**네트워크 프로그래밍**

채팅프로그램

20122382

박 준 은

[INDEX]

함수 목록 및 설명

프로그램 구조도

타임라인 다이어그램

메시지 구조

기능 구현 여부

**1. 함수 목록 및 설명**

[server.c]

(1) main()

메인함수이다. 서버의 port번호를 할당해 클라이언트와의 연결을 준비한다. 멀티플렉싱으로 채팅을 한다. Select 함수를 이용해 소켓들의 변화를 감지한다. 만약 리스링 소켓에서의 변화가 감지된다면 로그인 요청이 된 것이므로 login()를 호출한다. 그렇지 않다면 커넥티드 소켓으로부터 메세지가 전송된 것이다. 메세지가 전송된 경우 클라이언트는 처음 보낼때에는 자신의 아이디를 보내고 그다음으로 전송할 메세지를 보낸다. 아이디와 메세지의 크기는 각각 10과 100으로 NAME\_SIZE 와 BUF\_SIZE로 정의되어있다. 메세지를 읽었을 때 처음부분이 @list와 @quit 라면 각각 list()와 logout()을 호출한다. 그렇지 않다면 메세지를 보내기위해 mess()를 호출한다.

(2) login()

리스닝 소켓에서 변화가 감지되면 실행한다. 클라이언트로부터 로그인 요청이 들어온 것이다. Accept를 해주고 크기 NAME\_SIZE만큼 읽어서 사용자의 이름을 가져온다. 그리고 사용자의 이름을 user배열에 저장해 준다. 사용자는 최대 5명까지 로그인 할 수 있으며 커넥티드 소켓의 파일디스크립트 넘버로 이름을 구별한다. 로그인이 완료되었으면 모든 사용자들에게 새로 로그인한 클라이언트의 아이디를 전송한다. 이때 가장 처음에 “log”라는 배열을 크기 5만큼 전송해서 클라이언트가 정보를 구별할 수 있도록 하고 그 다음에 유저의 이름을 NAME\_SIZE 만큼 전송한다. 서버에도 출력한다.

strcpy(tmp,"log\0");

for(j=4;j<fd\_max+1;j++){

write(j,tmp,5);

write(j,user[clnt\_sock],NAME\_SIZE);

}

(3) mess()

커넥티드 소케세서 변화가 감지되어 클라이언트가 메세지를 전송했을 때 호출한다. 먼저 전송된 메세지가 귓속말인지 아닌지를 isWhisper()을 호출해 판별한다. 귓속말이 아닐경우 모든 사용자들에게 메세지를 다시 전송해 준다. 이때 “all”배열을 크기 5만큼 먼저 전송해 클라이언트가 정보를 구별 할 수 있도록 한다. 다음으로 메세지를 보내는 클라이언트의 아이디를 크기 NAME\_SIZE 만큼 전송하고, 끝으로 메세지를 전송한다.

strcpy(tmp,"all\0");

if(!isWhisper()){

for(j=4;j<fd\_max+1;j++){

write(j,tmp,5);

write(j,name, NAME\_SIZE);

write(j, buf, str\_len);

}

}

(4) isWhisper()

귓속말이 전송된 경우이다. @다음에 오는 아이디가 현재 접속된 아이디와 일치하면 귓속말을 보내준다. 서버가 받은 메세지를 strtok()를 이용해 앞부분을 잘라서 아이디만 token 배열에 저장한다. user배열과 strcmp()로 비교해 일치할경우 그 사용자에게만 메세지를 전달 해 준다. 그리고 서버에게 메세지를 보낸 클라이언트에게는 “drc”를 크기 5만큼 전송해 정보를 구별할 수 있도록 하게 하고, 나머지 정보를 받은만큼 보낸다. 그 후 귓속말이 완료되었음을 메세지를 전송한 소켓에 알린다. 모든 전송이 완료되었으면 1을 리턴 해준다. 만약 귓속말이 아닐 경우에는 0을 리턴한다.

