

Init. Download nlohmann/json

<https://github.com/nlohmann/json> release 에서 json.hpp 설치

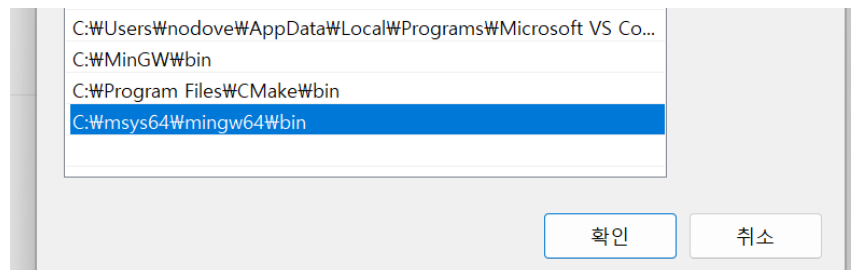
GCC 업데이트 방법

<https://www.msys2.org/> 에서 msys2.exe 다운로드

MSYS2 MinGW UCRT64 또는 MSYS2 MinGW x86_64를 선택

```
pacman -Syu  
  
64비트 GCC 설치:  
  
pacman -S mingw-w64-x86_64-gcc  
  
32비트가 필요한 경우:  
  
pacman -S mingw-w64-i686-gcc  
  
g++ --version
```

위의 명령어 입력 (모든 업데이트 완료 될 때 까지)



윈도우 환경 변수 편집 -> Path (편집) 에서

C:\msys64\mingw64\bin

추가 (*주의!* 기존 C:\MinGW\bin 보다 위에 올려야 함!)

1. 필수 도구 설치

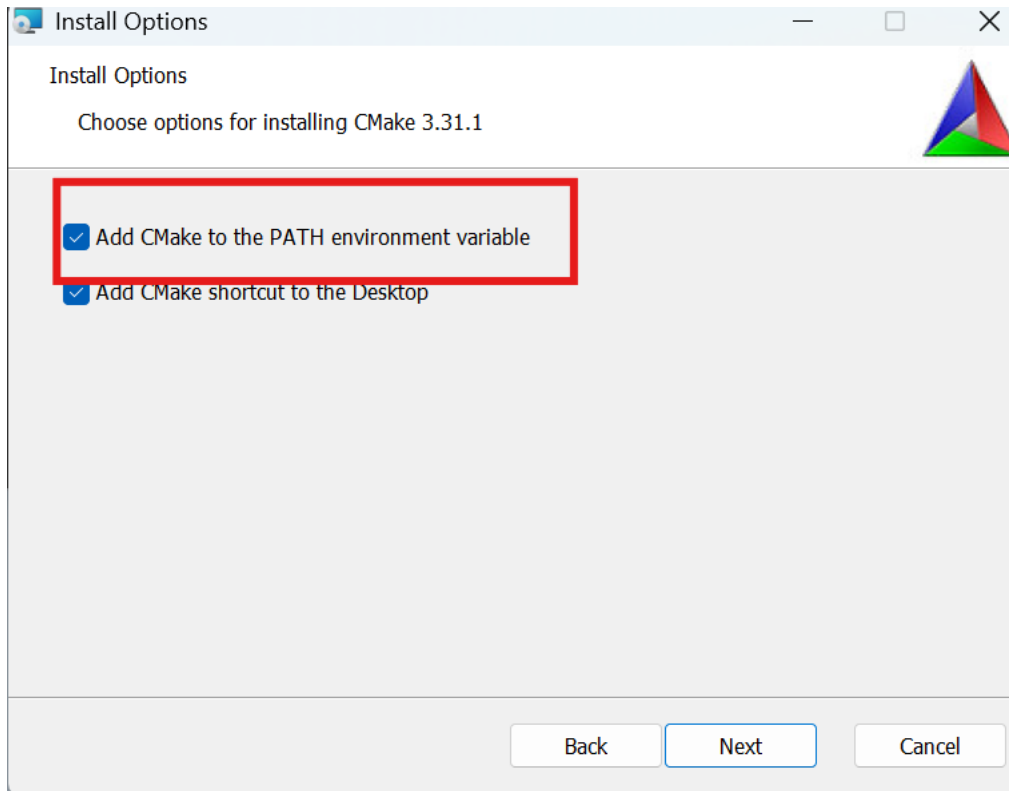
윈도우에서 CMake와 C++ 컴파일러를 사용하려면 다음 도구를 설치해야 합니다.

a. CMake 설치

1. CMake 다운로드:

- <https://cmake.org/download/> 에서 윈도우용 설치 파일을 다운로드

- 설치 중 Add CMake to system PATH 옵션을 활성화 (환경 변수 설정)



2. 설치 확인:

- 설치 후 cmd 또는 Git Bash에서 다음 명령어를 실행:
- `cmake --version`

```
C:\Users\nodove>cmake --version
cmake version 3.31.1

CMake suite maintained and supported by Kitware (kitware.com/cmake).
C:\Users\nodove>
```

- CMake 버전이 출력되면 정상적으로 설치된 것입니다.
-

b. C++ 컴파일러 설치

윈도우에서는 다음 컴파일러 중 하나를 설치해야 합니다:

1. MinGW 설치:

- <https://sourceforge.net/projects/mingw/>

에서 설치 프로그램을 다운로드합니다.

