

## 운영체제 과제 2

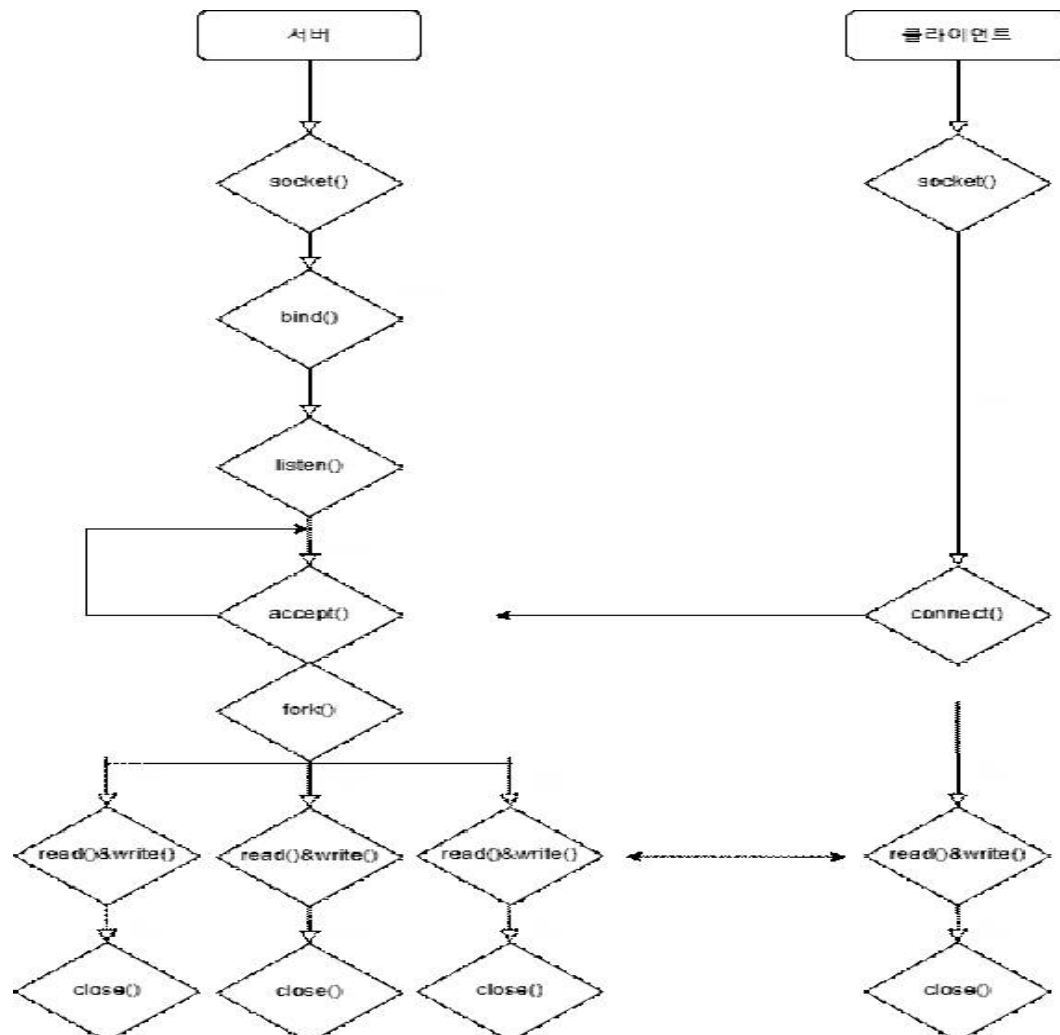
201921162 송민기

### 1. 소켓 프로그래밍 설계

#### 1.1 다중프로세스를 사용하는 이유

프로세스란 실행되고 있는 프로그램을 말한다. 다중 프로세스 즉 멀티프로세스는 다수의 프로세스를 생성하는 방식으로 설계된 설문 조사를 위한 소켓 프로그래밍의 서버에서 사용이 된다. 이것을 사용하는 이유는 cpu를 보다 효율적으로 사용하기 위함이고, 또한 설문조사를 진행할 경우 여러 명의 사람이 동시에 접속할 수 있다. 이 경우 하나의 프로세스만을 이용한다면 한 명이 접속해 있을 때 다른 사람들이 사용하지 못하는 상황이 발생한다. 따라서 부모 프로세스가 연결을 감지하면 자식 프로세스를 생성하여 한명의 소켓과 통신을 시도하고 다시 부모 프로세스는 listen을 한다. fork() 함수를 이용하여 프로세스를 생성할 수 있다.