(5) list()

현재 접속중인 사용자를 보내준다. 서버가 받은 메세지의 첫부분이 @list 일때 호출된다. 가장먼저 “lst”를 크기 5만큼 클라이언트에게 전송해 정보를 구별할 수 있도록 한다. 그 후 user 배열을 NAME\_SIZE 만큼 for문을 이용해 보낸다.

strcpy(tmp,"lst\0");

write(i,tmp,5);

for(j=4;j<9;j++)

write(i,user[j],NAME\_SIZE);

(6) logout()

서버가 받은 메세지의 첫부분이 @quit 일때 호출된다. “qutit”를 크기 5만큼 모든 사용자들에게 전송해 정보를 구별 할 수 있도록 하고, 사용자의 이름을 크기 NAME\_SIZE 만큼 모든 사용자 들에게 보내준다. 그 후 서버에 사용자가 접속을 종료했다는 글을 띄우고 사용자가 사용하던 아이디를 memset()으로 초기화시킨다. 끝으로 shutdown() 를 이용해 half-close를 한다. 접속을 종료하는 클라이언트가 BYE~라는 메세지를 보내면 서버에 출력하고 close()한다.

shutdown(i,SHUT\_WR); FD\_CLR(i, &reads);

read(i,tmp,5); close(i);

printf("%s\n",tmp);

[client.c]

클라이언트는 함수를 따로 만들지 않았다. 전부 main() 안에 있다. 프로그램이 실행되면 가장 먼저 서버와의 연결을 준비한다. 서버와의 연결이 완료되면 클라이언트는 사용할 아이디를 서버에게 NAME\_SIZE만큼 전송시켜준다. 그 후 본격적인 채팅이 시작된다.

Select()로 소켓의 변화를 감지한다. 서버는 클라이언트에게 메세지를 보낼 때 항상 정보를 구별하는 배열을 크기5만큼 보낸다. 이 배열은 all,list,drc,log,quit 으로 이루어져 있다. Strcmp()를 이용해 구별한다.

All은 서버가 모든 사용자들에게 메세지를 전송할 때 사용된다. 서버는 사용자 이름을 크기 NAME\_SIZE 만큼 먼저 전송하고 다음으로 메세지 내용을 전송한다. 클라이언트는 차례로 읽어들인 후 출력한다.

str\_len=read(sock, message, NAME\_SIZE);

message[str\_len]=0;

printf("<%s> : ", message);

str\_len=read(sock, message, BUF\_SIZE);

message[str\_len]=0;

printf("%s", message);

list는 서버가 접속중인 사용자 명단을 보낼 때 사용한다. 사용자는 서버에 최대 5명까지 있으므로 크기 50만큼 for문을 이용해 받아온다. 만약 받아온 배열이 null이 아니라면 사용자 이름을 출력해준다.

while(str\_len<50){

str\_len += read(sock,message,NAME\_SIZE);

if(message[0] != '\0')

printf("<%s> ",message);

