

12201922 이규민

System Programming Final Exam, Spring, 2024, Class 1

Submit one zip file that contains a word or pdf file that shows your answer (output result, and explanation) and the server and client source code files. The exam is open book and everything including lecture notes, sample answers, Internet, etc. are open except people and chatGPT.

zip파일 하나에 워드/pdf 파일(실행 결과, 설명)과 서버와 클라이언트 소스코드 파일을 담아서 제출하십시오. 시험은 강의노트, 모범답안, 인터넷 등 모든 것이 오픈북입니다. 다만 사람과 chatGPT에 대해서는 closed book 입니다.

Redo problem 6) in lect12, but change the code so that the client can specify the number of chat partners and send multiple chat partner names. The server should ask each chat partner name one by one. The server should relay the message from a client who is in chatting state to all chatting partners of this client. Use atoi() to convert a string number to an integer. You may need the client structure defined below. Also provide proper explanation for every line(including #include, variable declaration, statements, etc) of your code.

lect12의 6번을 다시 하되 코드를 변경하여 클라이언트가 채팅 상대자 수를 먼저 보내고 그이름들을 차례로 보내도록 하시오. 서버는 각 채팅 상대자 이름을 차례로 물어보아야 합니다. 서버는 채팅 상태에 있는 클라이언트가 메시지를 보내면 이 메시지를 이 클라이언트가 지정한 상대방에게 모두 보내야 합니다. atoi() 함수로 문자열을 숫자로 변환하기 바랍니다. 그리고 클라이언트는 아래 구조체를 사용하기 바랍니다. 또한 코드의 각 라인마다(including #include, variable declaration, statements, etc) 적절한 코멘트를 추가하기 바랍니다.

1. 코드

[finserv.c]

```
//소켓 통신에 필요한 헤더파일들이다.
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>
#include <stdlib.h>
#include <sys/time.h>
#include <unistd.h>
#include <sys/select.h>

#define SERV_TCP_PORT 31289 //여러 프로그램 중 프로그램 구별을 위해 포트 설정을 한다.
#define SERV_ADDR "192.168.0.5" //IP 주소를 입력 시 서버 컴퓨터의 주소를 입력한다

struct client{
    char name[20];
    int partners[10]; // socket number of partners. max 10 chatting partners are allowed
    int numPartners; // number of partners of this client
    int numPartnersSoFar; // initially 0. increased by one each time the server receives
    // the next chat partner name
};

void handle_protocol(int x, fd_set * pset, int state[], struct client cli[]);
void chat_protocol(int x, int s1, fd_set * pset, int state[], struct client cli[]);

int main(){
    int s1,s2, i, x, y;
    struct sockaddr_in serv_addr, cli_addr;
    char buf[50];
    socklen_t xx;

    printf("Hi, I am the server\n");

    bzero((char *)&serv_addr, sizeof(serv_addr));
    serv_addr.sin_family=PF_INET;
    serv_addr.sin_addr.s_addr=inet_addr(SERV_ADDR);
    serv_addr.sin_port=htons(SERV_TCP_PORT);
```

```

//open a tcp socket
if ((s1=socket(PF_INET, SOCK_STREAM, 0))<0){
    printf("socket creation error\n");
    exit(1);
}
printf("socket opened successfully. socket num is %d\n", s1);

//어떤 클라이언트에게 올지 모르므로 자신을 바인딩한다.
x =bind(s1, (struct sockaddr *)&serv_addr, sizeof(serv_addr));
if (x < 0){
    printf("binding failed\n");
    exit(1);
}
printf("binding passed\n");
listen(s1, 5);
xx = sizeof(cli_addr);

// now start ping-pong-pang-pung
// pset remembers all sockets to monitor
// rset is the copy of pset passed to select
fd_set rset, pset;

int maxfd = 50;

FD_ZERO(&rset); // init rset
FD_ZERO(&pset); // init pset

// step 1. s1 is a socket to accept new connection request.
// monitor connection request packet at s1
FD_SET(s1, &pset);

int state[50];
struct client cli[50];

// and loop on select
for(;;){
    rset = pset; // step 2
    select(maxfd, &rset, NULL, NULL, NULL); // step 3
    // now we have some packets
    for(x=0; x<maxfd; x++){ // check which socket has a packet
        if (FD_ISSET(x, &rset)){ // socket x has a packet
            // s1 is a special socket for which we have to do "accept"
            if (x == s1){ // new client has arrived
                //클라이언트 연결 요청을 기다린다.
                //이 경우는 부모 프로세스만이 작업을 한다.
                s2 = accept(s1, (struct sockaddr *)&cli_addr, &xx);
                printf("new cli at socket %d\n", s2);
                FD_SET(s2, &pset); //클라이언트 연결이 된 경우 이를 pset에서 해당 프로세스
                //디스크립터를 1로 바꿔준다.
                state[s2] = 1; //state 초기 설정은 1로 한다.
                cli[s2].numPartnersSoFar = 0;
            }else{ //자식 프로세스만이 프로토콜을 실행한다.
                if(state[x] > 0 && state[x] < 6)
                    handle_protocol(x, &pset, state, cli);

                else
                    chat_protocol(x, s1, &pset, state, cli);
            }
        }
    }
}
}

