

2017 년 2 학기 csed232 객체지향프로그래밍

프로그래밍 과제 컴파일 환경 및 설치 문서

작성: 한상도 (hansd@postech.ac.kr)

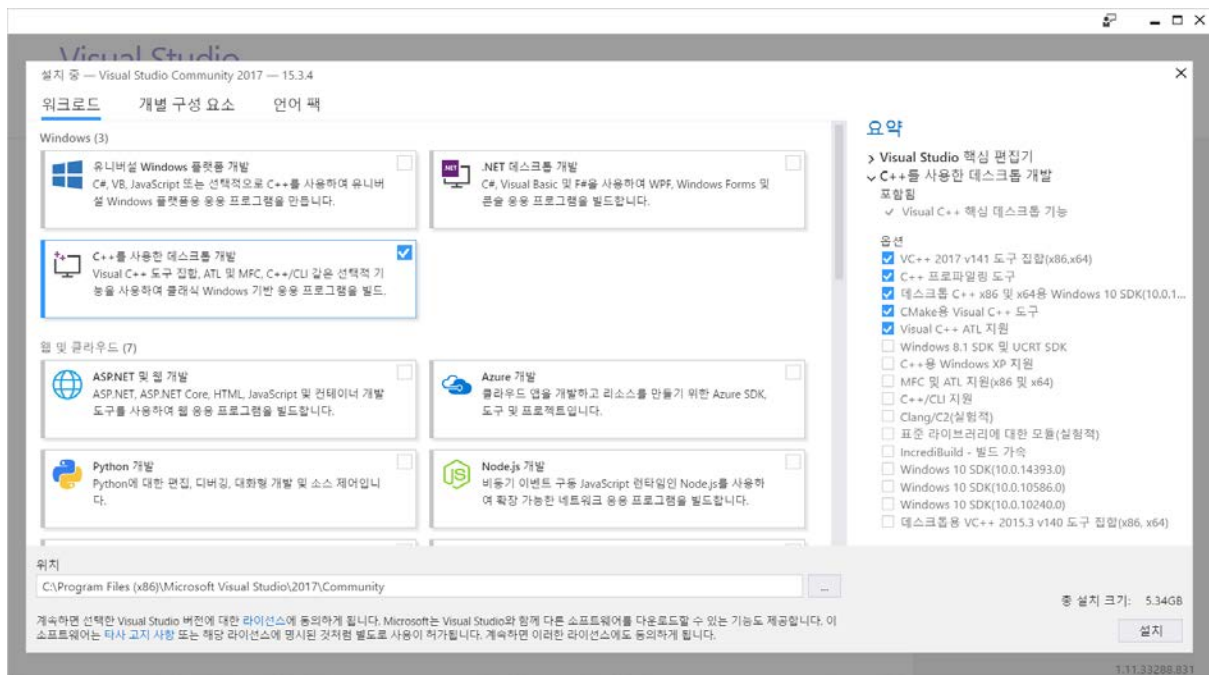
2017-9-13

Visual Studio 2017 기본 설정 하기

1. Visual studio download

<https://www.visualstudio.com/ko/vs/community/> 에서 vs community 2017 을 다운로드 합니다

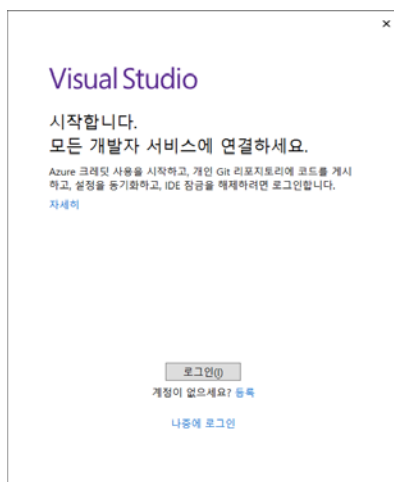
2. 설치 시 아래와 같은 선택을 합니다.



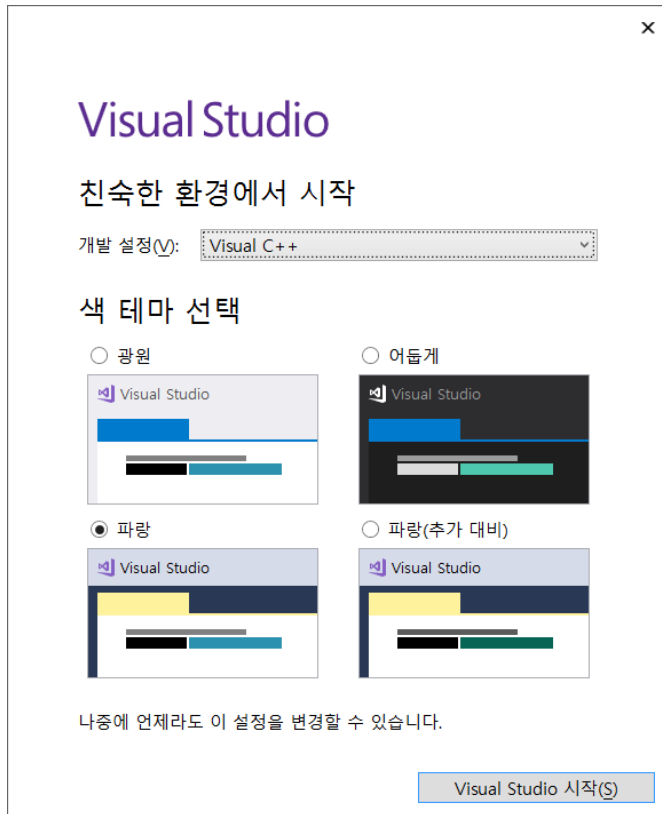
C++ 를 사용한 데스크톱 개발 클릭

3. 시스템 재부팅

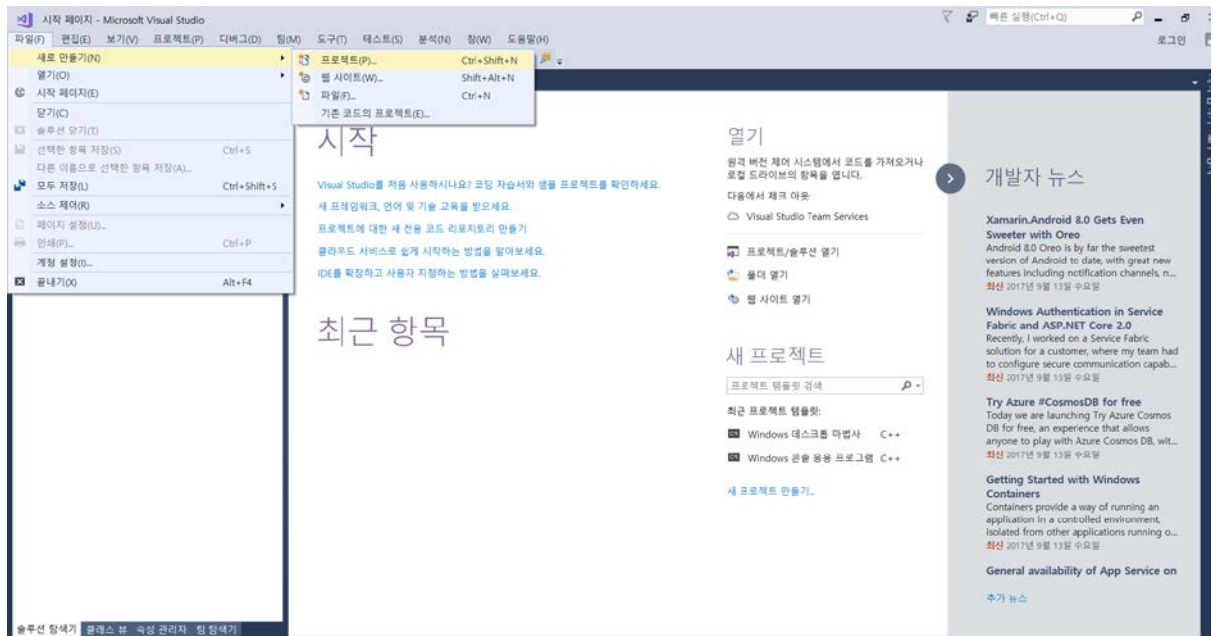
4. 재부팅 후 실행 시 “나중에 로그인” 을 클릭합니다. (계정 생성 후 로그인해도 무방)



