

# gcc 프로그래밍 환경구축

## 1. jdk 설치

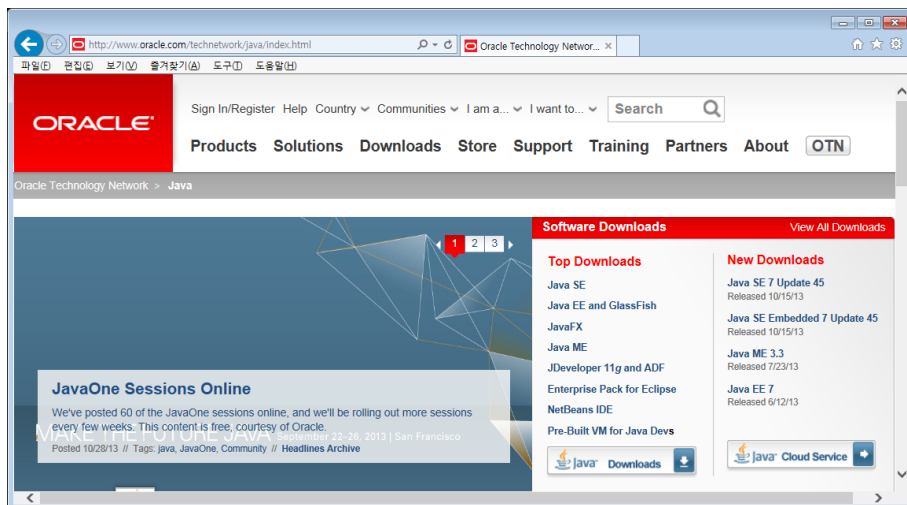
jdk는 java development kit 으로서 자바 클래스 파일을 구동하기 위한 jvm을 포함하여 자바 개발을 위한 필수 라이브러리 및 컴파일러와 실행기를 포함하고 있습니다.

본 과정은 자바과정이 아닌 C과정이므로 자세한 설명은 생략하도록 하겠습니다. 자바를 설치하는 이유는 C/C++개발을 위한 통합개발환경 툴로써 이클립스를 사용하는데 있어서 이클립스가 자바기반으로 생성된 툴이므로 jdk가 설치되지 않으면 이클립스를 구동할 수 없으므로 반드시 jdk를 설치하여야만 합니다.

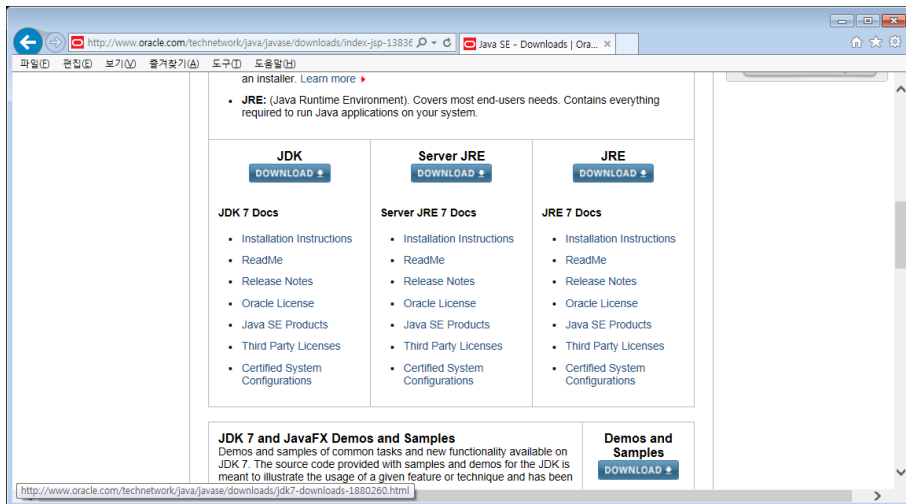
jdk 는 아래의 홈페이지를 통해 다운로드가 가능하다.

<http://java.sum.com>

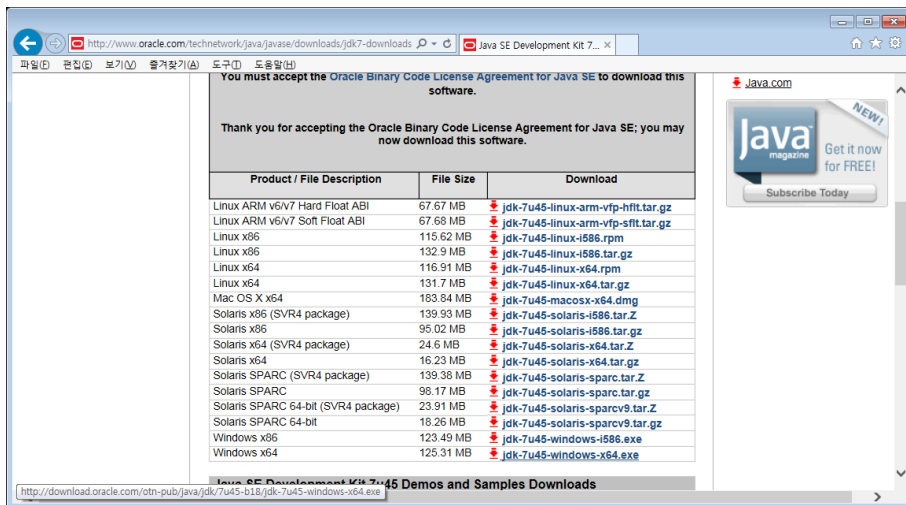
위의 페이지에 접속하였다면 페이지 좌측부분의 Java SE7 Update 45 링크를 클릭하도록 합니다.



아래의 페이지로 이동하였다면 JDK하단의 다운로드 아이콘을 클릭합니다.



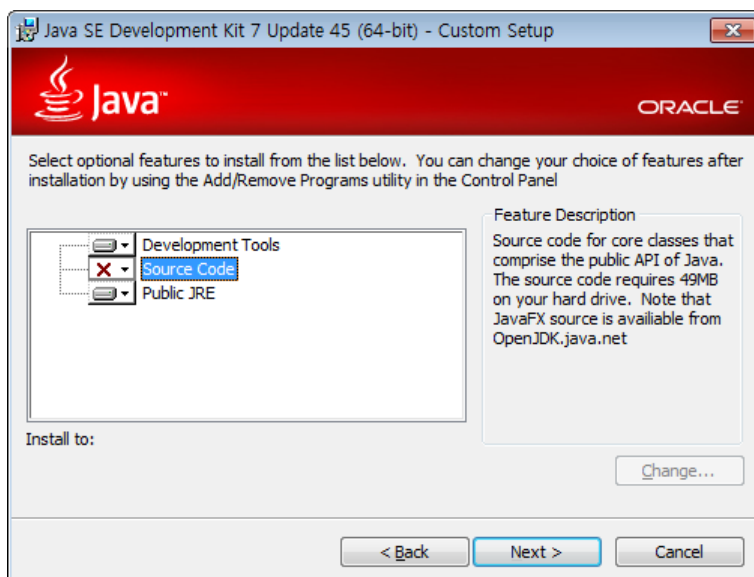
아래의 페이지에서 라이선스 동의 라디오 버튼을 클릭한 후, 자신의 시스템에 맞는 인스톨러를 다운로드 합니다.



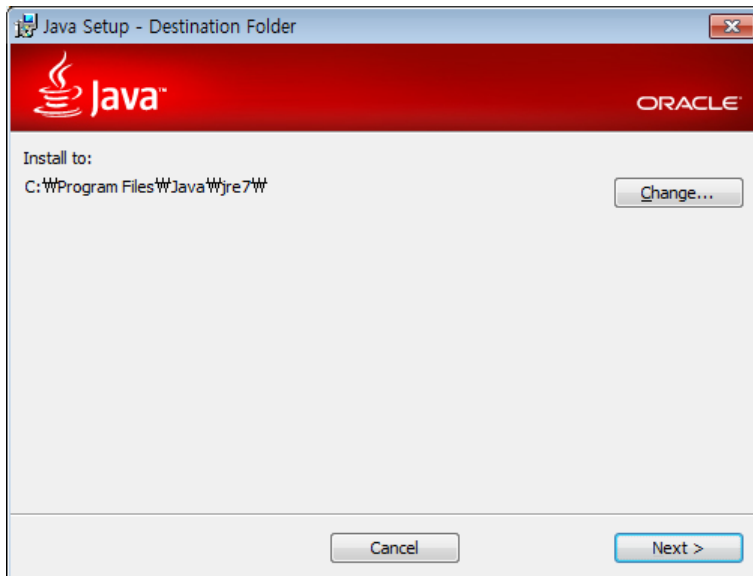
다운로드한 인스톨러를 실행하도록 합니다. 환영메세지를 보여주는 화면입니다. Next 버튼을 클릭하여 계속 진행합니다.



