

데이터 구조 개발환경

ECE20010

한동대학교
CSEE 자료구조
idebtor@gmail.com

g++ for macOS & Windows
git, github for Desktop
Atom

g++ for macOS

- Open a terminal. (command + space, type terminal in the search field.)
- Enter the following command and run it.
 - `$ xcode-select -install`
- Check the installation with the following command.
(If you see some version numbers, it was successfully installed.)
 - `$ g++ -v`
- Check the option such that all filename extensions are to be displayed.
[Finder] → [Preferences ...] → [Advanced] → Show all filename extensions

g++ for macOS

- Use this code to test that g++ can compile and run the program.

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello, World!\n";
    return 0;
}
```

1. Copy the above program and save the file as **hello.cpp** on your desktop.
2. Open a terminal and type **cd Desktop**
3. To compile, enter the following in the terminal and run it. We see the file **hello** appear on the desktop under an exec icon.

```
$ g++ -std=c++11 hello.cpp -o hello
```

4. Run the compiled program by typing the following in the terminal. You will see "Hello World!" printed in the terminal.

```
$ ./hello
```

g++ for Windows

- 두 종류의 패키지, 즉 **MSYS2** 와 **MinGW-w64** 를 설치해야 합니다.
MSYS2를 **먼저** 설치하지 않으면 어려움을 겪는 경우가 종종 발생합니다.

MSYS2 설치하기

- <http://www.MSYS2.org/>로 접속합니다.

MSYS2를 통해 MinGW-w64를 설치를 위해
아래 website를 참조해도 좋습니다.

- <https://vl0011.tistory.com/14>
- <https://stackoverflow.com/questions/30069830/how-to-install-mingw-w64-and-msys2/30071634#30071634>



MSYS2 installer
One click installer for msys2

클릭

msys2-i686-20190524.exe msys2-x86_64-20190524.exe View on GitHub

SHA256: c5a1881f7ac5a0449fe9b30d31... SHA256: 2dacadcc70cc122054e60914ct...

MSYS2 is a software distro and building platform for Windows

At its core is an independent rewrite of MSYS, based on modern Cygwin (POSIX compatibility layer) and MinGW-w64 with the aim of better interoperability with native Windows software. It provides a bash shell, Autotools, revision control systems and the like for building native Windows applications using MinGW-w64 toolchains.

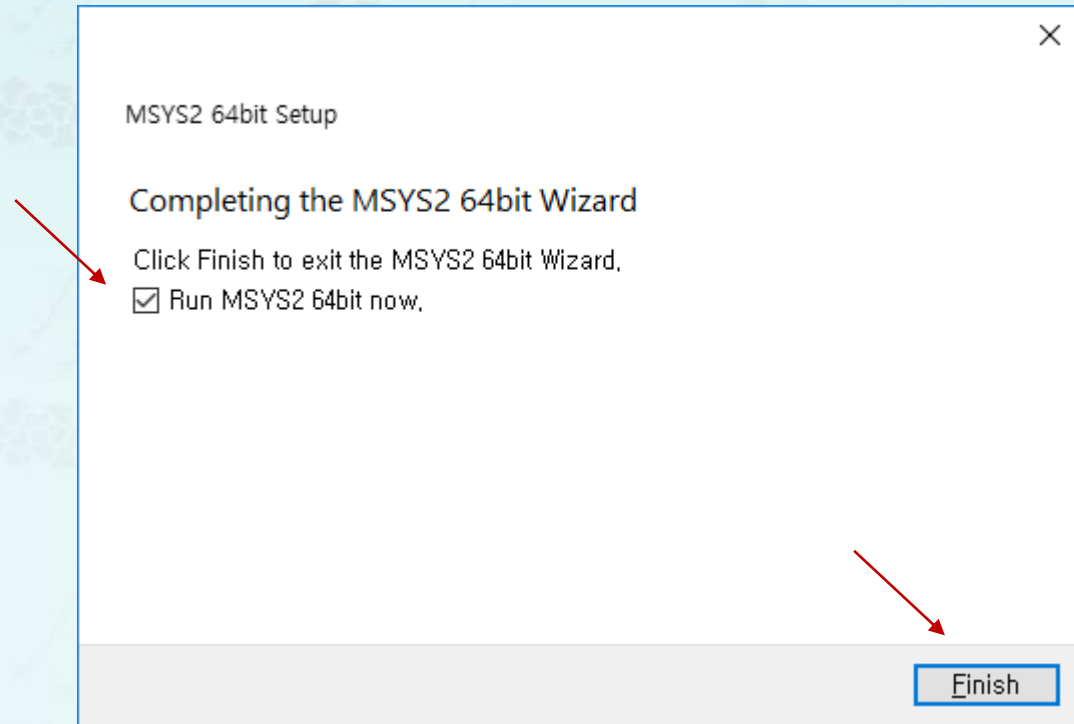