```

```

void handle_protocol(int x, fd_set * pset, int state[], struct client cli[]){
    // we have a data packet in socket x. do protocol.
    int y; char buf[50];
    y = read(x, buf, 50); // read data from socket x
    buf[y] = 0; // make it a string
    printf("we have received %s at socket %d\n", buf, x);

    //ping을 입력 받으면 답변을 한다 .
    if (state[x] == 1 && strcmp(buf, "ping") == 0){
        write(x, "pong", 4);
        printf("we have sent pong to socket %d\n", x);
        state[x]++;
    }

    //pang을 입력 받으면 이름을 묻는다 .
    else if (state[x] == 2 && strcmp(buf, "pang") == 0){
        write(x, "name?", 5);
        printf("we have sent name? to socket %d\n", x);
        state[x]++;
    }

    //이름을 구조체 name에 저장하고 , 원하는 파트너 수를 요청한다 .
    else if (state[x] == 3){
        strcpy(cli[x].name, buf);
        write(x, "how many chat partners?", 23);
        state[x]++;
    }
}

```

```

//파트너 수를 구조체에 저장하고 , 파트너 이름을 묻는다 .
else if(state[x] == 4){
    cli[x].numPartners = atoi(buf);
    write(x, "chat partner 1?", 15);
    state[x]++;
}

//파트너 이름을 구조체에 저장한다 .
else if(state[x] == 5){
    //모든 구조체의 name을 확인한다 .
    for(int i = 0; i < 50; i++)
    {
        if(strcmp(buf, cli[i].name) == 0)
        {
            //일치하는 이름이 있다면 구조체에 저장한다 .
            //이 경우 numPartnersSoFar은 몇 번째 파트너인지 의미한다 .
            cli[x].partners[cli[x].numPartnersSoFar] = i;
            cli[x].numPartnersSoFar++;
            break;
        }
    }

    //이전에 입력했던 파트너 수만큼 파트너를 찾았다면 채팅을 시작한다 .
    if(cli[x].numPartnersSoFar >= cli[x].numPartners)
    {
        state[x]++;
        write(x, "start chatting", 14);
    }
}

```

```

        //아직 파트너를 모두 찾지 않았다면 그 다음 파트너 이름을 요청한다.
        else
        {
            sprintf(buf, "chat partner %d?", cli[x].numPartnersSoFar + 1);
            write(x, buf, strlen(buf));
        }
    }

    //만약 위 내용 진행 중 다른 내용이 입력된다면 통신을 끝낸다.
    else{
        printf("protocol error. disconnecting socket %d\n", x);
        write(x, "protocol error", 14);
        state[x] = 0;
        close(x);
        FD_CLR(x, pset);
    }
}

void chat_protocol(int x, int s1, fd_set * pset, int state[], struct client cli[]){
    int y; char buf[50];
    y = read(x, buf, 50); //입력받은 내용을 buf에 저장한다.
    buf[y] = '\0';

    char message[50];

    //모든 파트너에게 client가 입력한 메시지를 전송한다.
    for(int i = 0; i < cli[x].numPartners; i++)
    {
        sprintf(message, "%s to %s : %s", cli[x].name, cli[cli[x].partners[i]].name, buf);
        write(cli[x].partners[i], message, strlen(message));
    }
}

```

[fincli.c]

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>
#include <stdlib.h>
#include <sys/time.h>
#include <unistd.h>
#include <sys/select.h>
#include <signal.h>

#define SERV_TCP_PORT 31289
#define SERV_ADDR "192.168.0.5"

int main(){
    int x,y;
    struct sockaddr_in serv_addr;
    char buf[50];
    printf("Hi, I am the client\n");

    bzero((char *)&serv_addr, sizeof(serv_addr));
    serv_addr.sin_family=PF_INET;
    serv_addr.sin_addr.s_addr=inet_addr(SERV_ADDR);
    serv_addr.sin_port=htons(SERV_TCP_PORT);

    //open a tcp socket
    if ((x=socket(PF_INET, SOCK_STREAM, 0))<0){
        printf("socket creation error\n");
        exit(1);
    }
    printf("socket opened successfully. socket num is %d\n", x);
```

아이피와 포트는 서버에서 지정해둔 값과 동일하게 입력해야한다.