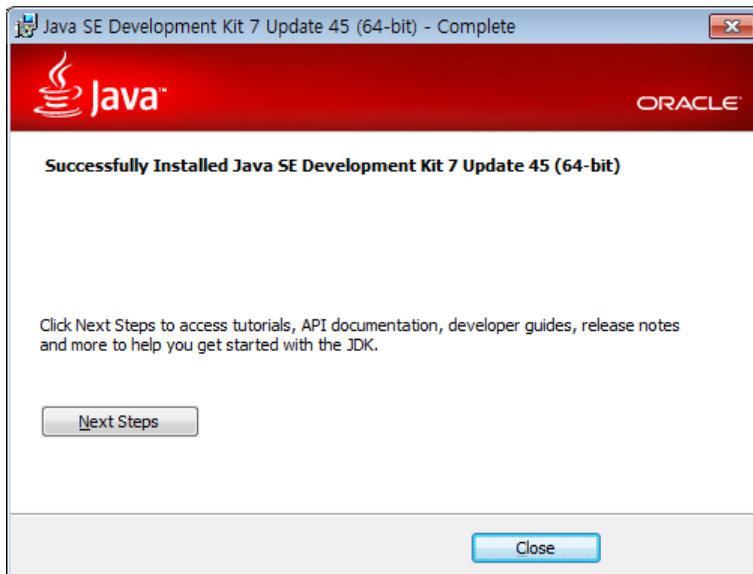
설치할 항목을 선택합니다. Java Source 는 필요없는 항목이므로 제외하고 설치하도록 합니다. 항목을 선택하셨다면 Next 버튼을 클릭하여 계속 진행합니다.



JDK 설치가 완료되면 JRE를 설치하게됩니다. JRE는 Java Runtime Environment 로써 자바를 실행하기 위한 실행기를 포함하고 있습니다. JRE역시 자세한 설명은 생략하도록 하겠습니다. Next 버튼을 클릭하여 설치를 진행합니다.



JRE가 설치되고 나면 이클립스를 구동하기 위한 준비가 끝납니다. Close 버튼을 클릭하여 JDK설치를 완료합니다.



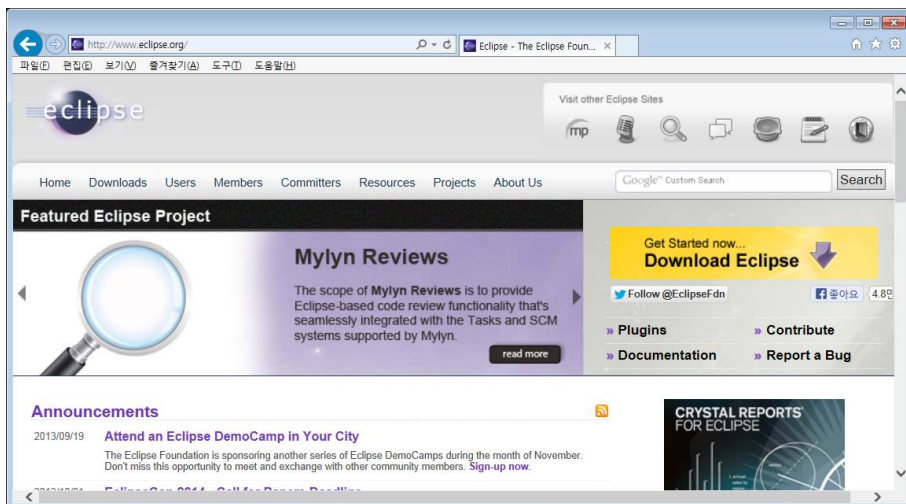
## 2. 이클립스 설치

이클립스는 통합개발툴로써 자바뿐만이 아니라 C나 C++ 그 밖에 php 등등 여러 프로그래밍을 개발할 수 있는 환경을 제공합니다.

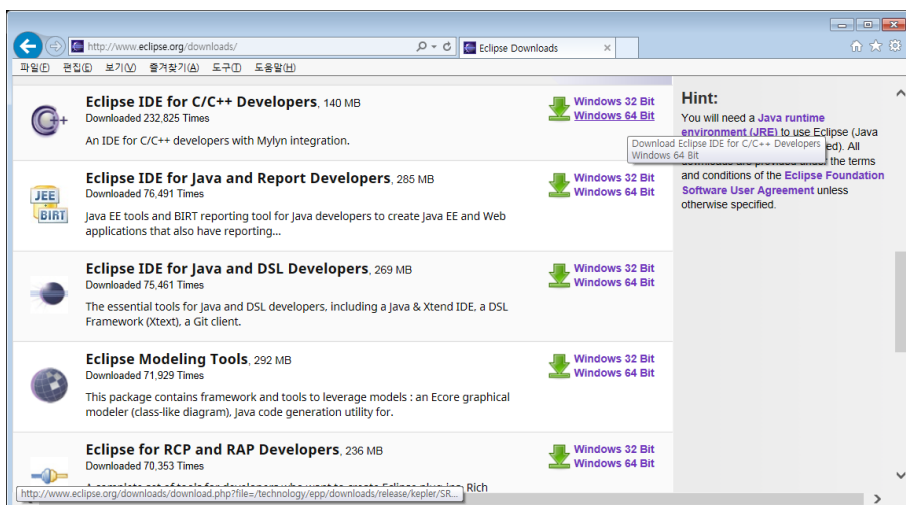
아래의 사이트에 접속하여 이클립스를 다운로드 받도록 합니다.

<http://www.eclipse.org>

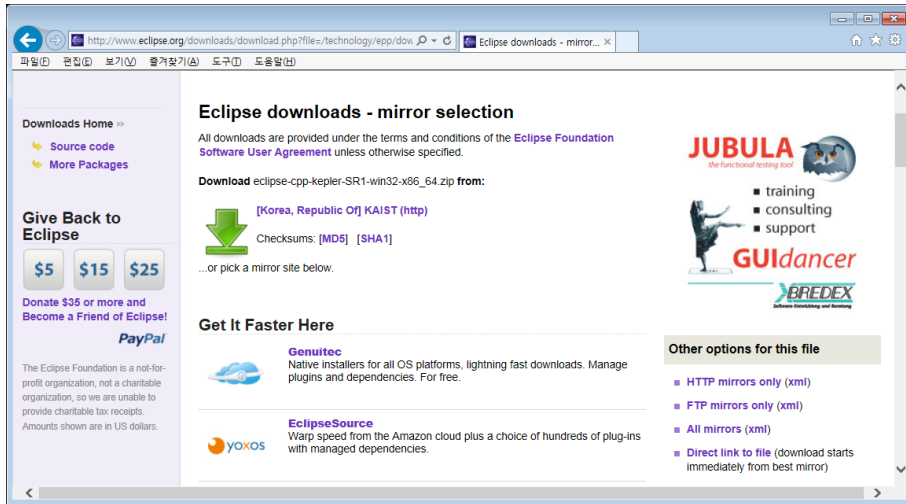
위의 주소에 접속하셨다면 Downloads 링크를 클릭하도록 합니다.



화면을 스크롤 하여 Eclipse IDE for C/C++ Developers(이하 CDT라 함)를 다운로드 합니다. 이때 역시 자신의 시스템에 맞는 링크 Windows 32Bit 또는 Windows 64Bit 를 클릭합니다.

