

```

//client.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#include<arpa/inet.h>
#include<sys/socket.h>

#define MAX 100
#define MON 3

int premenuue();
void oxmenuue();
void error_check(int validation, char *message);
int printmenuue();
void printcon();

FILE *fp;
char msg[MAX];
char sc[MON];
int top = 0;

int main(int argc, char *argv[])
{

    int ch,ch2;
    char con[MAX];
    int socket_fd, array_len, recv_len, recv_count;
    struct sockaddr_in server_adr;
    char result;

    if(argc!=3)
    {
        printf("./실행파일 IP주소 PORT번호 형식으로 실행해야 합니다.\n");
        exit(1);
    }
    socket_fd=socket(PF_INET, SOCK_STREAM, IPPROTO_TCP); //TCP소켓 생성
    error_check(socket_fd, "소켓 생성");

```

```

memset(&server_adr, 0, sizeof(server_adr)); //구조체 변수 값 초기화
server_adr.sin_family=AF_INET; //IPv4
server_adr.sin_port=htons(atoi(argv[2])); //포트 할당
server_adr.sin_addr.s_addr=inet_addr(argv[1]); //IP 할당

error_check(connect(socket_fd, (struct sockaddr *)&server_adr,
sizeof(server_adr)), "연결 요청");

while(1)
{
    ch = premenue();

    if(ch == 1)
    {
        printf("1<<<>>>");
        while(1)
        {
            ch2 = printmenue();
            if(ch2 == 1)
            {
                fp = fopen("con.txt", "r");
                printcon();
            }
            else if(ch2 == 2)
            {
                fp = fopen("con2.txt", "r");
                printcon();
            }
            else if(ch2 == 3)
            {
                fp = fopen("con3.txt", "r");
                printcon();
            }
            else if(ch2 == 4)
            {
                fp = fopen("con4.txt", "w");
                printf("개념 내용 추가\n");
                while(scanf("%s", con)==1)
                    fwrite(con, sizeof(con),1,fp);
            }
        }
    }
}

```

```

        fclose(fp);
    }
    else if(ch2 == 5)
    {
        printf("추가된 내용확인\n");
        fp = fopen("con4.txt","rb");
        while(fread(con, sizeof(con),1,fp)>0)
            printf("%s \n",con);
        fclose(fp);
    }
    else if(ch2 == 6)
    {
        puts("---end---\n");
        break;
    }
}

}
else if(ch == 2)
{
    printf("2<<<>>>\n");
    oxmenuue();
    array_len=write(socket_fd, sc, sizeof(sc));
    error_check(array_len, "데이터전송");
    recv_len =0;
    while(recv_len!=sizeof(result)) // 패킷이 잘려서 올수도 있으므로 예외처
리를 위한 조건문
    {
        recv_count=read(socket_fd, &result, sizeof(result));
        error_check(recv_count, "데이터수신");
        recv_len+=recv_count;
    }
    printf("-> 점수 : %c\n", result);

    close(socket_fd);

printf("프로그램을 종료합니다\n");
}
else if(ch == 3)
{
    printf("3<<<>>>\n");

```

```

        fp = fopen("pr1.txt", "r");
        while(fgets(msg, MAX, fp)!= NULL)
            printf("%s", msg);
        getchar();
        scanf("%c", &sc[top]);

        top++;
        fp = fopen("pr2.txt", "r");
        while(fgets(msg, MAX, fp)!= NULL)
            printf("%s", msg);
        getchar();
        scanf("%c", &sc[top]);
        array_len=write(socket_fd, sc, sizeof(sc));
        error_check(array_len, "데이터전송");
        recv_len =0;
        while(recv_len!=sizeof(result))
        {
            recv_count=read(socket_fd, &result, sizeof(result));
            error_check(recv_count, "데이터수신");
            recv_len+=recv_count;
        }
        printf("-> 점수 : %c\n", result);

        close(socket_fd);

    }else if(ch == 4)
    {
        printf("점수 확인하기 : ");
        fp = fopen("result.txt","r");
        while(fscanf(fp, "%s",con)==1)
            printf("%s \n",con);
        fclose(fp);
    }
}

void error_check(int validation, char *message)
{
    if(validation== -1)
    {
        fprintf(stderr, "%s 오류\n", message);
    }
}

