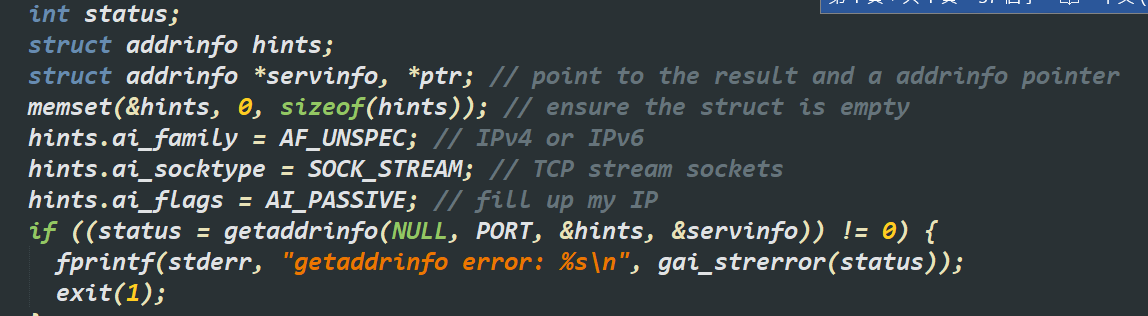
網路程式設計 作業一 學號/系級/姓名 4054100990/資工三/陳知余

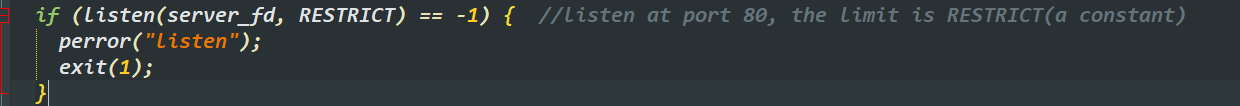
一、fork\_server.c:

(1)程式流程圖：

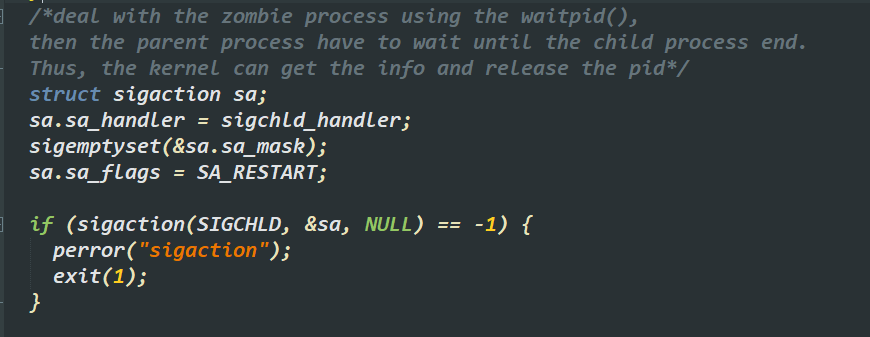
1.設定好server的address type：

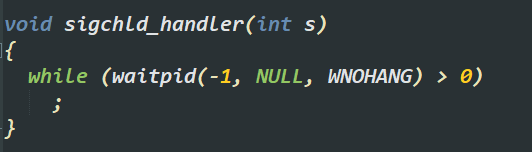


2.接著跑一個for迴圈，分別是socket函式設定socket，setsockopt設定socket option，bind函式定義在預設的port(設定80)，如果都沒error就break。

3. 在設定好的fd listen，最大限制連線RESTRICT設定為10。

4.設定signal handler，避免zombie產生(原理如圖中註解)。





5.最後進入主要的while loop，透過fork()產生child process提供client連線server(>0:child pid, ==0:child, <0:error)，並用read把client\_fd的內容讀入再輸出，網頁部分原本是想用fopen讀html檔再傳送，但發現會有問題(網路上也說不要用這種方法)，所以直接在code裡令一個char array取代，最後也順利顯示。





(2)功能實作技巧

Socket, setsockopt, bind, listen要照順序使用，用一個for迴圈才能保證使用的fd在這些函式沒有出現error。

(3)重要資料結構

1.struct addrinfo {

int ai\_flags;

int ai\_family;

int ai\_socktype;

int ai\_protocol;

socklen\_t ai\_addrlen;

struct sockaddr \*ai\_addr;

char \*ai\_canonname;

struct addrinfo \*ai\_next;

};