#### 1.2 다이어그램 설명



설계한 설문조사를 위한 소켓의 다중 프로세스 프로그래밍의 소켓의 다이어그램은 다음과 같다. 우선 서버와 클라이언트는 소켓에서 파생되는데, socket()을 통해 소켓을 처음에 생성을 하고 초기화를 시킨다. 이후 클라이언트의 연결을 기다리기 위해 서버가 먼저 실행된다. bind() 함수를 이용해서 소켓에 포트 번호를 부여한다. listen() 함수를 통해 클라이언트의 연결을 기다린다. 이후 클라이언트에서 connect() 함수를 통해 연결을 요청하고 연결이 확인 되면 accept() 함수를 통해서 클라이언트 연결에 대한 새로운 자식 프로세스를 생성한다. 생성된 자식 프로세스에서는 read()와 write() 함수를 이용하여 클라이언트와 통신하고 close() 함수를 통해 연결을 닫는다. 반면 accept() 이후에 fork()를 통해 자식프로세스가 생성되면 부모 프로세스는 다시 위로 올라가 다른 클라이언트의 연결을 기다리고 다시 연결에 성공하면 다른 자식 프로세스를 생성하여 다른 개별의 명령을 실행한다. 클라이언트도 수행을 마치면 close()를 통해서 소켓연결을 종료한다.

### 1.3 코드 설명

server.c

```
serv_sock = socket(PF_INET, SOCK_STREAM, 0);  
if (serv_sock == -1)  
{  
    error_handling("socket() error");  
}
```

소켓을 생성하는 코드이다. 프로토콜 체계를 설정하고 전송타입을 정하고 특정 프로토콜을 지정한다. 첫 번째 매개변수는 IPV4를 나타내고 스트림통신이 TCP를 이용하겠다는 것이다. 세 번째는 TCP/UDP를 정하는 부분인데, 앞서 TCP를 이용하겠다고 했으므로 0으로 선언한다.

```
memset(&serv_addr, 0, sizeof(serv_addr)); // s  
serv_addr.sin_family = AF_INET;  
serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);  
serv_addr.sin_port = htons(1983);
```

구조체 변수를 초기화하는 부분으로 htons괄호 안에 들어가는 부분은 자신의 ip 주소와 1983 포트번호를 이용하겠다는 것을 의미한다. htons()는 호스트 바이트를 네트워크 바이트로 변환시켜 준다.

```

if (bind(serv_sock, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) == -1)
{
    error_handling("bind() error");
}

if (listen(serv_sock, 10) == -1)
{
    error_handling("listen() error");
}

```

bind()에서 첫 번째 매개변수는 소켓 생성 뒤에 얻은 디스크립터이다. 두 번째는 앞서 저장시켰던 ip, port 등을 의미한다. 또한 세 번째는 구조체의 길이를 나타낸다.

listen()에서는 첫 번째 매개변수 역시 디스크립터를 나타내고 두 번째는 최대 수용 가능한 클라이언트의 수를 의미한다.

```

while (1)
{
    clnt_sock = accept(serv_sock, (struct sockaddr *)&clnt_addr, &clnt_addr_size);

    if (clnt_sock == -1)
    {
        error_handling("accept() error");
    }
    else
    {
        printf("소켓 연결 성공!\n");
    }

    pid = fork();
    people = people + 1;

    if (pid == 0)
    {
        sprintf(buffer, "현재 설문 참여 인원:%d\n", people);
        write(clnt_sock, buffer, sizeof(buffer));

        read(clnt_sock, buffer, sizeof(buffer));
        printf("클라이언트 메시지: %s\n", buffer);

        write(clnt_sock, "감사합니다.\n", sizeof("감사합니다.\n"));
        // memset(buffer, 0x00, sizeof(buffer));

        break;
    }
    else if (pid != 0)
    {
        continue;
    }
}

```

반복문 안에서 실행되며 accept()를 통해서 클라이언트의 연결을 받아들인다. 연결이 완료 되면 fork()를 통해 자식 프로세스를 생성하고 만약 pid(프로세스 id)가 0이며 자식 프로세스를 의미하고 설문조사를 진행한다. 우선 클라이언트의 메시지를 read() 하고 현재 설문 완료 인원을 write()를 통해서 클라이언트의 화면에 띄워준다. 만약 부모 프로세스 즉 id 가 0이 아닐 경우에는 다시 accept() 전으로 돌아가서 새로운 클라이언트의 접속을 기다린다. 수행이 모두 완료되면 서버는 close()를 통해 종료된다.

