

MFC MongoDB 카카오톡

김주리

클라이언트 : https://github.com/zlxlwnfl/MFC_Chat_Client

서버 : https://github.com/zlxlwnfl/MFC_Chat_Server

CONTENTS

1

프로그램 개요

2

맡은 부분

- 클라이언트 구현 (MFC)
- 서버 구현 (MongoDB)

3

어려웠던 점 / 해결방법

- MFC
- MongoDB C++ 드라이버

프로그램 개요

- MFC MongoDB 카카오톡 프로그램 (제작중)
- <제작자>
김주리
- <기술스택>
운영체제 : 윈도우 10(클라이언트), 리눅스 우분투 18.04(서버)
프로그래밍 언어 : C++
윈도우 프레임워크 : MFC
데이터베이스 : MongoDB
포맷 변환 라이브러리 : JsonCpp
- 회원가입/로그인/유저 채팅 리스트/채팅방 통신 구현 완료

구현

The image displays two side-by-side form mockups, each with a standard window header (title bar, maximize, minimize, close buttons) and a close button (X) in the top right corner.

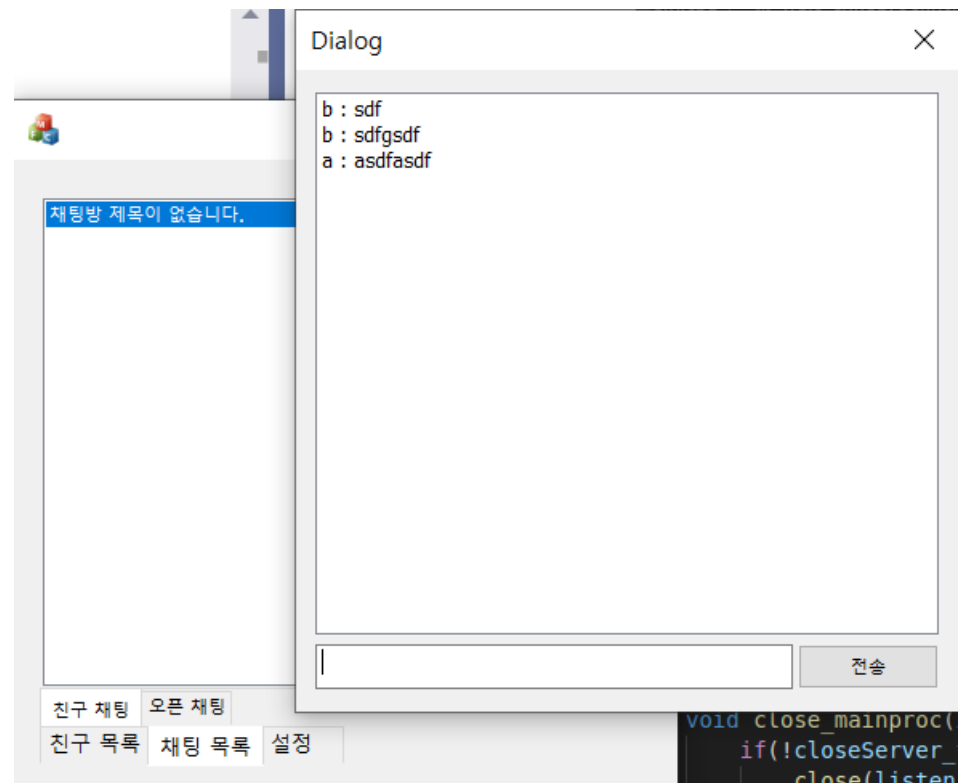
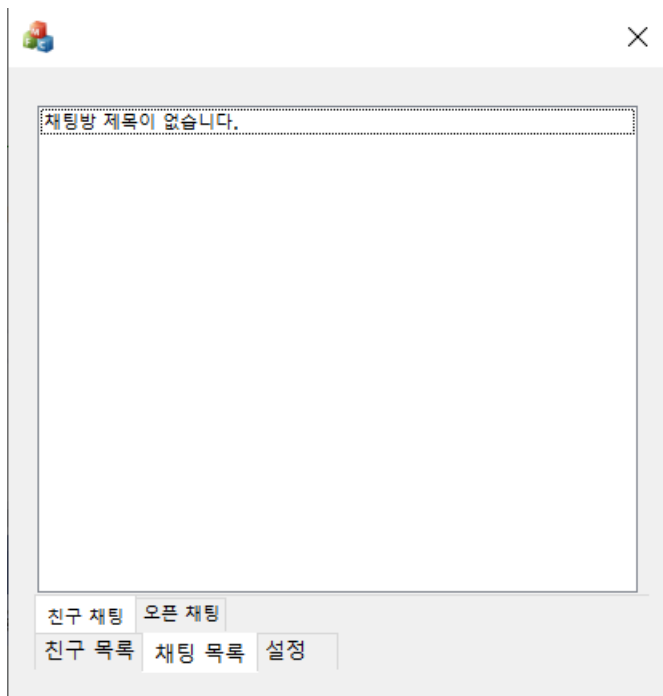
Left Form (Login/Sign In):