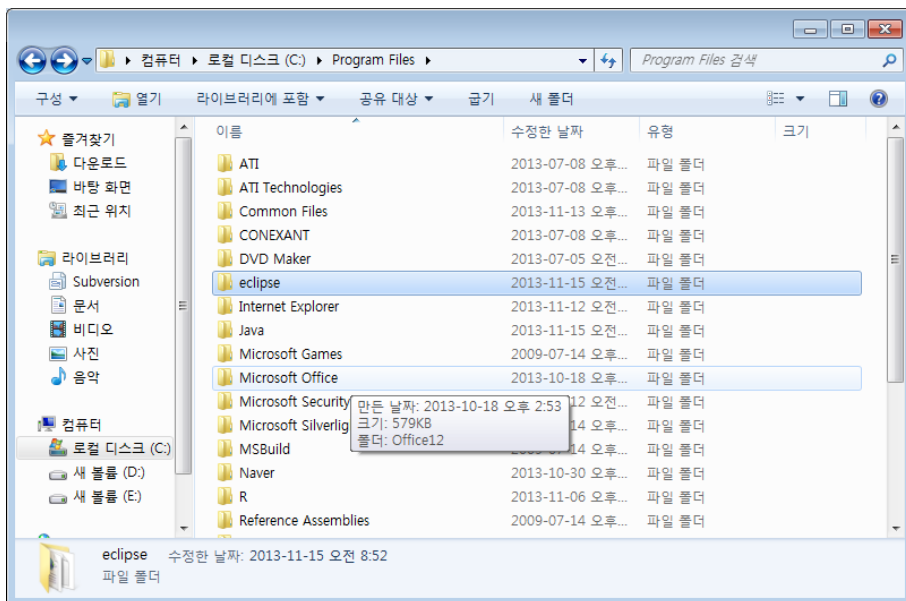


아래와 같이 다운로드 가능한 주소중 빠른 서버를 선택하신 후 다운로드 하도록 합니다.



다운로드가 완료되었다면 적당한 디렉토리에 압축을 해제하도록 합니다. 아래에서는 C:\Program Files 디렉토리에 압축을 해제한 예입니다.

압축해제가 완료되었다면 압축이 해제된 디렉토리에 있는 eclipse.exe 의 바로가기를 바탕화면에 생성하신 후, 사용하시면 보다 편하게 사용이 가능합니다. 이클립스가 설치되었다면 반드시 실행하여 구동여부를 확인하도록 합니다.



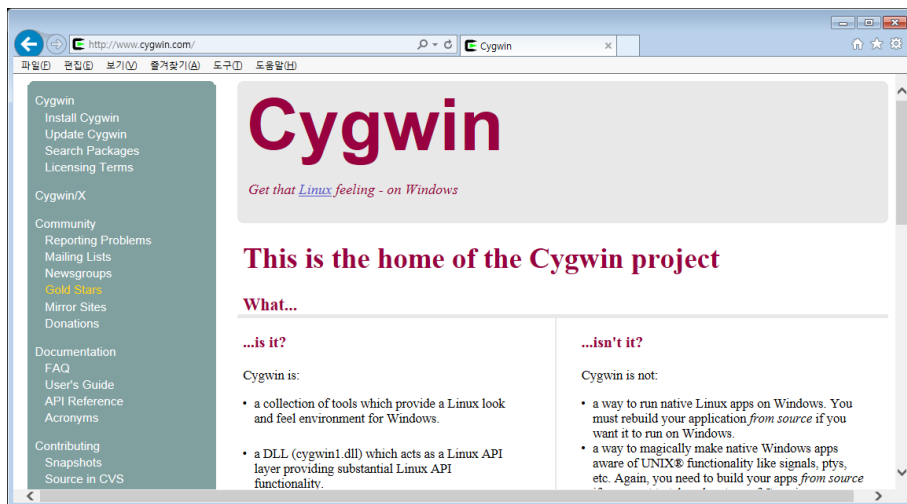
### 3. Cygwin 설치

Cygwin은 리눅스 환경을 제공하는 에뮬레이터 입니다. 보다 정확히 이야기 하자면 gcc 프로그램 개발환경을 제공하기 위한 에뮬레이터라고 볼 수 있습니다. 기본적으로 gcc는 리눅스 환경 또는 POSIX환경에서 구동되고 사용되는 컴파일러입니다. 하지만 Cygwin을 이용하면 실제 리눅스를 설치하지 않더라도 리눅스 환경을 에뮬레이팅하여 제공하므로 gcc 개발이 가능하게 됩니다.

우선 아래의 사이트에 접속하여 Cygwin 인스톨러를 다운로드 합니다.

<http://www.cygwin.com>

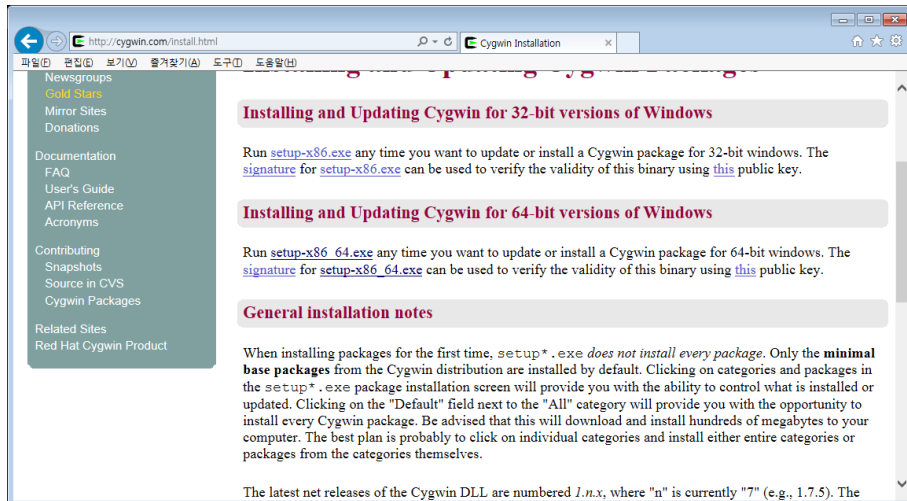
접속하셨다면 좌측 상단의 Install Cygwin 링크를 클릭합니다.



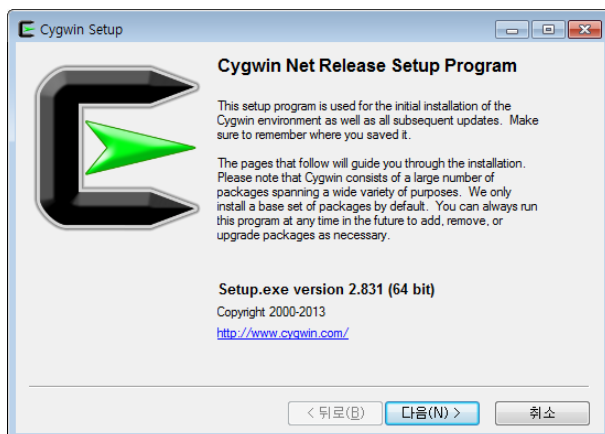
페이지를 이동하셨다면 자신의 시스템에 맞는 인스톨러를 다운로드 하도록 합니다.

x86 : 32bit 시스템

x64 : 64bit 시스템



인스톨러를 다운로드 하셨다면 파일을 실행하도록 합니다. 파일을 실행하면 아래의 화면과 같이 Cygwin의 릴리즈 정보가 나옵니다. 다음 버튼을 클릭하여 설치를 진행합니다.

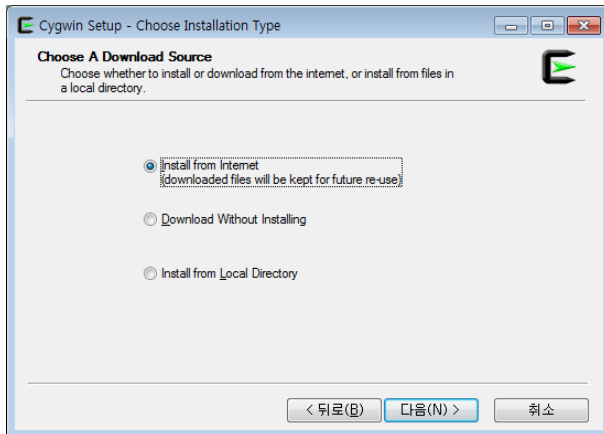


Cygwin 은 위에서도 설명하였듯이 gcc 개발을 위한 리눅스 환경을 제공하는 에뮬레이터 입니다. 그러므로 하나의 단일 프로그램이 기본 프로그램 위에 여러 프로그램을 설치하여 사용하는 Add-On 형식의 프로그램입니다. 즉 Cygwin 자체가

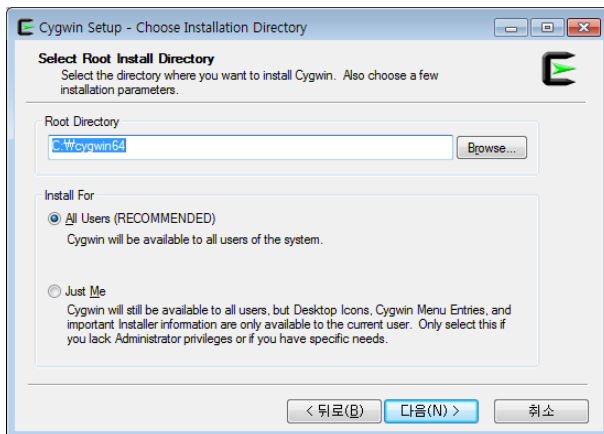


GCC개발을 가능하도록 해주는것이 아니라 단순히 GCC개발이 가능한 환경만을 제공한다라는 의미입니다. 그러므로 GCC개발을 위해서는 Cygwin만이 아니라 GCC개발을 위한 컴파일러나 라이브러리 그리고 디버거등의 프로그램이 추가로 설치되어야 합니다.