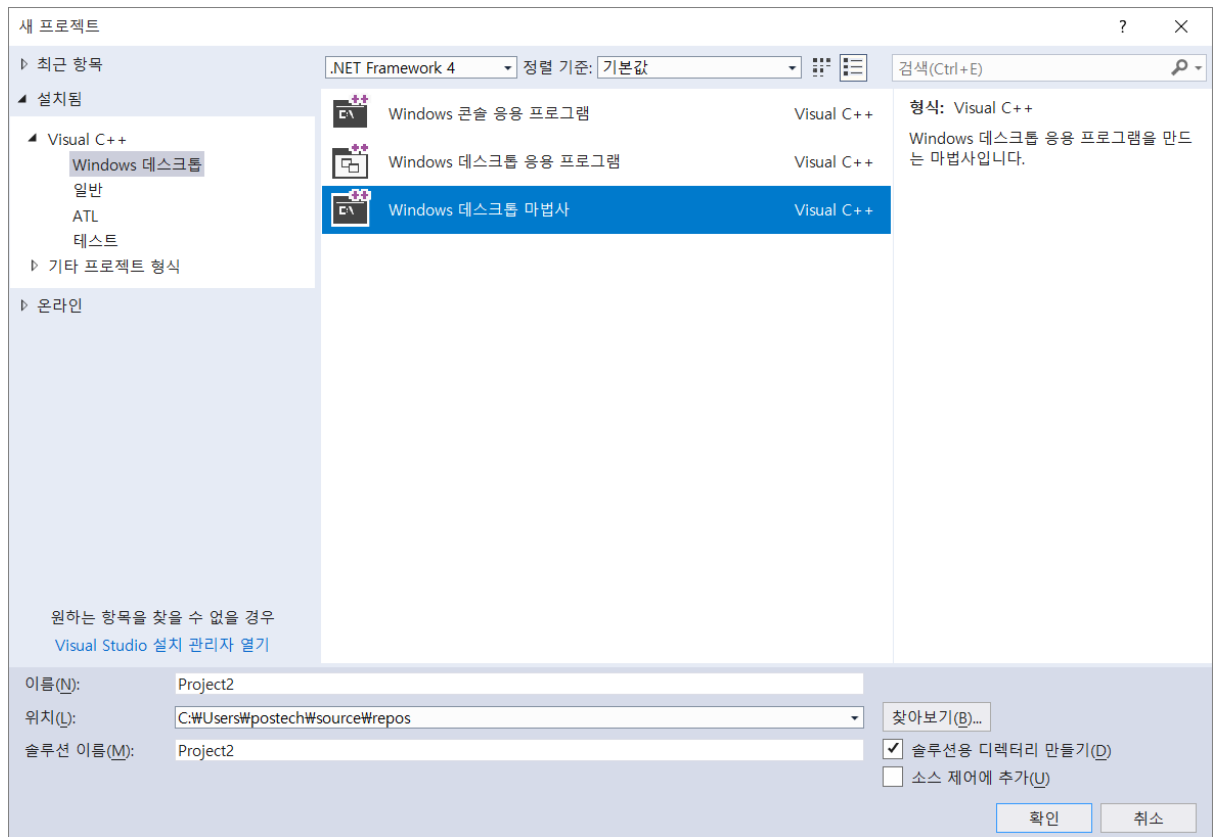
5. 개발 설정 : visual c++



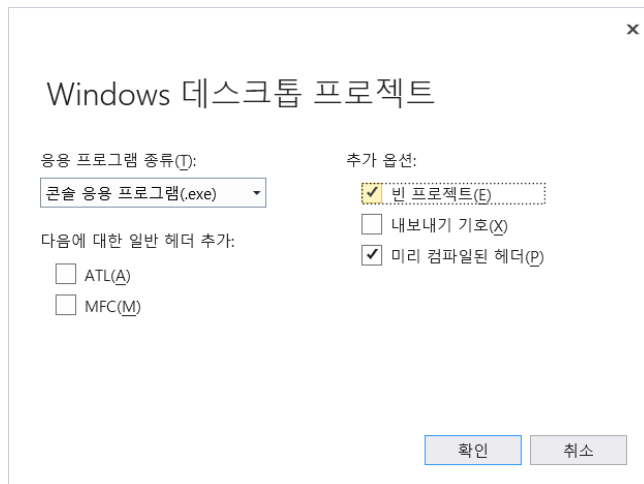
6. 파일, 새로만들기, 새 프로젝트 생성



7. Visual c++ > Windows 데스크톱 > windows 데스크톱 마법사

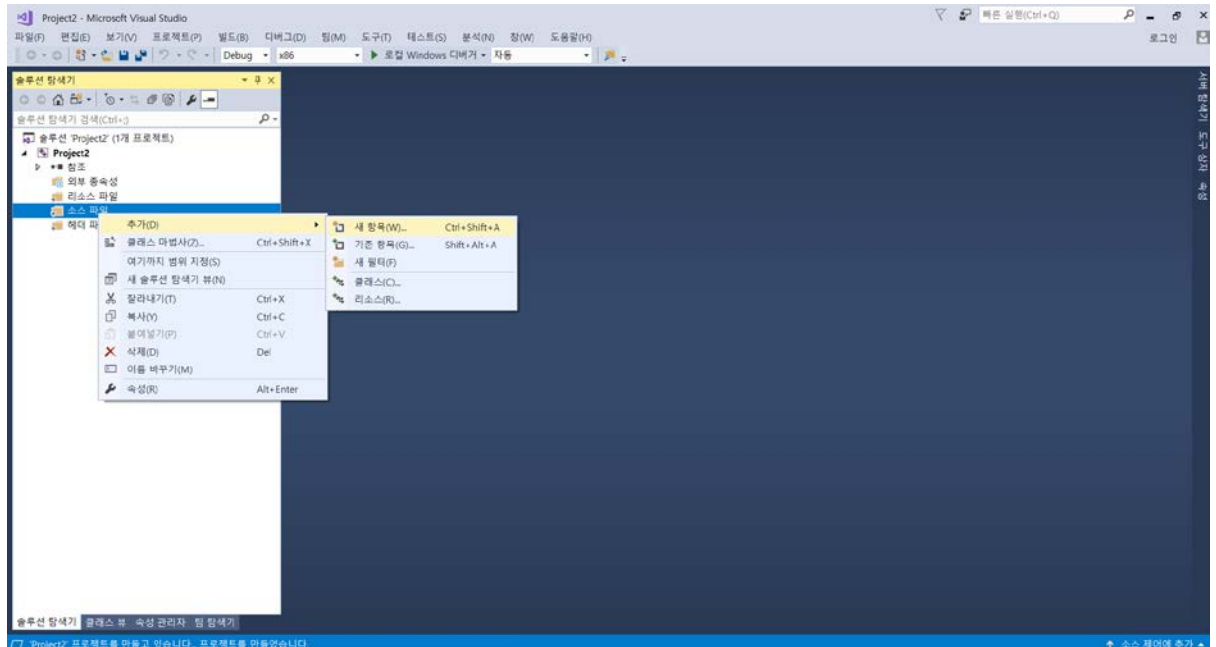


8. 빈 프로젝트 활성화

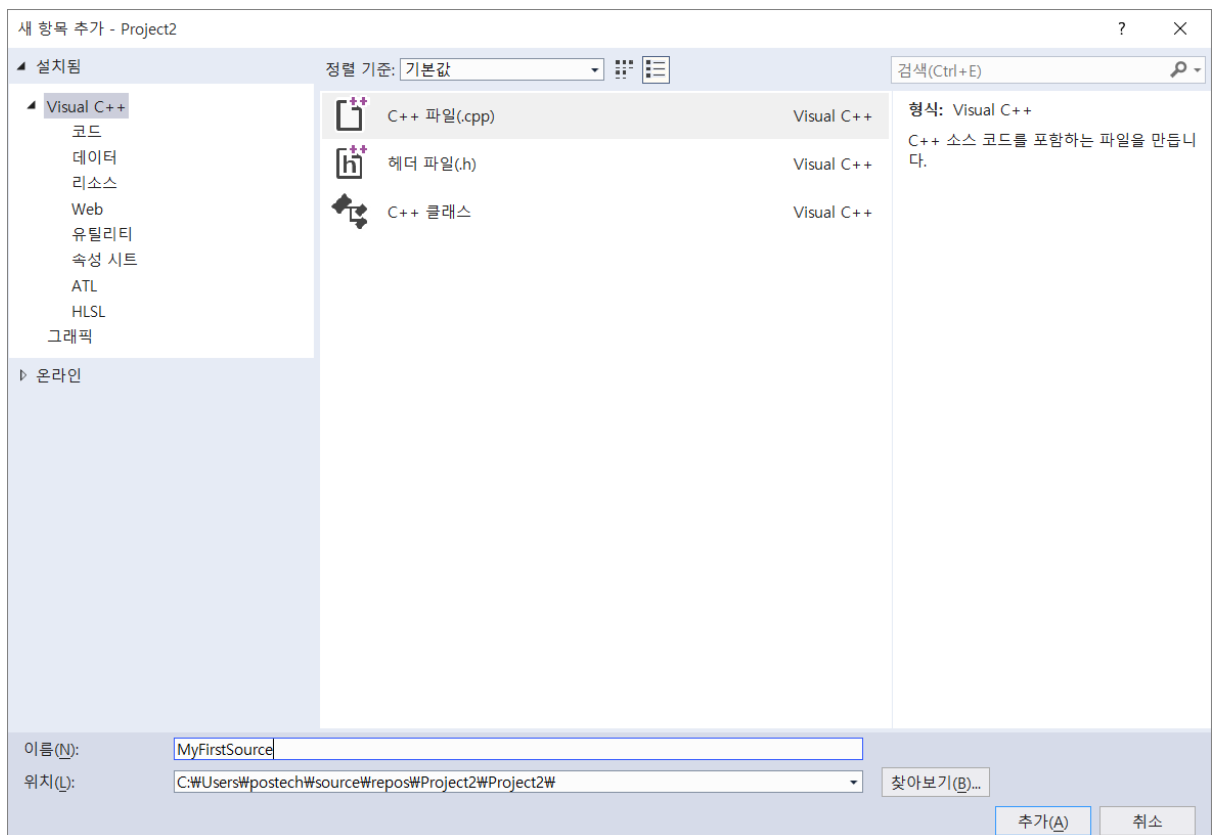


Hello World

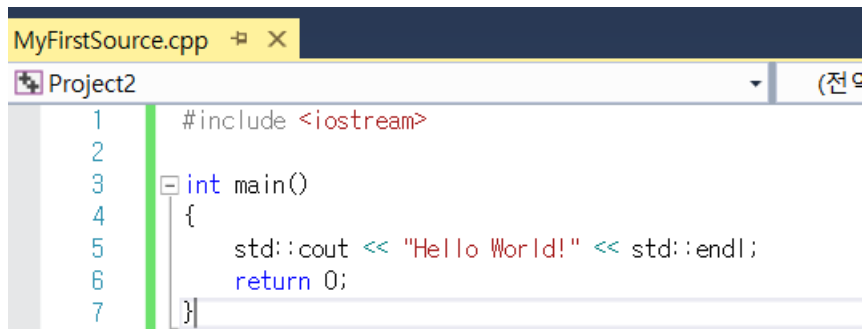
1. 소스파일 > 우클릭 > 추가 > 새 항목



2. MyFirstSource.cpp 파일을 생성합니다

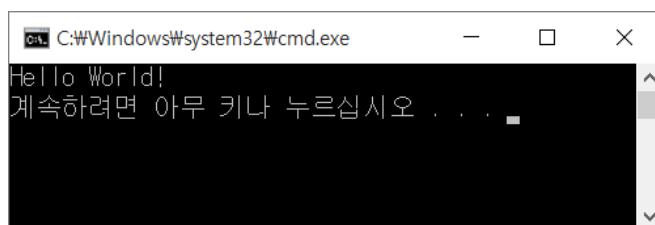


3. 아래와 같이 소스를 작성합니다.



```
MyFirstSource.cpp
Project2
1  #include <iostream>
2
3  int main()
4  {
5      std::cout << "Hello World!" << std::endl;
6      return 0;
7  }
```

