Quiz Show

Unix System Programming 2020

IT정보공학과 201812788_양건희

IT정보공학과 201846158_이영훈

Server

- Communication: TCP protocol
- Exit: SIGINT or Game over
- File: server.c, socklib.c, tcp.h, question.txt, answer.txt, (userList.txt, userScore.txt)
  (존재하지 않아도 게임 진행 중 생성 가능)
- Additional Function: give the questions randomly.
  Save user's id and score in txt File
  Select user with the highest score

Server

- Sequence:

- 코드 내 정의 된 port_num을 통해 서버를 실행하며 클라이언트의 접속을 기다린다.
- 클라이언트 접속 시 다섯개의 문제를 question.txt로부터 랜덤하게 추출하고, 그에 해당하는 정답을 answer.txt로부터 추출한다.
- 클라이언트로부터 name을 수신해서 userList.txt에 저장한다.
- 클라이언트의 준비상태를 확인 후 ready 수신 시 다섯개의 문제를 순차적으로 클라이언트에게 송신한다.
- ready외의 문자 수신 시 클라이언트에게 ready를 재입력할 것을 요청한다.

Server

- Sequence:

- 클라이언트로부터 수신한 답을 answert.txt와 비교하여 정답 시 score++한다.
- 다섯 문제가 끝난 후 Game Over를 클라이언트로 전송하고, 서버에서는 최종 점수를 출력한다.
- 각 클라이언트가 Game Over 될 때마다 접속했던 클라이언트들의 점수를 비교하여 가장 높은 점수를 획득한 클라이언트(WINNER)의 name과 score를 클라이언트에 송신한다.
- 각 클라이언트 Game Over 될 때마다score를 오름차순 정렬하여 name과 함께 서버에 출력한다. (세 번째 클라이언트 Game Over시 세 명의 이름-점수 확인)
- 세 번 클라이언트가 접속 후 종료하면 서버를 종료 시킨다.

Server_Additional Function 1

```
//이 름  입 력 받 기
if ((n=read(fd, (char *)&msg, sizeof(msg))) < 0){
    perror("read");
    exit(1);
}
strncpy(name, msg.data, sizeof(msg.data));

if(!strcmp(name, "end"))
    close(fd);

printf("\n\nclient's name is %s\n", name);
if(client_num ==1){
    if((fp3 = fopen("./userList.txt", "w")) == NULL){
        perror("fopen");
        exit(1);
    }
    fprintf(fp3, "%s", name);
    fclose(fp3);
}
else{
    if((fp3 = fopen("./userList.txt", "a+t")) == NULL){
        perror("fopen");
       exit(1);
    }
    fprintf(fp3, "\n%s", name);
    fclose(fp3);
}
```

- Save user's id and score in txt File

- Client 진입 시 마다 fopen(), fclose()를 통해 기존에 저장된 question.txt와 answer.txt를 랜덤으로 읽어 Client에 문제를 제공하고 답을 확인한다.

- Client로부터 수신한 user_id와 score는 fprintf를 통해 userList.txt/userScore.txt에 저장한다.

- 저장된 userList.txt와 userScore.txt를 읽어 들여 점수를 비교 후 정렬한다.

Server_Additional Function 2

```
printf("\n\n*************Score***************\n");
for(i=0; i<client_num; i++){
    tmp_score[i] = atoi(userscore[i]);
}

i=0;
do{
if(client_num==1)
        break;
    for(j=1; j<client_num; j++){
        if(tmp_score[i] > tmp_score[j]){
            strcpy(tmp, userlist[i]);
            strcpy(userlist[i], userlist[j]);
            strcpy(userlist[j], tmp);

            strcpy(tmp, userscore[i]);
            strcpy(userscore[i], userscore[j]);
            strcpy(userscore[j], tmp);
        }
    }
    i++;
}while(i<client_num-1);

for(i=0; i<client_num ; i++){
    printf("\nuserid: %s", userlist[i]);
    printf("\nuserscore: %s", userscore[i]);
}

strcpy(tmp2, userlist[client_num-1]);
strcat(tmp2, "   ");
strcat(tmp2, userscore[client_num-1]);
```

- Select user with the highest score
- Print sorted userid/userscore

- 선택 정렬을 통해 오름차순 정렬을 진행한다.
- 진행 후 server에 낮은 점수를 가진 user부터 차례로 출력한다.
- 가장 높은 값을 가진 user_id와 score는 client 에 전송 후 출력한다.

Sever, Client _ tcp.h file

▼ tcp.h

```
typedef struct  {
    char    str_rank[1000];
    char    data[256];
}MsgType;
```

```
if(write(fd, (char*)&msg, sizeof(msg)) <0)
{
    perror("write");
    exit(1);
}
sleep(2);
if((n=read(fd, (char*)&msg, sizeof(msg))) <0)
{
    perror("read");
    exit(1);
}
printf("\t%s\n", msg.data);
```

- Sever와 Client 사이 문자열을 송/수신 할 때 사용.

+tcp.h에 char str_rank[1000];가 포함되어 있지만 server.c 수정 중 미처 삭제하지 못한 코드로 인해 남겨둔 코드입니다.

- MsgType msg로 선언 후 이용.
- data 배열에 문자열을 담아 MsgType 구조체 자체를 송/수신 (write/read)
- 출력에도 MsgType 구조체의 data배열을 문자열로 사용
- fork()/thread를 사용하지 않아 Server와 Client의 연속되는 read, write가 겹쳐 에러가 발생하는 경우를 막기 위해 read – write 사이에 sleep(2)를 작성하였다. 이로 인해 게임 진행이 매우 느리다는 단점이 있다.

Client

- Communication: TCP protocol
- Exit: SIGINT or Game over
- File: client.c, socklib.c, tcp.h

Client

- Sequence:

- 저장된 서버 주소와 port_num을 통해 프로그램을 실행한다.
- 접속 후 name을 입력한다.
- 준비 상태를 알리는 ready를 입력한다. 다른 단어 입력 시 재입력을 요청 받는다.
- 5개의 문제를 순차적으로 수신 받고, 수신 받은 문제의 답을 입력 후 서버로 송신한다.
- 마지막 문제의 답을 서버로 송신하면 서버로부터 Game Over를 수신 받아 출력한다.
- 클라이언트의 총점(Final Score)을 서버로부터 수신 받아 출력한다.
- 이전 클라이언트들과 비교하여 가장 높은 점수를 획득한 클라이언트(WINNER)의 id와 score를 수신 받은 후 출력한다.

Compile

- Server: gcc socklib.c server.c -o 201812788_201846158_server

  ./201812788_201846158_server

- Client: gcc socklib.c client.c -o 201812788_201846158_client

  ./201812788_201846158_client

Part Sharing

코로나로 인해 비대면으로 프로젝트를 진행하여 서버/클라이언트 분배 시 대면보다 소통에 어려움이 있고, 한 사람에게 분량이 과도하게 치우칠 것 같아 서로 디렉토리를 공유해 서버 – 클라이언트를 협력하여 동시에 작성하였다.

- Server : 201846158_이영훈
- Client : 201812788_양건희