<채팅 서버 최종보고서>

과목명:객체지향프로그래밍

교수님: 조현철

팀　명: 상속자들

팀　장: 5881452 오예림

팀　원: 5881253 김인혜 | 5881603 최인경 | 5881298 김형민

1. 프로젝트 개요

본 프로젝트는 TCP 기반의 실시간 채팅 서버와 클라이언트를 C++ 언어로 구현한 것이다. 사용자는 회원가입과 로그인 후 서버에 접속하여 친구와 채팅하거나, 채팅방에 입장하여 실시간 대화를 주고받을 수 있으며, 명령어를 통해 간단한 게임 기능도 수행할 수 있도록 설계하였다. 서버는 멀티스레드 기반으로 다수의 클라이언트를 동시에 처리하며, 통신은 명령어 기반의 문자열 프로토콜로 이루어졌다. 본 시스템은 객체지향 설계 원칙을 바탕으로 설계되었으며, 사용자 관리, 친구 시스템, 채팅, 게임 기능이 클래스로 구분되어 모듈화되어 있다.

2. 개발 목적

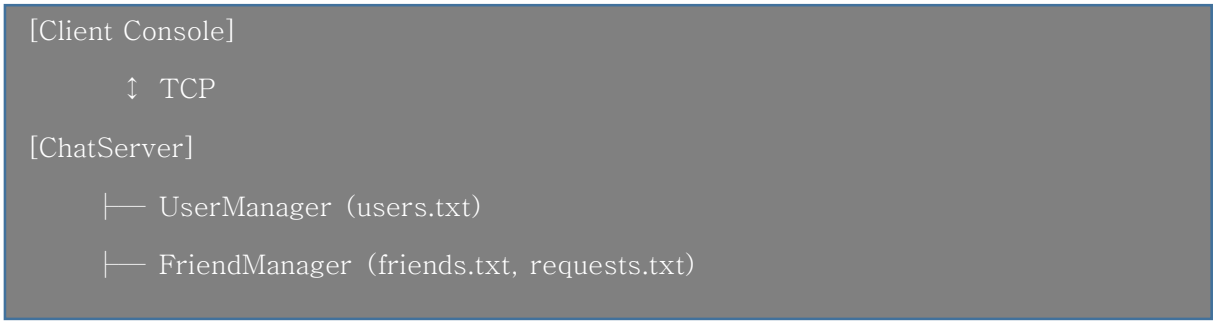
이 프로젝트는 네트워크 프로그래밍과 객체지향 설계 원리를 학습하고 실습하기 위한 목적으로 수행되었다. 주요 목표는 다음과 같다.

1. 객체지향 설계를 실제 네트워크 서비스에 적용하여 유지보수성과 확장성이 높은 코드를 작성하는 것이다.
2. WinSock2를 이용한 TCP/IP 통신 기반의 서버-클라이언트 모델을 구현하고, 실시간 메시지 송수신 기능을 직접 설계하고 테스트하는 것이다.
3. 스레드와 동기화 기법을 활용하여 멀티클라이언트 환경에서 발생하는 동시성 문제를 해결하는 것이다.
4. 사용자 중심의 콘솔 UI와 기능 중심 메뉴를 통해 직관적이고 반응성 높은 사용 환경을 구현하는 것

3. 개발 환경

항목	내용
운영체제	Windows 11
개발 언어	C++
컴파일러	Visual Studio 2022
통신 방식	TCP (WinSock2)
동시성 처리	std::thread, std::mutex
저장 방식	텍스트 파일 기반 (users.txt 등)

4. 시스템 아키텍처



5. 객체지향 설계

주요 클래스 및 역할

- ChatClient: 클라이언트 측에서 사용자 입력을 받아 서버에 전송하고, 서버로부터의 응답을 출력한다. 콘솔 기반 UI와 채팅방 입장 기능을 제공하며, 메시지 수신은 별도의 스레드에서 수행된다. 출력 시 콘솔 색상, 시간, 보낸 사람 정보 등을 함께 표시하도록 구현되어 있다.
- ChatServer: TCP 기반의 멀티스레드 서버 클래스이다. 클라이언트의 명령을 파싱하고, 그에 따라 사용자 로그인, 친구 요청, 채팅 송수신, 게임 처리 등의 기능을 수행한다. 사용자 상태를 관리하고, 각 클라이언트를 독립적인 스레드로 처리한다.
- UserManager: 회원가입 및 로그인, 로그인 상태 관리, ID 중복 검사 등을 수행하며, users.txt파일을 통해 사용자 정보를 영구 저장한다.
- FriendManager: 친구 요청, 수락, 친구 목록 조회 기능을 담당하며, 요청 및 친구 목록은 requests.txt와 friends.txt파일에 각각 저장된다.
- 게임 기능은 별도의 클래스로 분리되지 않고, ChatServer클래스 내부의 전용 영역에서 업다운 게임, 스피드 게임로직을 처리한다.

객체지향 설계 원칙 적용

- 캡슐화: 모든 클래스는 내부 데이터를 private또는 protected로 선언하고, public메서드를 통해서만 접근하도록 제한하였다.
- 추상화: 사용자 관리, 친구 시스템, 채팅, 게임 기능을 모듈화하여 복잡한 로직을 은닉하고 외부에서 단순한 명령으로 호출할 수 있도록 설계하였다.
- 단일 책임 원칙(SRP): 각 클래스는 하나의 명확한 역할만을 수행하며, 책임 분리가 명확하다. 예: UserManager는 사용자 관리만 담당.
- 동시성 제어: std::mutex를 통해 사용자 정보, 친구 관계, 게임 상태 등에 대한 동시 접근을 제어하여 스레드 간 충돌을 방지하였다.
- 의존성 관리: 서버 내부에서 각 관리 클래스를 포함하여 직접 호출하는 구조이며, 인터페이스 분리는 명확하지 않지만, 구조적으로 기능별 모듈화를 유지하고 있다.