- Labels: ID, PASSWORD
- Inputs: Two text input fields, one for ID and one for PASSWORD.
- Buttons: Two buttons, "SIGN IN" and "SIGN UP", stacked vertically.

Right Form (Registration/Sign Up):

- Labels: ID, PASSWORD, PASSWORD CHECK
- Inputs: Three text input fields, one for ID, one for PASSWORD, and one for PASSWORD CHECK.
- Buttons: Two buttons, "OK" and "CANCEL", stacked vertically.

구현



```
void close_mainproc(  
    if(!closeServer  
        close(listen
```

구현

■ 서버 구현

- 메인/컨트롤러/서비스 로 영역을 나누어 각각 기능 분배(컨트롤러/서비스 클래스화)
- 서버에 클라이언트가 접속할 때마다 새로운 프로세스를 생성하여 다중 멀티 접속 서버 구현(멀티스레드로 변경 예정)
- 새로 생성된 프로세스는 새로운 컨트롤러 객체를 생성하여 이를 통해 서버 기능 처리(FrontController 패턴 참고)
- 서버는 기본적으로 무한 루프를 돌며(새로운 클라이언트를 계속 받기 위해), 종료는 시그널 함수를 이용하여 처리

■ 컨트롤러 구현

- 컨트롤러 객체는 서비스 클래스의 객체를 멤버 변수로 갖고 있음
- recv 함수의 플래그로 'MSG_DONWAIT'을 주어 논블로킹 방식으로 처리
- 컨트롤러도 무한 루프를 돌며 서버 기능을 계속해서 처리하며, 클라이언트가 연결 종료 시, 지정된 type의 메시지를 보내게 하여 이를 받을 시 컨트롤러 종료 처리
- 받은 메시지와 보낼 메시지는 각각 수신 큐, 전송 큐를 생성하여 배치 방식으로 관리
- 서버와 클라이언트 간에는 type이 명시된 메시지를 주고 받음 (type의 예 : 회원가입, 로그인, 유저 채팅 리스트 등등)
- 메시지의 type을 switch 함수로 체크하여, 각각에 맞는 기능을 서비스 객체로 처리

■ 서비스 구현

- 서비스 객체는 서버의 구체적인 기능을 각각 메소드로 갖고 있음
- 나중에 스레드화할 것을 고려하여, 서비스 클래스 안에는 공유되는 멤버 변수 최소화(크리티컬 섹션 제거 목적)
- 서비스는 MongoDB와 연결하여 각종 기능을 처리

어려웠던 점 / 해결방법

■ MFC

- 윈도우 간 적절한 함수 호출 방법을 찾아 많이 헤멤
- 커스텀 메시지 방식이 제일 안전하고 괜찮은 것 같아,
모든 윈도우 간 함수 호출은 커스텀 메시지로 통일

■ MongoDB C++ 드라이버

- 설치 시, c++ 드라이버만 까니까 계속 오류가 나서 많이 헤멤
- 구글링 후, c 드라이버도 깔아야 정상 작동 한다는 걸 알게 됨

■ 전체적인 서버 구조 설계

- 현재 코드가 너무 비효율적이고 많이 부족한 것 같아 다중 채팅 서버 설계에 대해서 많이 알아봄
- 효율적인 다중 채팅 구현을 위해 멀티스레드화 및 다중 채팅방 클래스 따로 구현할 예정

Thank You