```

```

        exit(1);
    }
    else
    {
        fprintf(stdout, "%s 완료\n", message);
    }
}

void oxmenue()
{
    int i;
    printf("1. 고대 사회의 신분은 귀족 평민천민으로 구성되어있다. : ");
    getchar();
    scanf("%c", &sc[top]);
    top++;
    printf("\n 2. 고구려에서는 가난한 농민을 구제하기위한 시책으로 곡식을 빌려주었
다가 추수한뒤에 갚게하는 진대법을 고국원왕때 마련하였다. : ");
    getchar();
    scanf("%c", &sc[top]);
    top++;
    printf("\n 3. 신라의골품제도는 관등에 따라 관복의 색깔을달리하였다.:");
    getchar();
    scanf("%c", &sc[top]);
}

int premenue()
{
    int ch;
    puts("1.개념 2. OX문제 3.기출 문제 4. 점수확인하기 ");
    scanf("%d", &ch);
    return ch;
}

int printmenue()
{
    int ch;
    puts("1.고대(구.신) 2.청동기시대 3.고대사회 4. 내용추가 5. 내용확인 6. 종료");
    scanf("%d", &ch);
    return ch;
}

void printcon()
{

```

```

        while(fgets(msg, MAX, fp)!= NULL)
            printf("%s", msg);
    }

//Server.c OX문제처리
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define BACKLOG 3 //연결대기 큐 숫자
#define MAX 3 //스택 배열의 크기

void error_check(int validation, char *message); //실행 오류 검사
char check(int );

void main(int argc, char *argv[])
{
    FILE *fp;
    int server_fd, client_fd, array_len, i, recv_len, recv_count;
    struct sockaddr_in server_adr, client_adr;
    socklen_t client_adr_size;
    int top=0; //스택 포인터
    char sc[MAX];
    char re[3]={'x','o','o'};
    char result;
    int so =0 ;
    if(argc!=2)
    {
        fprintf(stderr, "./실행파일 port번호 형식으로 입력해야합니다\n");
        exit(1);
    }

    server_fd=socket(PF_INET, SOCK_STREAM, IPPROTO_TCP); //TCP 소켓 생성
    error_check(server_fd, "소켓 생성");

    memset(&client_adr, 0, sizeof(client_adr)); //구조체 변수 값 초기화
    memset(&server_adr, 0, sizeof(server_adr)); //구조체 변수 값 초기화
    server_adr.sin_family=AF_INET; //IPv4
    server_adr.sin_port=htons(atoi(argv[1])); //포트 할당

```

```

server_adr.sin_addr.s_addr=htonl(INADDR_ANY); //IP주소 할당

error_check(bind(server_fd, (struct sockaddr
*)&server_adr,sizeof(server_adr)), "소켓주소 할당"); //주소 바인딩
error_check(listen(server_fd, BACKLOG), "연결요청 대기");

client_adr_size=sizeof(client_adr);

client_fd=accept(server_fd, (struct sockaddr *)&client_adr, &client_adr_size);
//특정 클라이언트와 데이터 송수신용 TCP소켓 생성
printf("* %s:%d의 연결요청\n", inet_ntoa(client_adr.sin_addr),
ntohs(client_adr.sin_port));
error_check(client_fd, "연결요청 승인");

recv_len=0;
while(recv_len!=sizeof(sc)) // 패킷이 잘려서 올수도 있으므로 예외처리를 위한 조
건문
{
    recv_count=read(client_fd, sc, sizeof(sc));
    error_check(recv_count, "데이터수신");
    printf("%d 바이트를 수신하였습니다\n", recv_count);
    recv_len+=recv_count;
}
for(i=0; i<3; i++)
{
    if(sc[i] == re[i])
    {
        so++;
    }
}

result = check(so);
fp = fopen("result.txt", "w");
fprintf(fp, "%c", result);

array_len=write(client_fd, &result, sizeof(result));
error_check(array_len, "데이터전송");

close(client_fd);
printf("* %s:%d의 연결종료\n", inet_ntoa(client_adr.sin_addr),
ntohs(client_adr.sin_port));

```

```

        close(server_fd);

        printf("프로그램을 종료합니다\n");
    }

char check(int so)
{
    if(so == 0)
        return '0';
    else if(so == 1)
        return '1';
    else
        return '2';
}

void error_check(int validation, char* message)
{
    if(validation==-1)
    {
        fprintf(stderr, "%s 오류\n", message);
        exit(1);
    }
    else
    {
        fprintf(stdout, "%s 완료\n", message);
    }
}

//Server2.c 기출문제
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define BACKLOG 3 //연결대기 큐 숫자
#define MAX 2 //스택 배열의 크기

void error_check(int validation, char *message); //실행 오류 검사
char check(int );

void main(int argc, char *argv[])
{