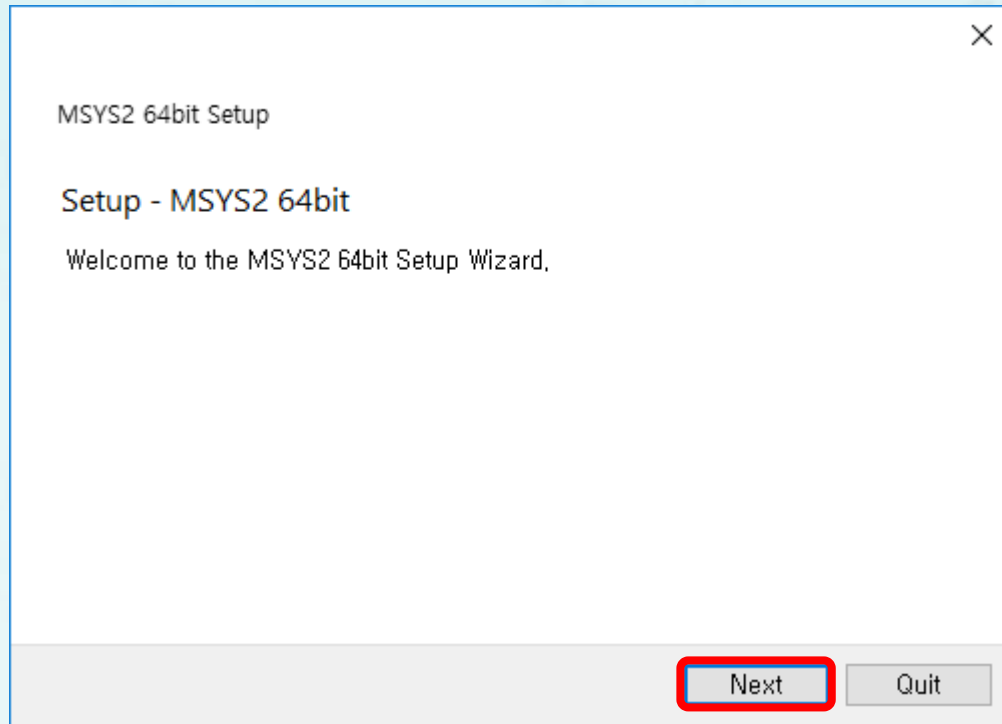
It features a package management system to provide easy installation of packages, Pacman. It brings many powerful features such as dependency resolution and simple complete system upgrades, as well as straight-forward package building.

Installation restrictions:

- MSYS2 can't be installed on FAT* partitions.
- Current MSYS2 can't be installed on Windows XP anymore.

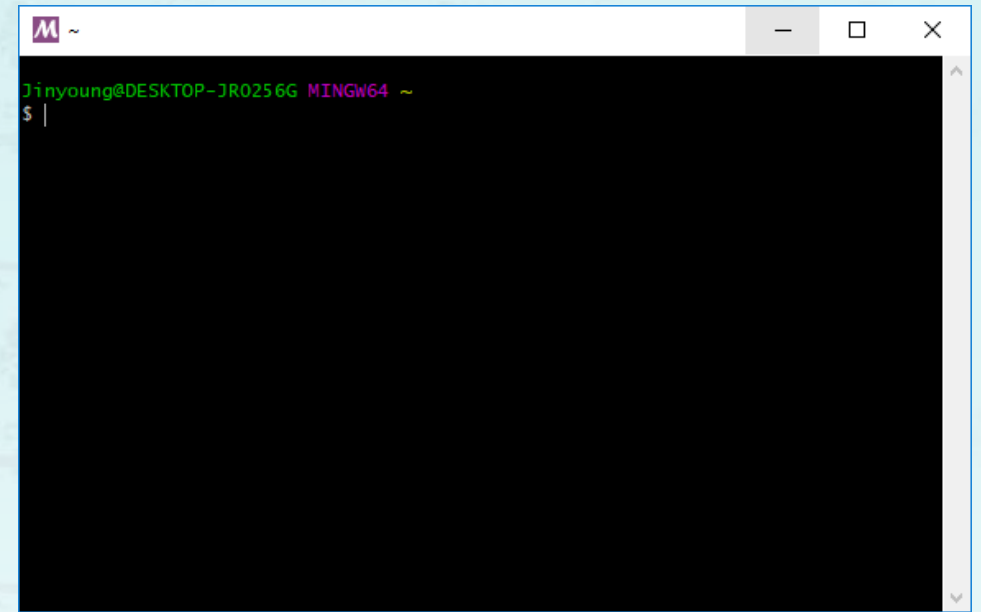
MSYS2 설치하기

- MSYS2-x86_64-20190524.exe를 실행합니다. 혹은 최신 버전으로.
- Finish를 클릭하여 MSYS2를 실행합니다.



MSYS2 설치하기

- MSYS2를 실행한 화면 입니다.



- Command Shell (cmd) 혹은 PowerShell에서 “**pacman -Syu**”를 입력하고 Enter 키를 눌러 패키지 목록과 MSYS2를 업데이트 합니다.
- “::설치를 진행하시겠습니까? [Y/N]”라는 질문이 나오면 Enter 키를 누릅니다.
- 실행 결과를 보면 “경고:terminate MSYS2 without returning....”, “경고: for example close your terminal window...”이라는 메시지가 나오면 경고를 무시하고 다음을 진행합니다.

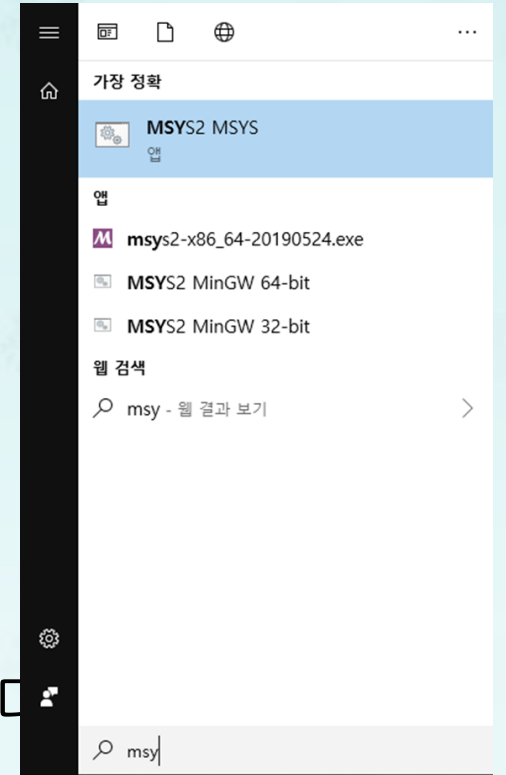
MSYS2 - GCC 설치하기

- MSYS2를 실행한 화면(콘솔)에서 다음을 입력하여 GCC를 설치하십시오.
`pacman -S mingw-w64-x86_64-gcc`