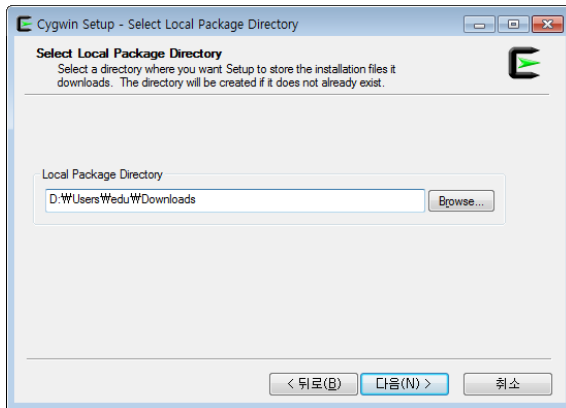
위에서 설명하였던 설치할 Add-On 프로그램을 다운로드 하는 방식을 선택합니다.



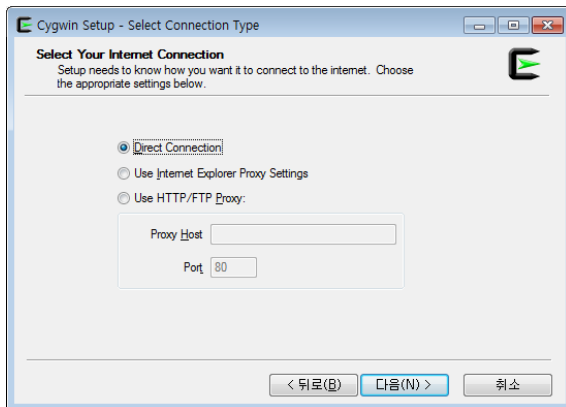
Cygwin을 설치할 디렉토리를 지정합니다.



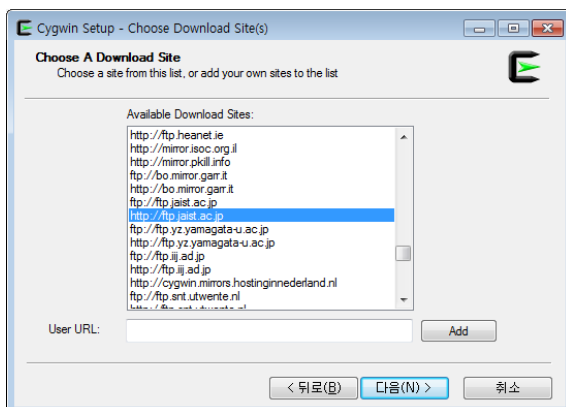
다운로드 한 Add-on 프로그램 패키지 파일을 저장할 디렉토리를 지정합니다. 한번 다운로드한 패키지 파일은 이후 다른 시스템에 복사하여 설치 시 다운로드 하는 번거로움을 덜을 수 있습니다. 이후 필요하시면 별도로 백업해 두셨다가 사용하시면 됩니다.



패키지 서버와의 연결방식을 선택합니다. 프록시 서버를 이용하시면 다소 다운로드 속도가 느려질 수 있으므로 직접연결을 선택하도록 합니다.



패키지를 다운로드 할 서버를 선택합니다. 당연히 빠른 서버가 좋겠지요...



설치할 패키지를 선택하도록 합니다. 아래의 패키지 만큼은 반드시 선택하도록 합니다.

Devel 하위의

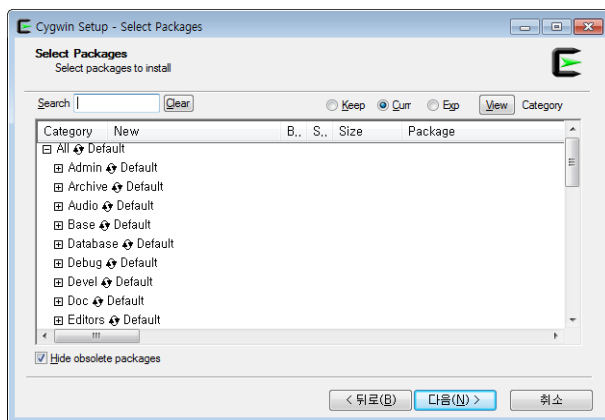
binutils

gcc-core

gcc-g++

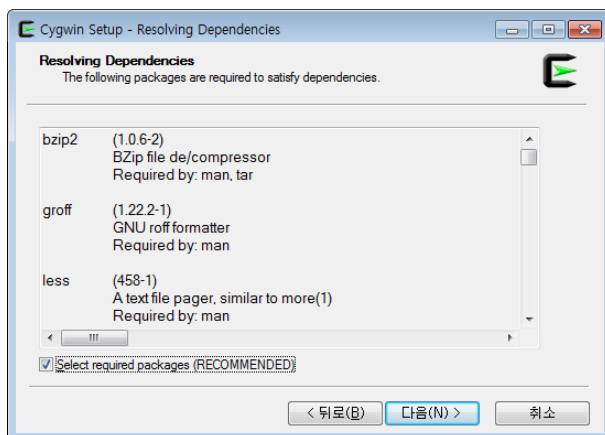
make

gdb

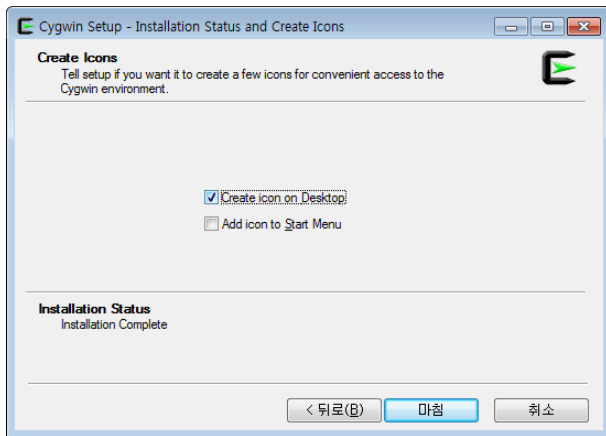


위에서 선택한 패키지를 사용하기 위해 필요한 패키지 입니다. 이러한 패키지를 종속된 패키지라고 합니다.

당연히 필요한 패키지이므로 설치가 되어야만 합니다. 다음 버튼을 클릭하여 계속 진행합니다.



설치가 완료되고난 후, 바탕화면에 바로가기 생성해 두도록 합니다.

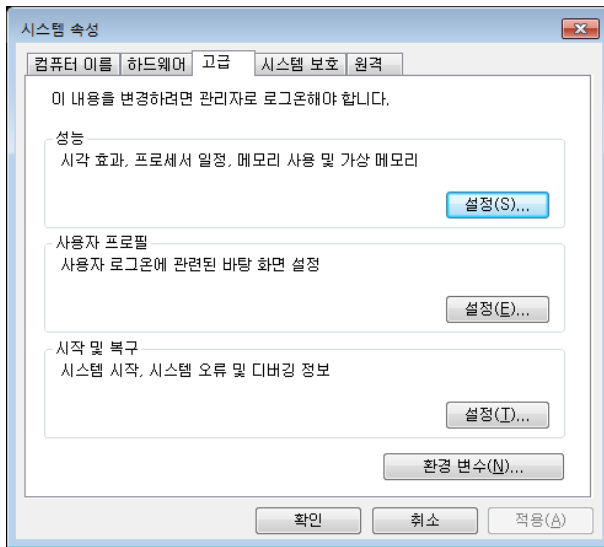


이클립스에서 작성한 프로젝트를 구동하기 위해서는 Cygwin 이 어디에 설치되어있는가를 명시하여야 합니다. 아래의 순서에 의해서 Cygwin 관련환경변수를 등록하도록 합니다.

