

개 발 계 획 서					
프로젝트 분야/주제	콘솔 기반 사용자 인증 및 도서정보 서비스 시스템 개발			총 투입인원	1 명
프로젝트 명	국문	LUMIS: 콘솔 기반 사용자 인증 및 도서정보 메뉴형 인터페이스 시스템			
	영문	Login & User/Menu-based Information System			
수행기간	2025. 04. 28 ~ 2025 . 05. 17				
참여자	소속	참여자성명	학년	학번	이메일
	컴퓨터공학부	이승준	3	2021243019	asraheesin@naver.com
<div>과제목표</div> <p>가) 필요성</p> <ul style="list-style-type: none"> 비대면 환경 확대 및 디지털 정보 접근 수요 증가 간결한 콘솔 인터페이스 기반 사용자 중심 정보 접근 구조 필요 구조체 기반 파일 처리 방식 활용을 통한 데이터 구조화 및 검색 효율성 확보 인증 기반 접근 절차 및 텍스트 기반 명령 흐름 설계를 통한 사용자 조작 명확화 <p>나) 개발 목표</p> <ul style="list-style-type: none"> 콘솔 환경에서 사용자 인증, 도서 검색, 평점 등록, 회원정보 수정·삭제 기능 구현 서버는 socket(), bind(), listen(), accept() 함수 기반 TCP 연결 처리 클라이언트 요청 수신 후 users.txt, booklist2-2.txt 파일 접근 및 구조체 변환 수행 문자열 비교 및 검색 기능은 strcmp(), strstr(), tolower() 함수 기반으로 구현 정렬 기능은 qsort() 알고리즘을 활용하여 평점 기준 내림차순 정렬 처리 클라이언트에서는 _beginthreadex()로 송수신 스레드 분리 구성 서버는 각 요청에 대해 독립 스레드로 처리하며, 블로킹 기반 recv() 함수 사용 키보드 입력 기반 명령어 처리 및 콘솔 출력 기반 사용자 인터페이스 구성 <p>다) 개발결과 창출</p> <ul style="list-style-type: none"> 콘솔 기반 도서정보 관리 시스템 구현 구조체 배열 기반 사용자 및 도서 데이터 관리 구조 확립 텍스트 기반 사용자 인증 흐름 및 키워드 검색 기능 구현 평점 기준 정렬 및 상위 랭킹 출력 기능 탑재 파일 동기화는 fgets(), fputs(), rewind() 활용 방식으로 처리 명령어 기반 입력 방식과 출력 응답 포맷 통일을 통한 사용성 확보 서버-클라이언트 간 간결한 문자열 기반 통신 프로토콜 적용 교육용 및 실습용 구조체 기반 콘솔 시스템으로 활용 가능 					

[표 1] 프로젝트 기본 정보 요약 표

항목	내용
프로젝트 명	LUMIS: 구조체 기반 콘솔형 도서정보 처리 시스템
개발 환경	Windows 11, VSCode, GCC
통신 방식	TCP/IP 기반 소켓 통신
개발 언어	C(Windows API 포함)
데이터 저장 방식	텍스트 파일 기반 구조체 처리

개발내용 요약

※ 사용자 인터페이스 기반 도서정보관리 시스템 구성 요소 및 기능에 대한 요약 정리

가) 사용자 측 구성 요소 및 기능

- 사용자 입력 및 연결 설정
 - 사용자로부터 서버 IP, 포트, 닉네임 입력을 순차적으로 수신
 - 닉네임 입력 후 sprintf()를 통해 포매팅하여 서버에 전송
 - connect() 호출 전 sockaddr_in 구조체에 접속 정보 저장
- 송신 및 수신 스레드 생성
 - _beginthreadex() 호출로 송신 및 수신 스레드를 각각 생성
 - 송신 스레드는 키보드 입력을 fgets()로 수신한 후 send()로 송신
 - 수신 스레드는 recv()로 메시지를 수신한 후 fputs()로 출력
- 주요 기능 요청
 - 사용자 명령에 따라 로그인, 회원가입, 도서 검색, 정렬, 평점 등록 기능 수행
 - 종료 요청 시 송수신 스레드 종료 대기 후 closesocket(), WSACleanup() 호출
- 사용자 명령 UI
 - 텍스트 기반 콘솔 인터페이스 구성
 - 명령어 목록을 상시 출력하여 사용자의 혼란 방지 및 입력값에 따른 예외 처리 수행