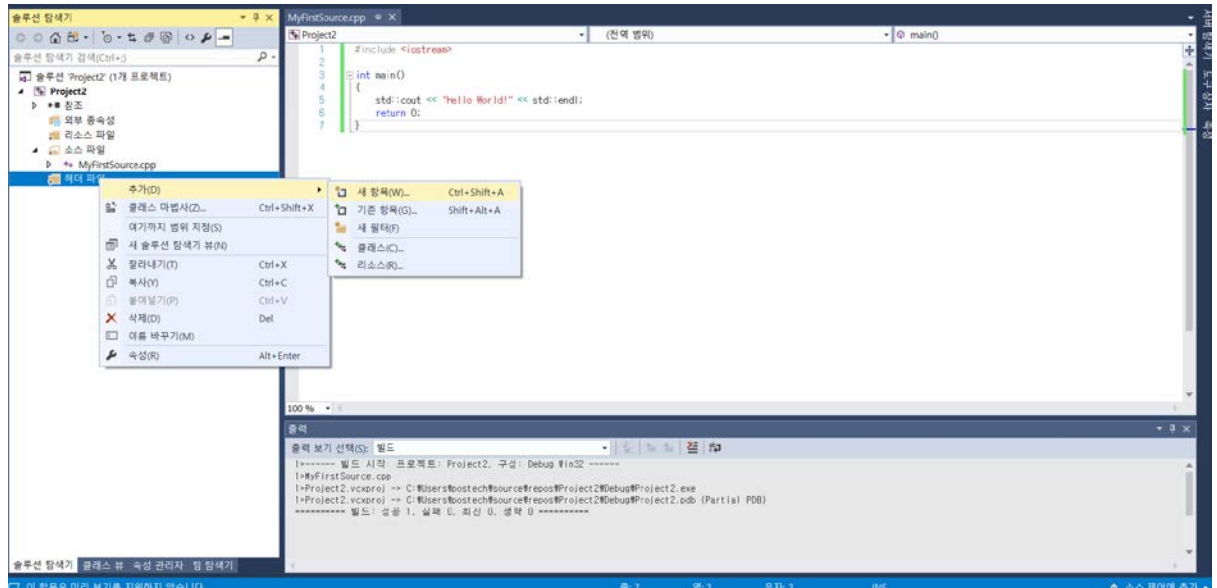
4. Ctrl+f5 로 실행 시 결과창



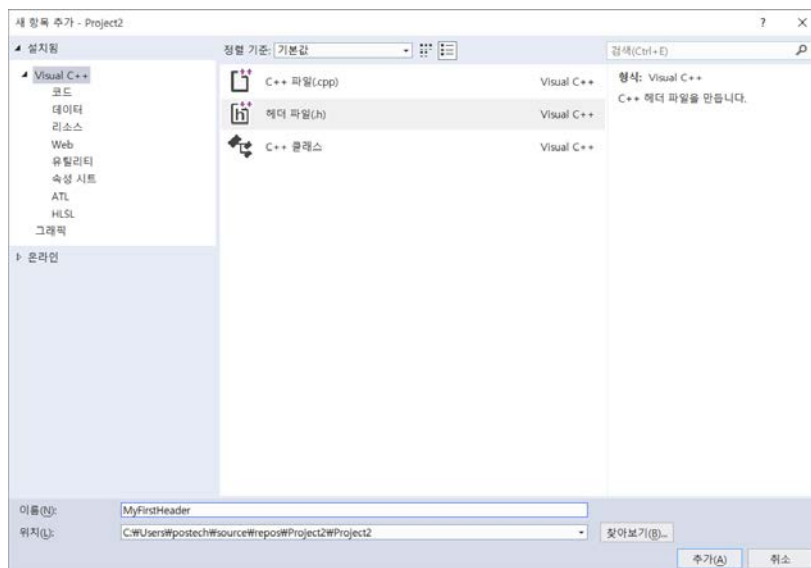
```
C:\Windows\system32\cmd.exe
Hello World!
계속하려면 아무 키나 누르십시오 . . .
```

헤더파일 만들기

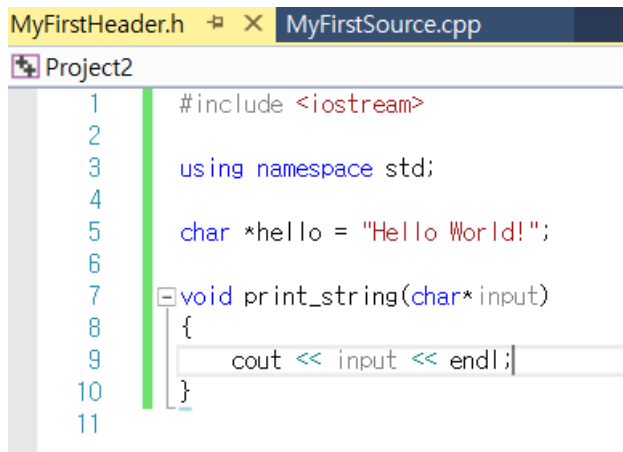
1. 헤더파일 > 우클릭 > 추가 > 새항목



2. 헤더파일 만들기



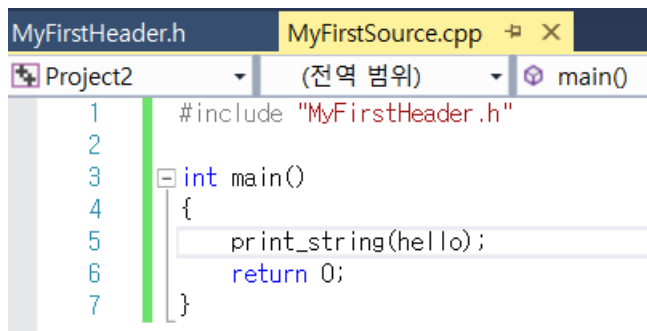
3. 아래와 같이 헤더파일을 작성합니다



The screenshot shows the Visual Studio IDE with the 'MyFirstSource.cpp' file open. The code is as follows:

```
1 #include <iostream>
2
3 using namespace std;
4
5 char *hello = "Hello World!";
6
7 void print_string(char* input)
8 {
9     cout << input << endl;
10 }
11
```

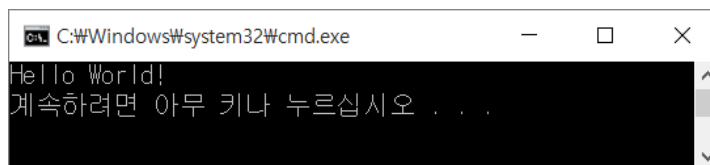
4. 아래와 같이 소스코드를 수정합니다.



The screenshot shows the Visual Studio IDE with the 'MyFirstSource.cpp' file open. The code is as follows:

```
1 #include "MyFirstHeader.h"
2
3 int main()
4 {
5     print_string(hello);
6     return 0;
7 }
```

5. Ctrl+f5 로 실행결과



The screenshot shows a Windows command prompt window titled 'C:\Windows\system32\cmd.exe'. The output is as follows:

```
Hello World!
계속하려면 아무 키나 누르십시오 . . .
```


2017 년 2 학기 csed232 객체지향프로그래밍

MinGW 환경 설정 문서

검토 : 한상도 (hansd@postech.ac.kr)

2017-9-13

업데이트 #3: 박영덕(ydpark@postech.ac.kr)

2016-9-1

업데이트 #2: 김해준 (khj809@postech.ac.kr)

2015-9-5

업데이트 #1: 김동은 (deiori@postech.ac.kr)

2014-9-2

작성: 이운성 (dnsrjd1212@postech.ac.kr)

