

CMake 자료 조사

2019102191 신주영

1. CMake란?

CMake는 오픈 소스 및 크로스 플랫폼을 지원하는 빌드 시스템 생성기(Build System Generator)로, 다양한 컴파일러와 운영 체제에서 소프트웨어 프로젝트의 빌드 프로세스를 관리하는 도구이다. CMake는 기존의 Makefile이나 Visual Studio 프로젝트 파일과 같은 네이티브 빌드 환경을 자동으로 생성하며, 이러한 기능을 통해 프로젝트의 복잡성을 줄이고 빌드 시스템 설정을 간소화 할 수 있다.

2. CMake 주요 특징

- 크로스 플랫폼 지원: CMake는 다양한 컴파일러(예: GCC, Clang, MSVC와 운영체제(Windows, Linux, macOS)에서 동일한 설정 파일로 프로젝트를 빌드할 수 있다
- 컴파일러 독립성: CMakeLists.txt 파일을 사용하여 빌드 프로세스를 정의하고 관리함으로써, 컴파일러나 플랫폼에 종속되지 않는 빌드 환경을 제공한다.
- 프로젝트 구성 및 의존성 관리: CMake는 프로젝트의 빌드 및 테스트 설정을 모듈화하여 프로젝트의 복잡성을 줄이고 유지보수를 쉽게 한다. 또한, 프로젝트 간의 의존성을 명확하게 정의하여 빌드 오류를 최소화할 수 있다.
- 빌드 시스템 자동화: CMake는 소스 코드 변경에 따라 필요한 파일만 재컴파일하고, 필요 시 타겟을 자동으로 링크하여 빌드 시간을 단축한다.
- 테스트 및 설치 지원: CMake는 CTest를 사용하여 테스트를 통합하고, CPack을 사용하여 소프트웨어 설치 패키지를 생성할 수 있다.

3. CMake 사용법

CMake를 사용하려면 프로젝트의 루트 디렉토리에 CMakeLists.txt라는 구성 파일을 생성해야 한다. 이 파일에는 프로젝트의 구조, 컴파일러 설정, 빌드 타겟 등의 정보가 포함된다. 다음은 간단한 CMakeLists.txt 예제이다.

```
# 최소 요구 CMake 버전 지정
cmake_minimum_required(VERSION 3.10)

# 프로젝트 이름과 사용 언어 설정
project(MyProject LANGUAGES CXX)

# 실행 파일 타겟 추가
add_executable(MyExecutable main.cpp)
```

- CMake 주요 명령어
 - `project()`: 프로젝트의 이름과 사용 언어를 설정
 - `add_executable()`: 지정된 소스 파일로부터 실행 파일을 생성하는 타겟을 추가
 - `target_include_directories()`: 타겟에 포함 경로를 추가하여 헤더 파일을 찾을 위치를 지정
 - `target_link_libraries()`: 타겟에 외부 라이브러리를 링크
 - `add_library()`: 지정된 소스 파일로부터 정적/동적 라이브러리를 생성
- CMake 고급 기능
 - 사용 요구 사항(Usage Requirements): `target_include_directories`, `target_compile_definitions`, `target_link_libraries`와 같은 명령어를 사용하여 라이브러리 또는 실행 파일이 다른 타겟과 어떻게 상호작용해야 하는지 정의할 수 있다. 예를 들어, `INTERFACE` 키워드를 사용하여 타겟 간의 종속성을 설정할 수 있다.
 - 패키지 설치 및 테스트: `install()` 명령어를 사용하여 라이브러리, 실행 파일, 헤더 파일 등의 설치 경로를 지정할 수 있다. 또한, `enable_testing()` 및 `add_test()` 명령어를 사용하여 프로젝트에 테스트를 추가하고, 빌드가 완료된 후 테스트를 자동으로 수행할 수 있다.
 - 빌드 환경 구성: CMake는 다양한 컴파일러 플래그 및 매크로 정의를 사용하여 디버그/릴리즈 모드와 같은 다양한 빌드 환경을 손쉽게 구성할 수 있다. 예를 들어, `target_compile_options()` 명령어를 사용하여 특정 타겟에만 적용되는 컴파일러 옵션을 설정할 수 있다.

※ 참고 문헌

- Wikipedia. (2024.07.19.). "CMake". <https://ko.wikipedia.org/wiki/CMake>.
- CMake. (version 3.30.4). "CMake Tutorial". <https://cmake.org/cmake/help/latest/guide/tutorial/index.html#>.
- CMake. (2024). "Getting Started with CMake". <https://cmake.org/getting-started/>.
- TUWLAB. (2018.06.13). "[CMake 튜토리얼] 1. CMake 소개와 예제, 내부 동작 원리". <https://www.tuwlab.com/ece/27234>.
- SCRIPTS BY. (2023.05.28). "CMake를 사용해보자!". <https://nx006.tistory.com/36>.