作業中有使用的是

ai\_family:設定IPv4或IPv6(AF\_INET and AF\_INET6)，不確定就用AF\_UNSPEC等getaddrinfo()回傳資訊。。

ai\_socktype:設定socket的type，例如SOCK\_STREAM (TCP Socket) or SOCK\_DGRAM(UDP Socket)。

ai\_flags:設定額外特殊的選項

(4)期望加分特點或新功能

1.利用read把連線的client資訊讀入buffer再印出。

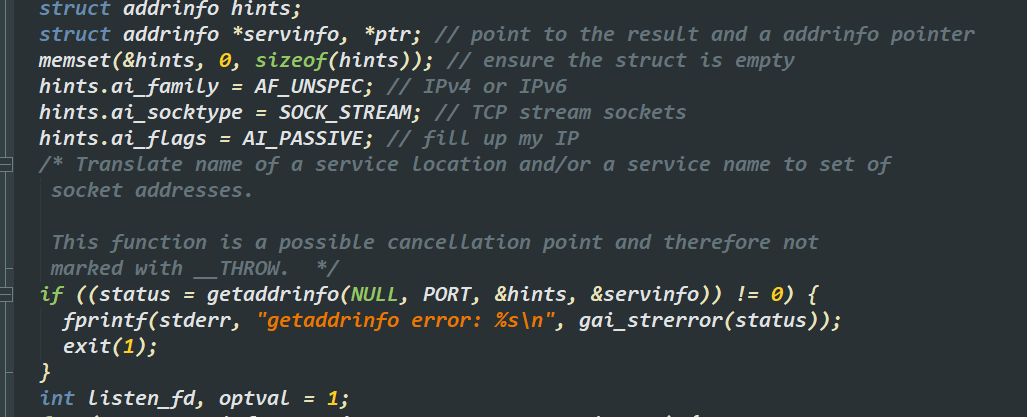
2.透過inet\_ntoa顯示連線者的address。

3.使用signal handler讓parent process wait直到child process結束，才能處理child process被結束後變成zombie的可能。

二、select\_server.c:

(1)程式流程圖：

1.基本上前面設定server的過程和fork\_server.c一樣





比較不同的是多了fd\_set的data type，能透過FD\_SET將合格的socket descriptors新增到set中，讓之後的select來選擇合格的連線。

(FD\_ZERO:清零)



(FD\_SET:將fd 加到set)



2.接著一樣進入while迴圈，首先令一個struct timeval，用來設定select執行間隔的時間。

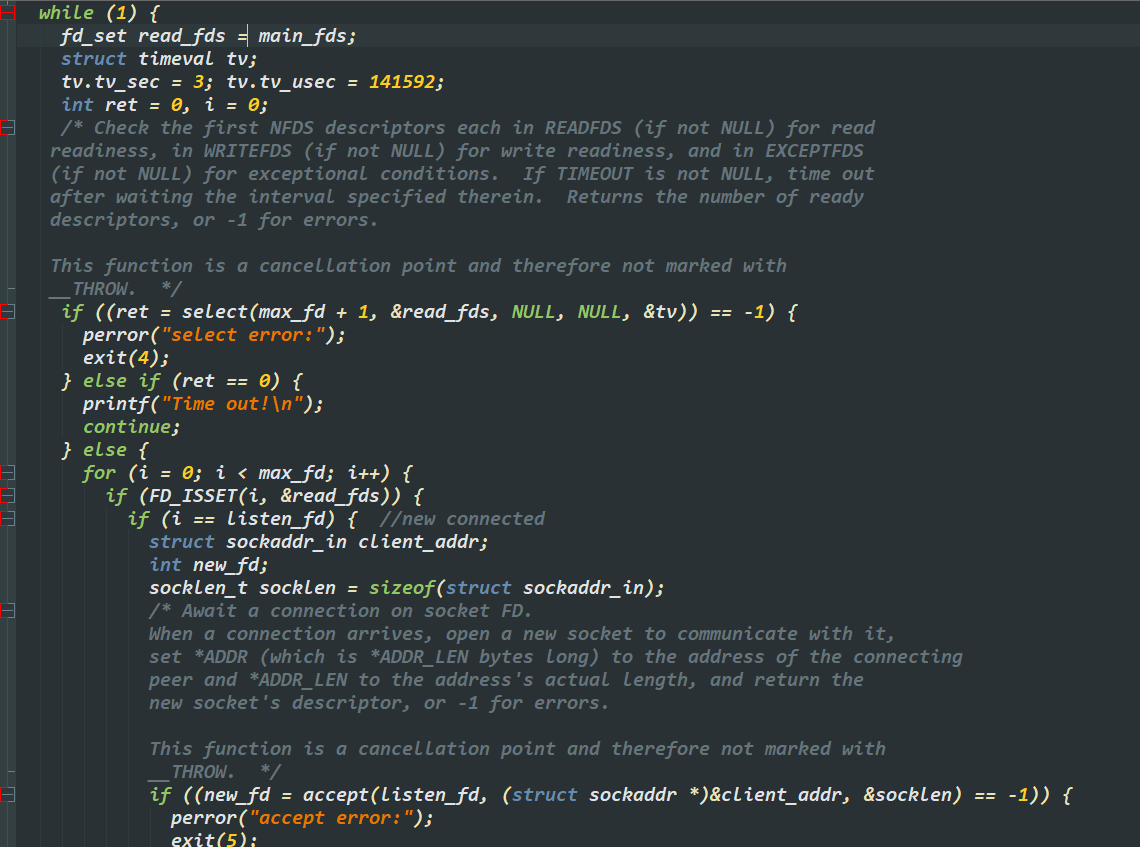
int select(int numfds, fd\_set \*readfds, fd\_set \*writefds,