```



```

FILE *fp;
int server_fd, client_fd, array_len, i, recv_len, recv_count;
struct sockaddr_in server_adr, client_adr;
socklen_t client_adr_size;
int top=0; //스택 포인터
char sc[MAX];
char re[MAX]='3','4';
char result;
int so =0 ;
if(argc!=2)
{
    fprintf(stderr, "./실행파일 port번호 형식으로 입력해야합니다\n");
    exit(1);
}

server_fd=socket(PF_INET, SOCK_STREAM, IPPROTO_TCP); //TCP 소켓 생성
error_check(server_fd, "소켓 생성");

memset(&client_adr, 0, sizeof(client_adr)); //구조체 변수 값 초기화
memset(&server_adr, 0, sizeof(server_adr)); //구조체 변수 값 초기화
server_adr.sin_family=AF_INET; //IPv4
server_adr.sin_port=htons(atoi(argv[1])); //포트 할당
server_adr.sin_addr.s_addr=htonl(INADDR_ANY); //IP주소 할당

error_check(bind(server_fd, (struct sockaddr *)&server_adr,sizeof(server_adr)), "소켓주소 할당"); //주소 바인딩
error_check(listen(server_fd, BACKLOG), "연결요청 대기");

client_adr_size=sizeof(client_adr);

client_fd=accept(server_fd, (struct sockaddr *)&client_adr, &client_adr_size);
//특정 클라이언트와 데이터 송수신용 TCP소켓 생성
printf("* %s:%d의 연결요청\n", inet_ntoa(client_adr.sin_addr), ntohs(client_adr.sin_port));
error_check(client_fd, "연결요청 승인");

recv_len=0;
while(recv_len!=sizeof(sc)) // 패킷이 잘려서 올수도 있으므로 예외처리를 위한 조
건문
{
    recv_count=read(client_fd, sc, sizeof(sc));

```

```

        error_check(recv_count, "데이터수신");
        printf("%d 바이트를 수신하였습니다\n", recv_count);
        recv_len+=recv_count;
    }
    for(i=0; i<2; i++)
    {
        if(sc[i] == re[i])
        {
            so = so+1;
        }
    }

    result = check(so);
    fp = fopen("result.txt", "w");
    fprintf(fp, "%c", result);

    array_len=write(client_fd, &result, sizeof(result));
    error_check(array_len, "데이터전송");

    close(client_fd);
    printf("*      %s:%d의      연결종료\n",      inet_ntoa(client_adr.sin_addr),
ntohs(client_adr.sin_port));
    close(server_fd);

    printf("프로그램을 종료합니다\n");
}

char check(int so)
{
    if(so == 0)
        return '0';
    else if(so == 1)
        return '1';
    else
        return '2';
}

void error_check(int validation, char* message)
{
    if(validation== -1)
    {
        fprintf(stderr, "%s 오류\n", message);
    }
}

```

```

        exit(1);
    }
    else
    {
        fprintf(stdout, "%s 완료\n", message);
    }
}

//Server 로그인기능
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define BACKLOG 2 //연결대기 큐 숫자
#define MAX 2 //스택 배열의 크기

void error_check(int validation, char *message); //실행 오류 검사

void main(int argc, char *argv[])
{
    int server_fd, client_fd, array_len,i,recv_len,recv_count,recv_count2, recv_len2;
    struct sockaddr_in server_adr, client_adr;
    socklen_t client_adr_size;
    int id, pw;
    int top=0; //스택 포인터
    int score[MAX]={0};
    double avg;
    char result;
    int ch;
    if(argc!=2)
    {
        fprintf(stderr, "./실행파일 port번호 형식으로 입력해야합니다\n");
        exit(1);
    }
    server_fd=socket(PF_INET, SOCK_STREAM, IPPROTO_TCP); //TCP 소켓 생성
    error_check(server_fd, "소켓 생성");

    memset(&client_adr, 0, sizeof(client_adr)); //구조체 변수 값 초기화
    memset(&server_adr, 0, sizeof(server_adr)); //구조체 변수 값 초기화
    server_adr.sin_family=AF_INET; //IPv4

```

```

server_adr.sin_port=htons(atoi(argv[1])); //포트 할당
server_adr.sin_addr.s_addr=htonl(INADDR_ANY); //IP주소 할당

error_check(bind(server_fd, (struct sockaddr *)&server_adr,sizeof(server_adr)), "
소켓주소 할당"); //주소 바인딩
error_check(listen(server_fd, BACKLOG), "연결요청 대기");

client_adr_size=sizeof(client_adr);

client_fd=accept(server_fd, (struct sockaddr *)&client_adr, &client_adr_size); //
특정 클라이언트와 데이터 송수신용 TCP소켓 생성
printf("*      %s:%d의      연결요청\n",      inet_ntoa(client_adr.sin_addr),
ntohs(client_adr.sin_port));
error_check(client_fd, "연결요청 승인");

recv_len=0;
while(recv_len!=sizeof(score)) // 패킷이 잘려서 올수도 있으므로 예외처리를 위한 조
건문
{
    recv_count=read(client_fd, score, sizeof(score));
    error_check(recv_count, "데이터수신");
    printf("%d 바이트를 수신하였습니다\n", recv_count);
    recv_len+=recv_count;
}

for(i=0;i<2;i++)
{
    printf("score[%d]=%d\n",i,score[i]); //수신된 값 출력, 디버깅용
}

id = score[0];
pw = score[1];

result = check(id, pw);

array_len = write(client_fd, &result, sizeof(result));
printf("%d 바이트를 전송하였습니다\n", array_len);
error_check(array_len, "데이터전송");

recv_len2=0;
while(recv_len2 != sizeof(ch))

