

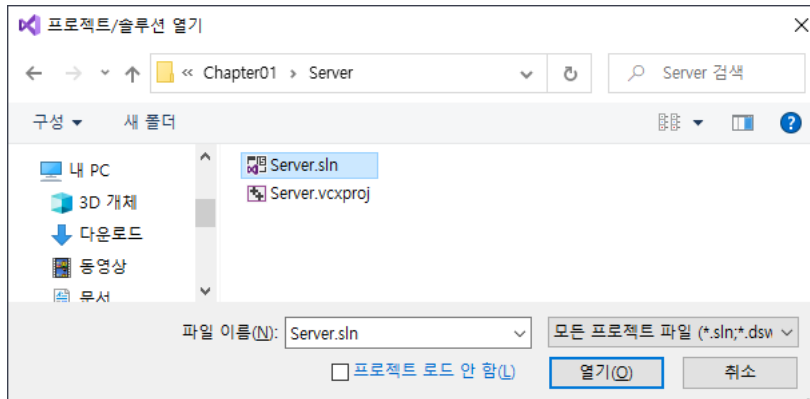
<TCP/IP 소켓 프로그래밍(2판)> 본문 소스 코드 사용법과 주의 사항

제공하는 소스 코드(<https://github.com/promche/TCP-IP-Socket-Prog-Book-2nd>)의 사용법과 주의 사항을 C/C++[윈도우], C/C++[리눅스], C#[윈도우]로 나누어서 소개한다. 윈도우에서는 압축된 소스 코드를 원하는 곳에 풀어두었다고 가정하며, 리눅스에서는 압축된 소스 코드를 다운로드만 해두었다고 가정한다.

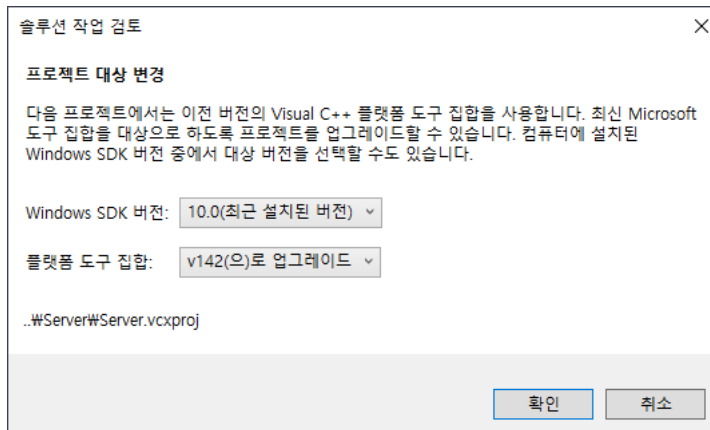
- [C/C++\[윈도우\]](#)
- [C/C++\[리눅스\]](#)
- [C#\[윈도우\]](#)

C/C++[윈도우]

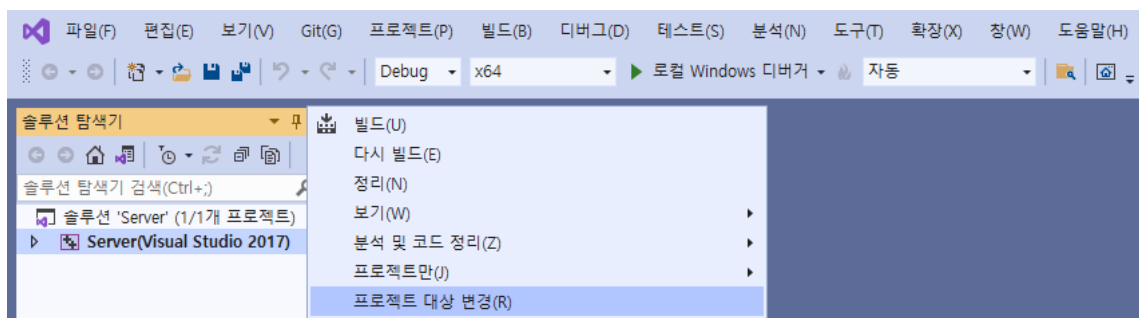
1. 비주얼 스튜디오의 [파일]-[열기]-[프로젝트/솔루션] 메뉴를 선택하고 .sln 파일을 연다.