```

//connect to the server
if (connect(x, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0){
    printf("can't connect to the server\n");
    exit(1);
}

//채팅 시작
int fd;
fd = fork();
for(int i = 0; i < 10; i++)
{
    if(fd == 0)
    {
        //자식 프로세스는 사용자의 입력을 대기한다.
        fgets(buf, 100, stdin);
        buf[strlen(buf) - 1] = '\0';
        write(x, buf, strlen(buf));
    }
    else
    {
        //부모 프로세스는 서버의 패킷을 대기한다.
        y = read(x, buf, 50);
        buf[y] = '\0';
        printf("%s\n", buf);
    }
}

close(x);
}

```

63 1

서버와 연결을하고, 프로세스 복제를 한다.

부모 프로세스는 쓰기 만을하고, 자식 프로세스는 읽기만을 한다.

2. 결과 화면

[finserv]

```
kyumin@DESKTOP-NUDFAPK ~  
$ finserv  
Hi, I am the server  
socket opened successfully. socket num is 3  
binding passed  
new cli at socket 4  
we have received ping at socket 4  
we have sent pong to socket 4  
we have received pang at socket 4  
we have sent name? to socket 4  
we have received aaa at socket 4  
new cli at socket 5  
we have received ping at socket 5  
we have sent pong to socket 5  
we have received pang at socket 5  
we have sent name? to socket 5  
we have received bbb at socket 5  
new cli at socket 6  
we have received ping at socket 6  
we have sent pong to socket 6  
we have received pang at socket 6  
we have sent name? to socket 6  
we have received ccc at socket 6  
new cli at socket 7  
we have received ping at socket 7  
we have sent pong to socket 7  
we have received pang at socket 7  
we have sent name? to socket 7  
we have received ddd at socket 7  
we have received 2 at socket 4  
we have received ccc at socket 4  
we have received ddd at socket 4  
we have received 2 at socket 6  
we have received eee at socket 6  
we have received eee at socket 6  
we have received ggg at socket 6
```


[fincli]

```
kyumin@DESKTOP-NUDFAPK ~  
$ fincli  
Hi, I am the client  
socket opened successfully. socket num is 3  
ping  
pong  
pang  
name?  
aaa  
how many chat partners?  
2  
chat partner 1?  
ccc  
chat partner 2?  
ddd  
start chatting  
hello  
hi  
how are you
```

Aaa 사용자는 파트너 수는 2명을 원하고, 파트너는 ccc와 ddd를 원한다. "start chatting" 알림이 나타나서 메시지를 입력했다.

```
kyumin@DESKTOP-NUDFAPK ~  
$ fincli  
Hi, I am the client  
socket opened successfully. socket num is 3  
ping  
pong  
pang  
name?  
bbb  
how many chat partners?
```

Bbb 사용자는 aaa의 파트너가 아니므로 아무런 메시지를 받지 못했다.

```

kyumin@DESKTOP-NUDFAPK ~
$ fincli
Hi, I am the client
socket opened successfully. socket num is 3
ping
pong
pang
name?
ccc
how many chat partners?
aaa to ccc : hello
aaa to ccc : hi
aaa to ccc : how are you
2
chat partner 1?
eee
chat partner 1?
eee
chat partner 1?
ggg
chat partner 1?

```

Ccc는 aaa로부터 메시지를 전송 받았다.

또한 등록되어있지 않은 사용자를 파트너로 요청할 경우 동일한 메시지가 반복되는 것을 볼 수 있다.

```

kyumin@DESKTOP-NUDFAPK ~
$ fincli
Hi, I am the client
socket opened successfully. socket num is 3
ping
pong
pang
name?
ddd
how many chat partners?
aaa to ddd : hello
aaa to ddd : hi
aaa to ddd : how are you

```

Ddd는 aaa로부터 메시지를 정상적으로 전송받았다.