}

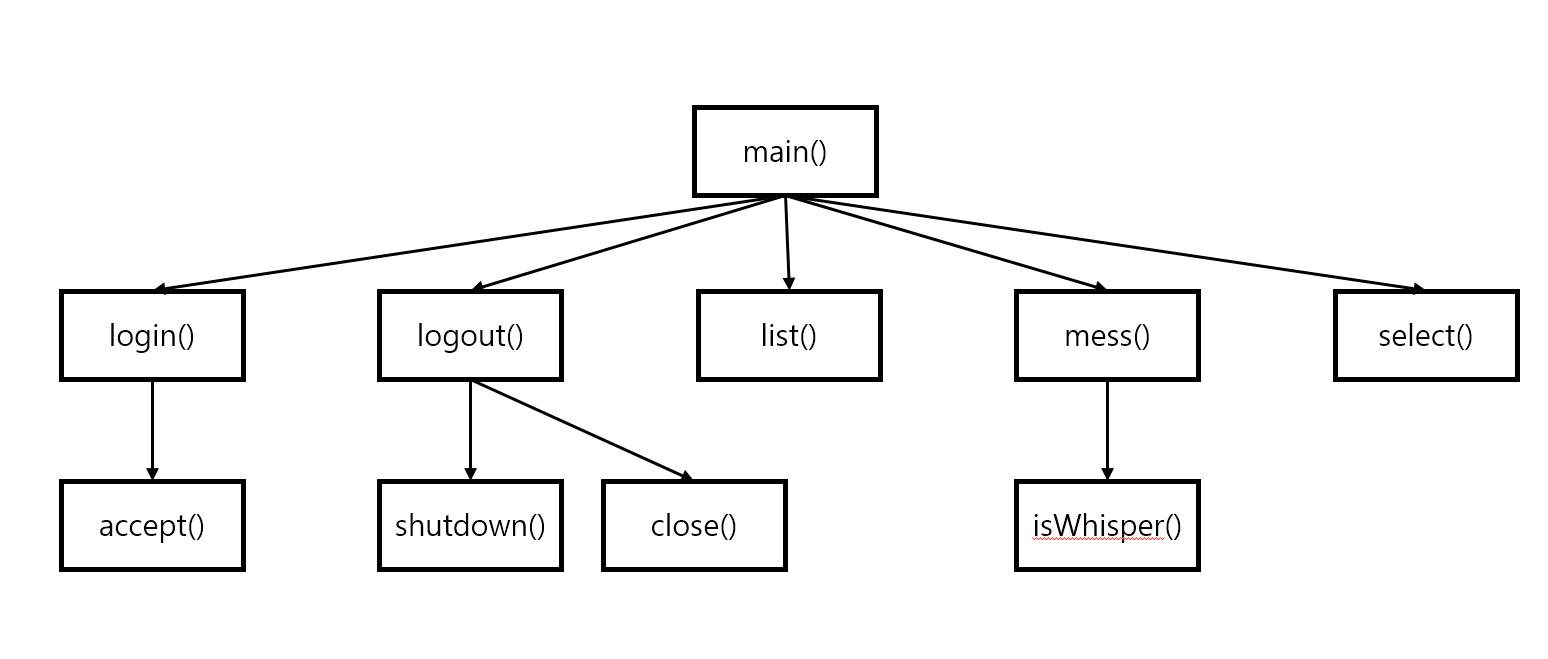
drc는 이전에 클라이언트가 귓속말을 보냈을 때 서버가 귓속말이 잘 전달되었음을 알려주는 정보이다. 화면에 귓속말이 잘 전달되었다는 문구를 띄워준다.

log는 사용자가 로그인했을 때 알려주는 정보이다. 사용자 이름을 NAME\_SIZE만큼 읽어와서 내 화면에 출력해준다.

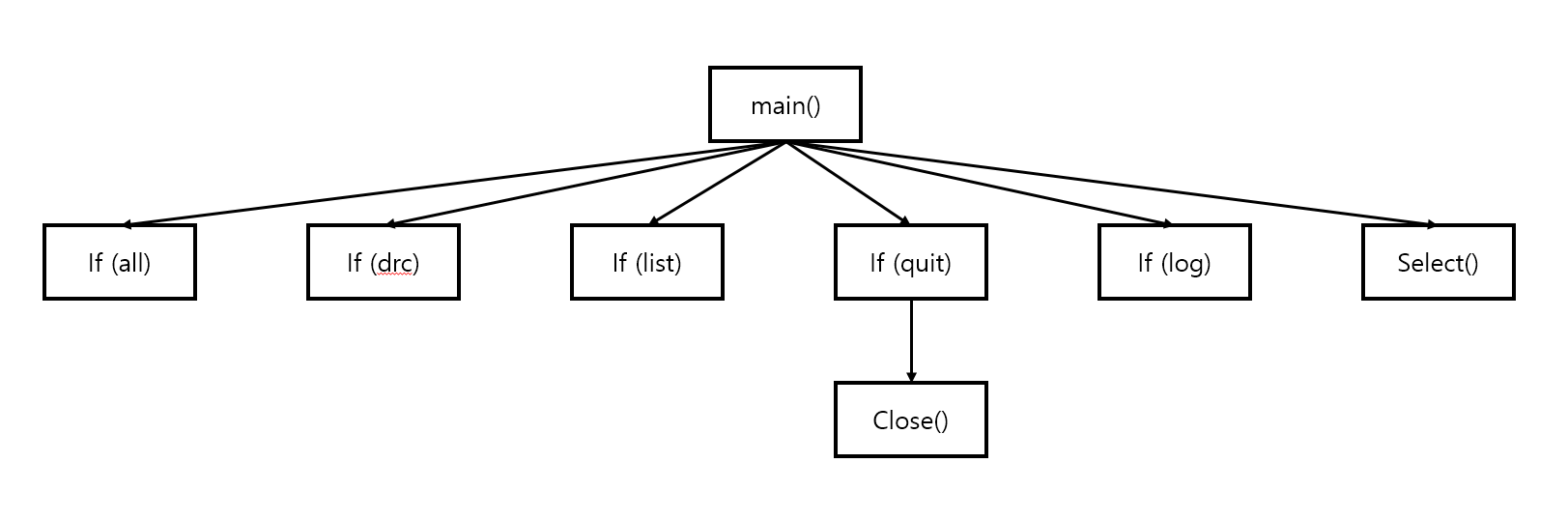
quit는 사용자가 로그아웃을 했을 때 알려주는 정보이다. 사용자 이름을 NAME\_SIZE만큼 읽어온다. 다른 클라이언트가 접속을 종료한 거라면 접속을 종료한 클라이언트 이름띄워주고, 만약 내가 접속을 종료한 것이라면 서버에게 BYE~ 를 전송하고 소켓을 닫아준다.