2. 비주얼 스튜디오 버전에 따라 다음과 같은 대화상자가 열릴 수 있다. 특별한 이유가 없다면 [확인]을 눌러 프로젝트 대상을 변경(업그레이드)하기를 권장한다.



만약 [취소]를 눌러 진행해도 문제가 없을 수 있지만, 나중에라도 문제가 생겨서(빌드 오류) 같은 작업(프로젝트 대상 변경)을 다시 하고 싶다면 솔루션 탐색기에서 마우스 오른쪽 버튼을 눌러 [프로젝트 대상 변경]을 선택한다.



3. 이후 빌드 및 실행 절차는 본문 1장 4절에서 소개한 것과 같다.

- 빌드: [빌드]-[솔루션 빌드] 메뉴
- 실행: [디버그]-[디버그하지 않고 시작] 메뉴

C/C++[리눅스]

다음은 본문 1장 4절의 내용과 다소 중복되지만, 1장 4절을 건너뛰고 곧바로 부록을 보는 독자를 위해 개발 도구 설치부터 소개하기로 한다. 리눅스는 특정 배포판(예: Ubuntu, Fedora, CentOS, Red Hat Enterprise Linux 등)을 선택해서 설치했다고 가정한다.

1. 본문 예제를 빌드하려면 g++ 컴파일러와 make 명령이 설치되어 있어야 한다. 다음 명령을 실행하면 g++와 make 설치 여부와 현재 설치된 버전을 확인할 수 있다.

```
student@vm:~$ g++ --version
g++ (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 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.

student@vm:~$ make --version
GNU Make 4.1
Built for x86_64-pc-linux-gnu
Copyright (C) 1988-2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

만약 설치되어 있지 않다면 리눅스 배포판의 패키지 관리 명령을 실행해서 설치하면 된다. 예를 들어 Ubuntu 리눅스라면 다음과 같이 **apt**(또는 **apt-get**) 명령을 사용하면 된다. 필자의 PC에는 이미 설치되어 있어서 현재 버전이 표시되지만 그렇지 않다면 패키지 다운로드에 이어 설치가 진행될 것이다.

```
student@vm:~$ sudo apt install g++ make
[sudo] password for student: 암호 입력(보이지 않음)
Reading package lists... Done
Building dependency tree
Reading state information... Done
make is already the newest version (4.1-9.1ubuntu1).
g++ is already the newest version (4:7.4.0-1ubuntu2.3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

레드햇 리눅스 계열(Fedora, CentOS, Red Hat Enterprise Linux 등) 사용자라면 **yum** 명령을 사용하면 된다. 이미 설치되어 있다면 현재 버전이 표시되지만 그렇지 않다면 패키지 다운로드에 이어 설치가 진행될 것이다. 다음은 집필 시점의 최신 Fedora 리눅스를 기본 옵션으로 설치한 상태에서 g++를 설치하는 화면이다. 패키지 이름이 'gcc-c++'임에 유의한다. 패키지 이름을 Ubuntu 리눅스처럼 'g++'로 해도 설치되지만, 구버전의 Fedora 리눅스에서는 오류가 생길 수 있다. make는 이미 설치되어 있어서 따로 설치하지 않았다.

```

[swkim@fedora ~]$ sudo yum install gcc-c++
[sudo] password for swkim:
Last metadata expiration check: 0:41:37 ago on Thu 02 Jun 2022 01:38:14 AM KST.
Dependencies resolved.
=====
Package                        Architecture      Version           Repository        Size
=====
Installing:
gcc-c++                       x86_64            12.1.1-1.fc36    updates          13 M
Installing dependencies:
libstdc++-devel               x86_64            12.1.1-1.fc36    updates          2.3 M
=====
Transaction Summary
=====
Install 2 Packages

Total download size: 15 M
Installed size: 47 M
Is this ok [y/N]: y
Downloading Packages:
(1/2): libstdc++-devel-12.1.1-1.fc36.x86_64.rpm    181 kB/s | 2.3 MB  00:13
(2/2): gcc-c++-12.1.1-1.fc36.x86_64.rpm           325 kB/s | 13 MB   00:39
-----
Total                                              375 kB/s | 15 MB   00:40