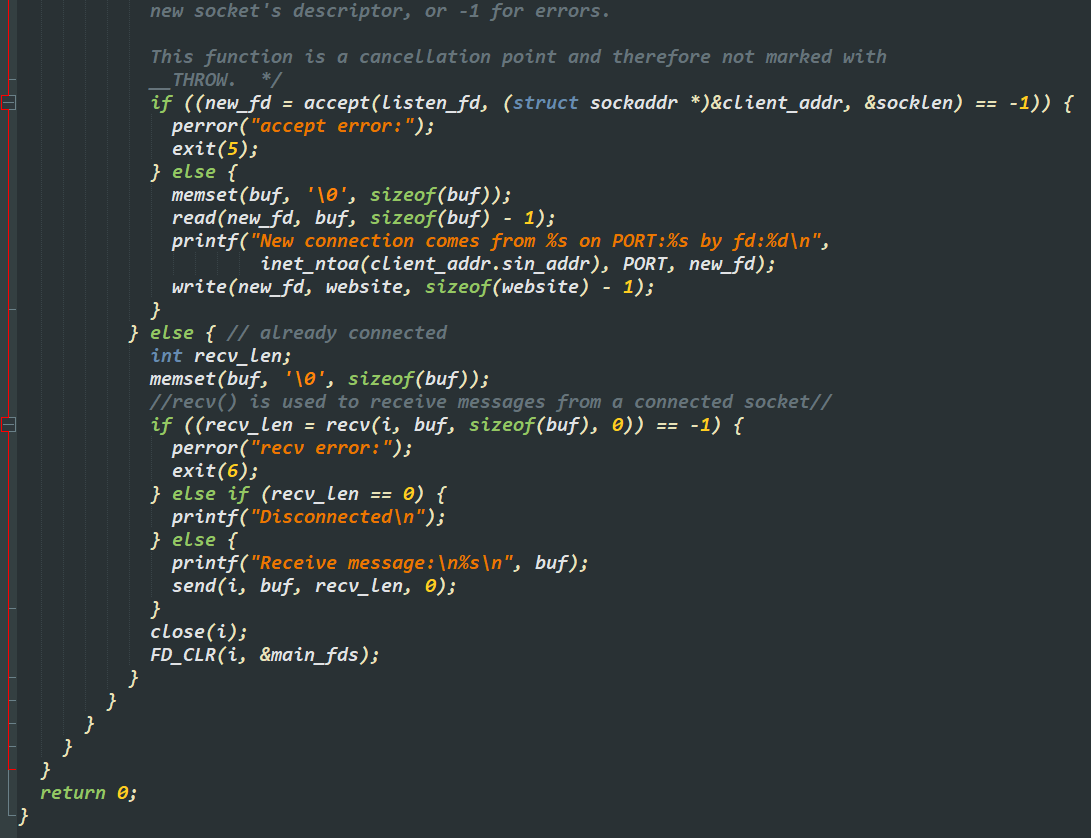
fd\_set \*exceptfds, struct timeval \*timeout);

readfds、writefds 及 exceptfds 分別表示要監視的那一組 file descriptor set。

如果成功，select() 回傳number of file descriptors contained in the three returned descriptor sets，如果等於0則代表時間到期在事件發生之前，-1則是有error。

剩下如果fd是新的連線，就用和fork一樣的accept將網頁傳給client，如果是已連線的狀況則改用excv()傳送，最後用FD\_CLR()將該fd移出set。





(2)功能實作技巧:

和fork\_server.c一樣，server設定時Socket, setsockopt, bind, listen要照順序使用，另外select的最後一個參數可以設定時間間隔，如果設定為 0，select() 會在輪詢問過sets中的每個file descriptors之後就timeout。如果設定為 NULL，就永遠不會 timeout並且陷入等待，直到至少一個 file descriptor 已經準備好了。

(3)重要資料結構

1. struct timeval {

time\_t tv\_sec; /\* seconds \*/

suseconds\_t tv\_usec; /\* microseconds \*/

};

定義在<time.h>裡，前者單位是秒，後者單位是微秒，

1 microsecond = 1 /1000000 second.

2. struct sockaddr\_in {

sa\_family\_t sin\_family; /\* address family: AF\_INET \*/

in\_port\_t sin\_port; /\* port in network byte order \*/

struct in\_addr sin\_addr; /\* internet address \*/

};

主要是給IPv4用的。

(4)期望加分特點或新功能：

1.有對幾乎全部的函數做error檢查，並使用perror印出錯誤訊息。

2.有查詢的函數都有附上註解方便review功能、參數、return value…等等。