우선 제어판에서 시스템을 선택하여 실행합니다. 그리고 좌측에서 고급 시스템 설정을 클릭하여 시스템 속성창을 띄우도록 합니다.

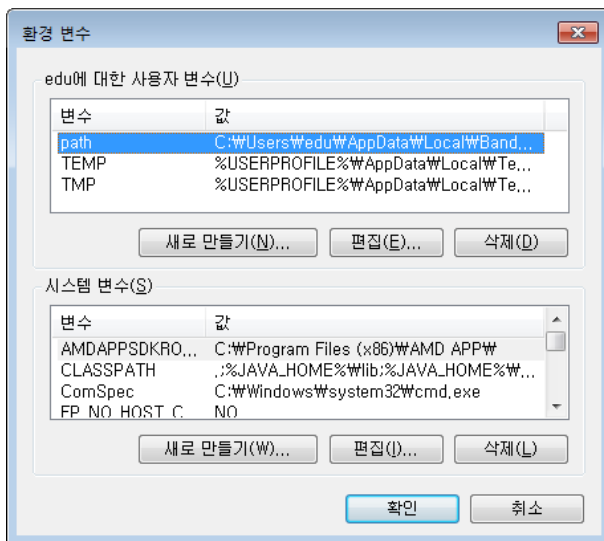


시스템 속성창을 띄우셨다면 고급탭을 선택하신 후, 환경변수 버튼을 클릭합니다.



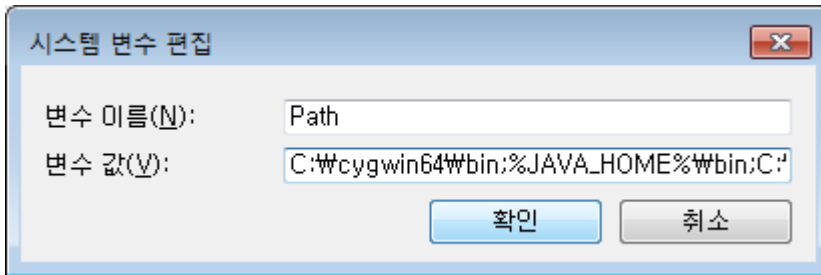
환경변수는 사용자환경변수와 시스템환경변수 두 종류가 있습니다. 사용자 환경변수는 현재 유저에게만 적용되는 환경변수이며 시스템 환경변수는 모든 유저에게 적용되는 환경변수입니다. 물론 시스템 환경변수의 의미가 "모든 유저에게 적용되는" 의미는 아닙니다. 자세한 내용은 생략하도록 하겠습니다.

아래쪽의 시스템 환경변수에서 path 변수를 선택하신 후, 편집 버튼을 클릭합니다.

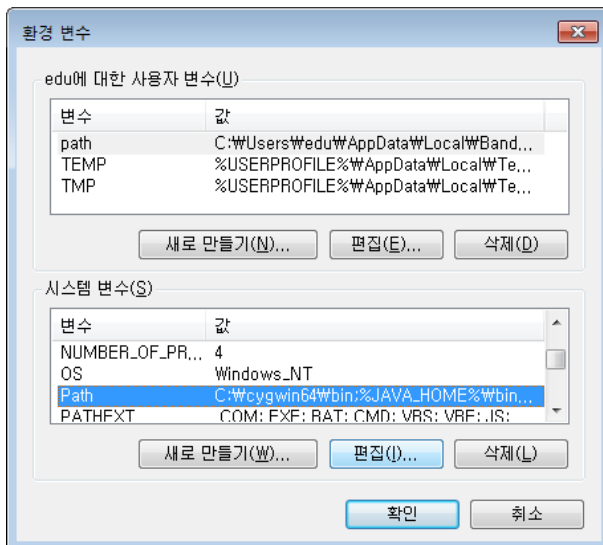


아래와 같이 Cygwin 이설치된 디렉토리 하위에 있는 bin 디렉토리를 추가합니다.  
 필자는 환경변수의 앞부분에 추가하였습니다. 환경변수의 추가는 세미콜론(;)으로  
 구분되어야 하므로 반드시 추가하신 후에 추가된 경로의 앞과 뒤에 세미콜론으로  
 구분되어있는가를 확인하도록 합니다.

또한 Cygwin이 설치된 디렉토리를 반드시 확인하여 등록하도록 합니다.

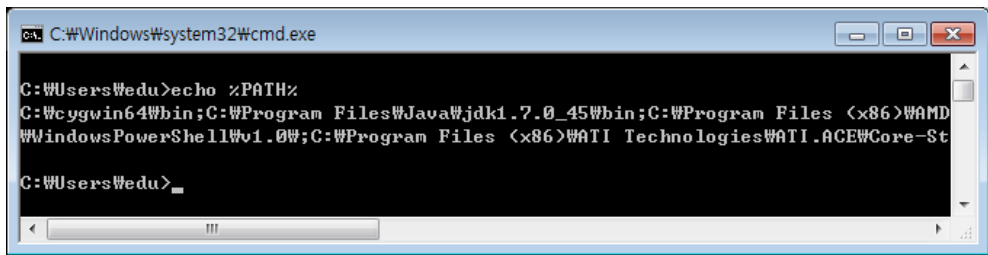


path 환경변수의 편집이 완료되었다면 확인 버튼을 클릭하여 환경변수창을 닫습니다.



환경변수의 등록여부는 cmd 명령을 실행하여 프롬프트 창을 띄운 후, 아래의 명령을  
 이용하여 확인이 가능합니다.

Dos/Windows 시스템에서의 환경변수(리눅스나 유닉스의 쉘 환경변수)의  
 사용은 %환경변수명%(리눅스나 유닉스에서는 \$환경변수명)으로 참조가 가능합니다.

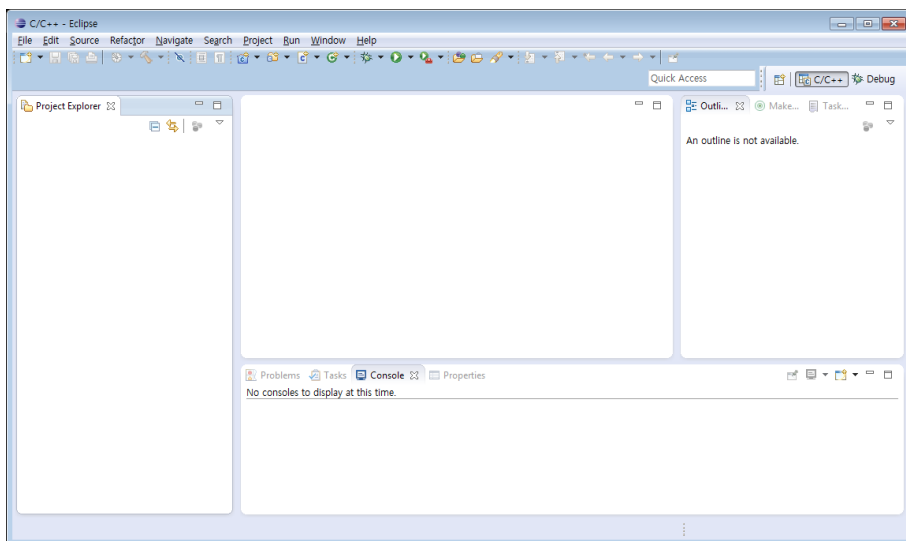


```
C:\Windows\system32\cmd.exe

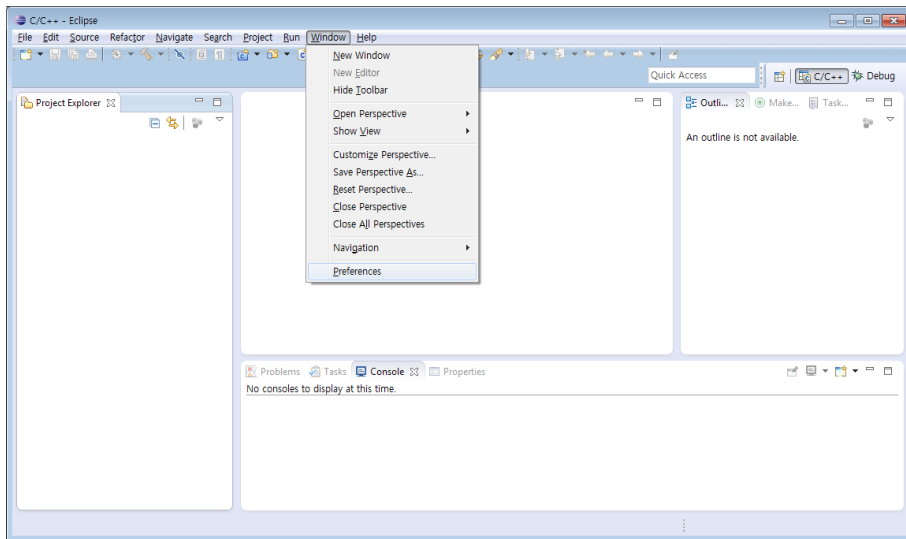
C:\Users\Wedu>echo %PATH%
C:\cygwin64\bin;C:\Program Files\Java\jdk1.7.0_45\bin;C:\Program Files (x86)\AMD\WindowsPowerShell\v1.0\;C:\Program Files (x86)\ATI Technologies\ATI.ACE\Core-St
C:\Users\Wedu>
```