```

- 출판사 웹사이트(GitHub 링크)에서 다운로드한 소스 코드의 압축을 푼다. 다음 **unzip** 명령은 ~/Downloads/ 디렉터리에 TCP-IP-Socket-Prog-Book-2nd-Source.zip 파일이 있는 상태에서 현재 사용자의 홈 디렉터리에서 실행한다. 본문 실습은 ~/Source 디렉터리에 소스 코드가 있는 상태에서 진행하므로 **mv** 명령으로 디렉터리 이름을 변경해두기를 권장한다(필수는 아님).

```

student@vm:~$ unzip ~/Downloads/TCP-IP-Socket-Prog-Book-2nd-Source.zip
Archive: /home/student/Downloads/TCP-IP-Socket-Prog-Book-2nd-Source.zip
12c39abe2f42e114f9b1c7a276438b00557afd29
  creating: TCP-IP-Socket-Prog-Book-2nd-Source/
  creating: TCP-IP-Socket-Prog-Book-2nd-Source/C#/
... 중간 출력 부분은 생략 ...
  inflating: TCP-IP-Socket-Prog-Book-2nd-Source/Windows/Common.h
student@vm:~$ mv TCP-IP-Socket-Prog-Book-2nd-Source/ Source

```

- 실습하기를 원하는 장의 디렉터리로 이동하고 **make** 명령으로 빌드하면 실행 파일이 만들어진다. 이후 실행 절차는 본문 1장 4절에서 소개한 것과 같다.

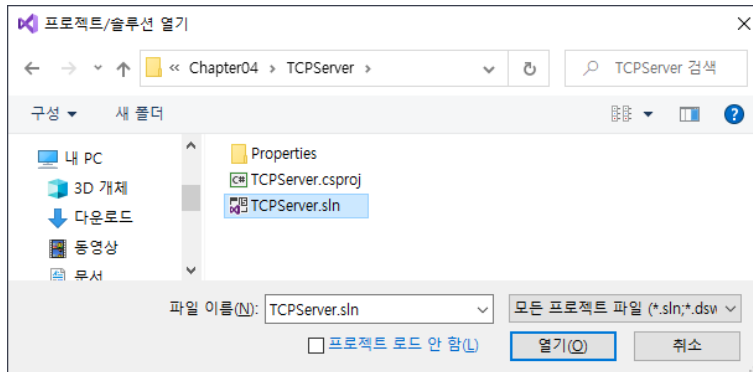
```

student@vm:~$ cd Source/
student@vm:~/Source$ cd Linux/
student@vm:~/Source/Linux$ cd Chapter01/
student@vm:~/Source/Linux/Chapter01$ make
g++ -Wall Server.cpp -lpthread -o Server
student@vm:~/Source/Linux/Chapter01$ ./Server

```

C#[윈도우]

1. C# 코드는 C/C++ 코드와 마찬가지로 비주얼 스튜디오의 [파일]-[열기]-[프로젝트/솔루션] 메뉴를 선택하고 .sln 파일을 열면 된다.



2. 이후 빌드 및 실행 절차는 C/C++ 프로젝트와 같으므로 본문 1장 4절을 참고한다.
 - 빌드: [빌드]-[솔루션 빌드] 메뉴
 - 실행: [디버그]-[디버그하지 않고 시작] 메뉴