6. 주요 기능 설명

클라이언트 기능

- 회원가입 및 로그인
    - /signup [ID] [비밀번호]명령으로 회원가입을 요청
    - /login [ID] [비밀번호]명령으로 로그인 수행 후 메인 메뉴 진입
- 친구 시스템
    - /friendrequest [ID]로 친구 요청
    - /viewrequests로 받은 요청 확인 후, /acceptfriend [ID]로 수락 가능

- /getfriends로 친구 목록 및 접속 상태 확인

- 채팅 기능
  - /entertalk명령으로 채팅방 입장 후 실시간 채팅 가능
  - [닉네임] 메시지형태로 채팅 출력되며, 귓속말(/귓속말)과 게임 메시지는 색상으로 구분됨
- 게임 기능
  - /게임 업다운: 1~100 범위의 숫자 맞히기 게임
  - /게임 스피드: 무작위 숫자 입력 경쟁 게임
  - /입력 [숫자]로 게임 응답 제출 가능
- 콘솔 기반 UI
  - 시작 메뉴, 메인 메뉴, 친구 메뉴를 구성하고 사용자는 숫자 입력을 통해 조작
  - 채팅 메시지 수신은 실시간으로 별도 스레드에서 수행
  - 시간, 발신자, 메시지 내용을 출력 시 함께 표시

서버 기능

- 클라이언트 연결 및 스레드 분기 처리
  - 각 클라이언트는 별도의 스레드에서 독립적으로 처리됨
- 사용자 정보 관리
  - 로그인 상태 추적, ID 중복 검사, 파일 기반 정보 저장
- 친구 기능 처리
  - 요청, 수락, 목록 조회 기능을 처리하며, 파일을 통해 데이터 유지
- 채팅 메시지 브로드캐스트
  - 채팅방 참가자들에게 메시지를 일괄 송신하며, 귓속말 및 게임 메시지는 구분 처리
- 게임 처리
  - 업다운 및 스피드 게임 로직을 자체 구현하며, 참여자 추적, 승자 판단 등 기능 수행
- 에러 및 예외 처리
  - 잘못된 명령, 접속 해제, 게임 중 중복 요청 등 다양한 상황에 대한 대응 메시지 출력

7. 테스트 시나리오

- 회원가입 중복 ID 확인: 같은 ID로 두 번 /signup→ 오류 메시지 출력 확인
- 친구 요청 및 수락: 한 명이 요청, 상대방이 수락 후 /friend_list로 상태 확인
- 채팅방 색상 출력 확인: /빨강 안녕입력 시 색상 출력 여부 확인
- 업다운 게임 테스트: /입력숫자명령 후 정답 여부에 따른 응답 출력 확인
- 서버 자동 검색 기능 테스트: 클라이언트에서 브로드캐스트로 서버 IP 자동 탐색

8. 팀원 역할 분담

이름	역할
오예림	팀장, 보고서 작성, 포스터 제작
김인혜	자료 조사
김형민	발표, PPT 제작
최인경	회의록 작성
공동 작업	코드는 전원이 협력하여 구현하였음

9. 제외된 코드 및 제외 사유(git-hub 참조)

- 오예림 코드: 서버 구조 및 게임 초안 설계에 기여하였으며, 메시지 처리 흐름을 먼저 구상하였다. 그러나 구조가 최종 통합 구조와 달라 직접 통합되지는 않고 로직 일부만 반영되었다.
- 김형민 코드: TCP 채팅과 로그 저장 기능이 구현되어 있으나 로그인 및 친구 기능이 누락되어 제외되었다.
- 최인경 코드: 기능이 통합된 형태로 작성되었으나 채팅방 입장 흐름 및 UI 흐름 처리에서 프로젝트 방향성과 차이가 있어 일부 로직만 참조되었다.
- 기타 테스트 코드: 네트워크 연결 검증 목적의 단일 TCP 연결 코드로 최종 기능 구현에는 포함되지 않았다.
- 최종 선발된 코드를 중점적으로 개발하였다.

10. 향후 개선 및 확장 계획

- GUI 기반 클라이언트로 전환하여 사용자 친화성 강화
- 사용자 인증 보안을 위한 비밀번호 암호화 및 세션 기반 로그인 적용
- 채팅방 분리 구조 도입을 통한 다중 채팅방 지원
- SQLite 또는 JSON 기반 저장 시스템 도입
- 웹소켓 기반 구조 확장을 통한 플랫폼 독립성 확보

11. 결과 및 회고

이번 프로젝트를 통해 객체지향 설계, TCP 네트워크 통신, 멀티스레딩과 동기화, 사용자 인터페이스 설계 등 소프트웨어 공학 전반에 대한 실전 경험을 쌓을 수 있었다. 특히 코드의 구조화, 역할 분담의 명확화, 기능별 모듈 분리를 통해 협업의 중요성을 체감하였다. 또한 예외 처리, 사용자 편의성 향상, 커맨드 설계 등 세부적인 구현에서의 고민은 향후 보다 안정적이고 확장 가능한 소프트웨어를 개발하는 데 밑거름이 될 것이다. 본 프로젝트는 네트워크 기반 실시간 통신 시스템의 핵심 개념을 실습한 귀중한 경험이었으며, 학문적 기반 위에 실무적 감각을 더하는 계기가 되었다.