

Network Programming 과제 #03

주의 사항

아래의 조건(TCP 소켓, 구조체)을 따르지 않고 구현한 경우, 0점 처리함

각 소스 파일에 **학번, 이름(영문 가능) 주석이 없는 경우, 파일당 -1점 감점**

각 기능에 대한 소스 코드만 있고, 동작이 되지 않는 경우에는 아래의 점수를 받을 수 없음
화면 출력 과정에서 쓰레기 값이 출력되거나, 출력 내용이 맞지 않으면 항목당 -5점 감점함

1. TCP 데이터 송수신(SEQ, ACK)을 이용한 파일 전송 프로그램 (20점)

- 파일제출: hw3_server.c, hw3_client.c

■ 공통 사항

- ✓ 클라이언트, 서버의 read/write BUF_SIZE는 100 바이트로 고정
- ✓ 파일 입출력 함수는 저수준(read, write)함수 또는 고수준(fread, fwrite) 함수 모두 사용 가능함
- ✓ 서버의 파일 크기와 클라이언트의 파일 크기는 동일해야 되며, 파일 이름이 화면에 입력한 내용과 동일해야 됨 (버퍼 사용 주의)
- ✓ 클라이언트와 서버는 서로 다른 폴더에서 동작을 시킴

■ 클라이언트, 서버 공용 데이터 구조

```
#define BUF_SIZE 100
#define SEQ_START 1000
typedef struct {
    int seq;          // SEQ number
    int ack;          // ACK number
    int buf_len;      // File read/write bytes
    char buf[BUF_SIZE+1]; // 파일 이름, "File Not Found" 또는 파일 내용 전송
}Packet;
```

■ 클라이언트 (2점)

- ✓ 클라이언트는 사용자로부터 전송 받을 파일의 이름을 입력 받고, 서버에 해당 파일 이름을 전송함 (Packet 구조체의 buf[]에 파일 이름을 저장하여 전송)
 - seq=0, ack=0으로 설정
 - buf_len: 파일 이름의 길이
- ✓ 클라이언트는 서버에서 전송한 파일 수신이 완료되면 소켓을 종료

■ 서버 (3점)

- ✓ 서버는 클라이언트에게 전송 받은 파일이름을 확인해서 파일이 있는 경우, 해당 파일을 전송함
- ✓ 파일이 서버에 없는 경우, buf[]에 **"File Not Found"**를 저장하여 클라이언트에게 전송하고 소켓 종료
 - seq=0, ack=0으로 설정

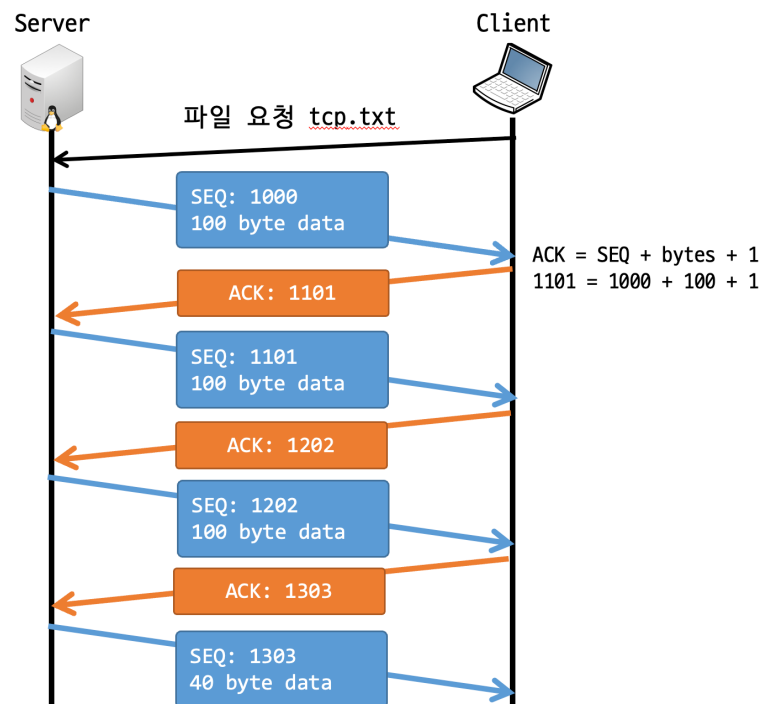
- buf_len: “File Not Found”의 문자열 길이

■ TCP 데이터 송수신 과정 출력 (15점)

