Multiplexing Web Server

The Assignment 2 report from NTU102-1 Systems Programming (Cloud Computing Program) course

by NTU Michael Hsu

執行環境

- Public Domain Workstation Lab (R217).
- Linux linux19 3.10-3-amd64 #1 SMP Debian 3.10.11-1 (2013-09-10) x86_64
- gcc version 4.8.1 (Debian 4.8.1-10)

如何執行

1. GCC compiler `makefile`

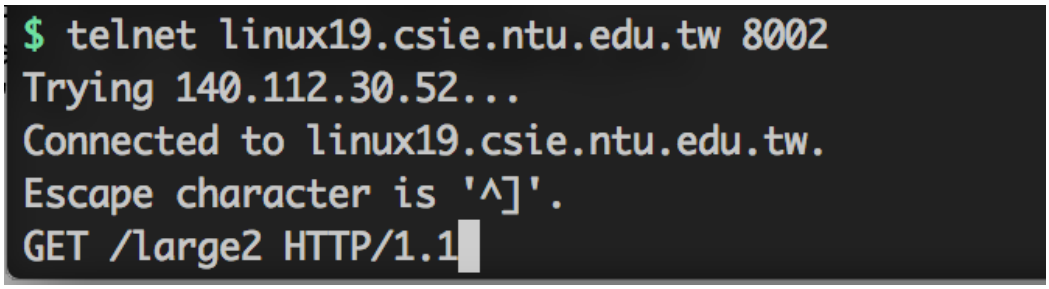
```
$ make
>> gcc -Wall sp_pa2_httpd.c -o run
```

2. run httpd server [`port` `log_file`]

```
$ ./run 8002 logs.txt
```

3. telnet connect

```
$ telnet linux19.csie.ntu.edu.tw 8002
$ > GET /large1 HTTP/1.1
```



4. Browser `url:port/file`

- http://linux19.csie.ntu.edu.tw:8002/large2

Problem description

Browser Busy：

在連線建立完成，開始傳輸（write）資料的時候，不知道為什麼 telnet 都可以很順利的完成，但是 Browser 總是會出現 Broken pipe，然後只印出部分的內容，這個問題讓我想了很久，最後請教了助教才發現原來會出現 Browser busy 的情況，所以 `nwritten = write( requestP[conn_fd].conn_fd,` 回傳的值可能會是 `-1` ，後來我也發現的確會如此。

```
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = -1 / 2042128
>> nwritten / buf_len = 349577 / 2042128
2042128 \ 2042128
complete writing 2042128 bytes on fd 4
```

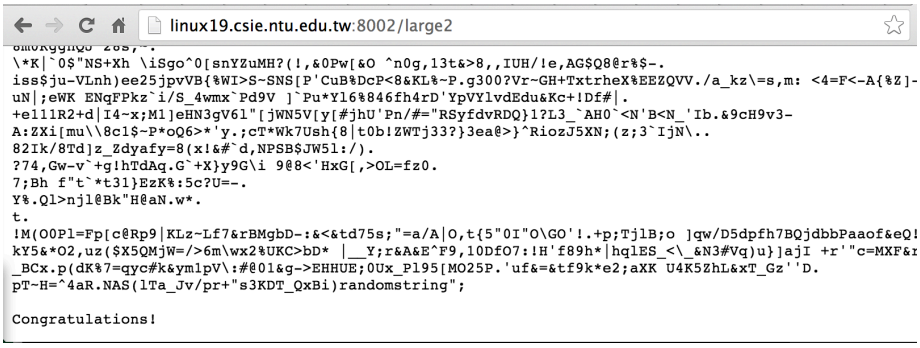
Solution：

因為這個情況有時候會發生會時候不會，所以在考慮上需要把 connection 的 `File Descriptor (fd)` 給 `FD_SET` 到 `$writefds` 的 set 裏面，然後當碰到 `-1` ，的情況的時候就可以直接排除 `-1` 的回傳結果了。

```
writing (buf 0x7fdd44280010, idx 0) 2042510 bytes to request fd 4
 >> nwritten / buf_len = 78512 / 2042510
 >> nwritten / buf_len = 39256 / 2042510
 >> nwritten / buf_len = 92532 / 2042510
 >> nwritten / buf_len = 82718 / 2042510
 >> nwritten / buf_len = 64492 / 2042510
 >> nwritten / buf_len = 74306 / 2042510
 >> nwritten / buf_len = 64492 / 2042510
 >> nwritten / buf_len = 64492 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 128984 / 2042510
 >> nwritten / buf_len = 62886 / 2042510
2042510 \ 2042510
complete writing 2042510 bytes on fd 4
```