## client.c

```
if (connect(sock, (struct sockaddr*) &serv_addr, sizeof(serv_addr)) == -1) {  
    error_handling("connect() error!");  
}
```

클라이언트는 소켓 생성과, memset() 구조체 변수 선언 부분은 코드가 같고, connect를 통해 listen 하고 있는 서버에 신호를 보낸다. 이후 read(), write()를 통해서 메시지를 주고 받는다.

## MakeFile

```
all:  
    gcc -o client client.c  
    gcc -o server server.c  
clean:  
    rm -f server client
```

all 은 make를 실행할 경우 실행되는 모든 코드 부분이다. gcc를 통하여 서버와 클라이언트의 오브젝트 파일을 생성하였다. clean 부분에서 make clean 코드를 통해 명령어 rm을 통해 생성되었던 오브젝트, 실행파일을 삭제한다.

## 2. 프로그램 실행

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\USER> ssh osc@192.168.56.1
osc@192.168.56.1's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-190-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 package can be updated.
1 update is a security update.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

>> Welcome to the Linux VM for Operating System Concepts!
Last login: Thu Oct  8 10:41:15 2020 from 10.0.2.2
osc@ubuntu:~$ cd homework/a
osc@ubuntu:~/a$ make
gcc -o client client.c
client.c: In function 'main':
client.c:32:26: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
   char buffer[200] = { NULL }; // 버퍼 초기화
                        ^
client.c:32:26: note: (near initialization for 'buffer[0]')
client.c:40:15: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[200]' [-Wformat=]
   scanf("%s", &buffer); // 설문 값 입력
               ^
gcc -o server server.c
server.c: In function 'main':
server.c:49:25: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
   char buffer[200] = {NULL};
                         ^
osc@ubuntu:~/a$ ./server
연결을 기다리는 중...
소켓 연결 성공!
클라이언트 메시지: 1

 * Management:  https://landscape.canonical.com
 * Support:     https://ubuntu.com/advantage

1 package can be updated.
1 update is a security update.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

>> Welcome to the Linux VM for Operating System Concepts!
Last login: Thu Oct  8 10:58:45 2020 from 10.0.2.2
osc@ubuntu:~$ cd homework/d
osc@ubuntu:~/d$ cd
osc@ubuntu:~$ cd homework
osc@ubuntu:~/homework$ cd a
osc@ubuntu:~/a$ ./client
설문 참여한 인원: 1

본인이 생각하는 운영체제 과제의 난이도를 선택해주세요
1. 매우 어렵다
2. 어렵다
3. 보통이다
4. 쉽다
5. 매우 쉽다
번호를 입력해주세요 : 1
osc@ubuntu:~/homework/a$
```

버추얼 머신 하나만으로 서버와 클라이언트를 동시에 실행하는 것은 힘들 것이라 생각하여 ssh를 통하여 접속하여 서버와 클라이언트 두 개를 실행할 환경을 준비했다. 다음은 클라이언트 하나가 접속했을 때이다. 소켓 연결에 성공하면 ‘소켓 연결 성공’ 문구를 서버에서 띄운다.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\USER> ssh osc@192.168.56.1
osc@192.168.56.1's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-190-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 package can be updated.
1 update is a security update.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

>> Welcome to the Linux VM for Operating System Concepts!
Last login: Thu Oct  8 10:41:15 2020 from 10.0.2.2
osc@ubuntu:~$ cd homework/a
osc@ubuntu:~/a$ make
gcc -o client client.c
client.c: In function 'main':
client.c:32:26: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
   char buffer[200] = { NULL }; // 버퍼 초기화
                        ^
client.c:32:26: note: (near initialization for 'buffer[0]')
client.c:40:15: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[200]' [-Wformat=]
   scanf("%s", &buffer); // 설문 값 입력
               ^
gcc -o server server.c
server.c: In function 'main':
server.c:49:25: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
   char buffer[200] = {NULL};
                         ^
osc@ubuntu:~/a$ ./server
연결을 기다리는 중...
소켓 연결 성공!
클라이언트 메시지: 1

 * Management:  https://landscape.canonical.com
 * Support:     https://ubuntu.com/advantage

1 package can be updated.
1 update is a security update.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

>> Welcome to the Linux VM for Operating System Concepts!
Last login: Thu Oct  8 10:58:45 2020 from 10.0.2.2
osc@ubuntu:~$ cd homework/d
osc@ubuntu:~/d$ cd
osc@ubuntu:~$ cd homework
osc@ubuntu:~/homework$ cd a
osc@ubuntu:~/a$ ./client
설문 참여한 인원: 1

본인이 생각하는 운영체제 과제의 난이도를 선택해주세요
1. 매우 어렵다
2. 어렵다
3. 보통이다
4. 쉽다
5. 매우 쉽다
번호를 입력해주세요 : 1
간사합니다.
osc@ubuntu:~/homework/a$
```

