

第21組

組長：鍾昀誼

組員：張惟筑

組員：陳麒懋



Trace Code

Main()

將參數讀入 `argv` 分別對應

`argv[1]` = IP

`argv[2]` = Port

`argv[3]` = rate

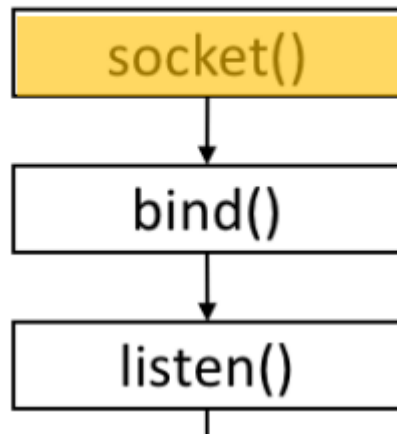
```
sscanf(argv[1], " %d.%d.%d.%d", &proxy_IP[0], &proxy_IP[1], &proxy_IP[2], &proxy_IP[3]);  
port = atoi(argv[2]);  
// assign rate  
rate = atoi(argv[3]);
```

```
maomao@maomao-VirtualBox:~/2016-ITCN-FTP-Proxy-Template-master$ ./proxy 127.0.0.1 8888 200
```

Trace Code

Create server ()

Server Side



```
ctrlfd = create_server(port);
```

```
int create_server(int port)
```

```
listenfd = socket(AF_INET, SOCK_STREAM, 0);
```

```
if (bind(listenfd, (struct sockaddr *)&servaddr ,  
        sizeof(servaddr)) < 0)
```

```
listen(listenfd, 3);
```

Trace Code

`Listen(int sockfd , int backlog)`

```
listen(listenfd, 3);
```

這裡的 3 表示 listen 函數可以利用的queue的容量。

Trace Code

proxy_func()

```
int proxy_func(int ser_port, int clifd, int rate , int loads)
```

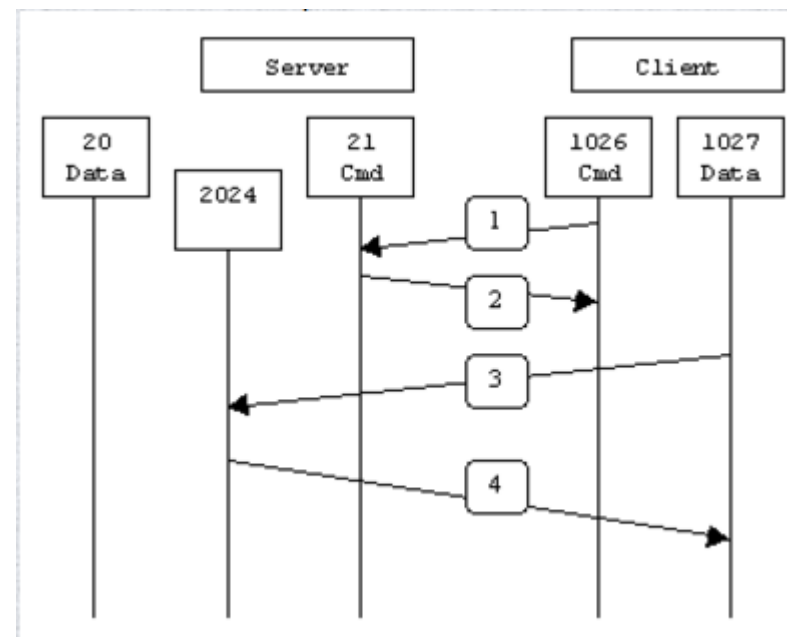
Proxy的主體，進入 proxy_func() ,並將main function的connfd 傳進去。

Trace Code

`connect_FTP()`

這個 function 是實作1、2 這兩個步驟。

至於第3、4 步驟在 `proxy_func()`
裡頭實作。



Trace Code

proxy_func()

```
// initialize select vars  
FD_ZERO(&allset);  
FD_SET(clifd, &allset);  
FD_SET(serfd, &allset);
```

FD_ZERO : 將 allset 全部清成 0

FD_SET : 將 clifd 與 serfd 放入 allset 監聽。

Trace Code

proxy_func()

```
maxfdp1 = max(clifd, serfd) + 1;  
// select descriptor  
nready = select(maxfdp1, &rset, NULL, NULL, NULL);
```

Select：監聽多重的 I/O port，如果有 I/O 就返回就緒的 FD 個數。如果 timeout 仍無 I/O 則返回 0

Trace Code

proxy_func()

```
// reset select vars  
rset = allset;  
// check FTP client socket fd  
if (FD_ISSET(clifd, &rset)) {  
    :
```

FD_ISSET : 確認第一個參數(file descriptor) 有被加入第二個參數(descriptor set)監聽，若有就回傳1，否則回傳0

Trace Code

proxy_func()

FD_ISSET

```
if ((byte_num = read(clifd, buffer, MAXSIZE)) <= 0) {  
    printf("[!] Client terminated the connection.\n");  
    break;  
}  
  
if (write(serfd, buffer, byte_num) < 0)
```

假如 FD_ISSET return 1，就開始傳輸資料，做 read(from client) 和 write(from server) 的動作。

Trace Code

proxy_func()

```
if (FD_ISSET(scrfd, &rset)) {  
    memset(buffer, 0, MAXSIZE);  
    if ((byte_num = read(scrfd, buffer, MAXSIZE)) <= 0) {  
        printf("[!] Server terminated the connection.\n");  
        break;  
    }  
}
```