2013-9-2

1) Windows MinGW

2) 설치

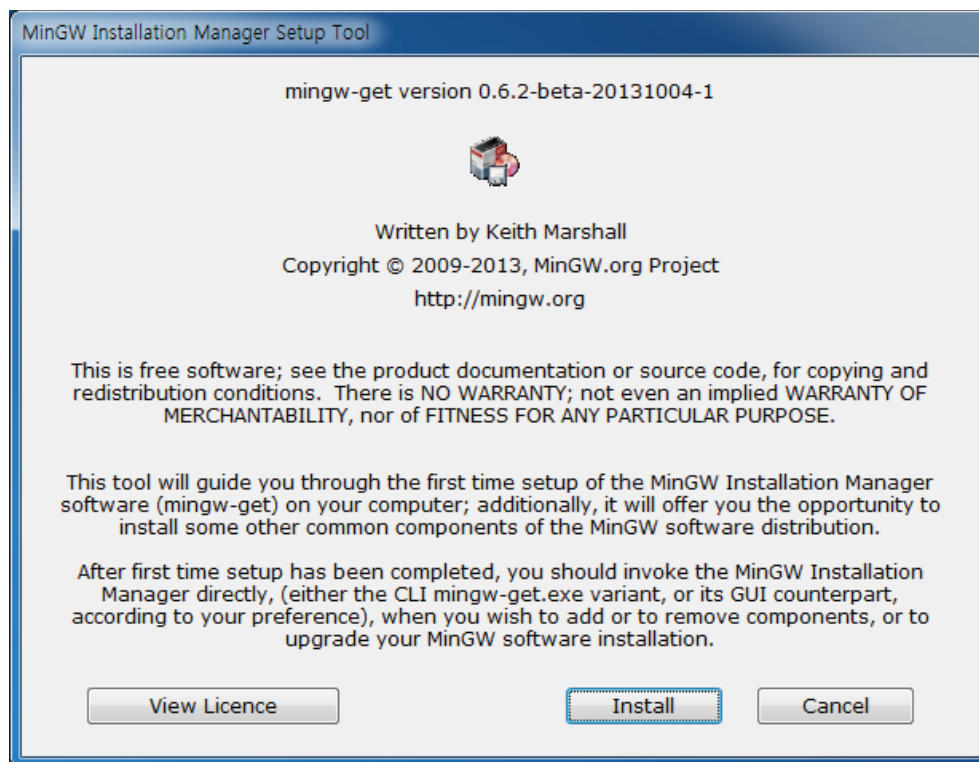
MinGW (<http://www.mingw.org>) 는 Windows에서 gcc, g++을 컴파일러를 이용해 프로그램을 개발할 수 있도록 만들어 놓은 오픈 소스 툴 모음입니다. 다음 주소에서 최신 버전을 받을 수 있습니다.

→ <http://sourceforge.net/projects/mingw/files/latest/download?source=files>

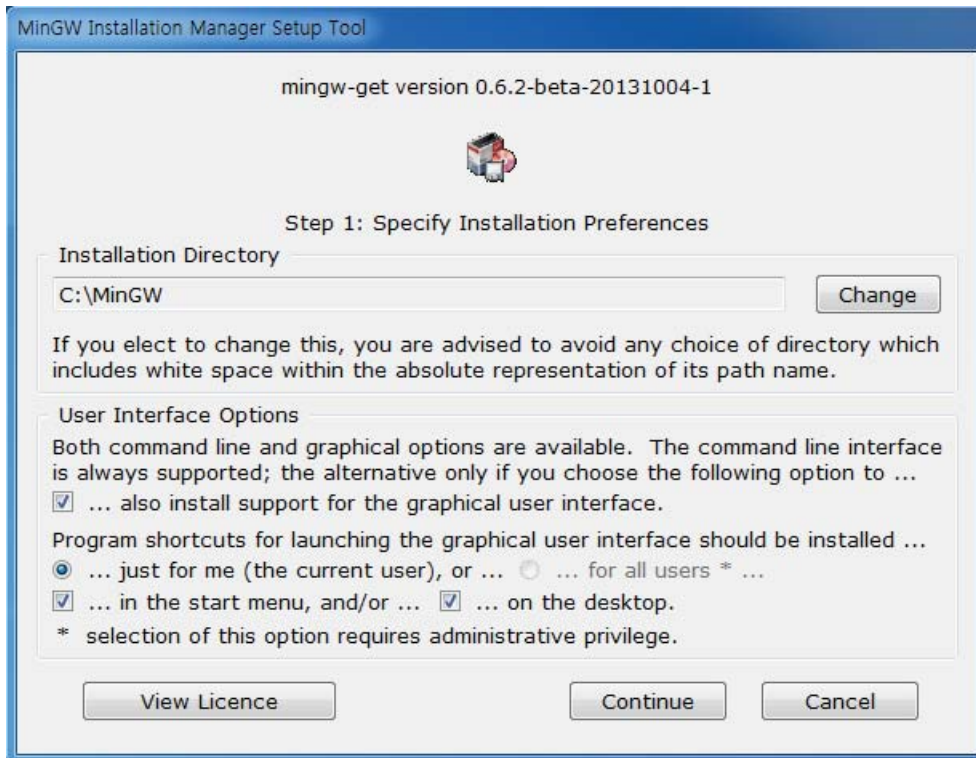
(파일명: mingw-get-setup.exe)

이 프로그램은 MinGW, MSYS, gvim 설치를 포함하고 있습니다.

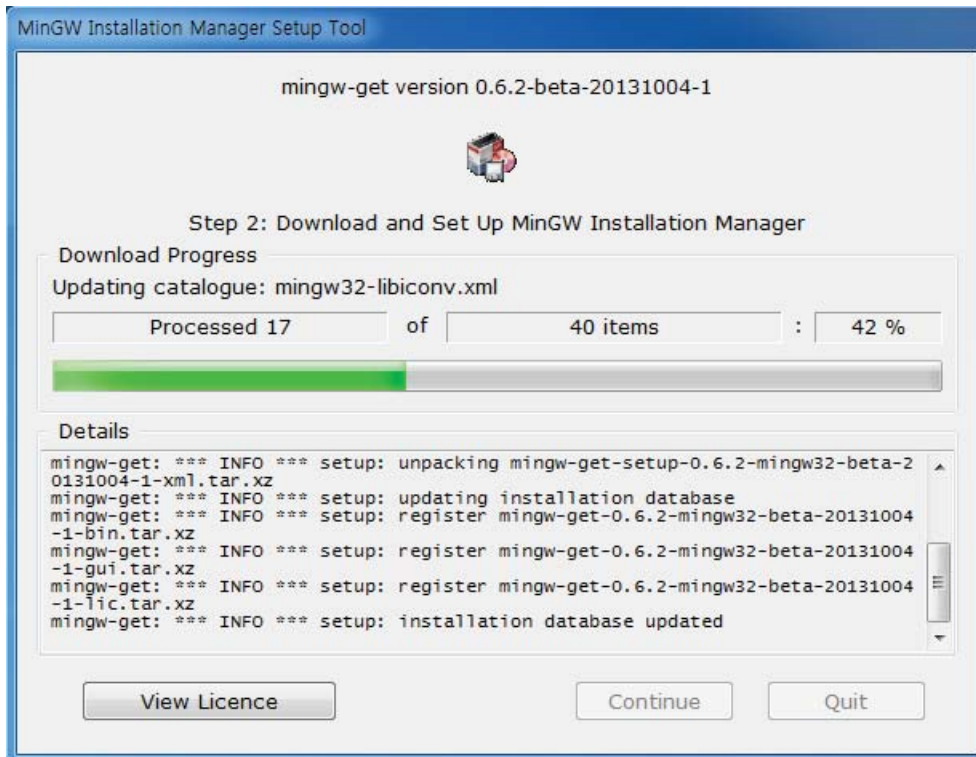
설치 과정은 다음과 같습니다.



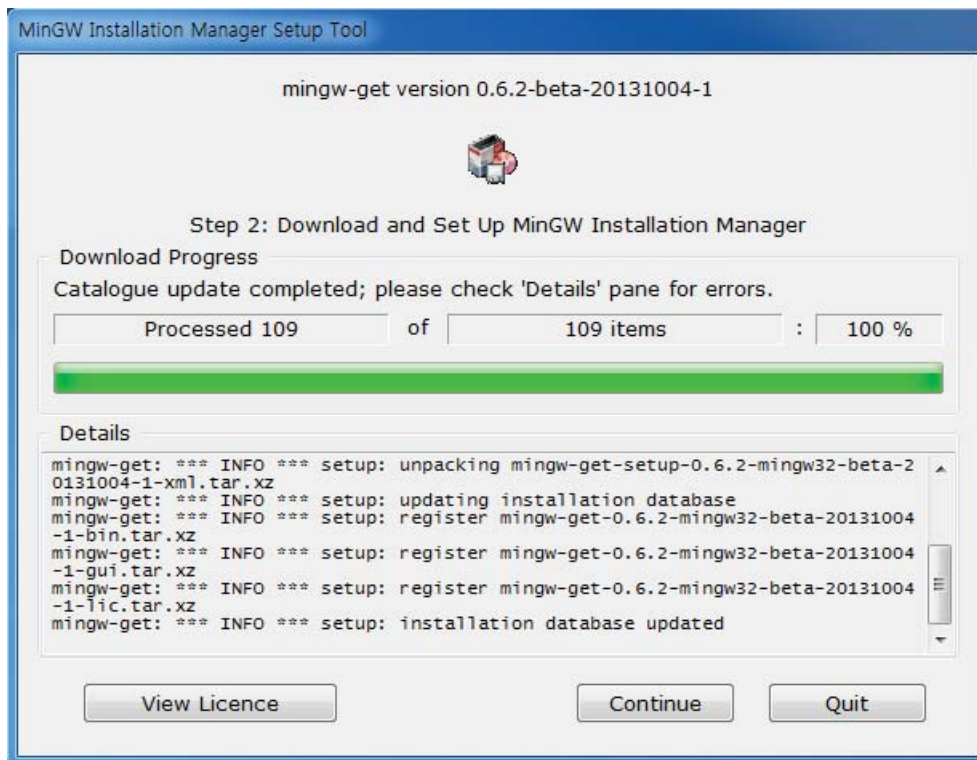
Install.



