

IOT프로그래밍 3차과제

20150327 이진수

1. 실행 방법

서버의 경우

- gcc -o chat_server chat_server.c -pthread
- ./chat_server <PORT> ex) ./chat_server 54321

클라이언트의 경우

- gcc -o chat_client chat_client.c -pthread
- ./chat_client <IP> <PORT> <NAME> ex) ./chat_client 127.0.0.1 54321 C1

2. 서버와 클라이언트의 하는 일

채팅 서버의 경우 총 3가지의 일을 한다.

- 새로운 클라이언트의 접속 허용
- 클라이언트에게 전달 받은 DATA를 다른 클라이언트에게 전송
- 기존의 클라이언트가 접속을 종료하면 종료한 클라이언트를 서버에서 삭제

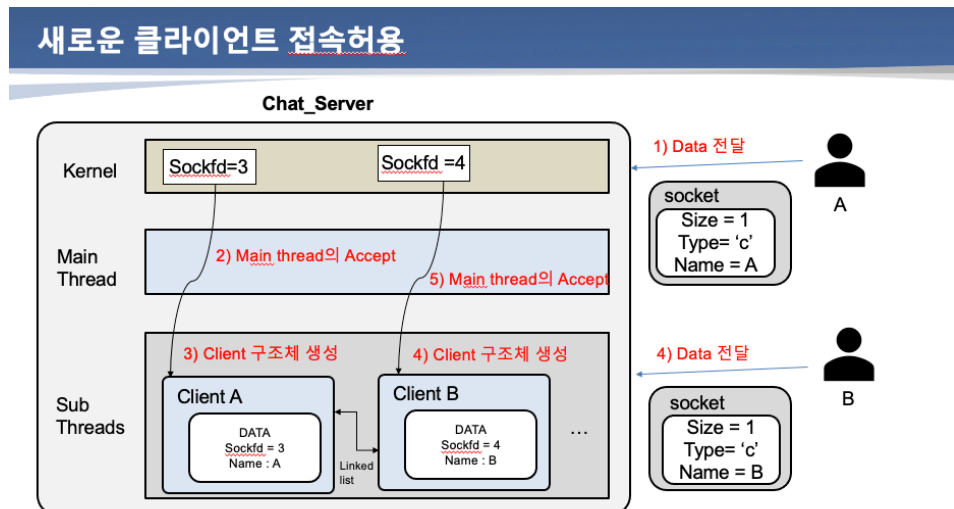
클라이언트의 경우 2가지의 일을 한다.

- Server로 부터 소켓을 연결한 뒤 server로부터 오는 내용 받아오기
- Server에게 키보드로 입력한 데이터 전달하기

3. 구현방법

A. 서버

1. 새로운 클라이언트의 접속 허용



새로운 클라이언트의 접속의 경우 Main Thread에서 accept를 한뒤 sub Thread를 통해서 socket번호와 각 클라이언트의 주소를 가져온다. Sub Thread를 사용하여 client 구조체에 소켓번호와 이름을 기억하여 저장한다.

2. 클라이언트에게 전달 받은 Data를 다른 클라이언트에게 전송

Socket을 통해 전달 받은 데이터를 server에 head_pointer를 통해서 linked list로 구현되어있는 각 클라이언트를 돌면서 Data를 모든 클라이언트에게 전송

3. 기존의 클라이언트가 접속을 종료하면 종료한 클라이언트를 서버에서 삭제
Linked list에 연결되어있는 것을 끊어서 앞뒤에 있는 노드들을 서로 연결 시켜준
뒤 접속을 종료한 클라이언트를 free함

B. 클라이언트

1. Server로 부터 소켓을 연결한 뒤 server로 부터 오는 내용 받아오기
Connect로 소켓을 연결했다면 recv로 server가 보내는 것을 받아옴
2. Server에게 키보드로 입력한 데이터 전달하기
Detach한 thread를 통해서 키보드 입력을 받은뒤 입력받은 데이터를 server로
send하면 됨