這裡的動作跟上面幾張 ppt 做的是一樣的，這裡做的是從server讀取data。

Trace Code

```
proxy_func()
```

```
if (status == FTP_PASV_CODE && ser_port == FTP_PORT)
```

判斷是 command packet or data packet

如果是 command packet 就開一個 data port 讓他去連線。

Trace Code

proxy_func()

```
if ((childpid = fork()) == 0) {  
    data_port = pasv[5] * 256 + pasv[6];  
    datafd = create_server(data_port);  
    printf("[-] Waiting for data connection!\n");  
    clilen = sizeof(struct sockaddr_in);  
    connfd = accept(datafd, (struct sockaddr *)&cliaddr, &clilen);
```

實作上一頁提到的開一個 data port

Trace Code

proxy_func()

```
proxy_func(data_port, connfd, rate , 1);
```

再遞迴一次proxy，傳入 data_port，這裡的 1 代表passive mode 已經設定完成，可以開始傳輸資料。

The understanding of the proxy

Proxy(代理伺服器) 是建立起 client 和 server 的連線，再利用 proxy 傳資料。

而且 proxy 可以做到限制速度的功能，甚至是保護 server，隱藏 server 的地址，讓有心人士無法直接攻擊 server。

How to implement the approach of controlling transmission rate?

在 proxy function 裡面先以 clock() 求出上一個 read 到下一個 read 的時間差與收到的 byte 數，再把這兩個 data 傳入 rate_control function。

在 rate_control function 裡先求出理想狀態所需要的時間(target_time)，接著算出實際速度與理想速度的差(error_speed)

How to implement the approach of controlling transmission rate?

再以微積分的方式，求出usleep function裡面調整delay時間的公式

```
void rate_control(clock_t t , int rate , int buf_size) {  
  
    long double target_time , time_delay, error_speed, errors;  
    target_time = (long double)((long double)size*1000000.0)/((long double)rate*1024.0);  
    error_speed = (long double)size/(long double)t-(long double)rate;  
    errors = (error_speed/(long double)rate)/  
    (long double)((long double)rate*((long double)rate+(long double)(error_speed/(long double)rate)));  
    printf("----errors speed: %LF----\n", error_speed);  
    if (t <= target_time){  
        time_delay = (long double)(target_time-t);  
        printf("----delay: %LF----\n", time_delay+errors*1000000*(long double)(error_speed/(long double)rate));  
        usleep(time_delay + errors*1000000*(long double)(error_speed/(long double)rate));  
    }  
}
```

Problems you confront and how do you solve them?

✓我們一開始完全看不懂 `Fork()` 這個 function 在做

什麼，於是我們就去問助教，加上我們在網路上到處查資料，後來比較了解程式整體之後才開始 coding。

我們 `upload` 資料的時候，瞬時速率符合我們的期待，但是平均速率跑出來卻是瞬時速率從來沒出現過的數字。(例如：瞬時：100~195 平均：3XX...)

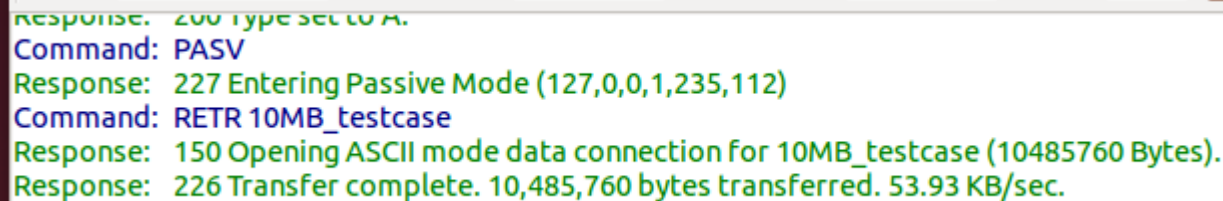
後來我們發現是我們看錯地方..... 我們看到的是它重新整理的速度。

How to run your code and show some experimental results

```
gcc ftp_proxy -o proxy
```

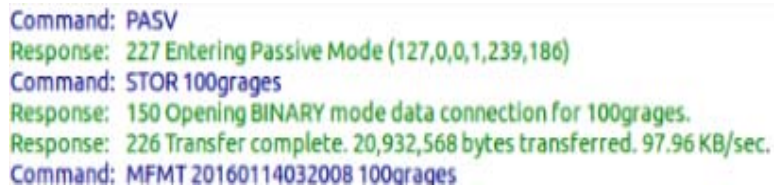
```
./proxy 127.0.0.1 8888 50
```

(download)(50KB/s)



```
Response: 200 Type set to A.  
Command: PASV  
Response: 227 Entering Passive Mode (127,0,0,1,235,112)  
Command: RETR 10MB_testcase  
Response: 150 Opening ASCII mode data connection for 10MB_testcase (10485760 Bytes).  
Response: 226 Transfer complete. 10,485,760 bytes transferred. 53.93 KB/sec.
```

(upload) (100KB/s)



```
Command: PASV  
Response: 227 Entering Passive Mode (127,0,0,1,239,186)  
Command: STOR 100grages  
Response: 150 Opening BINARY mode data connection for 100grages.  
Response: 226 Transfer complete. 20,932,568 bytes transferred. 97.96 KB/sec.  
Command: MFMT 20160114032008 100grages
```

What is the responsibility of each member?

組長：鍾昀誼：撰寫程式、修改參數。

組員：張惟筑：撰寫程式、修改參數。

組員：陳麒懋：修改參數、report。

共同：理解template code