환경변수의 등록이 완료되었다면 이클립스를 구동합니다. 이클립스는 통합개발환경을 제공하는 툴로서 화면을 크게 다음과 같이 구분됩니다.

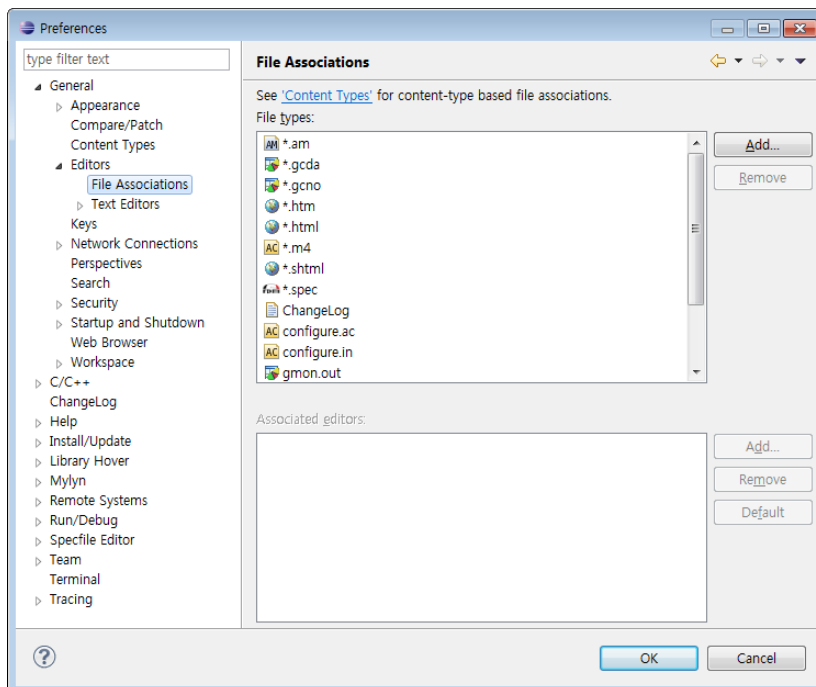
- 프로젝트 익스플로러 : 좌측
- 도큐먼트 영역 : 중앙
- 콘솔 : 하단



Windows 메뉴에서 Preferences 메뉴를 선택합니다.

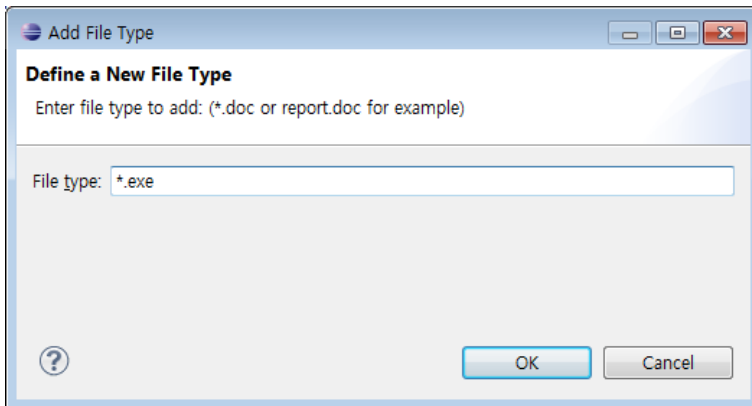


좌측의 카테고리에서 General > Editor > File Associations 을 선택하신 후, 좌측의 File Associations 의 Add 버튼을 클릭합니다.

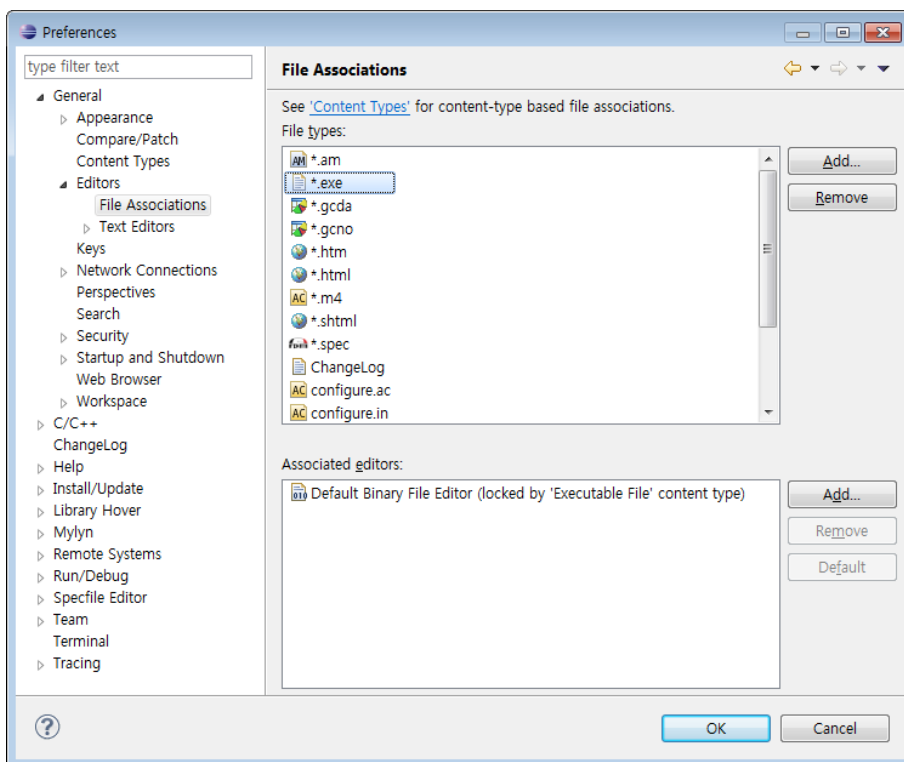




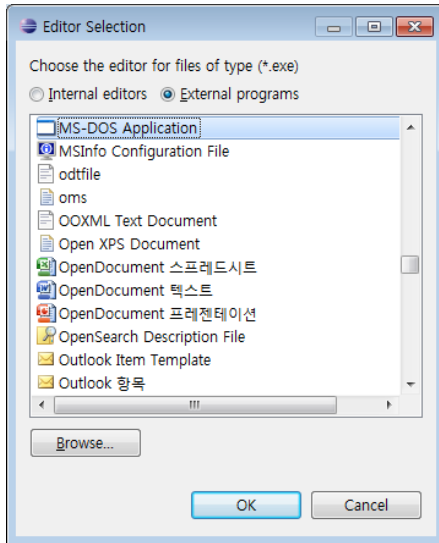
Add File Type 의 File Type 란에 아래와 같이 exe 확장자를 등록한 후 OK 버튼을 클릭합니다.