- 만약, MSYS2 실행 화면(콘솔)이 없어졌으면, 윈도우 검색창에 “MSYS2”를 입력하거나 MSYS2를 찾아가서 MSYS2 msys를 실행하여 콘솔을 다시 생성하고 명령어를 입력 하십시오.

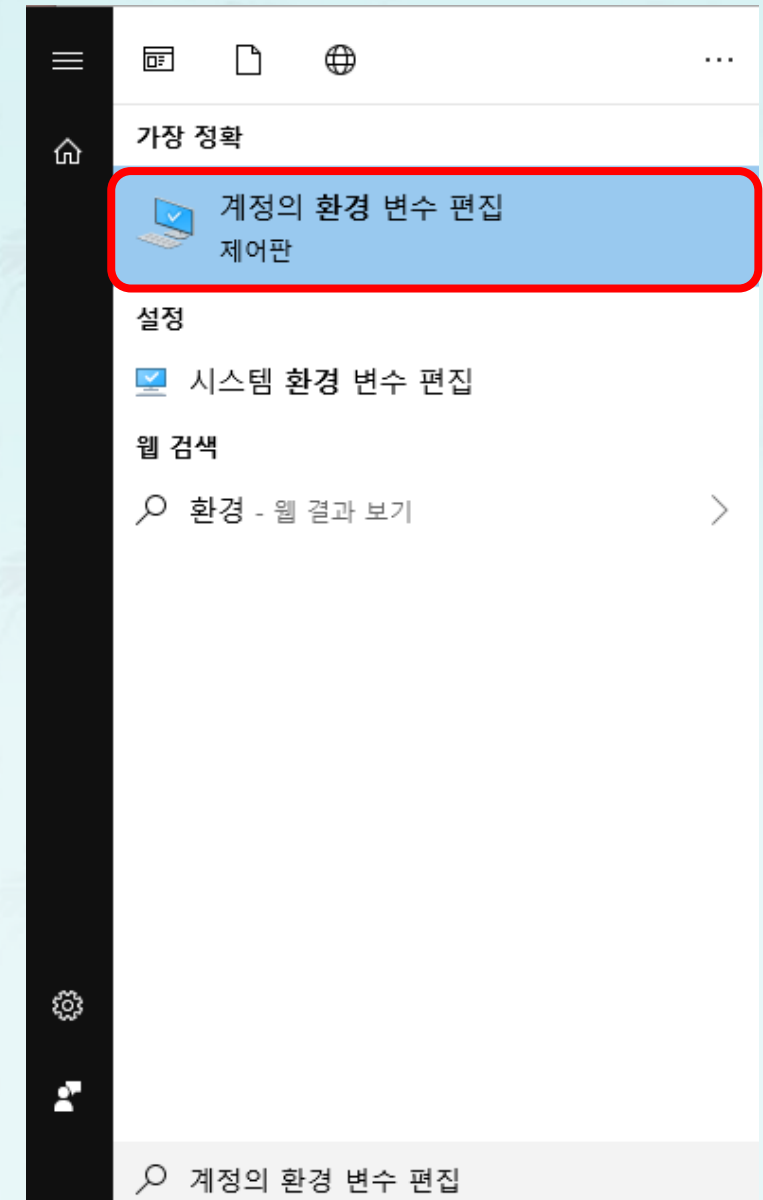
- 설치할 때 질문이 나오면, <Enter>로 응답하면 됩니다.

- Toolchain 설치 - make, gdb 등 gcc와 함께 쓰이는 툴들을 설치합니다
`pacman -S mingw-w64-x86_64-toolchain`
설치할 때 질문이 나오면, <Enter>로 응답하면 됩니다.



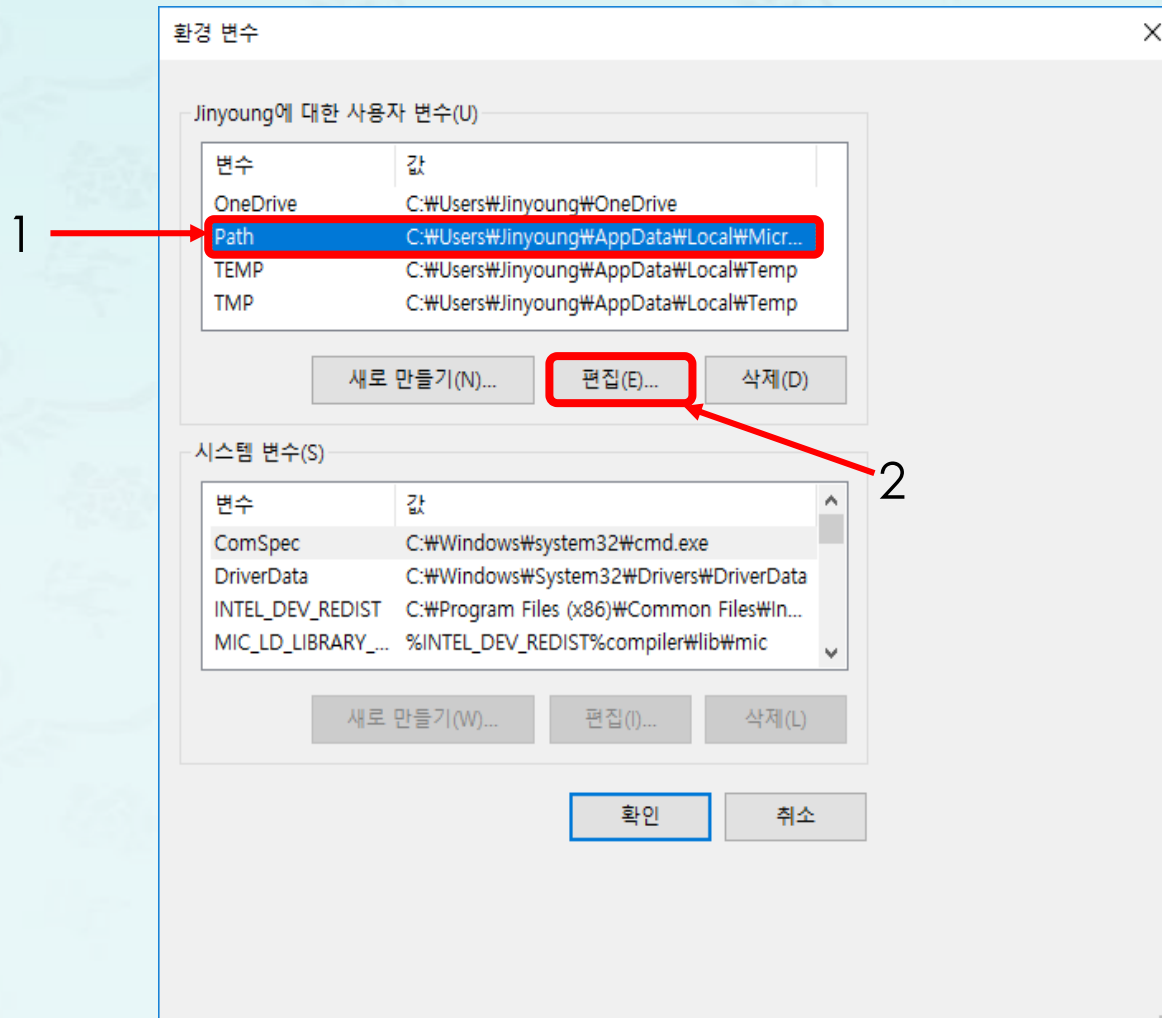
MSYS2 - 환경 변수 설정하기

- 다음은 시스템 환경 변수를 설정해야 합니다.
- 환경변수를 설정해주기 위해 윈도우 검색에서 “환경 변수”를 입력하고 계정의 환경 변수 편집으로 들어갑니다.



MSYS2 - 환경 변수 설정하기

- 사용자 변수 창 혹은 시스템 변수 창에서 Path를 선택한 후 편집을 클릭합니다.



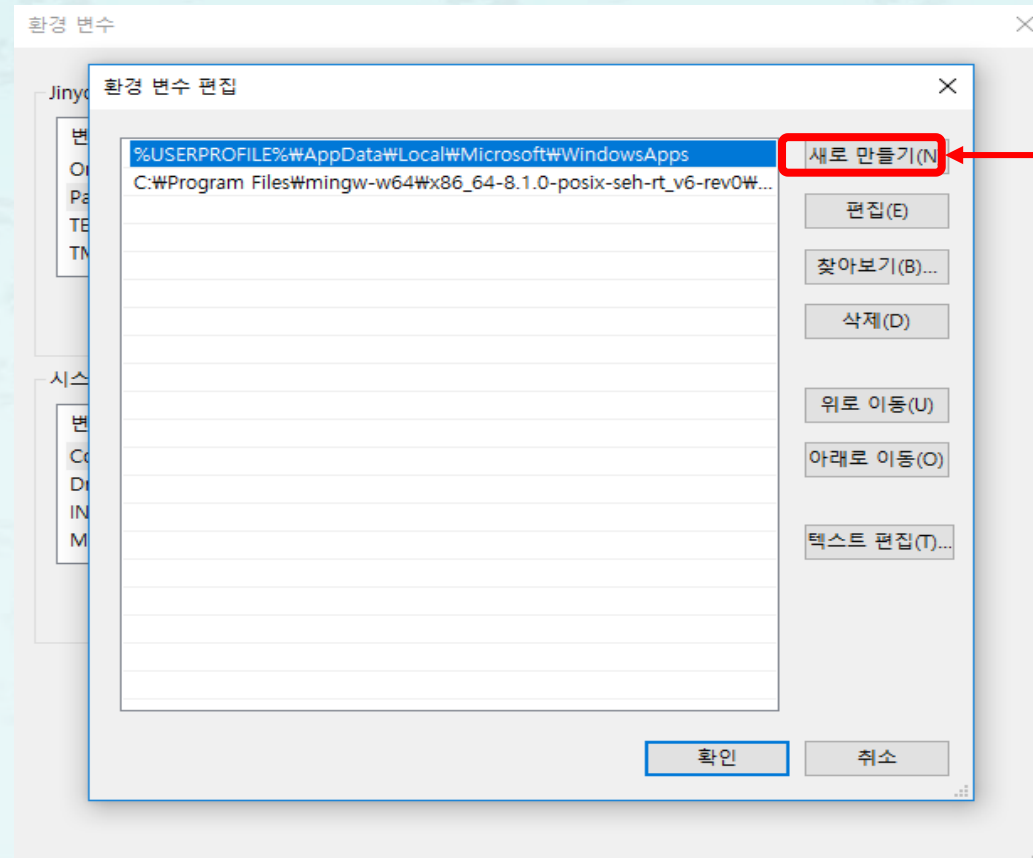
MSYS2 - 환경 변수 설정하기

- [새로 만들기]를 클릭 후 g++.exe 가 존재하는 path를 입력하거나 [붙여넣기] 합니다.
 - msys2와 mingw64를 설치한 곳에서, **g++.exe** 가 존재하는 폴더를 찾아서, 그 **Path**를 복사하여 붙여 넣기를 하는 것이 안전합니다.
 - 만약, msys2 설치 시에 디폴트를 사용했다면, 다음과 같지만, 한번 확인해 보는 것이 좋습니다

C:\msys64\mingw64\bin

C:\msys64\usr\bin

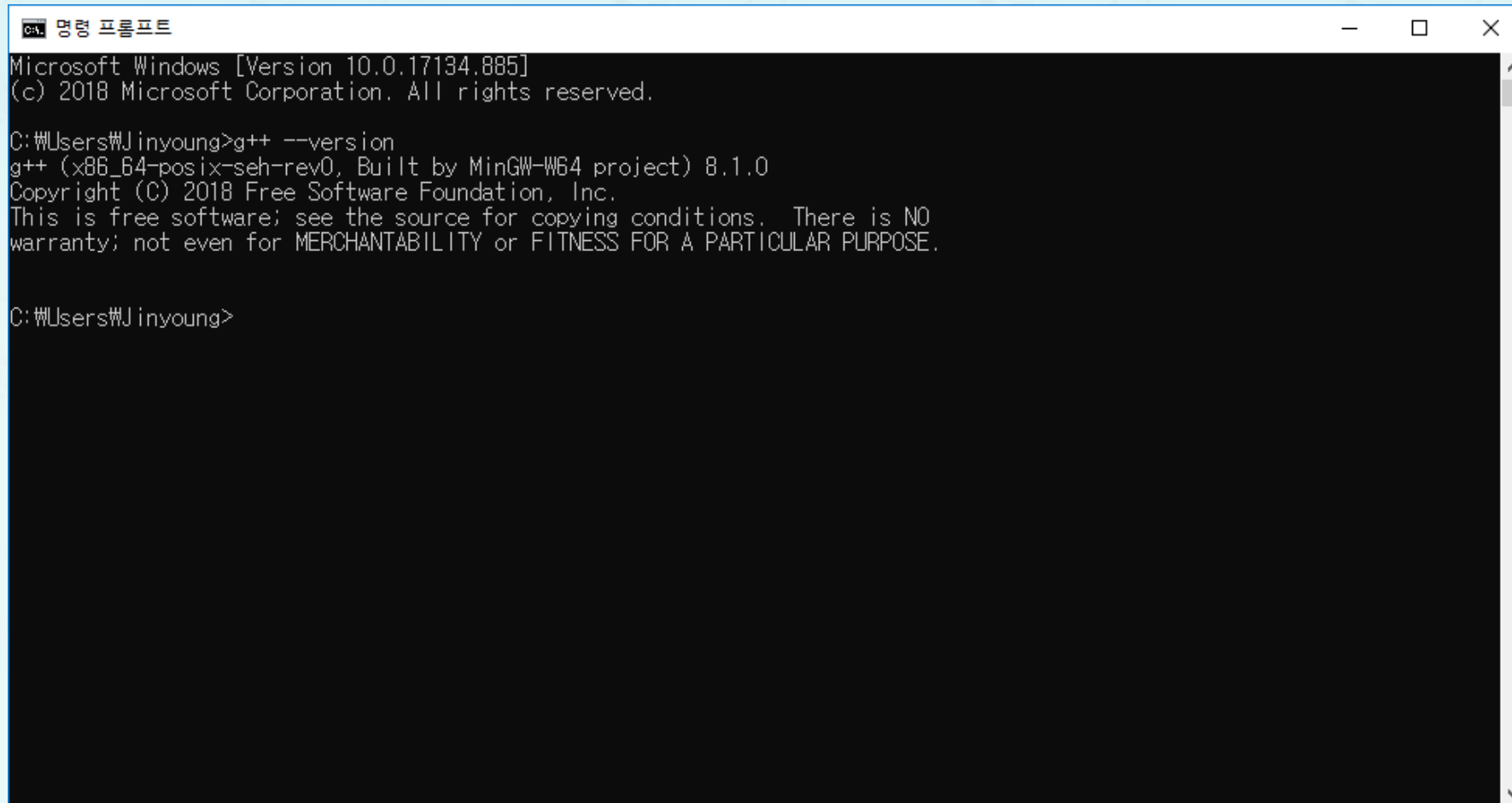
위의 두번째 폴더는 ls, cp, rm 같은 Bash 명령어를 사용하기 위해 필요한 폴더입니다.



클릭 후 path 입력

MSYS2 - 환경 변수 설정하기

- 윈도우 검색창에 “**cmd**” 명령어를 실행하여, 콘솔(console or terminal)을 띄웁니다.
- 프롬프트 창에 “**g++ --version**”을 입력하여 다음과 같이 화면이 나온다면 path추가가 정상적으로 된 것 입니다.