**2. 프로그램 구조도**

Server

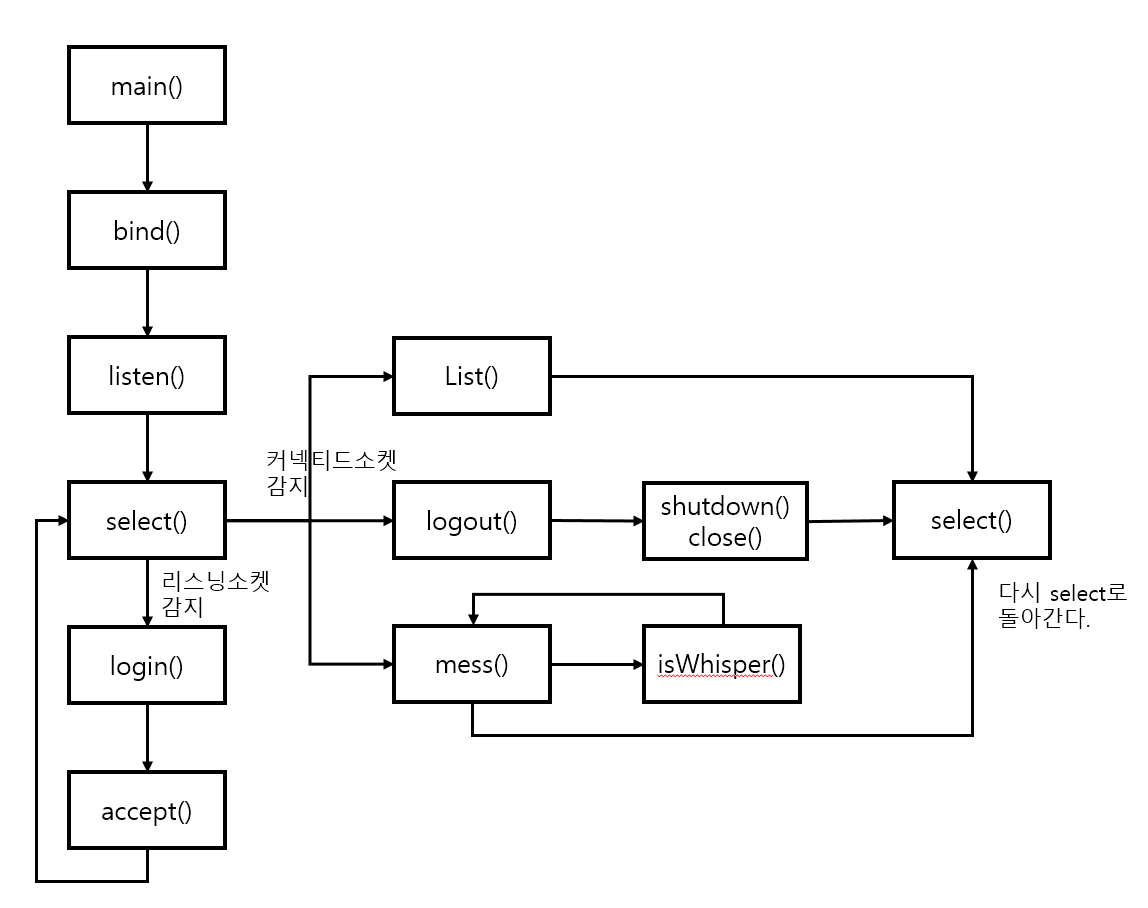


Client

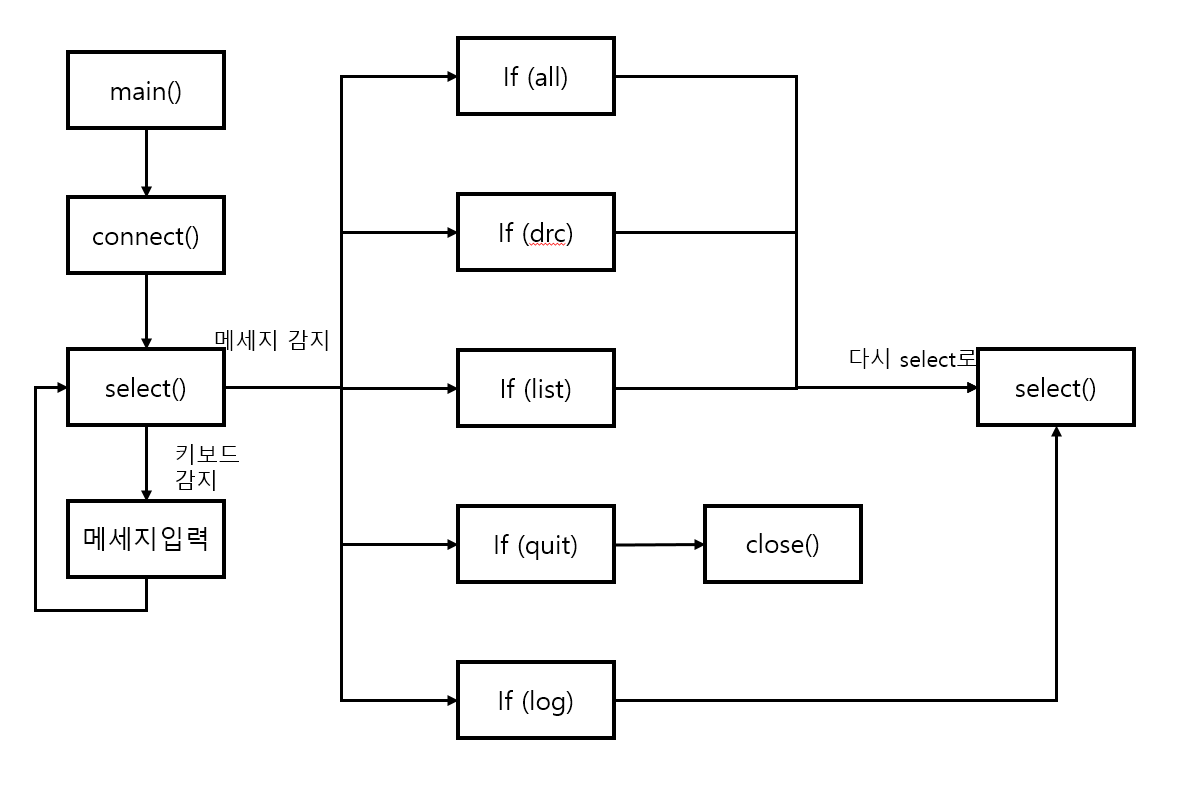


**3. 타임라인 다이어그램**

Server



client



**4. 메시지 구조**

서버에서 보낼때에는 가장먼저 무조건 5의 크기만큼 정보를 알려주는 값을 보낸다. 클라이언트는 그것을 보고 자신이 받은 정보를 판별하고 대응한다.

받은길이 10 5

정보

사용자 이름

메세지 내용

위의 그림은 서버가 클라이언트들에게 메세지를 보낼 때와 귓속말 할때의 구조이다. 정보부터 보낸다.

사용자 이름

정보

위의 그림은 list 보내줄 때와 logout 할 때와 login 할 때의 구조이다.

클라이언트는 항상 다음과 같은 메세지 구조를 서버에 보낸다.

메세지 길이 10

사용자 이름

메세지 내용

**5. 기능 구현 여부**

|  |  |  |
| --- | --- | --- |
| **기능** | **완료** | **미완** |
| **로그인** | **o** |  |
| **커맨드** | **모두완료** |  |
| **다중접속 지원** | **o** |  |
| **클라이언트 메세지 출력** | **모든메세지 출력** |  |

**모든기능 사용가능.**

**스크린샷**