클라이언트는 설문조사의 보기에서 번호를 선택하여 전송하면 전송한 결과가 서버의 화면에 출력이 된다. 또한 출력이 완료된 후에 현재 설문에 참여한 인원을 화면에 띄워준다. 전송을 완료한 클라이언트는 자동으로 설문조사가 종료된다.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\USER> ssh osc@192.168.56.1
osc@192.168.56.1's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-190-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 package can be updated.
1 update is a security update.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

>> Welcome to the Linux VM for Operating System Concepts!
Last login: Thu Oct  8 10:41:15 2020 from 10.0.2.2
osc@ubuntu:~$ cd homework/a
osc@ubuntu:~/homework/a$ make
gcc -o client client.c
client.c: In function 'main':
client.c:32:26: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
   char buffer[200] = { NULL }; // 버퍼 초기화
                        ^
client.c:32:26: note: (near initialization for 'buffer[0]')
client.c:40:15: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[200]' [-Wformat=]
   scanf("%s", &buffer); // 설문 값 입력
               ^
gcc -o server server.c
server.c: In function 'main':
server.c:49:25: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
   char buffer[200] = {NULL};
                       ^
server.c:49:25: note: (near initialization for 'buffer[0]')
osc@ubuntu:~/homework/a$ ./server
연결을 기다리는 중...

소켓 연결 성공!
클라이언트 메시지: 1
소켓 연결 성공!
소켓 연결 성공!
```

두 개의 클라이언트가 동시에 접근을 하여도 다중 프로세스를 이용하였기 때문에 끄떡없이 프로그램이 잘 실행된다.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\USER> ssh osc@192.168.56.1
osc@192.168.56.1's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-190-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 package can be updated.
1 update is a security update.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

>> Welcome to the Linux VM for Operating System Concepts!
Last login: Thu Oct  8 10:41:15 2020 from 10.0.2.2
osc@ubuntu:~$ cd homework/a
osc@ubuntu:~/homework/a$ make
gcc -o client client.c
client.c: In function 'main':
client.c:32:26: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
   char buffer[200] = { NULL }; // 버퍼 초기화
                        ^
client.c:32:26: note: (near initialization for 'buffer[0]')
client.c:40:15: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[200]' [-Wformat=]
   scanf("%s", &buffer); // 설문 값 입력
               ^
gcc -o server server.c
server.c: In function 'main':
server.c:49:25: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
   char buffer[200] = {NULL};
                       ^
server.c:49:25: note: (near initialization for 'buffer[0]')
osc@ubuntu:~/homework/a$ ./server
연결을 기다리는 중...

소켓 연결 성공!
클라이언트 메시지: 1
소켓 연결 성공!
소켓 연결 성공!
클라이언트 메시지: 2
클라이언트 메시지: 3
```

두 번 연속으로 설문 내용을 전송해도 전송이 잘 되는 것을 확인할 수 있다. 또한 처음 예시에 이어 두 개의 클라이언트가 동시에 접근했기 때문에 설문 완료 인원이 3명으로 표시되는 것을 확인할 수 있다.