```
명령 프롬프트
Microsoft Windows [Version 10.0.17134.885]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Jinyoung>g++ --version
g++ (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 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.

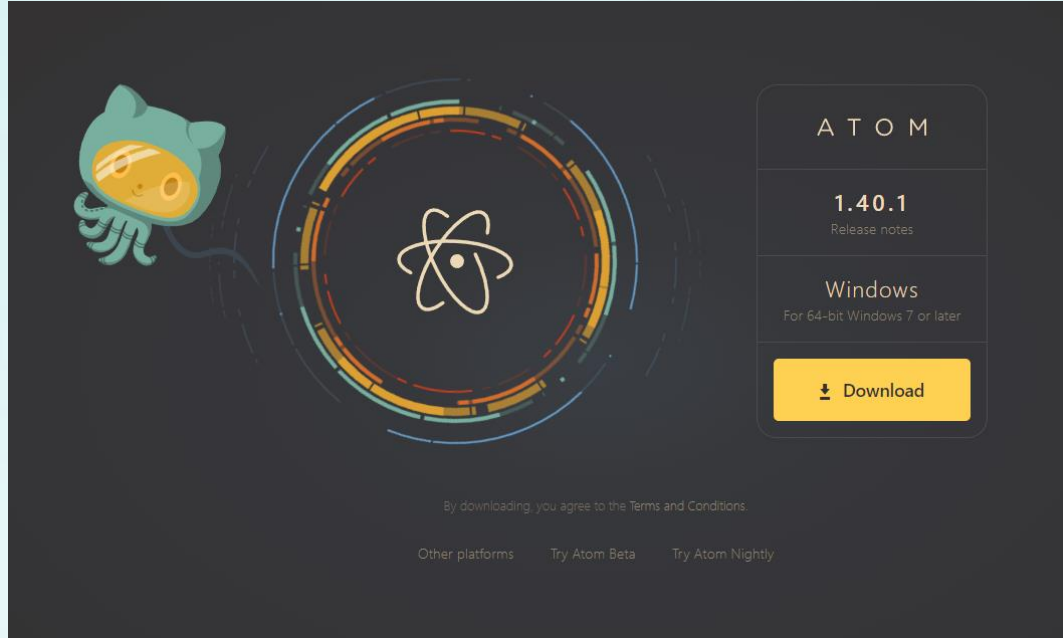
C:\Users\Jinyoung>
```

Mingw-w64 설치 오류 관련

- MSYS2가 아니라 **MinGW-w64**를 먼저 설치할 경우, 설치 오류가 발행하는 경우가 종종 있습니다.
- 또한 한글로 사용자 이름이 설정되어 있는 경우, 어려움을 많이 겪습니다. 이런 경우, 특히 블로그의 설명에 따르면 재설치하는 경우 문제가 발생한다고 합니다.
