

7일차 복습

1. 외부 라이브러리(MySqlClient)

- 검색 API 문서(레퍼런스 문서)
- [헤더파일(.h) / 오브젝트파일(.o, .so, .a) / 기타파일]을 debian 패키지로 받음

2. MySql server 설정 + libsqlclient c programming

- c와 c++은 호환이 잘되는 언어기에 위 내용은 c++가서도 써먹을 수 있다.

3. Cmake

- makefile을 생성하는 도구
- 부가기능: 디버깅, 데비안[.deb] 배포(CPACK), Install, ...

=====

8일차

1교시 복습 및 part1 cmake로 바꾸기

2교시 속제 같이 만들어보기

3교시 동적할당

4교시 동적할당 예시

5교시 inner class 공용체 열거/booksql 동적할당 적용

6교시 enum

7교시 booksql 제거, 수정, 쿼리요청 추가 기능

8교시 booksql 제거, 수정, 쿼리요청 추가 기능

정적할당<=메모리에 저장소 확보 할당<= 프로그램 시작 순간 확보<= 컴파일시 순간 결정

동적할당<=프로그램 동작 중간에 메모리 저장소 확보

위 확보 장소를 미리 마련함(힙)

c언어 동적할당 함수:

malloc() 바로 할당

free(),

calloc()메모리 0으로 초기화 후 할당

realloc() 늘려서 할당할 때 써먹음

*주의사항: 반드시 free를 해라: 메모리 리킹이 일어나기때문

- 아레나 구조: 속도도빨라지고 실수할 확률 내려간다더라, 찾아보려면 찾아봐라

구조체의 성격 (본질):

복합 데이터 타입: 구조체는 하나 이상의 변수(멤버)를 묶어서 새로운 하나의 데이터 타입을 정의합니다.

사용자 정의: 사용자가 원하는 대로 자유롭게 데이터 묶음을 설계할 수 있습니다.

데이터 조직화: 관련 있는 데이터를 묶어 체계적으로 관리하여 코드의 가독성과 유지보수성을 높여줍니다.

구조체의 사용처:

복잡한 정보 표현: 사람 정보, 상품 정보, 날짜 정보, 좌표 정보 등 여러 속성을 가진 현실 세계의 객체를 프로그램에서 표현할 때 유용합니다.

데이터베이스 레코드: 데이터베이스의 테이블 레코드처럼, 각각 다른 종류의 컬럼으로 이루어진 데이터를 구조체로 표현할 수 있습니다.

설정 정보 관리: 프로그램 설정 값들을 구조체로 묶어서 관리하면 편리합니다.

Enum의 성격 (본질):

symbolic names for integer values: Enum은 내부적으로 정수 값에 이름을 붙여놓은 것과 같아요. MONDAY는 0, TUESDAY는 1, WEDNESDAY는 2... 와 같은 식으로 숫자가 자동으로 할당됩니다. (물론 사용자가 직접 값을 지정할 수도 있어요.)

고정된 값 집합: Enum은 미리 정의된 값들의 목록 안에서만 값을 선택하도록 제한합니다.

코드 가독성: 숫자를 직접 사용하는 것보다 의미 있는 이름 (예: MONDAY, TUESDAY) 을 사용함으로써 코드의 의미를 명확하게 하고 가독성을 높여줍니다.

Enum의 사용처:

상태 표현: 프로그램의 상태 (예: LOADING, READY, ERROR), 동작 상태 (예: PLAYING, PAUSED, STOPPED) 등을 나타낼 때 유용합니다.

옵션 선택: 프로그램 설정 옵션 (예: COLOR_MODE_DARK, COLOR_MODE_LIGHT), 메뉴 선택 등을 나타낼 때 사용됩니다.

범주형 데이터: 요일, 월, 계절, 직업 유형, 에러 코드 등 미리 정해진 범주에 속하는 데이터를 표현할 때 적합합니다.

특징	구조체 (Structure)	Enum (열거형)
목적	여러 종류의 정보 묶어서 복합 데이터 만들기	정해진 선택지 중에서 하나를 선택하는 타입 만들기
본질	여러 변수(멤버)의 묶음	정수 값에 이름을 붙여놓은 것
구성 요소	다양한 타입의 변수 (멤버)	이름이 붙은 정수 값 목록
사용 예시	책 정보 (제목, 저자, 가격), 사람 정보 (이름, 나이, 주소)	요일 (월, 화, 수, ...), 계절 (봄, 여름, 가을, 겨울), 상태 코드
비유	데이터 묶음 상자 (서랍, 도시락)	선택 메뉴판 (메뉴, 옵션 목록)
공통점	사용자 정의 데이터 타입을 만드는 도구	
	코드 가독성과 유지보수성 향상	
	프로그램의 논리적인 구조를 명확하게 표현	

비슷한 점:

둘 다 새로운 '데이터 타입'을 만들어준다: struct와 enum 모두 기본 데이터 타입(int, char 등)만으로는 표현하기 어려운 복잡하거나 의미 있는 데이터를 다루기 위해 사용자가 직접 정의하는 데이터 타입입니다.

코드 가독성 및 유지보수성 향상: 둘 다 코드를 더 명확하고 이해하기 쉽게 만들어주고, 나중에 코드를 수정하거나 관리할 때 편리하게 해줍니다.

프로그램 구조 명확화: 데이터를 의미 있는 단위로 묶거나, 선택지를 명확히 정의함으로써 프로그램의 전체적인 구조를 더 잘 드러나게 해줍니다.

다른 점:

무엇을 묶는가?: 구조체는 다양한 종류의 데이터를 묶는 반면, Enum은 정수 값에 이름을 부여한 목록을 묶습니다.

무엇을 표현하는가?: 구조체는 복합적인 정보를 표현하는 데 사용되는 반면, Enum은 고정된 선택지나 범주를 표현하는 데 사용됩니다.

성격: 구조체는 '상자'처럼 데이터를 담는 그릇의 성격이 강하고, Enum은 '메뉴판'처럼 선택지를 제공하는 성격이 강합니다.

4단계: 언제 구조체를 쓰고, 언제 Enum을 써야 할까요?

구조체: 여러 속성을 가진 '하나의 대상' (객체)을 표현하고 싶을 때 사용하세요. 예를 들어, '사람', '자동차', '주문' 같은 실세계의 객체는 여러 가지 정보(이름, 나이, 모델, 가격 등)를 가지므로 구조체로 표현하는 것이 적합합니다.

Enum: '몇 가지 중 하나' 를 선택해야 하는 상황이나, '상태' 를 표현하고 싶을 때 사용하세요. 예를 들어, '요일', '신호등 색깔', '로그인 상태', '파일 형식' 등은 Enum으로 표현하는 것이 코드 가독성과 안정성을 높여줍니다.