### 3. 개발 HISTORY 및 소감

이번 과제를 하면서 vm 상에서 c파일의 permission denied, 부모 프로세스의 자식 프로세스 두 배 생성 등 많은 고난과 마주했다. 사실 소켓 통신 프로그래밍이라는 과제를 마주했을 때 객체 지향 프로그래밍 당시 했던 ATM 프로젝트가 떠올랐지만 사실 C와 JAVA의 문법은 다르기 때문에 다이어그램, 전체적인 플로우만 알 수 있었다. 전체적인 소켓 통신의 함수는 인터넷과 서적을 참고하여 코드를 완성하였다. 소켓 통신의 코드는 물론 가볍게 짤 수 있었지만, 다중 프로세스는 처음 접해보는 기법인지라 쉽게 접근할 수가 없었다. 이후 많은 인터넷 검색을 통해서 다중 프로세스가 코드 상에서 어떻게 사용되는 것인지 확인할 수 있었다. fork() 함수의 영감은 운영체제 책에서 얻을 수 있었다. 그리고 부모 프로세스가 accept 한 다음 fork()를 통해 자식 프로세스를 만들고 만약 id 가 0(자식프로세스 id)이면 클라이언트와의 통신을 진행하고 0이 아닐 경우에는 부모 프로세스이므로 다시 listen()을 통해 새로운 클라이언트의 접속을 받아들이도록 하였다. 또한 코드 작성 후 c 파일을 실행하려는데, permission denied가 계속해서 발생하였다. 이것은 파일의 속성이 실행을 허용하지 않는 특장값을 갖는다는 것을 알고 명령어 chmod 775를 통해서 파일의 속성값을 변경해주었다.

클라이언트가 설문을 완료할 때 마다 마지막에 현재까지 설문을 진행한 클라이언트의 수를 띄워주도록 코드를 설계 했는데, 수가 1명, 2명, 2명, 3명, 3명, 3명, 3명 이렇게 1,2,4,8~ 2의 제곱수로 떠서 제대로 실행되지 않는다는 것을 알 수 있었다. 코드를 분석하니 부모 프로세스가 만든 자식 프로세스가 다시 while 문을 통해 break 되지 않고 accpet()부터 시작해서

1. 부모 프로세스 -> 자식 프로세스
2. 부모 프로세스, 자식 프로세스->자식 프로세스 \*2
3. 부모 프로세스, 자식 프로세스\*3 ->자식 프로세스\*4

다음과 같이 코드가 실행된다는 것을 알 수 있었다. 따라서 if(id ==0)일 경우에 클라이언트와 통신을 수행하고 통신이 끝나면 break 하여 while 문에서 자식 프로세스를 종료 시키도록 하였다. 그리고 부모 프로세스는 continue 하도록 하였다.

```
people = people + 1;
pid = fork();

if (pid == 0)
{
    read(clnt_sock, buffer, sizeof(buffer));
    printf("클라이언트 메시지: %s\n", buffer);
    write(clnt_sock, "감사합니다.", sizeof("감사합니다.));
    sprintf(buffer, "현재 설문 완료 인원:%d\n", people);
    write(clnt_sock, buffer, sizeof(buffer));
    // memset(buffer, 0x00, sizeof(buffer));

    break;
}
else if (pid != 0)
{
    continue;
}
```

다음으로 처음에는 코드를 통신을 마칠 때 “감사합니다”를 먼저 띄우고 “현재 설문 완료 인원: ” 이렇게 클라이언트로 write()하여 read()하도록 하였다. 하지만 프로그램을 실행시키니 “ 설문 참여한 인원 :” 이 아래와 같이 출력되지 않는다는 것을 확인했다.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\USER> ssh osc@192.168.56.1
osc@192.168.56.1's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-190-generic x86_64)

+ Documentation:  https://help.ubuntu.com
+ Management:     https://landscape.canonical.com
+ Support:        https://ubuntu.com/advantage

1 package can be updated.
1 update is a security update.

Now release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

>> Welcome to the Linux VM for Operating System Concepts!
Last login: Thu Oct  8 01:46:25 2020
osc@ubuntu:~$ cd homework/
osc@ubuntu:~/homework$ cd realSocket/
osc@ubuntu:~/homework/realSocket$ ./server
연결을 기다리는 중...