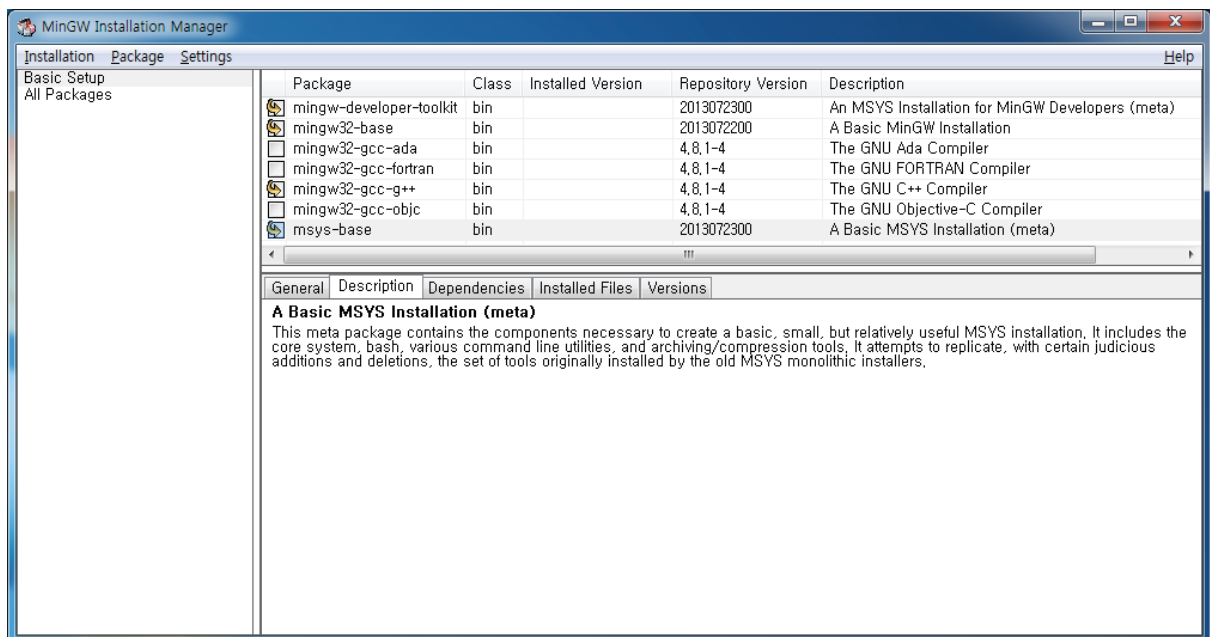
경로, GUI설치 옵션, 그리고 shortcut 설정 후 Continue.



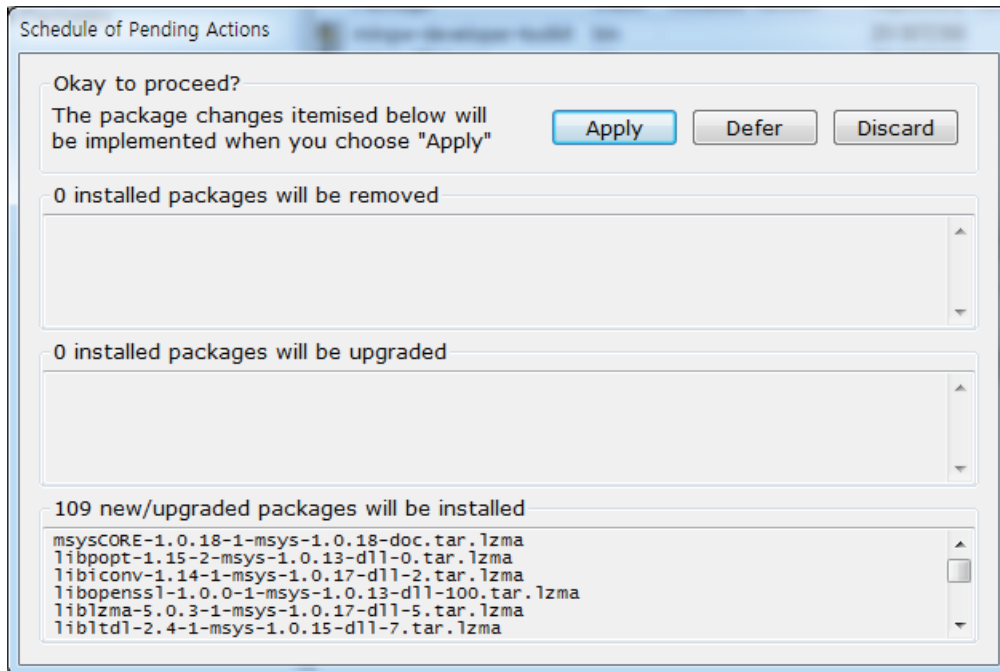
수 초 동안의 다운로드를 수행 후,



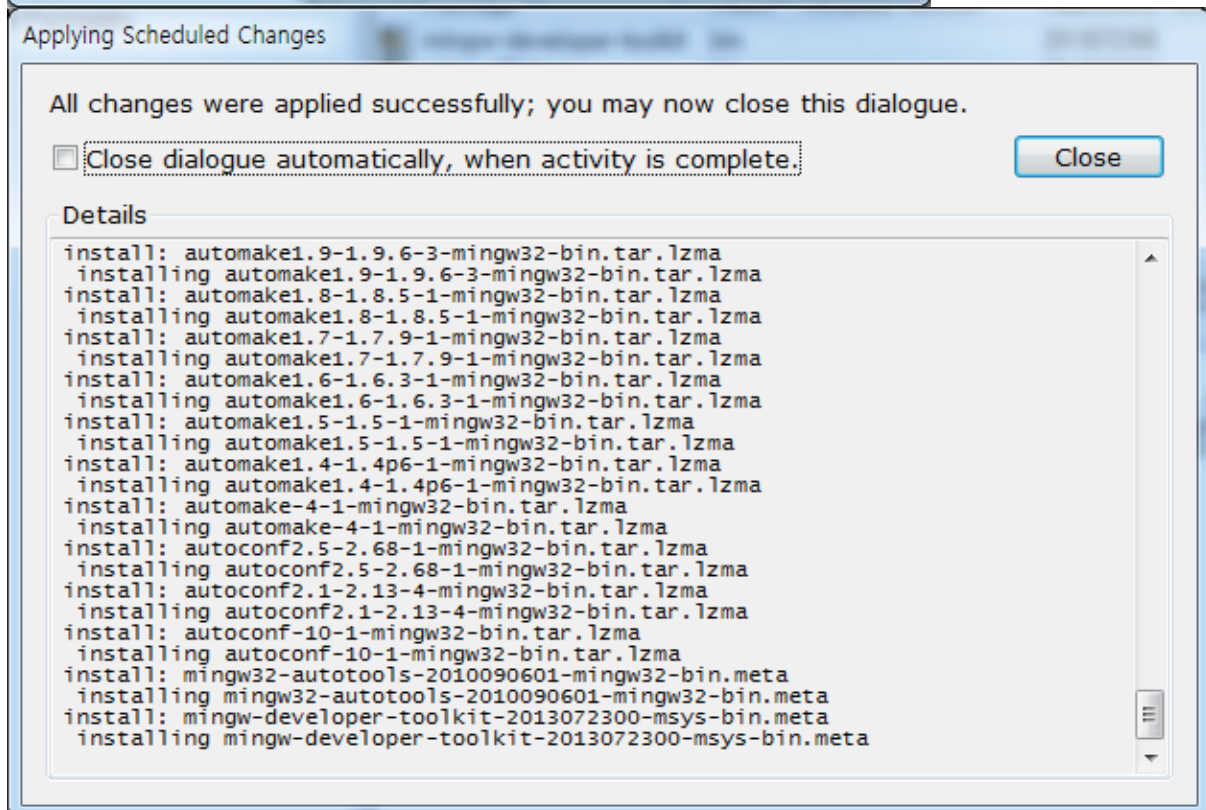
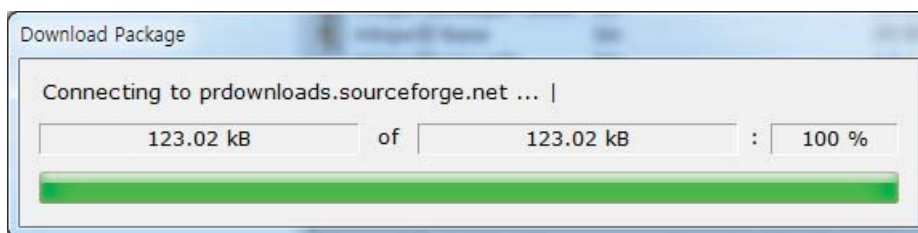
다운이 완료되면 Continue.



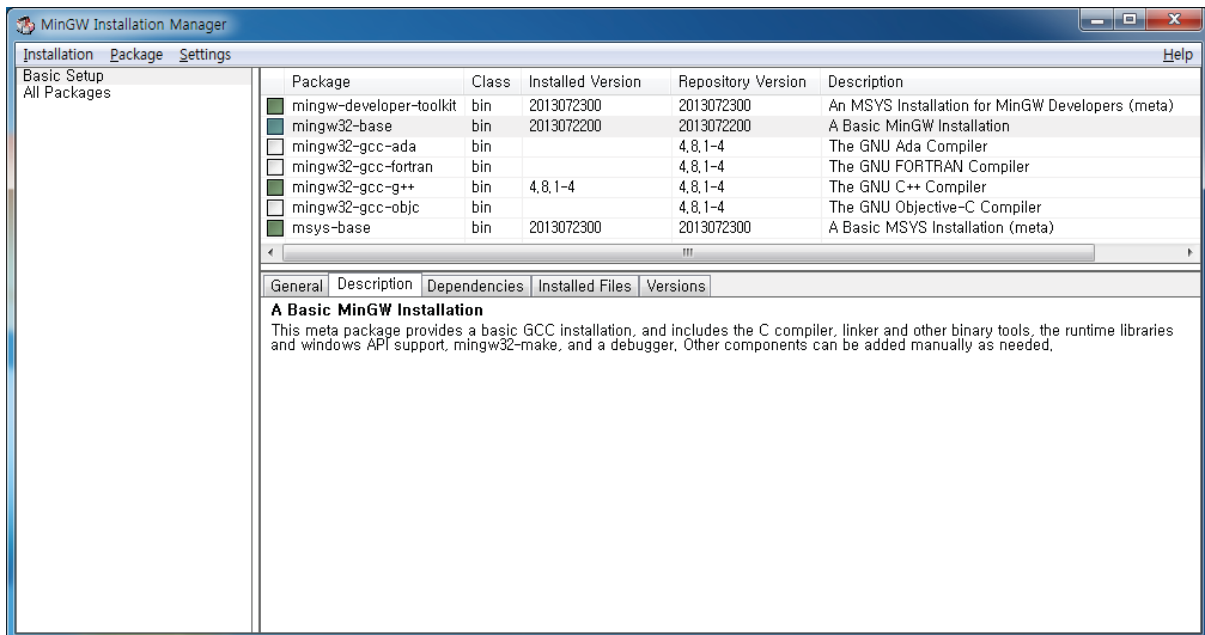
위와 같이 MinGW Installation Manager가 자동실행 됩니다. MSYS와 개발 툴킷, 그리고 C와 C++ 컴파일러를 체크해 주고 (Mark for Installation), 상단 메뉴의 Installation → Apply Changes.