나) 서버 측 구성 요소 및 기능

- 서버 초기화 및 연결 수락
 - socket(), bind(), listen() 호출을 통해 서버 소켓 생성 및 포트 바인딩
 - 클라이언트 접속 요청은 accept()로 수락하며, 요청마다 _beginthreadex()로 독립 스레드 생성
- 데이터 파일 불러오기 및 구조체 변환
 - 사용자 정보는 users.txt, 도서 정보는 booklist2-2.txt에서 fgets()로 수신 후 구조체 배열에 저장
- 사용자 요청 처리
 - 로그인, 회원가입, 사용자 수정·삭제 요청은 strcmp() 및 strcpy() 기반 문자열 비교와 파일 재작성 방식으로 처리
 - 도서 검색은 strstr() 기반 키워드 탐색 수행
 - 평점 등록은 도서 구조체에 평점 필드를 추가하여 누적 및 평균 처리
 - 정렬 요청 시 qsort() 호출을 통해 내림차순 정렬 수행

○ 응답 송신 및 종료 처리

- 사용자 요청 응답은 send() 함수로 송신하며, 문자열 구분 기호를 활용한 명령·데이터 구분 적용
- 종료 시 모든 소켓 자원 해제 및 서버 루틴 종료 루프 유지

[표 2] 기능 구성 요약 표

구분	주요 기능	설명
사용자	로그인 처리	서버로 ID/PW 전송 후 응답 처리
사용자	회원가입	ID 중복 확인 및 사용자 정보 전송
사용자	도서 검색	키워드 입력 → 검색 결과 수신 및 출력
사용자	도서 정렬	평점 기준 정렬 요청 후 순위 출력
서버	로그인 응답	ID/PW 확인 → 성공 여부 응답
서버	회원가입 처리	중복 확인 → 구조체 및 파일에 저장
서버	도서 처리	검색, 정렬, 평점 등록 등 기능 수행

상세 개발 사항

가) 클라이언트 프로그램 구성

○ 사용자 입력 및 연결 설정

- 사용자로부터 서버 IP, 포트, 닉네임 입력을 순차적으로 수신
- 닉네임 입력 후 sprintf()를 통해 포맷팅하여 서버에 전송
- connect() 호출 전 sockaddr_in 구조체에 접속 정보 저장

○ 송신 및 수신 스레드 생성

- _beginthreadex() 호출로 송신 및 수신 스레드를 각각 생성
- 송신 스레드는 키보드 입력을 fgets()로 수신한 후 send()로 송신
- 수신 스레드는 recv()로 메시지를 수신한 후 fputs()로 출력

○ 클라이언트 주요 기능 처리

- 사용자 명령에 따라 로그인, 회원가입, 도서 검색, 정렬, 평점 등록 기능 수행
- 로그인 요청 시 입력값을 서버에 전송한 후 응답값을 기반으로 성공 여부 판단
- 회원가입 요청 시 중복 확인, 포맷 검증 후 서버에 정보 전송
- 도서 검색은 키워드 입력을 통해 서버로 요청 전송, 서버의 검색 결과를 수신 후 출력
- 정렬 기능은 평점 기준 내림차순 정렬 요청을 서버로 전송하고, 순위 결과 수신 후 출력
- 종료 요청 시 송수신 스레드 종료 대기 후 closesocket(), WSACleanup() 호출

나) 서버 프로그램 구성

○ 서버 초기화 및 연결 수락

- socket(), bind(), listen() 호출을 통해 서버 소켓 생성 및 포트 바인딩
- 클라이언트 접속 요청은 accept()로 수락하며, 요청마다 _beginthreadex()로 독립 스레드 생성

○ 데이터 파일 불러오기 및 구조체 변환

- 사용자 정보는 users.txt, 도서 정보는 booklist2-2.txt에서 fgets()로 수신 후 구조체 배열에 저장
- 사용자 구조체에는 ID, 비밀번호, 닉네임 정보를 포함하고, 도서 구조체에는 제목, 저자, 평점, 등록 수 필드 구성