소켓 연결 성공!
클라이언트 메시지: 2
현재 질문 완료 인원: 2
^C
osc@ubuntu:~/homework/realSocket$ ./server
연결을 기다리는 중...

소켓 연결 성공!
클라이언트 메시지: 1
현재 질문 완료 인원: 1
^C
osc@ubuntu:~/homework/realSocket$ ./server
연결을 기다리는 중...

소켓 연결 성공!
클라이언트 메시지: 1
현재 질문 완료 인원: 1
^C
osc@ubuntu:~/homework/realSocket$

osc@ubuntu:~/homework/realSocket$ ./client
>> Welcome to the Linux VM for Operating System Concepts!
Last login: Thu Oct  8 01:48:33 2020 from 10.0.2.2
osc@ubuntu:~$ cd homework/realSocket/
osc@ubuntu:~/homework/realSocket$ ./client
본인이 생각하는 운영체제 과제의 난이도를 선택해주세요
1. 매우 어렵다
2. 어렵다
3. 보통이다
4. 쉽다
5. 매우 쉽다
번호를 입력해주세요: 1
현재 질문 완료 인원: 1
osc@ubuntu:~/homework/realSocket$

osc@ubuntu:~/homework/realSocket$ ./client
>> Welcome to the Linux VM for Operating System Concepts!
Last login: Thu Oct  8 01:48:33 2020 from 10.0.2.2
osc@ubuntu:~$ cd homework/realSocket/
osc@ubuntu:~/homework/realSocket$ ./client
본인이 생각하는 운영체제 과제의 난이도를 선택해주세요
1. 매우 어렵다
2. 어렵다
3. 보통이다
4. 쉽다
5. 매우 쉽다
번호를 입력해주세요: 4
현재 질문 완료 인원: 3
osc@ubuntu:~/homework/realSocket$

osc@ubuntu:~/homework/realSocket$ ./client
bash: ./client: No such file or directory
osc@ubuntu:~/homework/realSocket$ ./client
본인이 생각하는 운영체제 과제의 난이도를 선택해주세요
1. 매우 어렵다
2. 어렵다
3. 보통이다
4. 쉽다
5. 매우 쉽다
번호를 입력해주세요: 2
osc@ubuntu:~/homework/realSocket$

```

많은 시행착오를 거치다

write(clnt\_sock, "감사합니다.\n", sizeof("감사합니다.\n"));

sprintf(buffer, "설문 참여한 인원:%d\n", people);

write(clnt\_sock, buffer, sizeof(buffer)); 를

sprintf(buffer, "설문 참여한 인원:%d\n", people);

write(clnt\_sock, buffer, sizeof(buffer));

write(clnt\_sock, "감사합니다.\n", sizeof("감사합니다.\n")); 로 수정하였다. 그랬더니 실행이 되었다.

```

people = people + 1;
pid = fork();

if (pid == 0)
{
    read(clnt_sock, buffer, sizeof(buffer));
    printf("클라이언트 메시지: %s\n", buffer);
    sprintf(buffer, "현재 설문 완료 인원:%d\n", people);
    write(clnt_sock, buffer, sizeof(buffer));
    write(clnt_sock, "감사합니다.", sizeof("감사합니다."));
    // memset(buffer, 0x00, sizeof(buffer));

    break;
}
else if (pid != 0)
{
    continue;
}

close(clnt_sock);

```

그 이유를 계속해서 찾아봤지만 이유를 알아낼 수 없었다. 결국 내린 결론은 write()와 read()



를 연속해서 사용하는 과정에서 버퍼에서 오류가 발생했을 것이라고 예상하였다. 결국은 오류의 발생을 막기 위해 연결이 되면 바로 출력하기로 하였다. 그래서 동시에 접속 중이어도 충돌이 일어나는 일이 발생하지 않게 된다.

마지막에 여태까지의 설문 내용을 다 종합하여 비율로 결과를 알려주는 코드를 작성하고 싶었지만, 계속해서 버퍼, 출력오류가 발생하는 바람에 완성하지는 못하였다. 오랜만에 접한 소켓 프로그래밍이라 코드를 작성하는 데에는 어려움이 있었고, 다중 프로세스를 구현하는 과정에서도 많은 시간을 소요했다. 하지만 스스로 끝내 구현에 성공했고, 이 과정에서 사고력이 향상되는 것을 느꼈다.