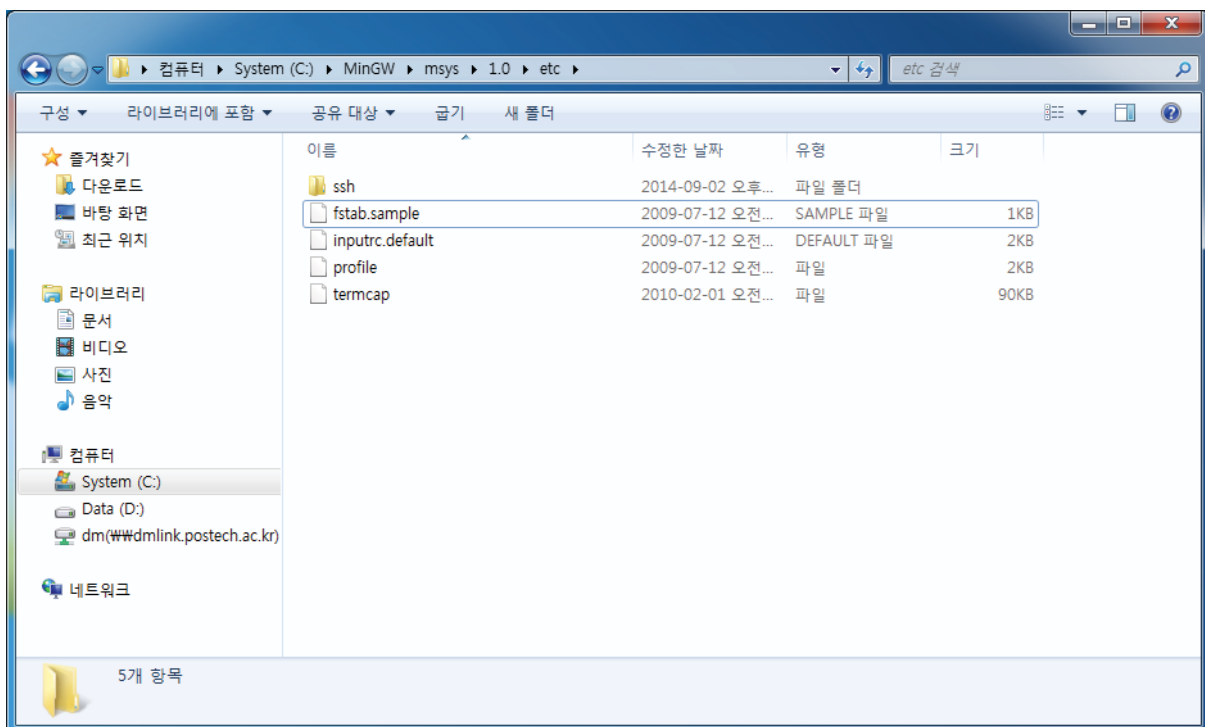
Apply를 누르면 체크한 패키지를 아래와 같이 다운로드 및 설치 하게 됩니다.



완료되면 Close.



설치가 완료된 패키지의 체크박스가 바뀐 모습.

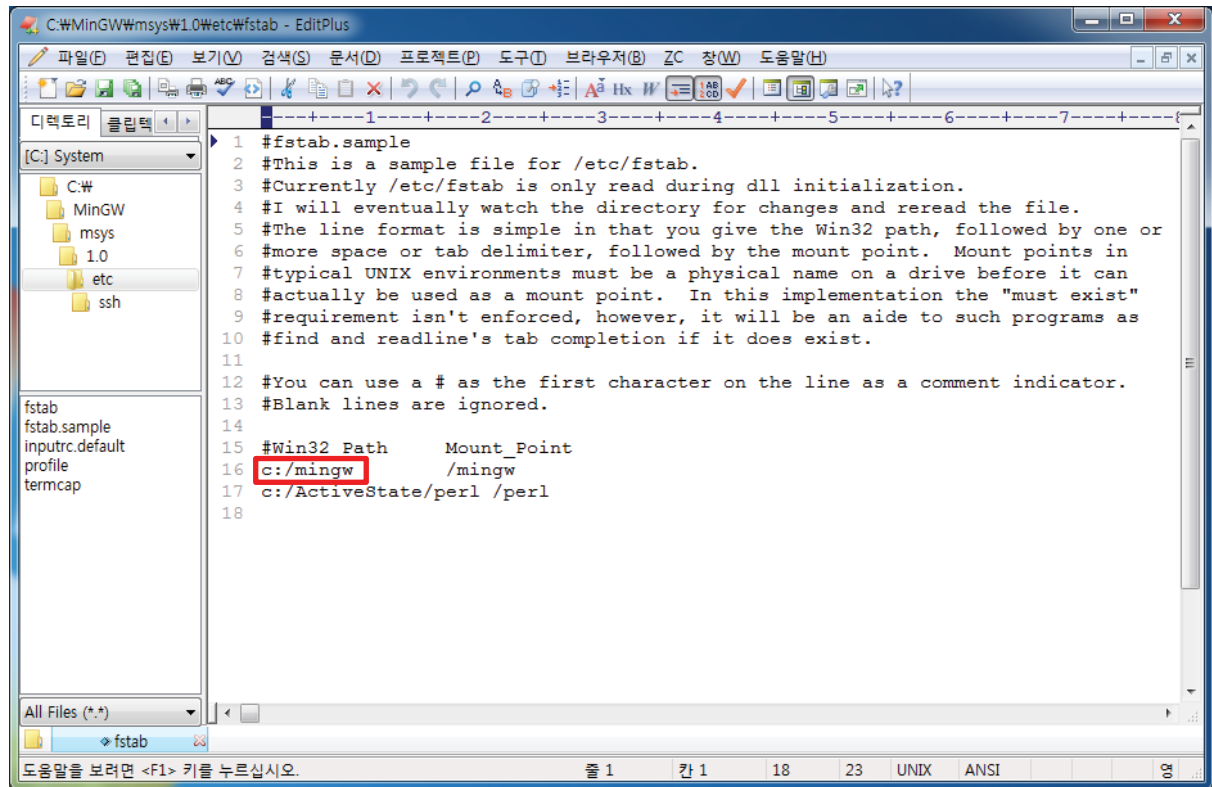


설치 후 설치경로(기본값 c:\MinGW)로 가서, [설치경로]\msys\1.0\etc 로 가면 "fstab.sample" 파일이 있습니다. 해당 파일을 복사 또는 이름 바꾸기를 통하여 "fstab" 파일을 생성합니다.

Note) 만일 fstab 파일이 이미 존재할 경우, 그대로 사용하시면 됩니다.

만약, MinGW의 설치경로를 기본값으로 하지 않았다면, 임의의 텍스트에디터(메모장도 가능)를 이용하여, "fstab" 파일을 열어서 아래 그림의 MinGW 경로 설정 부분을 수정한 후 저장합니다.