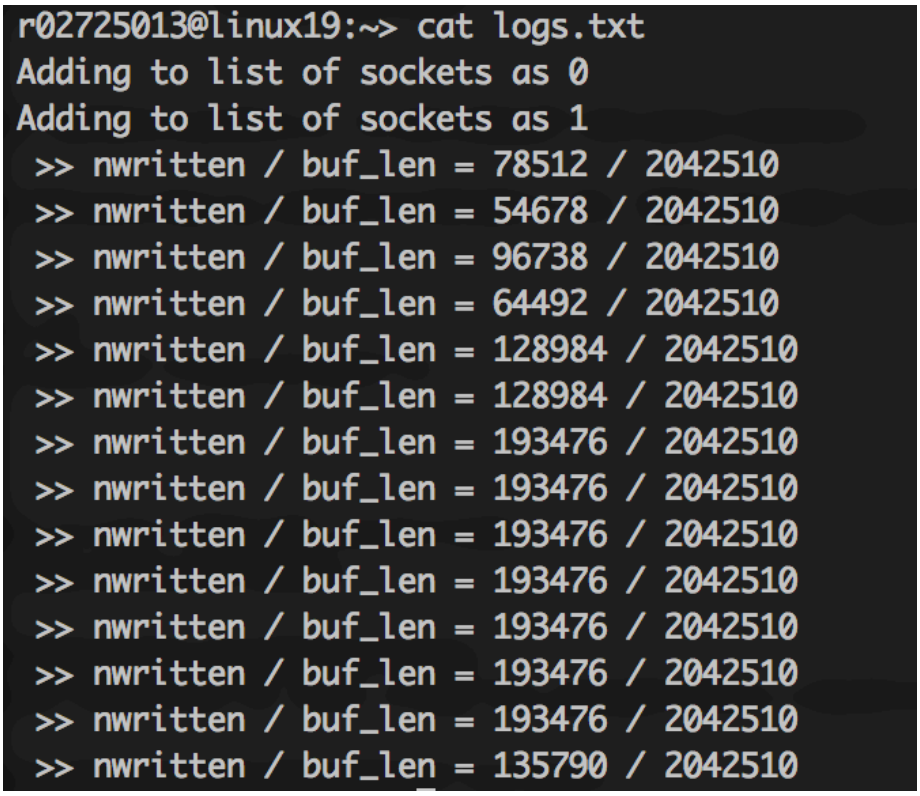
Result

按照 Assignment 2 -- Multiplexing Web Server `Sample Input.pdf` 這份文件依序做模擬操作的結果如下圖：

- large files Congratulations (browser)
  - 

- large files Congratulations (telnet)

- logs file [`cat logs.txt`]



Discussions

這次的實作讓我學習到非常多的 C system call function，但也因為如此過程實在非常的不順利，也查了很多使用 function 的資料。但整體來說這次是一個有趣的作業，以前寫網頁都是直接拿現成的軟體來架站，但是這次終於有比較清楚 Http server 到底是怎麼運作的了，從一個 `A Simple Web Server` 單一連線並且限制傳輸大小的 code 要改成 `A Multiplexing Web Server` 多人連線不會被 block 住並且沒有檔案傳輸的限制，做完非常有成就感！ 最後在

輸出到 logs file 檔的時候又碰到了難題，不管怎麼試都不行，後來經過助教提示才發現原來忘記了 standard output 會先暫存在緩衝區，直到緩衝區滿了才會真正地寫出來，後來使用 `fflush(stdout);` 才能夠立馬寫出來，這一點其實老師上課的時候有提到過，但是寫程式的時候完全忘了這件事。

Reference

- Handle multiple socket connections with fd_set and select on Linux
- Is a return value of 0 from write(2) in C an error
- ◎read與write系統呼叫◎
- ptt2 `看板《SysProgram》` 的精華區。
- Makefile簡易教學...

Source Code

- https://github.com/evenchange4/102-1_SP_PA2_Multiplexing-Web-Server