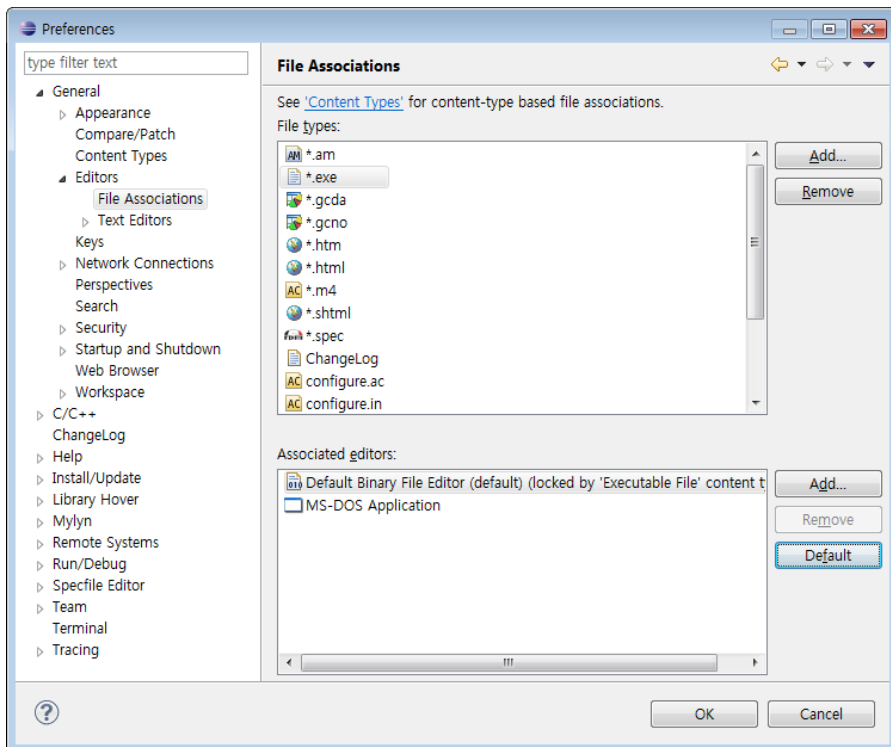
exe 확장자의 등록을 확인하신 후, Associated editors 의 좌측에 있는 Add 버튼을 클릭하여 해당 확장자에 대한 어플리케이션을 지정하도록 합니다.



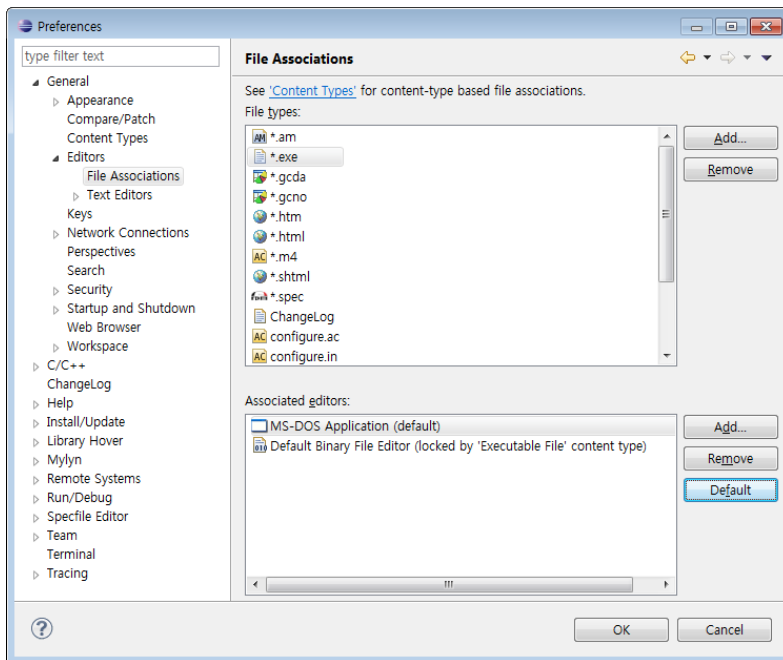
External programs 를 선택하신 후, 아래의 프로그램 목록에서 MS-DOS 어플리케이션을 선택하신 후, OK를 클릭하세요.



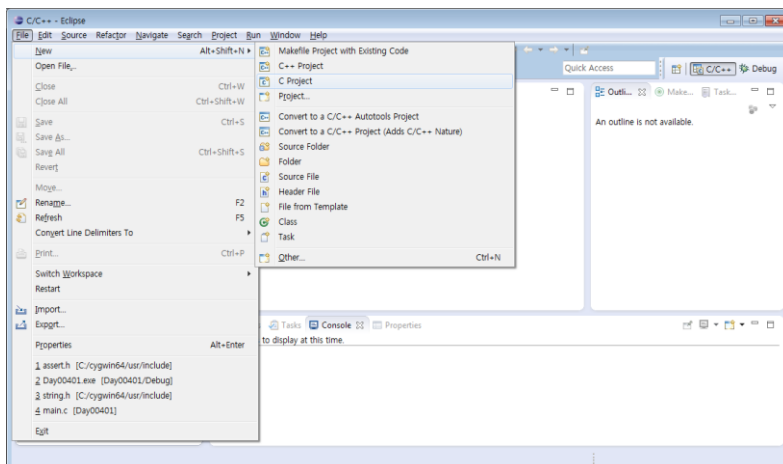
exe 확장자에 대한 어플리케이션으로 MS-DOS Application이 등록되었다면 MS-DOS Application을 선택하시고 Default 버튼을 클릭합니다.



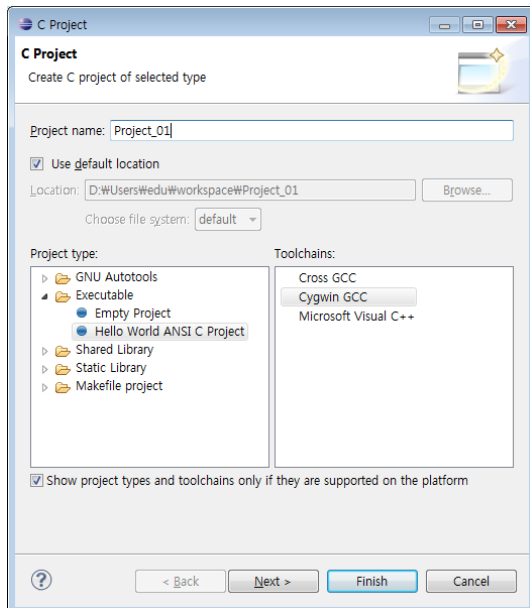
추가설명 : 이클립스는 통합개발환경 툴로서 프로그램의 작성 뿐만이 아니라 컴파일 및 디버깅 그리고 실행환경을 제공합니다. 작성된 프로그램 소스를 컴파일 하고 실행하게되면 실행결과를 아래쪽의 콘솔을 통해 출력을 하게 됩니다. 이때 프로그램 소스내에 scanf()함수와 같은 입력 함수가 존재하면 자동 플러쉬(flush)가 되지 않는 현상(일종의 버그?)가 존재합니다. 그러므로 소스를 컴파일 하고 실행할때 결과를 콘솔이 아닌 MS-DOS 프롬프트창을 띄우고 그 창을 통해 확인하기 위한 설정으로 exe확장자를 등록하고 어플리케이션을 등록한 것입니다.



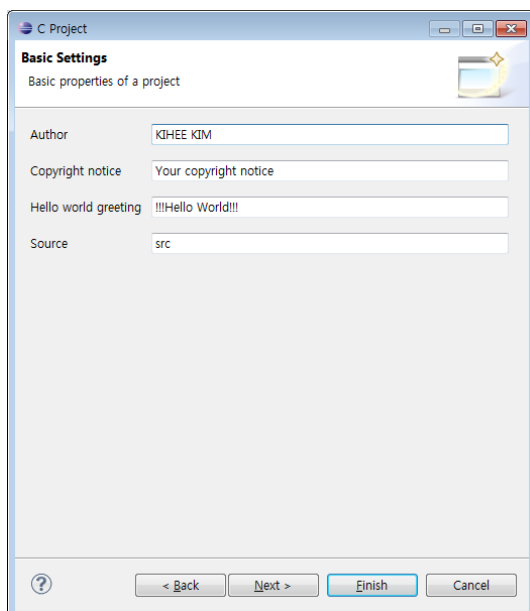
이제는 프로젝트를 생성해보도록 하겠습니다. File > New > C Project를 선택합니다.



