

컴퓨터망 프로그래밍 과제 #03

1. TCP 데이터 송수신(SEQ, ACK)을 이용한 파일 전송 프로그램 (15점)

- 파일제출: hw3_server.c, hw3_client.c

■ 공통 사항

- ✓ 클라이언트, 서버의 read/write 버퍼 크기는 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]; // 파일 이름 또는 파일 내용 전송
}Packet;
```

■ 클라이언트 (2점)

- ✓ 클라이언트는 사용자로부터 전송 받을 파일의 이름을 입력 받고, 서버에 해당 파일 이름을 전송함 (buf[BUF_SIZE]에 파일 이름을 저장하여 전송)
- ✓ 클라이언트는 서버에서 전송한 파일 수신이 완료되면 소켓을 종료

■ 서버 (3점)

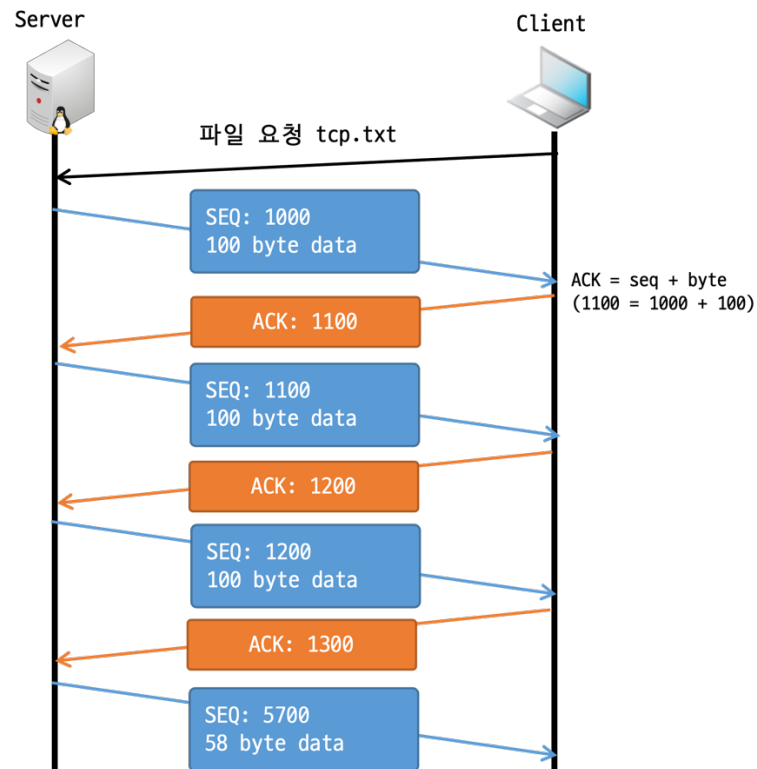
- ✓ 서버는 클라이언트에게 전송 받은 파일이름을 확인해서 파일이 있는 경우, 해당 파일을 전송함
- ✓ 파일이 서버에 없는 경우, "File Not Found"를 클라이언트에게 전송하고 소켓 종료

■ TCP 데이터 송수신 과정 출력 (10점)

- ✓ TCP 데이터 송수신 과정(SEQ, ACK 내용)을 반드시 화면 출력해야 됨
- ✓ 송수신 과정에서 SEQ, ACK 출력이 없는 경우: -10점
- ✓ SEQ, ACK의 출력은 있지만, 번호가 정확하지 않는 경우: -5점
- ✓ 서버의 시작 SEQ번호: 1000
- ✓ 서버의 파일 크기와 클라이언트가 수신한 파일 크기가 동일하지 않는 경우 (-5점 감점)

■ 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>\$./hw3_server 9190</pre> <hr/> <p>File Transmission Server</p> <hr/> <p>a.txt File Not Found</p>	<pre>\$./hw3_client 127.0.0.1 9190</pre> <p>Input file name: a.txt</p> <p>[Client] request a.txt</p> <p>File Not Found</p>

실행 결과 #02: 클라이언트가 tcp.txt 파일을 요청한 경우

서버	클라이언트
<pre>\$./server 9191</pre> <hr/> <p>File Transmission Server</p> <hr/> <p>[Server] sending tcp.txt</p> <p>[Server] Tx: SEQ: 1000, 100 byte data</p> <p>[Server] Rx ACK: 1100</p> <p>[Server] Tx: SEQ: 1100, 100 byte data</p> <p>[Server] Rx ACK: 1200</p>	<pre>\$./client 127.0.0.1 9191</pre> <p>Input file name: tcp.txt</p> <p>[Client] request tcp.txt</p> <p>[Client] Rx SEQ: 1000, len: 100 bytes</p> <p>[Client] Tx ACK: 1100</p> <p>[Client] Rx SEQ: 1100, len: 100 bytes</p> <p>[Client] Tx ACK: 1200</p>

[Server] Tx: SEQ: 1200, 100 byte data [Server] Rx ACK: 1300	[Client] Rx SEQ: 1200, len: 100 bytes [Client] Tx ACK: 1300
[Server] Tx: SEQ: 1300, 100 byte data [Server] Rx ACK: 1400	[Client] Rx SEQ: 1300, len: 100 bytes [Client] Tx ACK: 1400
[Server] Tx: SEQ: 1400, 100 byte data [Server] Rx ACK: 1500	[Client] Rx SEQ: 1400, len: 100 bytes [Client] Tx ACK: 1500
[Server] Tx: SEQ: 1500, 100 byte data [Server] Rx ACK: 1600	[Client] Rx SEQ: 1500, len: 100 bytes [Client] Tx ACK: 1600
.
[Server] Tx: SEQ: 5300, 100 byte data [Server] Rx ACK: 5400	[Client] Rx SEQ: 5300, len: 100 bytes [Client] Tx ACK: 5400
[Server] Tx: SEQ: 5400, 100 byte data [Server] Rx ACK: 5500	[Client] Rx SEQ: 5400, len: 100 bytes [Client] Tx ACK: 5500
[Server] Tx: SEQ: 5500, 100 byte data [Server] Rx ACK: 5600	[Client] Rx SEQ: 5500, len: 100 bytes [Client] Tx ACK: 5600
[Server] Tx: SEQ: 5600, 100 byte data [Server] Rx ACK: 5700	[Client] Rx SEQ: 5600, len: 100 bytes [Client] Tx ACK: 5700
[Server] Tx: SEQ: 5700, 58 byte data tcp.txt sent (4758 Bytes)	[Client] Rx SEQ: 5700, len: 58 bytes tcp.txt received (4758 Bytes)

클라이언트에서 파일 수신 및 파일 크기 확인

```
$ ll
total 44
drwxrwxr-x 2 changsu changsu 4096 3월 31 10:58 ./
drwxr-xr-x 3 changsu changsu 4096 3월 31 10:49 ../
-rwxrwxr-x 1 changsu changsu 16944 3월 31 10:49 client*
-rw-r--r-- 1 changsu changsu 4166 3월 31 10:58 hw3_client.c
-rw-rw-r-- 1 changsu changsu 4758 3월 31 10:51 tcp.txt
```

주의) 정상적으로 데이터가 보였을 때
파악 종료!!
그 다음