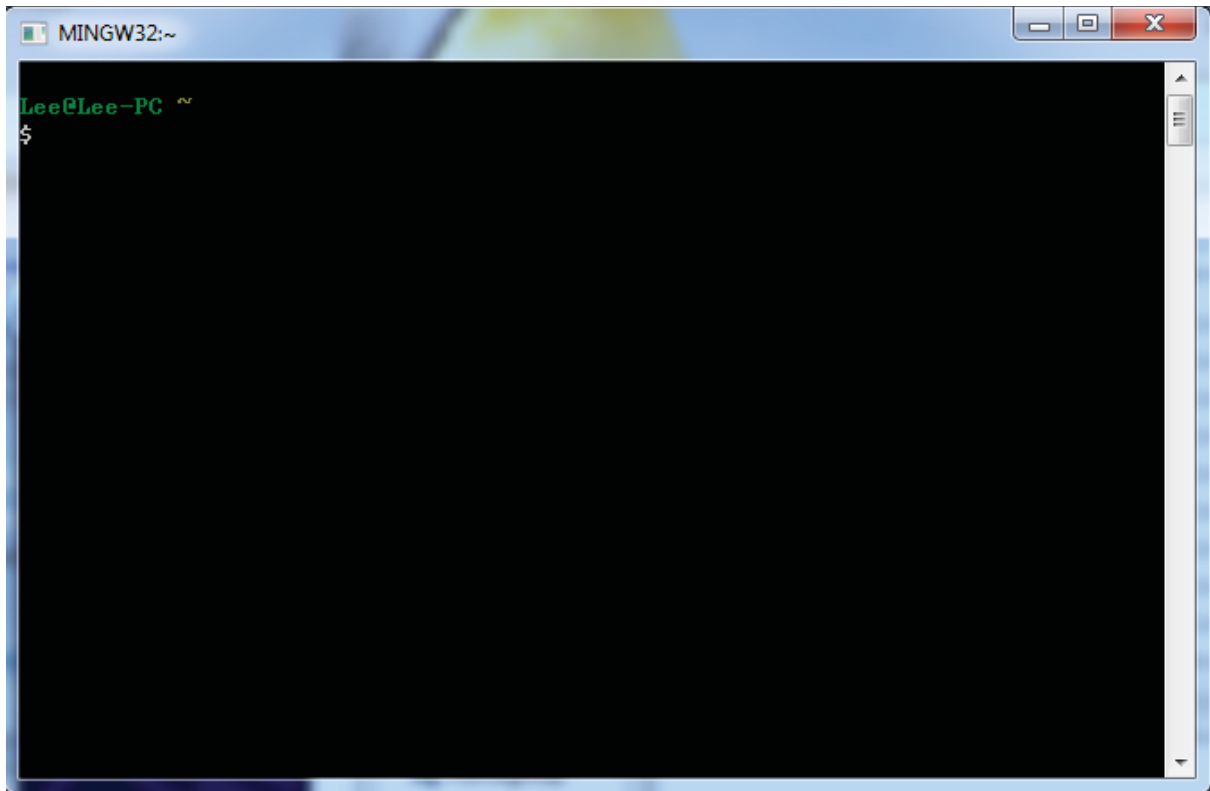
(설치 후, "fstab.sample" 파일이 존재하지 않는다면, 텍스트에디터로 파일 생성 후 아래 그림의 "c:/mingw/mingw" 부분을 자신의 설치 경로에 맞게 입력하고 줄바꿈을 넣은 후 저장합니다.)



```
1 #fstab.sample
2 #This is a sample file for /etc/fstab.
3 #Currently /etc/fstab is only read during dll initialization.
4 #I will eventually watch the directory for changes and reread the file.
5 #The line format is simple in that you give the Win32 path, followed by one or
6 #more space or tab delimiter, followed by the mount point. Mount points in
7 #typical UNIX environments must be a physical name on a drive before it can
8 #actually be used as a mount point. In this implementation the "must exist"
9 #requirement isn't enforced, however, it will be an aide to such programs as
10 #find and readline's tab completion if it does exist.
11
12 #You can use a # as the first character on the line as a comment indicator.
13 #Blank lines are ignored.
14
15 #Win32 Path      Mount_Point
16 c:/mingw         /mingw
17 c:/ActiveState/perl /perl
18
```


3) 실행

설치 후 MinGW설치 폴더(아마도 대부분 default로 C:\MinGW를 선택하였을 것입니다.)로 갑니다. 그러면 msys 폴더가 있습니다. 그리고 그 폴더의 하위에 1.0 폴더가 있습니다(MinGW\msys\1.0). 완료되면 msys.bat (파일 유형: Windows 배치 파일)을 실행시킵니다.



이제 C++ 개발을 위한 기본 준비는 끝이 났습니다.

MSYS 콘솔 내에서 사용되는 명령어는 리눅스 터미널에서 사용되는 명령어와 비슷합니다.

pwd: 현재 경로를 출력합니다.

ls [옵션] [파일명]: 현재 경로에서의 하위 디렉토리 및 파일 표시합니다.

cd [디렉토리]: 경로 이동 명령어로, "."은 현재 디렉토리, ".."은 상위 디렉토리를 의미합니다.

cp [source] [Destination]: Source파일을 Destination의 디렉토리 or 파일명으로 복사합니다.

mv [source] [Destination]: Source에서 Destination으로 디렉토리 or 파일명으로 이동합니다.

mkdir [옵션] [디렉토리명]: 디렉토리 생성합니다.

rm [옵션] [파일 or 디렉토리]: 파일이나 디렉토리 제거, 해당 디렉토리의 모든 하위 디렉토리 및 파일을 제거 하고 싶다면 rf 옵션을 주면 됩니다. (rm -rf 디렉토리명)

find [옵션] [파일명, 디렉토리]: 찾고자 하는 파일 혹은 디렉토리의 경로를 보여줍니다. -name 옵션이 잘 이용됩니다. 이는 파일의 이름으로 검색합니다.

grep [옵션] [패턴] [파일명]: 이 명령어는 파일 내부에 해당 word를 검색해서 찾아줍니다. 이 때 파일명을 *로 나타낼 경우 모든 파일에 해당하는 word를 찾아줍니다. 패턴은 찾고자 하는 word

를 입력합니다.

(예) `grep -r -n -i "Hello world" *`,

현재 디렉토리의 하위 파일들을 대상으로 Hello world가 들어간 파일들의 경로를 보여줍니다. 이 때 l 옵션은 대소문자 무시, n 옵션은 해당 파일을 찾았을 때 해당 단어가 포함된 line number를 표시, r 옵션은 현재 디렉토리의 하위 디렉토리까지 모두 포함시켜서 찾아달라는 것입니다.

cat [파일명]: 파일의 내용을 출력할 때 사용됩니다.

exit: 종료

* 추가적인 명령어

mingw-get install [프로그램]: 추가 패키지를 설치하기 위한 명령어입니다.

[프로그램]:

1. `msys-wget`: `wget` 명령어를 사용하여 해당 주소로부터 파일을 local pc로 받아옵니다.

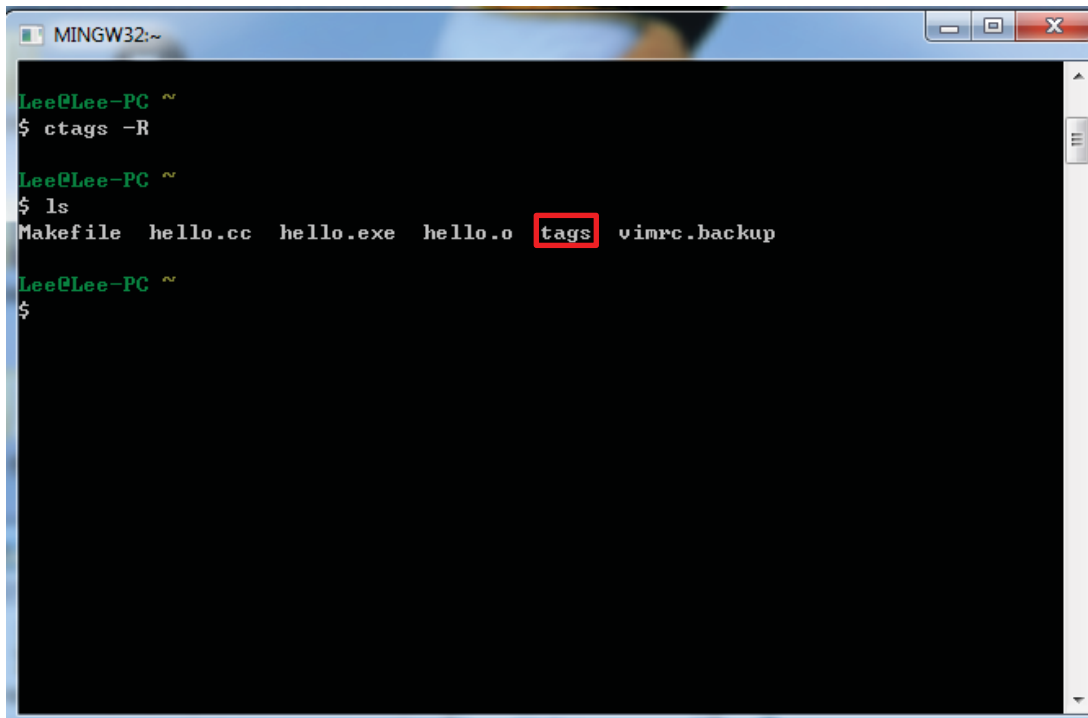
예) `wget http://.....`

2. `msys-zip`: 압축 명령어
3. `msys-unzip`: 압축해제 명령어

And so on.

추가적인 프로그램

디버깅 혹은 개발의 편의성을 위해서 리눅스에서는 `ctags`를 제공합니다. 이것은 (전역 변수, 함수 정의, 매크로 선언)에 대한 tag를 생성하여 vim과 같은 편집기를 사용하는 사용자로 하여금 쉽게 이것들의 선언문 혹은 선언문에서 이것들이 호출 된 곳을 쉽게 따라 갈 수 있도록 환경을 제공합니다. http://sourceforge.net/projects/ctags/files/ctags/5.8/ctags58.zip/download?use_mirror=jaist에서 윈도우용 `ctagsxx.zip` 파일을 다운받아서 실행 파일(`ctags.exe`)을 `msys`의 bin폴더(MinGW설치 폴더\msys\1.0\bin)에 옮겨 놓습니다. 그리고 나서 콘솔 창으로 돌아와서 tag를 만들고 싶은 최상위 디렉토리에서 `ctags -R` 명령어를 입력합니다. 그러면 tag가 생성됩니다.



```
MINGW32:~
Lee@Lee-PC ~
$ ctags -R
Lee@Lee-PC ~
$ ls
Makefile  hello.cc  hello.exe  hello.o  tags  vimrc.backup
Lee@Lee-PC ~
$
```

Vim 과 같은 편집기에서 다음과 같이 사용 할 수 있습니다.