 - 이런 경우, 기존의 있는 폴더를 완전히 삭제한 후, MSYS2로 다시 설치해보십시오.
 - **MSYS2를 먼저 설치하고**, 그 안에서 MinGW-w64를 추가 설치하는 방법을 택하길 적극 권장합니다.
 - 그래도, 설치에 어려움이 있다면, (특히 한글 사용자 이름으로 인하여), 새로운 사용자를 만들어 시도해보십시오.
- <https://taeguu.tistory.com/3>

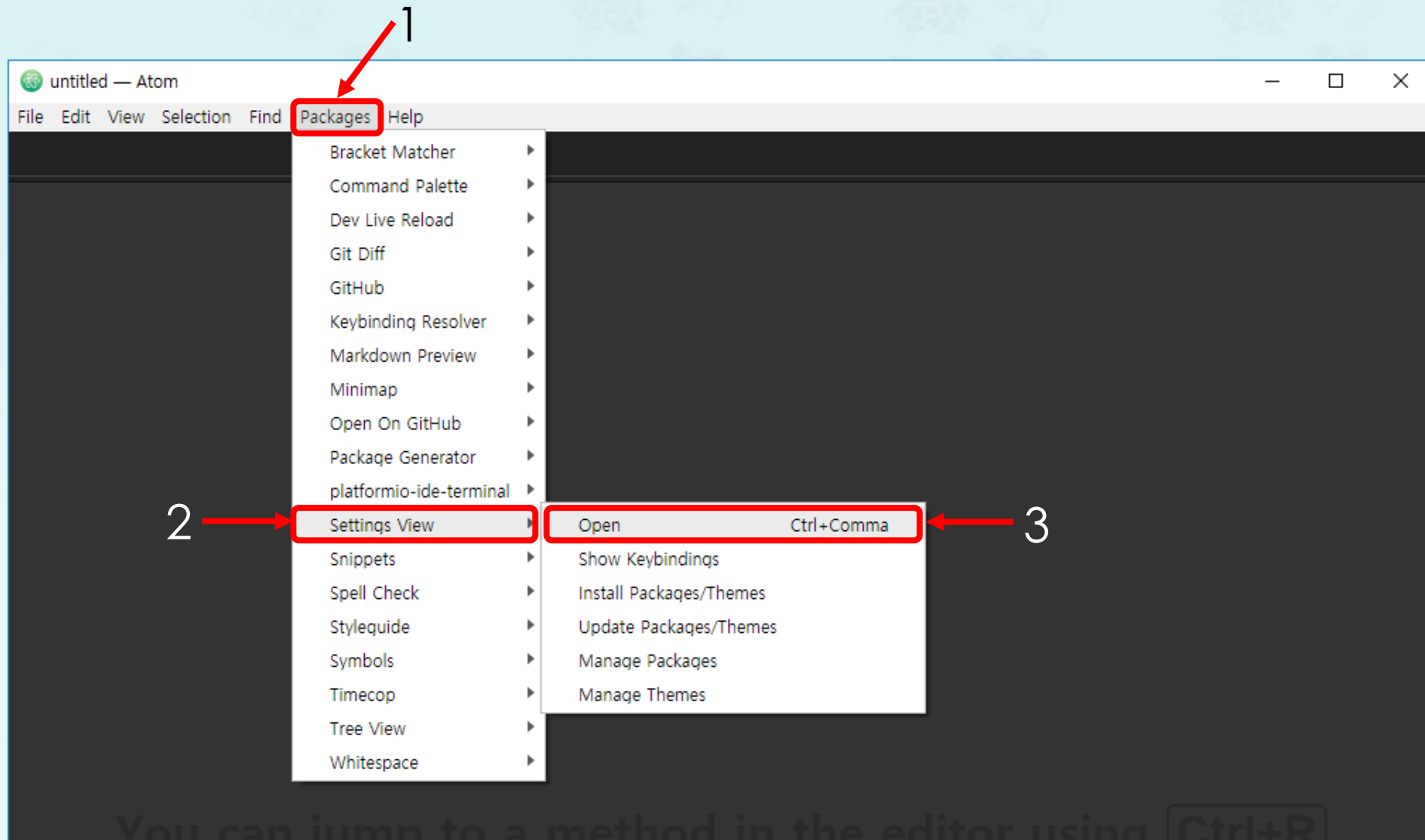
Atom 설치하기

- <https://atom.io/>로 접속합니다.



- AtomSetup-x64.exe를 실행하면 자동으로 Atom이 설치가 완료됩니다.
- 다음 폴더를 사용자 혹은 시스템의 환경 변수 **Path**에 추가하면 편리합니다.
C:\Users\<username>\AppData\Local\atom\bin

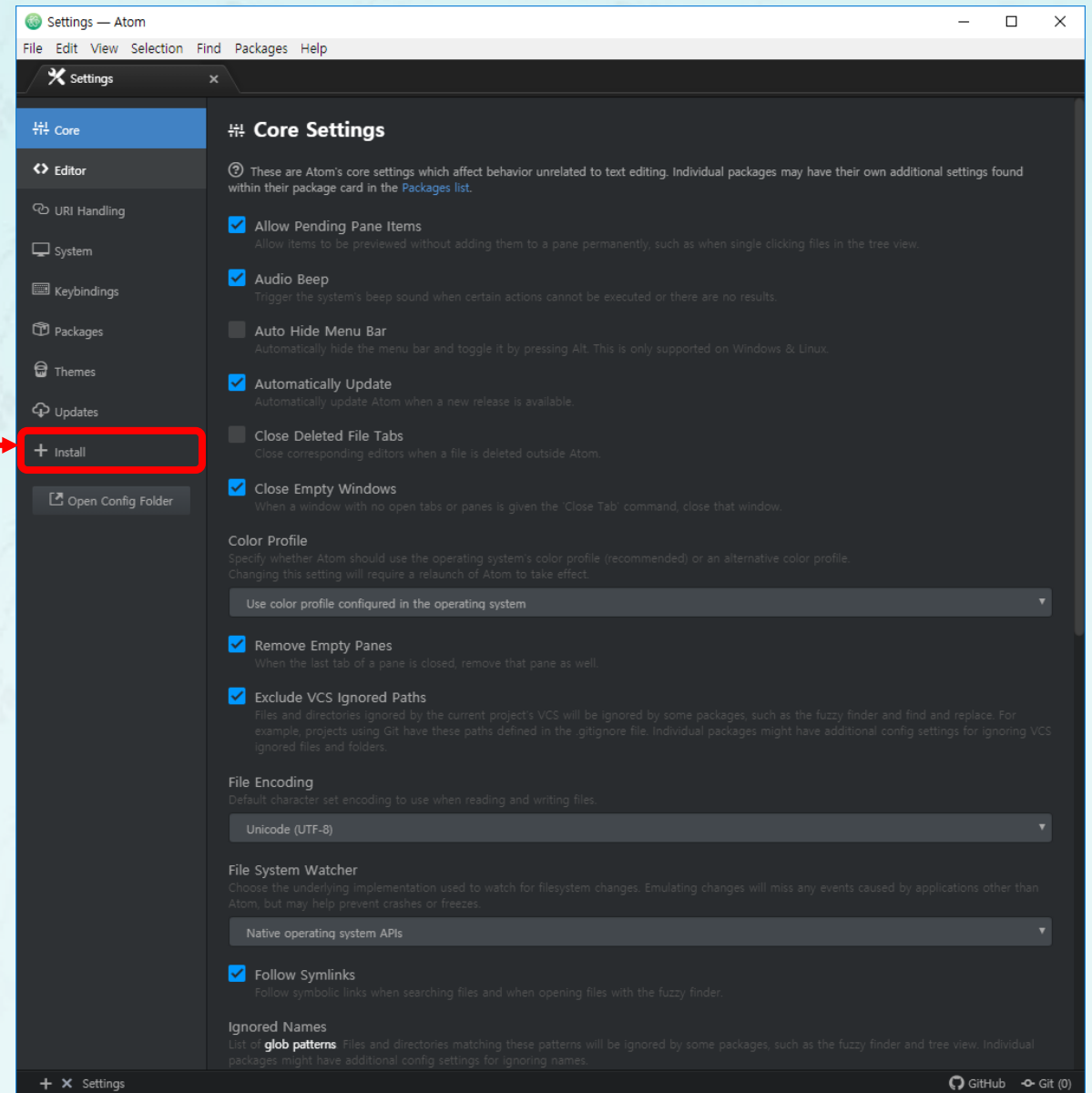
Atom 패키지 설치하기



Atom 패키지 설치하기

