碼，當客戶中斷連接時設置為-1。

[0]	[1]	[2]	...	[FD_SETSIZE-1]
4	5	-1	...	-1

2. rset：fd0、fd1、fd2、fd3 分別為 sdtin、sdtout、sdtout、listion socket 的 fd 號碼，此時變數 fdmaxfd=3，每當有新的 client 加入，變數 maxfd=maxfd+1，然後設置 fd(maxfd)=1，當某客戶中斷連線時，fd(該客戶 fd)設為 0。

fd0	fd1	fd2	fd3	fd4	fd5
0	0	0	1	1	1

上述兩個資料結構描述伺服器開啟初期有兩個客戶程序連接時的情形。

## 五、改進的地方：

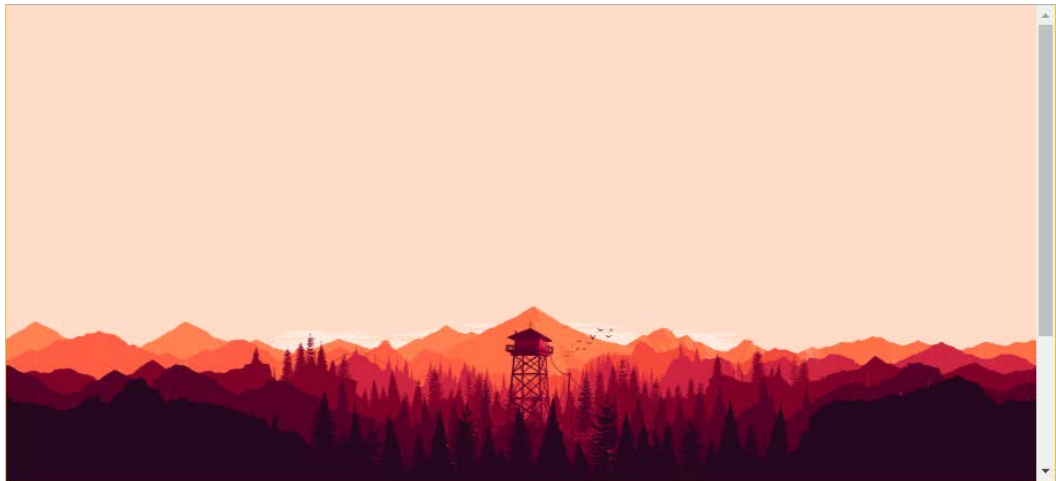
1. 關掉 server 再重開的話會產生 "bind error: Address already in use" 錯誤訊息，一段時間內無法啟動 server。

原因：程式結束後，雖然已經正常把 socket 給 close() 了，但是 kernel 並不會釋放這個 TCP 的連線，它的狀態會保持在 TIME\_WAIT 狀態。

改良方法：在 bind() 之前加上這段程式碼：

```
if((setsockopt(listenfd,SOL_SOCKET,SO_REUSEADDR,&on,sizeof(on)))<0)
{
    perror("setsockopt failed");
    exit(EXIT_FAILURE);
}
```

2. 較美觀的網站，隨著滾輪往下滾會有角度改變的效果：



滾輪往下後：

