

[506489] 시스템프로그래밍

실습 #10 문제 및 보고서

이름	곽영주
학번	20175105
소속 학과/대학	빅데이터
분반	01 (담당교수: 김태운)

<주의사항>

- 개별 과제입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)

- 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.

- 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.

- SmartLEAD 제출 데드라인:

- 월요일 분반: 다음 실습 시간 전날까지(일요일 까지)

- 수요일 분반: 다음 실습 시간 전날까지 (화요일 까지)

- 데드라인을 지나서 제출하면 24 시간 단위로 20%씩 감점(5 일 경과 시 0 점)

- 주말/휴일/학교행사 등 모든 날짜 카운트 함

- 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함

- 예외 없음

- SmartLEAD 에 아래의 파일을 제출해 주세요

- 보고서(PDF 파일로 변환 후 제출 권장하나, WORD 형식으로 제출도 가능)

- 보고서 파일명에 이름과 학번을 입력해 주세요.

- 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (zip 파일로 압축하여 하나의 파일로 제출)

<개요>

이번 과제는 소켓 프로그래밍을 사용해서 프로그램을 개발하는 내용입니다.

<실습 과제>

[Q 0] 요약 [10 점]

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.

답변: 이번 과제를 통해 TCP 프로토콜 소켓 프로그래밍에 대해 개념을 제대로 잡을 수 있었습니다. 또한, UDP 프로토콜 소켓 프로그래밍을 해보면서 두 프로토콜 소켓의 차이점을 배웠습니다. 그리고 멀티 서비스와 동시 동작 서버를 구현해보면서 텀프로젝트 하는데 큰 도움이 되었습니다.

[Q 1] 1:1 대화 [20 점]

TCP 프로토콜 + 인터넷 도메인 소켓을 이용해 1:1 대화가 가능한 프로그램을 작성하세요. 메시지를 입력할 때, 서버-클라이언트-서버-클라이언트-... 방식으로, 서로 번갈아 가며 메시지를 입력합니다. 대화 상대 중 한명이 <QUIT> 이라고 입력하고 메시지를 전송하면 1:1 대화가 종료됩니다. 터미널에는 본인이 입력하는 메시지와 상대방이 보내준 메시지 모두가 표시되어야 합니다. 상대방이 입력한 메시지는 아래와 같이 터미널에 출력합니다.

YOU: Nice to meet you.

프로그램을 구동하고, 각 사용자마다 3 번 이상의 메시지를 입력하세요. 대화에 참여하는 두명의 터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w10/q01$ ./client
=== 1 on 1 ===
YOU: hello client.
MY: hi, server
YOU: hi hallym
MY: yes
YOU: no
MY: ok
YOU: <QUIT>
=== exit ===
yeongju@vm-ubuntu20:~/sp2021-2/w10/q01$
```

```
yeongju@vm-ubuntu20:~/sp2021-2/w10/q01$ ./server
=== 1 on 1 ===
MY: hello client.
YOU: hi, server
MY: hi hallym
YOU: yes
MY: no
YOU: ok
MY: <QUIT>
=== exit ===
yeongju@vm-ubuntu20:~/sp2021-2/w10/q01$
```

[Q 2] 서버 시간 조회 [20 점]

UDP 프로토콜 + 인터넷 도메인 소켓을 이용해서, 서버에서 시간 정보를 읽어오는 프로그램을 작성하세요. 서버의 시간을 어떤 형식으로 표현할지는 자유롭게 정하세요. 클라이언트 프로그램은, 시작하는 즉시 서버에게 시간을 조회하는 요청 메시지(TIME-REQUEST)를 전송합니다. 서버는 메시지를 받으면 자신의 터미널에 클라이언트의 IP 주소를 출력하고, 메시지에 대한 응답으로 현재 시간 정보를 클라이언트에게 전송합니다. 클라이언트는 서버로부터 받은 메시지를 터미널에 출력하고 종료합니다.

프로그램을 구동하고, 터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야

합니다.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w10/q02$ ./client
Server Time: Mon Nov  1 22:48:48 2021
yeongju@vm-ubuntu20:~/sp2021-2/w10/q02$
```

```
yeongju@vm-ubuntu20:~/sp2021-2/w10/q02$ ./server
Client IP: 127.0.0.1
^C
yeongju@vm-ubuntu20:~/sp2021-2/w10/q02$
```

[Q 3] 멀티 서비스 [25 점]

TCP 프로토콜 + 인터넷 도메인 소켓을 사용하여, 아래와 같이 동작하는 프로그램을 작성하세요.

- 클라이언트가 접속하면, 아래와 같이 서비스 목록을 클라이언트에게 보내줍니다 (문자열의 형태로 보내주면 됩니다). 클라이언트는 아래의 메시지를 받으면 터미널에 출력합니다.

<Available Services>

1. Echo server <ECHO>
2. Get server info <SINFO>
3. Get server time <STIME>

- 클라이언트는 <ECHO>, <SINFO>, <STIME> 중에서 하나를 서버로 전송하여 어떤 서비스를 사용하는지를 알립니다.
- 서버는 클라이언트의 메시지를 받은 후, 해당하는 서비스를 클라이언트에게 제공합니다. 각각의 서비스에 대한 설명은 아래와 같습니다.