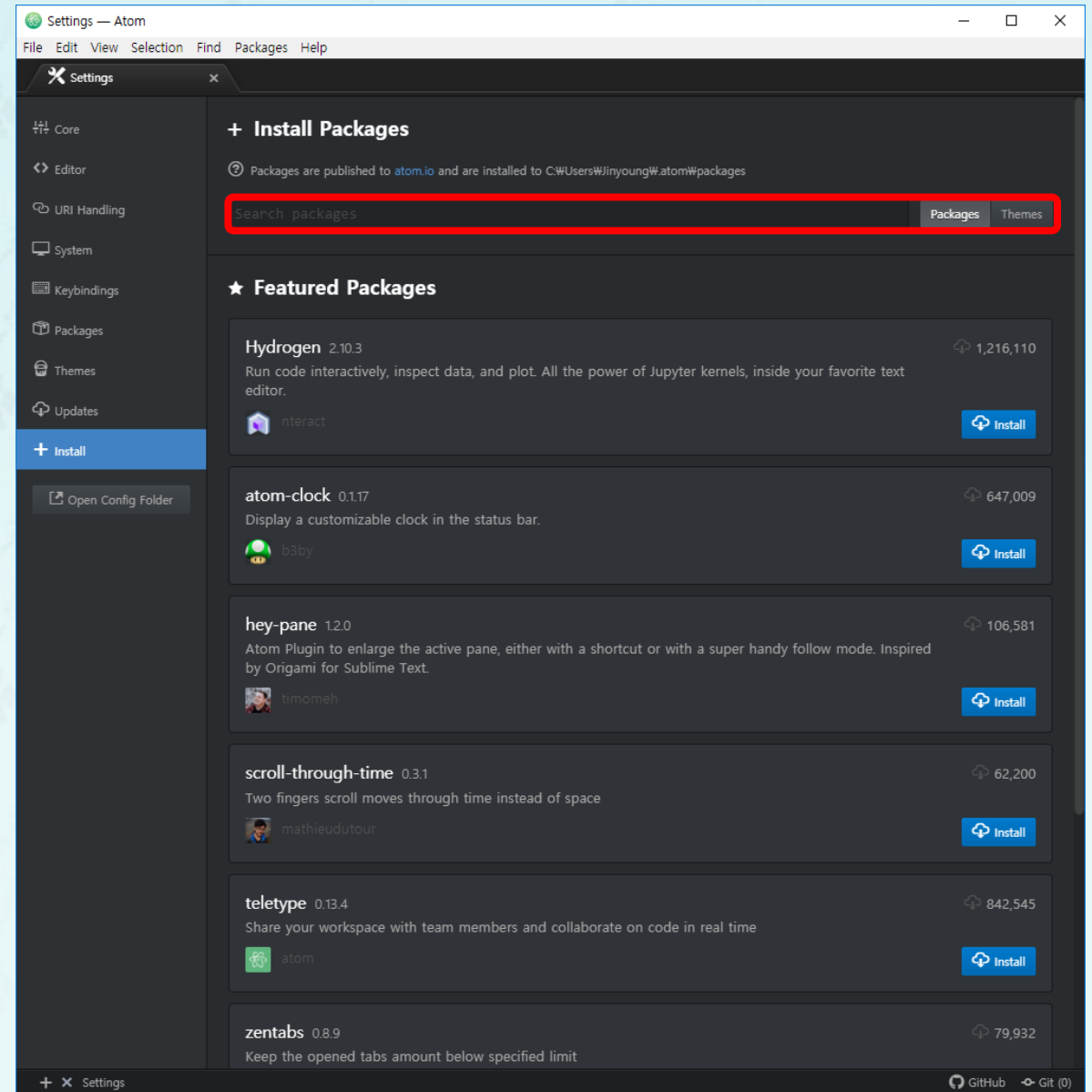
- Install를 클릭합니다.

클릭



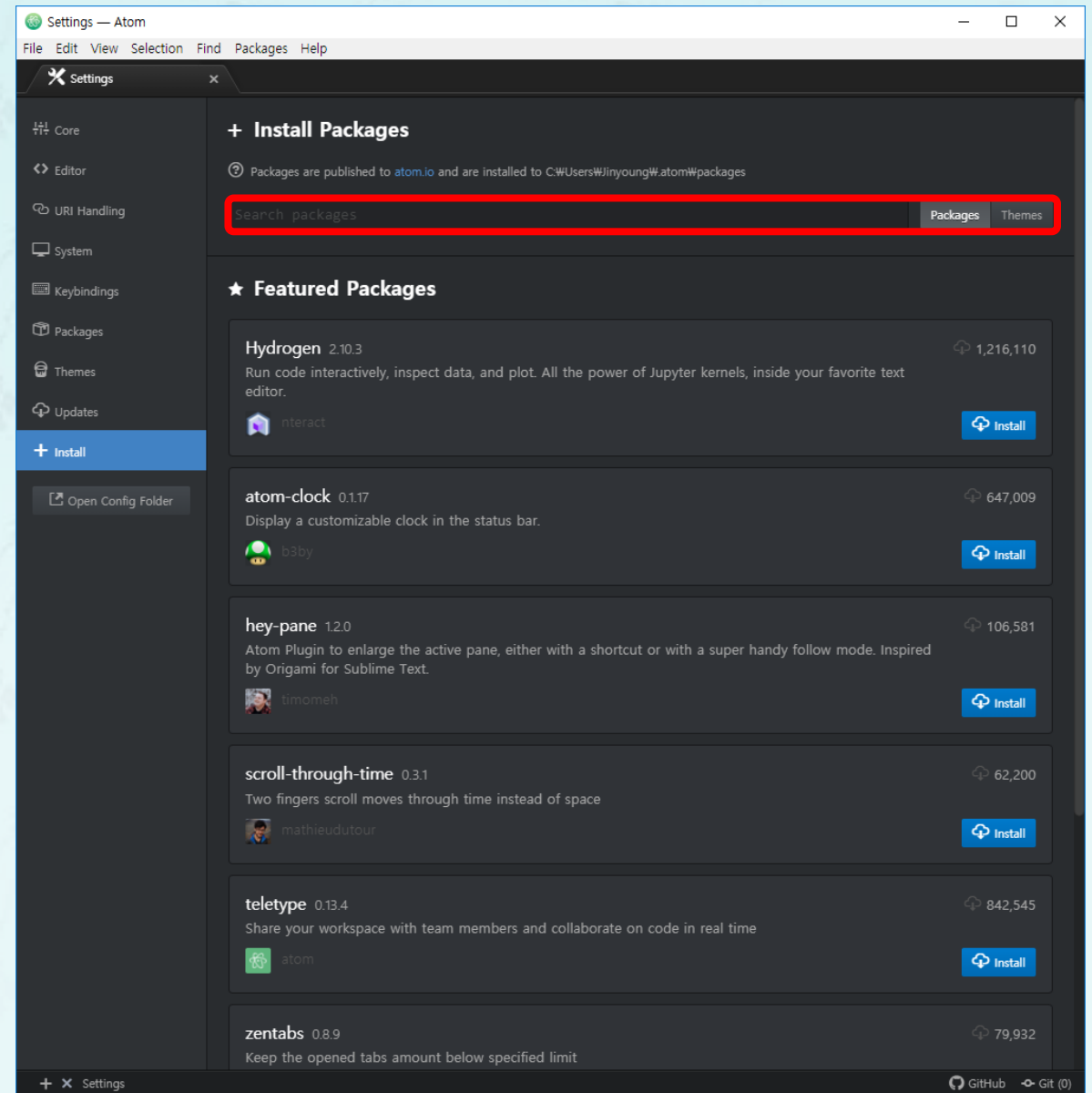
Atom 패키지 설치하기

- 검색창에 다음의 패키지 이름들을 검색하여 설치합니다.
- platformio-ide-terminal
- file-icons
- minimap
- autocomplete-clang



Atom 패키지 설치하기

- 검색창에 다음의 패키지 이름들을 검색하여 설치합니다.
- platformio-ide-terminal
- file-icons
- minimap
- autocomplete-clang



g++ for Windows

- Use this code to test that g++ can compile and run the program.

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello, World!\n";
    return 0;
}
```