```

```

    {
        recv_count2 = read(client_fd, ch, sizeof(ch));
        error_check(recv_count, "데이터수신");
        printf("%d byte recv \n", recv_count2);
    }
    printf("%d \n", ch);
    close(client_fd);

    printf("*      %s:%d의      연결종료\n",      inet_ntoa(client_adr.sin_addr),
ntohs(client_adr.sin_port));
    close(server_fd);

    printf("프로그램을 종료합니다\n");
}

void error_check(int validation, char* message)
{
    if(validation== -1)
    {
        fprintf(stderr, "%s 오류\n", message);
        exit(1);
    }
    else
    {
        fprintf(stdout, "%s 완료\n", message);
    }
}

int check(int id, int pw)
{
    if(id == 1234 && pw == 1234)
    {
        return 1;
    }
    else
        return 0;
}

//client 로그인기능을하는 기능
#include <stdio.h>
#include <stdlib.h>

```

```

#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define MAX 3 //스택 배열의 크기

void error_check(int validation, char *message); //실행 오류 검사
void menu();

void main(int argc, char *argv[])
{
    int socket_fd, array_len, recv_len, recv_count, array_len2;
    struct sockaddr_in server_adr;
    int top=0; //스택포인터
    int ch;
    int s[MAX]; //id, pw, ch
    char result;

    if(argc!=3)
    {
        printf("./실행파일 IP주소 PORT번호 형식으로 실행해야 합니다.\n");
        exit(1);
    }

    socket_fd=socket(PF_INET, SOCK_STREAM, IPPROTO_TCP); //TCP소켓 생성
    error_check(socket_fd, "소켓 생성");

    memset(&server_adr, 0, sizeof(server_adr)); //구조체 변수 값 초기화
    server_adr.sin_family=AF_INET; //IPv4
    server_adr.sin_port=htons(atoi(argv[2])); //포트 할당
    server_adr.sin_addr.s_addr=inet_addr(argv[1]); //IP 할당

    error_check(connect(socket_fd, (struct sockaddr *)&server_adr,
sizeof(server_adr)), "연결 요청");

    printf("ID 를 입력해주세요 : ");
    scanf("%d", &s[top]);
    top++;

    printf("PW 를 입력해주세요 : ");
    scanf("%d", &s[top]);

```

```

top++;

array_len=write(socket_fd, s, sizeof(s));
printf("%d 바이트를 전송하였습니다\n", array_len);
error_check(array_len, "데이터전송");

recv_len=0;
while(recv_len!=sizeof(result)) // 패킷이 잘려서 올수도 있으므로 예외처리를 위한 조
건문
{
    recv_count=read(socket_fd, &result, sizeof(result));
    error_check(recv_count, "데이터수신");
    printf("%d 바이트를 수신하였습니다\n", recv_count);
    recv_len+=recv_count;
}
if(result == 1)
{
    menu();
    scanf("%d", &ch);
    array_len2=write(socket_fd, s, sizeof(s));

    printf("%d 바이트를 전송하였습니다\n", array_len);
    error_check(array_len, "데이터전송");
}
close(socket_fd);

printf("프로그램을 종료합니다\n");
}

void error_check(int validation, char *message)
{
    if(validation== -1)
    {
        fprintf(stderr, "%s 오류\n", message);
        exit(1);
    }
    else
    {
        fprintf(stdout, "%s 완료\n", message);
    }
}

```

```

int value_check(int score)
{
    if(score<0||score>100)
    {
        printf("값이 유효하지 않습니다. 다시입력해주세요 : ");
        scanf("%d", &score);
        score=value_check(score); //유효값 입력할때까지 재귀호출
    }
    return score;
}

void menu()
{
    puts("로그인 성공");
    puts("****리눅스 시스템 프로그래밍 ****");
    puts("**** 히라 국사 ***");
    puts("1.연도표 2. 개념 3. OX 4. 기출문제 5. 정오답표");
}

```