○ 클라이언트 요청 처리

- 로그인 요청 시 사용자 ID와 비밀번호 일치 여부를 strcmp()로 확인하여 응답 송신

- 회원가입 요청 시 ID 중복 여부 확인 후 구조체 배열에 추가하고 파일로 갱신
- 사용자 수정 및 삭제는 요청에 따라 구조체 정보 갱신 및 해당 정보 제외 후 파일 재작성
- 도서 검색은 strstr() 기반 키워드 탐색 수행 후 일치 항목 필터링하여 응답 송신
- 평점 등록은 해당 도서 구조체의 평점과 평가 수를 누적한 후 평균 계산 및 파일 갱신
- 정렬 요청 시 qsort() 호출을 통해 평점 기준 내림차순 정렬 처리 후 순위 응답 송신

○ 응답 송신 및 종료 처리

- 클라이언트 요청 처리 후 send()를 사용해 응답 송신, 문자열 기반 명령 구분 적용
- 서버는 다중 클라이언트 지원을 위해 각 요청을 독립 스레드에서 처리
- 종료 조건에 따른 소켓 정리 후 closesocket()과 WSACleanup()을 호출하여 자원 해제
- 서버 루프는 외부 입력 없이 무한 루프로 유지되며, 명시적 종료는 포함하지 않음

[표 3] 기술 요소 구성표

기술 요소	적용 위치	설명
socket(), bind(), listen()	서버 초기화	TCP 소켓 서버 구성
send(), recv()	양방향 통신	클라이언트-서버 간 메시지 전송
구조체 배열	서버 내부	사용자 및 도서 데이터 관리
파일 입출력	서버 초기화/종료 시	users.txt, booklist2-2.txt 읽기/쓰기
_beginthreadex()	스레드 처리	비동기 송수신 및 요청 병렬 처리

과제수행방법

가) 개발 방법

- 클라이언트 요구사항 분석을 기반으로 전체 기능을 2단계 구조(클라이언트-서버)로 분할
- 명령어 중심의 입력 기반 시스템 구성으로 키보드 조작 중심 사용자 인터페이스 구성
- 텍스트 파일 기반 데이터 저장 방식을 구조체 배열로 처리
- 각 기능 모듈별 테스트를 병행하며 점진적으로 통합
- 전체 기능 완료 이후 예외 상황 및 경계 조건 테스트 수행

주요 내용	추진 일정 (주)				
	1	2	3	4	5
클라이언트 요구사항 분석 및 기능 명세 확정					
서버 구조체 설계 및 데이터 파일 처리 구현					
클라이언트 통신 기능 및 사용자 입력 구현					
기능 통합 및 테스트 진행					
예외 상황 보완 및 보고서 작성					

결과활용계획

가) 기술 내재화 및 학습 효과

- 네트워크 기반 클라이언트-서버 구조 설계를 통해 소켓 프로그래밍의 실전 적용 능력을 배양
- 구조체 기반의 텍스트 파일 데이터 관리, 사용자 인증 절차, 명령어 기반 인터페이스 구현 등을 통해 전반적인 C언어 시스템 프로그래밍 능력을 내재화 가능
- 실습 중심의 설계를 통해 스레드 동기화, 다중 사용자 처리, 예외 상황 대응 등 복합적 기술이 통합된

경험을 확보

나) 향후 프로젝트 확장 및 재사용 가능성

- 본 시스템은 도서 정보 관리라는 도메인에 국한되지 않고, 사용자 정보 처리, 파일 기반 데이터 저장 구조, 클라이언트 명령어 처리 구조를 통해 다양한 콘솔형 관리 시스템으로 확장 가능
- 예: 영화 정보 시스템, 회원제 리뷰 플랫폼, 학사 관리 시스템 등으로 재사용이 가능

다) 실무 환경 대응력 강화

- 사용자 계정 등록, 로그인, 데이터 조회/수정/삭제 등은 실무 웹서비스 및 API의 기본 구성과 유사하며, 이러한 기능 구현 경험을 통해 실제 현업에서 요구되는 논리 흐름과 기능 블록 설계 능력을 강화
- 서버 다중 접속 처리 구조 및 동기화 제어 경험은 백엔드 개발 및 분산처리 환경 설계에 기반

라) 팀 프로젝트 기반 협업 역량 확보

- 명확한 기능 분담 및 통합 테스트를 통해 협업 시 요구되는 의사소통, 문서 작성, 인터페이스 표준화 능력 체득
- 이는 향후 산업 현장에서 요구되는 프로젝트 관리 및 팀워크 능력으로 연결 가능