== 간단한 ctags 명령어 목록 ==

:ta [name] :: name과 일치하는 태그 위치로 이동

ctrl +] :: 커서가 위치하고 있는 함수나 구조체의 정의로 이동

ctrl + t :: 이전 위치로 돌아오기

:ts [name] :: name과 일치하는 태그 목록 출력

:ta /[name]:: name과 일치하는 태그 목록 출력

:tj [name] :: 목록이 한개인 경우 이동, 여러개인 경우 목록 출력

:tn :: 다음 태그로 이동 (tag next)

:tp :: 이전 태그로 이동 (tag previous)

:tags :: 이동한 태그 히스토리 목록 출력

출 처: <http://sungto.tistory.com/34>

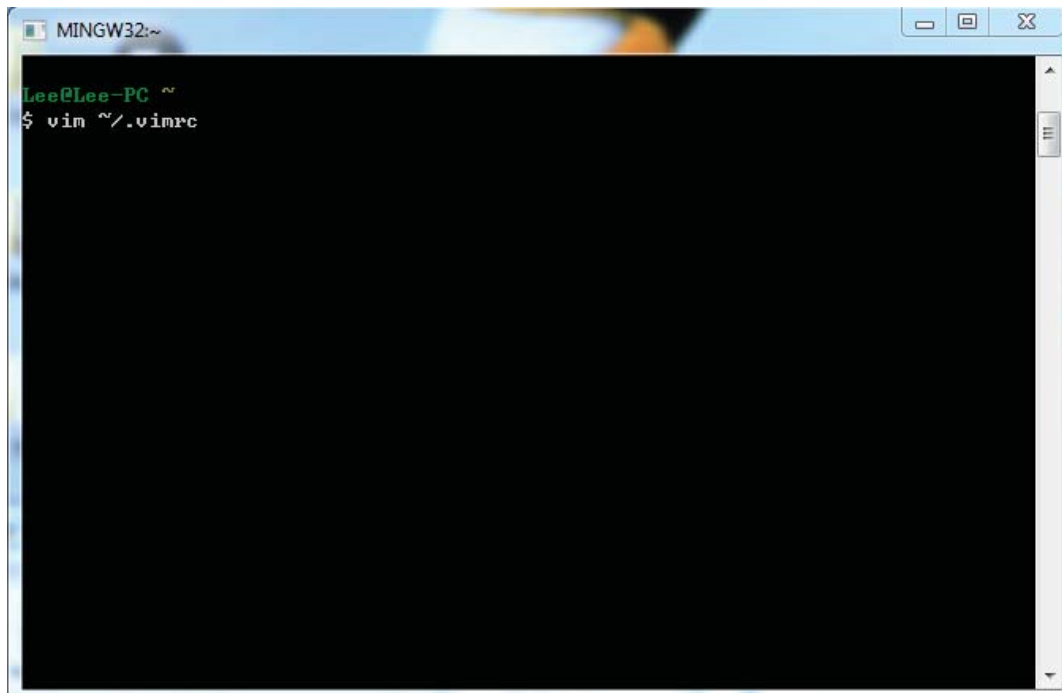
VIM 편집기 이용

콘솔 창에서 vim [파일명]을 입력하면 해당 파일을 vim 편집기로 엽니다. (MSYS에는 vi 대신 vim 만 내장되어 있어서, 별도의 다운로드가 필요)

VIM 편집기 사용 tip

Vim을 처음 사용 할 경우 초기 설정을 해주어, 좀 더 편리하게 프로그래밍을 할 수 있도록 합니다.

Vim ~/.vimrc 명령어를 입력합니다.



해당 편집기를 열어서 다음과 같이 기능을 추가 해주면 됩니다.

:set autoindent	"자동 들여쓰기
:set cindent	"c문법의 들여쓰기
:set smartindent	"좀 더 스마트한 들여쓰기
:set nu	"편집기 상에서 line number 표시
:set bs=2	"back space 허용
:set ts=4	"tab을 사용시 이동을 4칸으로 고정
:set sw=3	"들여쓰기시에 3칸을 띄웁니다.
:set bg=dark or light	"배경을 어둡게 혹은 밝게
:set nobackup	"backup 파일 금지
:set visualbell	"키를 잘 못 눌렀을 때 소리 대신 깜빡임
:set ruler	"오른쪽 하단에 현재 커서의 row, column 표시
:set hlsearch	"vim 내에서 word 검색 시 강조해서 표시
:set laststatus=2	"vim 내에서 해당 파일의 제목을 항상 표시
:syntax on	"문법 강조

색상 (syntax on과 함께 사용 됩니다.)

colo or colorscheme [이름]	"해당 color scheme을 이용
hi [해당 type] [속성]	"해당 type의 word를 속성처럼 표시
→ 예) hi Type cterm=underline guifg=Blue "변수의 type에 밑줄을 긋고, 파란색으로 표시	

매핑 (단축키 설정)

예)

Map <F1> :wq<CR>

"F1 function key를 누르면 vim editor를 저장하고 종료하라,
<CR>은 enter를 의미합니다.

Makefile 생성

make 는 프로그램의 컴파일 및 기타 작업을 도와주는 툴입니다. make를 이용하면 g++을 이용해 커맨드라인에서 직접 컴파일하는 것보다 편하게 프로그램을 컴파일할 수 있습니다.

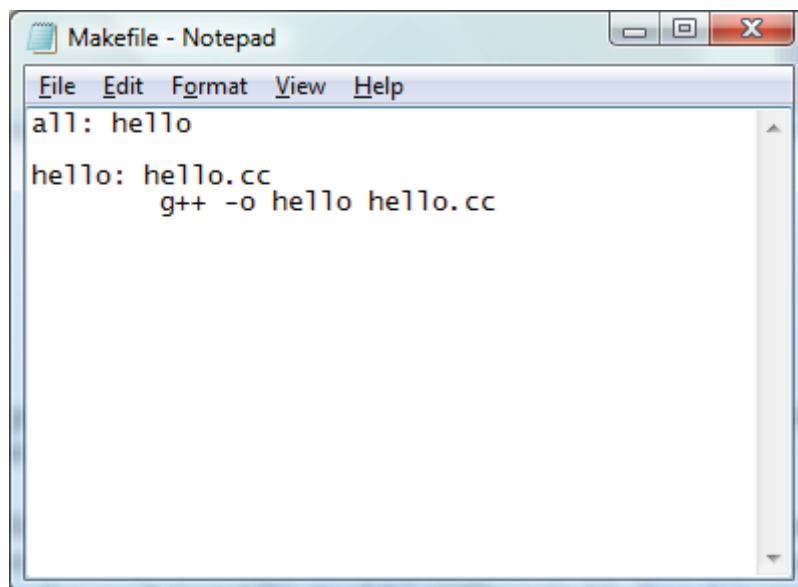
make를 이용하기 위해서는 우선 Makefile을 만들어야 합니다. Makefile은 컴파일을 하는 방법에 대한 규칙을 기술하는 텍스트 파일입니다. 규칙은 다음과 같습니다.

```
Target: dependencies ...  
      commands  
      ...
```

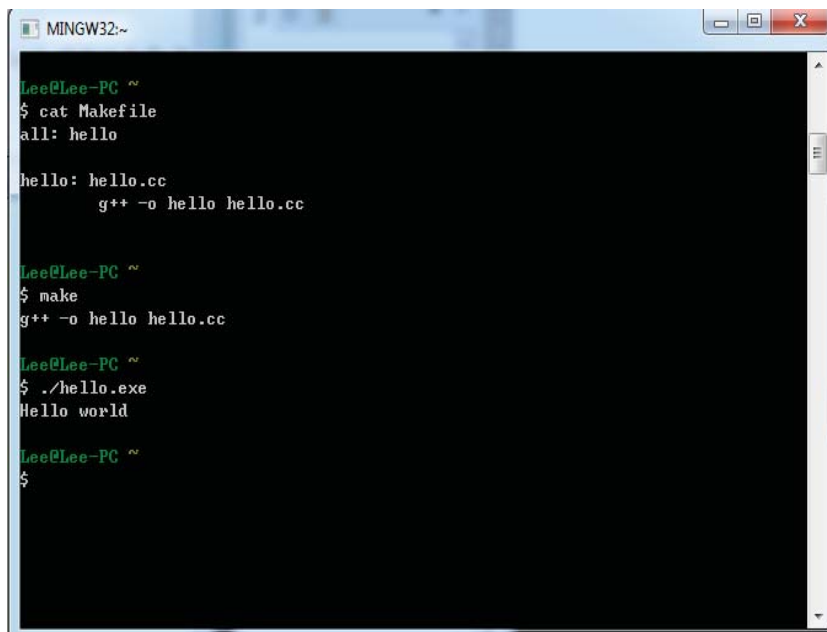
예를 들어, hello 라는 프로그램을 만들고 싶다면 다음과 같이 Makefile을 작성하면 됩니다. (commands 앞은 tab키를 이용해서 띄우셔야 합니다.)

```
hello: hello.cc  
      g++ -o hello hello.cc
```

그리고 실행은 커맨드라인에서 make hello를 치시면 Makefile에 지정된 규칙에 따라 실행이 되게 됩니다. 즉, hello는 hello.cc에 dependency가 있으므로, hello의 파일수정시간이 hello.cc의 파일수정시간보다 이전이면 컴파일 명령이 실행됩니다. hello 파일이 hello.cc 파일보다 최신이면 컴파일 명령이 실행되지 않습니다. 혹은 좀 더 편하게 이용하기 위해서는 다음과 같이 할 수도 있습니다.



이렇게 하면 커맨드라인에서 make만 치시면 make all과 같은 효과가 나고 따라서 hello가 만들어 지게 됩니다.



```
MINGW32:~
Lee@Lee-PC ~
$ cat Makefile
all: hello

hello: hello.cc
    g++ -o hello hello.cc

Lee@Lee-PC ~
$ make
g++ -o hello hello.cc

Lee@Lee-PC ~
$ ./hello.exe
Hello world

Lee@Lee-PC ~
$
```

make 는 매우 유용한 툴로 유닉스나 리눅스 환경 등에서 매우 편리하게 사용하실 수 있습니다.
보다 자세한 내용은 <http://www.gnu.org/software/make/manual/> 를 참고하시기 바랍니다.