- 설치 중 g++, gcc, make 등 필수 컴파일러 도구를 선택.

Package	Class	Installed Version	Repository Version	Description
<input type="checkbox"/> mingw-developer-toolkit	bin		2013072300	An MSYS Installation for MinGW Developers (meta)
<input checked="" type="checkbox"/> mingw32-base	bin	2013072200	2013072200	A Basic MinGW Installation
<input type="checkbox"/> mingw32-gcc-ada	bin		6.3.0-1	The GNU Ada Compiler
<input type="checkbox"/> mingw32-gcc-fortran	bin		6.3.0-1	The GNU FORTRAN Compiler
<input checked="" type="checkbox"/> mingw32-gcc-g++	bin	6.3.0-1	6.3.0-1	The GNU C++ Compiler
<input checked="" type="checkbox"/> mingw32-gcc-objc	bin	6.3.0-1	6.3.0-1	The GNU Objective-C Compiler
<input type="checkbox"/> msys-base	bin		2013072300	A Basic MSYS Installation (meta)

- 환경 변수(PATH)에 MinGW의 bin 디렉토리 경로 추가.
- 설치 확인:
- g++ --version

```
C:\Users\nodove>g++ --version
g++ (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

2. Visual Studio 설치:

- [Visual Studio Community](#)를 다운로드합니다.
- 설치 중 C++ 개발 도구를 선택.
- Visual Studio 설치 시 MSVC 컴파일러가 포함됩니다.

2. 빌드 및 실행

a. CMake를 사용한 빌드

1. 빌드 디렉토리 생성:

- CMake는 빌드 파일을 프로젝트 디렉토리 외부에 생성하는 것이 일반적입니다.
- team_report_24 디렉토리 내에서 다음 명령어를 실행:
- mkdir build

- cd build

2. CMake로 빌드 시스템 생성:

- MinGW를 사용할 경우:
- cmake -G "MinGW Makefiles" -DCMAKE_C_COMPILER=gcc -DCMAKE_CXX_COMPILER=g++ ..

```
nodove@DESKTOP-MLFT0L3 MINGW64 /c/git/CPP_LoadBalancing_algorithm/team_report_24/build (win)
$ cmake -G "MinGW Makefiles" -DCMAKE_C_COMPILER=gcc -DCMAKE_CXX_COMPILER=g++ ..
-- Configuring done (0.2s)
-- Generating done (0.0s)
-- Build files have been written to: C:/git/CPP_LoadBalancing_algorithm/team_report_24/build
```

3. 컴파일 실행:

- MinGW:
- mingw32-make

4. 생성된 실행 파일 실행:

- 빌드가 성공하면 실행 파일이 생성됩니다.
- MinGW:
- ./CPP_LoadBalancing_algorithm.exe
- Visual Studio:
- ./Release/CPP_LoadBalancing_algorithm.exe

b. Visual Studio GUI 사용

1. Visual Studio를 실행하고, File -> Open -> CMake를 선택.
2. 프로젝트의 루트 디렉토리(CPP_LoadBalancing_algorithm)에서 CMakeLists.txt 파일을 엽니다.
3. Visual Studio가 CMake 프로젝트를 로드하고, 빌드 및 실행 버튼을 제공합니다.

3. 빌드 에러 해결

1. cmake: command not found 문제:

- CMake가 설치되지 않았거나 PATH 환경 변수에 등록되지 않았습니다.
- PATH 등록:
- `export PATH=/c/Program Files/CMake/bin:$PATH`

2. MinGW 설치 문제:

- g++ 또는 make 명령어가 없으면 MinGW를 올바르게 설치하지 않았을 가능성이 있습니다.
- PATH 설정:
- `export PATH=/c/MinGW/bin:$PATH`

3. CMakeLists.txt 오류:

- CMake 설정 중 오류가 발생하면 CMakeLists.txt를 확인해 정확히 작성되었는지 검토하세요.

최종 확인

윈도우에서 CMake 프로젝트를 빌드하려면:

1. CMake 및 컴파일러 설치.
2. CMake 명령으로 빌드 시스템 생성.
3. 컴파일 및 실행.