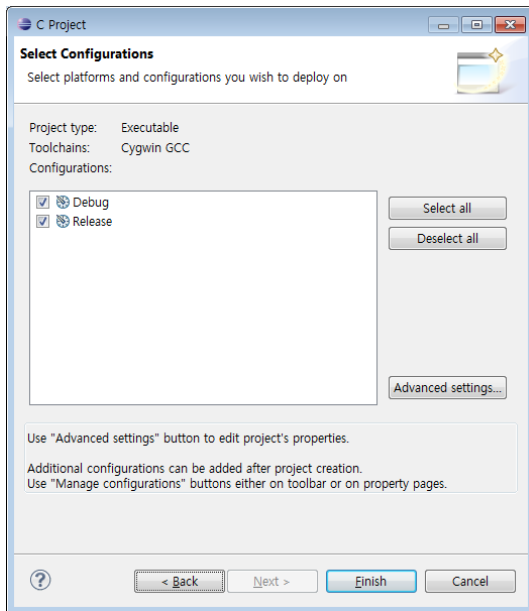
프로젝트 이름을 등록하고 프로젝트 타입과 해당 프로젝트에서 사용할 컴파일러 종류를 선택하신 후 Next 버튼을 클릭합니다.



프로젝트에 대한 기본정보를 입력한 후 Next 버튼을 클릭합니다.



아래의 환경설정창에서 Advanced settings... 버튼을 클릭합니다.

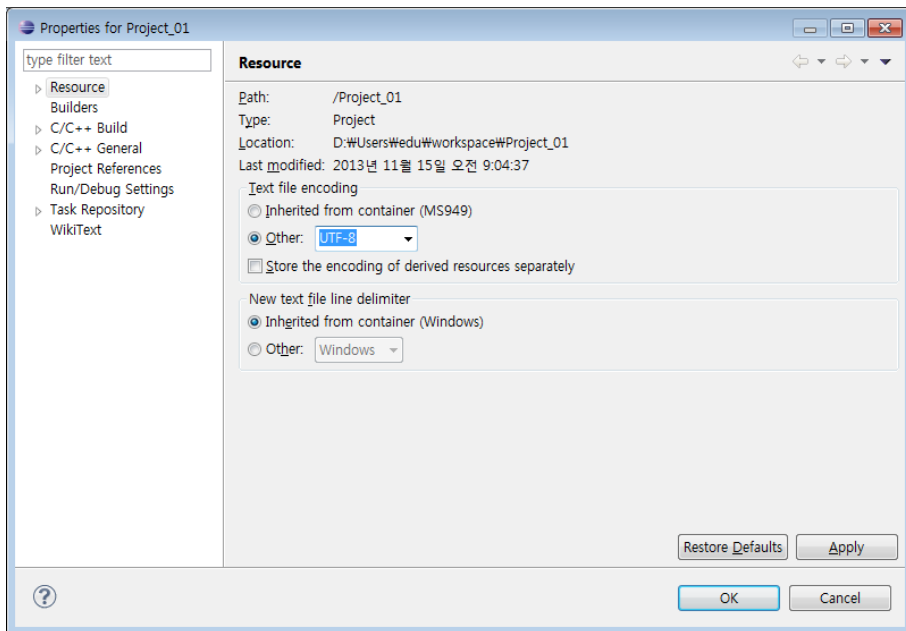


좌측의 Resource에서 Text file encoding 을 UTF-8 로 변경하도록 합니다. 만약 이부분을 설정하지 않는다면 이후 프로그램 파일 실행 시, 한글이 제대로 출력되지 않습니다. 이는 이클립스는 Windows 환경에서 구동되고 있으며 Cygwin은 POSIX환경이므로 두 시스템간의 문자셋 차이로 인해 발생하는 문제입니다.

Windows : cp949

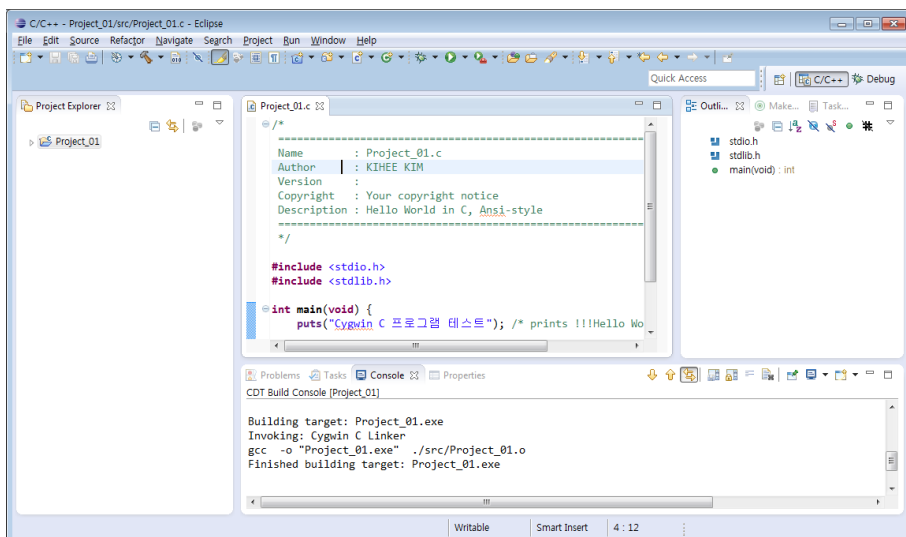
POSIX : utf-8

그러므로 이부분을 반드시 utf-8로 변경해주시기 바랍니다. 물론 이부분을 잊어버리고 그냥 두었다라도 추후 변경이 가능합니다만 이부분의 변경이 발생하는 경우, 기존파일의 한글이 모두 깨지는 현상이 발생하는 수도 있으므로 프로젝트 생성시 지정해주도록 하시기를 바랍니다.



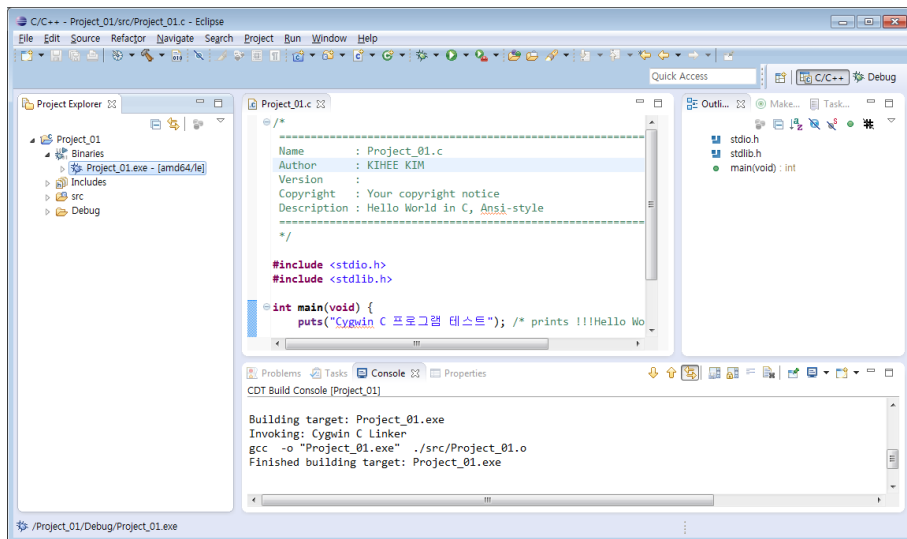
프로젝트 생성시 Hello GCC Project를 선택하셨다면 아래와 같이 기본코드가 생성이 되어있습니다. 기본생성 코드를 지우고 테스트를 위한 코드를 작성하도록 합니다. 이때 main()함수의 return 구문 이전에 반드시 getchar()함수 호출을 두어 종료전에 멈춰있도록 해줍니다.

작성된 프로그램 소스는 상단 메뉴바의 Project > Build All 또는 단축키 Ctrl + B 키를 이용하여 컴파일 합니다. 또한 이렇게 컴파일(빌드)된 결과는 하단의 콘솔창을 통해 확인이 가능합니다.



컴파일이 정상적으로 종료되었다면 좌측 프로젝트 익스플로러에서 트리를 오픈한 후, Binaries 또는 Debug 하위에 있는 실행파일을 더블클릭함으로써 실행이 가능합니다.

Binaries 하위에 있는 실행파일은 릴리스용 Debug 하위에 있는 실행파일은 디버깅용이라고 보시면 됩니다.



아래는 위의 프로젝트를 컴파일(빌드)한 후, 실행한 결과입니다.