1. Copy the above program and save the file as **hello.cpp** on your folder.
2. Open a terminal at the folder where **hello.cpp** exists.
Use the [File] → [Windows PowerShell 열기] menu at the top left corner of the **File Explorer**.
3. To compile, enter the following in the terminal and run it. We see the file **hello** appear on the desktop under an exec icon.

```
$ g++ hello.cpp -o hello
```

4. Run the compiled program by typing the following in the terminal. You will see "Hello World!" printed in the terminal.

```
$ ./hello
```

g++ for Windows

The screenshot displays a Windows development environment. On the left, a File Explorer window shows a directory named 'nowic' with a context menu open, highlighting 'Windows PowerShell 열기(R)' (Open Windows PowerShell). In the center, a Windows PowerShell terminal window shows the command `PS C:\GitHub\nowic> atom hello.cpp`. On the right, the Atom IDE is open, displaying the file `hello.cpp` with the following code:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Hello World!\n";
6 }
```

The IDE's left sidebar shows a project view with various files, including `02ArgcArgv.md`, `02GettingInput.md`, `02UsingMintty.md`, `02UsingStaticLib.md`, `03callByRef.md`, `03overload.md`, `03PowerShell.md`, `DS_DevEnvSetup64.p`, `ds.sln`, `ECE20010DS-Syllabu`, `GitHubWikis.md`, `graph_euler_hamiltor`, `hello.cpp`, and `README.md`. The bottom status bar of the IDE shows the file path `C:\GitHub\nowic\hello.cpp` and other details like encoding (UTF-8) and language (C++).

Annotations with yellow boxes and red arrows point to specific elements:

- File-icons**: Points to the file list in the Atom IDE sidebar.
- Platformio-ide-terminal**: Points to the terminal window at the bottom of the IDE.
- Platformio-ide-terminal**: Points to the terminal window in the background.
- Minmapl**: Points to the minimap in the Atom IDE.



Installing Git – the easy way

- Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.



git fast-version-control

<https://git-scm.com/>

- If you've never used git or github before, there are a bunch of things that you need to do.
- Get a **github** account.
- Download and install **git**.

Installing Github Desktop

- Focus on what matters instead of fighting with Git.
- Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.
- Installing GitHub Desktop:

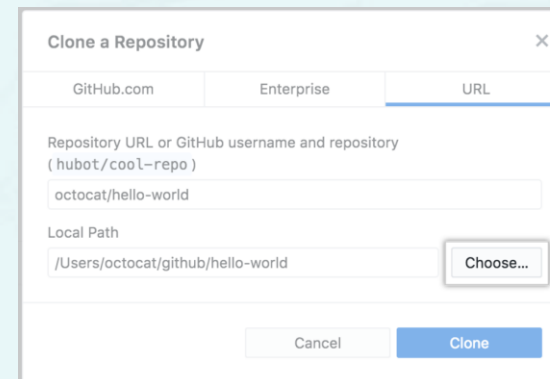
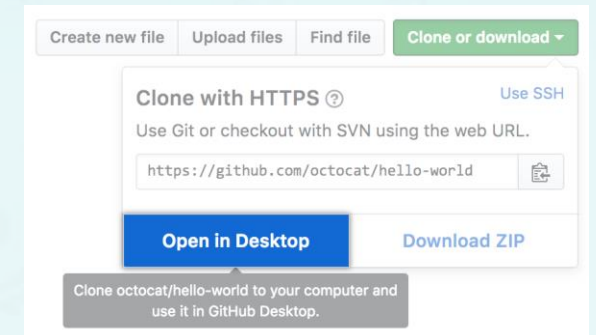


<https://desktop.github.com/>

- Repository for this course:
 - github.com/idebtor/nowic

Cloning a repository from GitHub to GitHub Desktop

- <https://help.github.com/en/desktop/contributing-to-projects/cloning-a-repository-from-github-to-github-desktop>
1. Sign in to GitHub and GitHub Desktop before you start to clone.
 2. On GitHub, navigate to the main page of the repository.
 3. Under your repository name, click Clone or download.
 4. Click Open in Desktop to clone the repository and open it in GitHub Desktop.
 5. Click **Choose...** and, using the Finder window, navigate to a local path where you want to clone the repository.
 6. Click **Clone**.



데이터 구조 개발환경

ECE20010

한동대학교
CSEE 자료구조
idebtor@gmail.com