<서비스 설명>

1. 에코 서버 <ECHO> : 클라이언트만 메시지를 입력할 수 있습니다. 클라이언트가

메시지를 입력하면, 서버는 동일한 메시지를 클라이언트에게 돌려줍니다. 클라이언트는 수신한 메시지를 터미널에 출력합니다. 클라이언트가 <QUIT> 이라는 메시지를 보내면 모든 서비스가 종료됩니다.

2. 서버 정보 조회 <SINFO> : 클라이언트에게 서버의 정보(호스트명, IP 주소, 포트 번호)를 보내주고 모든 서비스가 종료됩니다. 클라이언트는 수신한 메시지를 터미널에 출력하고 종료합니다.

3. 서버 시간 조회 <STIME> : 클라이언트에게 서버의 시간을 보내주고 모든 서비스가 종료됩니다. 클라이언트는 수신한 메시지를 터미널에 출력하고 종료합니다. 서버의 시간을 어떤 형식으로 표현할지는 자유롭게 정하세요.

프로그램을 구동하고, 각각의 서비스를 이용하는 터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w10/q03$ ./client
<Available Services>
1. Echo server <ECHO>
2. Get server info <SINFO>
3. Get server time <STIME>
Enter a service: <ECHO>
SEND: hi
RECV: hi
SEND: hello
RECV: hello
SEND: bye
RECV: bye
SEND: <QUIT>
RECV: <QUIT>
yeongju@vm-ubuntu20:~/sp2021-2/w10/q03$
```

```
yeongju@vm-ubuntu20:~/sp2021-2/w10/q03$ ./server
<ECHO>
RECV: hi
RECV: hello
RECV: bye
RECV: <QUIT>
yeongju@vm-ubuntu20:~/sp2021-2/w10/q03$
```

```
yeongju@vm-ubuntu20:~/sp2021-2/w10/q03$ ./client
<Available Services>
1. Echo server <ECHO>
2. Get server info <SINFO>
3. Get server time <STIME>
Enter a service: <SINFO>
hostname: localhost
IP: 127.0.0.1
Port: 9002
yeongju@vm-ubuntu20:~/sp2021-2/w10/q03$
```

```
yeongju@vm-ubuntu20:~/sp2021-2/w10/q03$ ./server
<SINFO>
```

```
yeongju@vm-ubuntu20:~/sp2021-2/w10/q03$ ./client
<Available Services>
1. Echo server <ECHO>
2. Get server info <SINFO>
3. Get server time <STIME>
Enter a service: <STIME>
Server time: Tue Nov  2 01:24:41 2021
yeongju@vm-ubuntu20:~/sp2021-2/w10/q03$
```

```
yeongju@vm-ubuntu20:~/sp2021-2/w10/q03$ ./server
<STIME>
```

[Q 4] 동시 동작 ECHO 서버 [25 점]

Echo 서버를 동시에 여러 개 서비스 할 수 있는 프로그램을 작성하시오(TCP, 인터넷 도메인 소켓). Echo 서버는 클라이언트로부터 메시지를 받으면 동일한 메시지를 클라이언트로

전달합니다. 클라이언트는 서버로부터 받은 메시지를 터미널에 출력합니다. Fork 또는 Fork + Exec 를 사용하여 동시 동작 서버를 구현하고, 서버가 동시에 다수의 클라이언트를 서비스 할 수 있게 구현하세요.

3 명의 사용자가 동시에 각각의 Echo 서버와 통신하는 방식으로 실행하세요. 각 사용자마다 3 번 이상의 메시지를 입력하세요. 총 4 개의 터미널을 한 화면에 캡처해서 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

The image displays four terminal windows arranged in a 2x2 grid, showing the execution of a concurrent echo server and three client programs.

- Top-Left Window:** Running the client program `./client`. It shows a sequence of messages: `SEND: i'm 1.`, `RECV: i'm 1.`, `SEND: hi`, `RECV: hi`, `SEND: end`, `RECV: end`, `SEND: <QUIT>`, and `RECV: <QUIT>`.
- Top-Right Window:** Running the client program `./client`. It shows a sequence of messages: `SEND: i'm 3.`, `RECV: i'm 3.`, `SEND: hello`, `RECV: hello`, `SEND: apple`, `RECV: apple`, `SEND: <QUIT>`, and `RECV: <QUIT>`.
- Bottom-Left Window:** Running the client program `./client`. It shows a sequence of messages: `SEND: i'm 2.`, `RECV: i'm 2.`, `SEND: ho`, `RECV: ho`, `SEND: haha`, `RECV: haha`, `SEND: <QUIT>`, and `RECV: <QUIT>`.
- Bottom-Right Window:** Running the server program `./server`. It shows the server's log of connections and disconnections for the three clients:
 - `Client[1] Connect`, `Client[2] Connect`, `Client[3] Connect`
 - `Client[1]: i'm 1.`, `Client[2]: i'm 2.`, `Client[3]: i'm 3.`
 - `Client[3]: hello`, `Client[3]: apple`, `Client[1]: hi`
 - `Client[1]: end`, `Client[2]: ho`, `Client[2]: haha`
 - `Client[2]: <QUIT>`, `Client[2] exit`, `Client[3]: <QUIT>`
 - `Client[3] exit`, `Client[1]: <QUIT>`, `Client[1] exit`
 - `^C`

끝! 수고하셨습니다 ☺