- ✓ TCP 데이터 송수신 과정(SEQ, ACK 내용)을 반드시 화면 출력해야 됨
- ✓ 송수신 과정에서 SEQ, ACK 출력이 없는 경우: -15점
- ✓ SEQ, ACK의 출력은 있지만, 번호가 정확하지 않는 경우: -10점
- ✓ 서버의 시작 SEQ번호: 1000
- ✓ 서버의 파일 크기와 클라이언트가 수신한 파일 크기가 동일하지 않는 경우 (-10점 감점)
- ✓ 파일 송수신이 종료되면, 서버와 클라이언트는 종료됨

■ 동작 과정

- 클라이언트가 tcp.txt 파일을 서버에게 요청한 경우



[실행 결과] #01: 클라이언트가 요청한 파일이 없는 경우

[서버] : 서버와 클라이언트는 서로 다른 폴더에서 실행 시킴

| 서버 | 클라이언트 |
|--|--|
| <pre>\$./server 9190 ----- File Transmission Server ----- a.txt File Not Found Exit server \$</pre> | <pre>\$./client 127.0.0.1 9190 Input file name: a.txt [Client] request a.txt File Not Found Exit client \$</pre> |

[실행 결과] #02: 존재하는 파일을 요청한 경우 (파일 전송이 끝나면 서버/클라이언트 종료)

| 서버 | 클라이언트 |
|--|---|
| <pre>\$./server 9191 ----- File Transmission Server ----- [Server] sending tcp.txt [Server] Tx: SEQ: 1000, 100 byte data [Server] Rx ACK: 1101 [Server] Tx: SEQ: 1101, 100 byte data [Server] Rx ACK: 1202 [Server] Tx: SEQ: 1202, 100 byte data [Server] Rx ACK: 1303 [Server] Tx: SEQ: 1303, 100 byte data [Server] Rx ACK: 1404 [Server] Tx: SEQ: 1404, 100 byte data [Server] Rx ACK: 1505 [Server] Tx: SEQ: 1505, 100 byte data [Server] Rx ACK: 1606 . . . [Server] Tx: SEQ: 5343, 100 byte data [Server] Rx ACK: 5444 [Server] Tx: SEQ: 5444, 100 byte data [Server] Rx ACK: 5545 [Server] Tx: SEQ: 5545, 100 byte data [Server] Rx ACK: 5646 [Server] Tx: SEQ: 5646, 100 byte data [Server] Rx ACK: 5747 [Server] Tx: SEQ: 5747, 58 byte data tcp.txt sent (4758 Bytes) Exit server \$</pre> | <pre>\$./client 127.0.0.1 9191 Input file name: tcp.txt [Client] request tcp.txt [[Client] Rx SEQ: 1000, len: 100 bytes [Client] Tx ACK: 1101 [Client] Rx SEQ: 1101, len: 100 bytes [Client] Tx ACK: 1202 [Client] Rx SEQ: 1202, len: 100 bytes [Client] Tx ACK: 1303 [Client] Rx SEQ: 1303, len: 100 bytes [Client] Tx ACK: 1404 [Client] Rx SEQ: 1404, len: 100 bytes [Client] Tx ACK: 1505 [Client] Rx SEQ: 1505, len: 100 bytes [Client] Tx ACK: 1606 . . . [Client] Rx SEQ: 5343, len: 100 bytes [Client] Tx ACK: 5444 [Client] Rx SEQ: 5444, len: 100 bytes [Client] Tx ACK: 5545 [Client] Rx SEQ: 5545, len: 100 bytes [Client] Tx ACK: 5646 [Client] Rx SEQ: 5646, len: 100 bytes [Client] Tx ACK: 5747 [Client] Rx SEQ: 5747, len: 58 bytes tcp.txt received (4758 Bytes) Exit client \$</pre> |

[최종] 클라이언트에서 파일 수신 및 파일 크기 확인

| |
|---|
| <pre>\$ ll total 44 drwxrwxr-x 2 netprog netprog 4096 3월 26 17:25 ./ drwxr-xr-x 3 netprog netprog 4096 3월 26 17:24 ../ -rwxrwxr-x 1 netprog netprog 16944 3월 26 17:24 client*</pre> |
|---|

| | | | | | | |
|------------|---|---------|---------|------|-------------|--------------|
| -rw-r--r-- | 1 | netprog | netprog | 4168 | 3월 26 17:06 | hw3_client.c |
| -rw-rw-r-- | 1 | netprog | netprog | 4758 | 3월 26 17:25 | tcp.txt |