Installing a C/C++ Compiler

We encourage you to install a C/C++ compiler on your personal computer so that you can compile your code without an internet connection. This guide will help you install the GNU Compiler Collection (GCC) or Clang ("klăng"), depending on your system. However, you do not have to install anything before the first week of class, but feel free to give it a try. Do not hesitate to ask an instructor or teaching assistant if you need help or if you have questions about the instructions below. You may also ask your questions and share your tips and tricks here on the Piazza course webpage.

Remark: In the following instructions, a shell command such as `gcc --version` is sometimes shown as

```
$ gcc --version
```

where the dollar sign is commonly used to indicate the command line prompt. In other words, it is not part of the command, so whenever you see a line that starts with a dollar sign, leave it out when you type the command. Similarly, on Windows, the Command Prompt typically looks like this:

```
C:\>gcc --version
```

The character `>` (and everything that precedes it) indicates the command line prompt, and it is not part of the command. Thus, whenever you see a line that starts with a path followed by `>`, leave out this part when you type the command.

Linux

Linux systems generally come with a package manager that can be used to install a compiler and other useful development tools. We will use the GNU Compiler Collection which is very common on Linux systems.

Debian Derivatives

Debian Linux derivatives such as *Linux Mint* and *Ubuntu* come with a tool called APT (*Advanced Package Tool*) that can be used to install the GNU Compiler Collection (GCC). Open a terminal window and execute the following command (requires admin rights):

```
$ sudo apt-get update
$ sudo apt-get install build-essential
```

This installs GCC along with a number of dependencies and development tools.

Red Hat Derivatives

Red Hat Linux derivatives such as *CentOS*, *Scientific Linux*, and *Amazon Linux* come with a tool called YUM (*Yellowdog Updater, Modified*) that can be used to install the GNU Compiler Collection (GCC). Open a terminal window and execute the following command (requires admin rights):

```
$ sudo yum group install "Development Tools"
```

This installs GCC along with a number of dependencies and development tools.

openSUSE

OpenSUSE comes with a package manager called Zypper. Open a terminal window and execute the following command (requires admin rights):

```
$ sudo zypper install gcc make
```

This installs GCC and a tool called Make.

Verify the Installation

Verify that GCC and a tool called Make are installed:

```
$ gcc --version
$ make --version
```

macOS

Clang

The C compiler provided by Apple is called *Clang*, and it can be installed either with or without the Xcode IDE. To install Clang without the Xcode IDE, install the so-called *Command Line Tools* by executing the following command in the Terminal app:

```
$ xcode-select --install
```

Click *Install* to download and install Xcode Command Line Tools. This installs the Clang compiler along with a number of command line tools. Verify that Clang and a tool called Make are installed:

```
$ clang --version
$ make --version
```

If you would like to install the Xcode IDE as well, launch the App Store and install Xcode. Note that we will only provide very limited support for specific IDEs such as Xcode.

GNU Compiler Collection

If you prefer to use GCC instead of Clang, you can install it using the Homebrew package manager as follows:

```
$ brew install gcc make
```

Verify that GCC and a tool called Make are installed:

```
$ gcc-8 --version
$ gmake --version
```

Notice that you will need to use the command `gcc-8` instead of simply `gcc`. This is because `gcc` is a symbolic link that actually refers to `clang`. Similarly, to avoid conflicts with Apple's version of Make the Homebrew version is named `gmake` by default.

Windows

MSYS2 and the GNU Compiler Collection (recommended)

MSYS2 is a build environment for Windows that provides a Bash shell and several useful tools, including a package manager `pacman` that can be used to install the GNU Compiler Collection. Follow these steps to set up MSYS2 and the GNU Compiler Collection:

1. Go to msys2.org and download the MSYS2 installer (follow these steps if you do not know if you are running a 32-bit or 64-bit version of Windows). Complete the steps described in the installation instructions at msys2.org.

   Alternatively, if you use *Chocolatey* as a package manager, you can install MSYS2 using the Command Prompt:

   ```
   C:\>choco install msys2
   ```

2. Click on the Start menu and launch **MSYS2**. We will now install the GNU Compiler Collection (GCC) and a tool called Make. If your OS is 64-bit, you can install the 64-bit version as follows:

   ```
   $ pacman -S --needed mingw-w64-x86_64-gcc make
   ```

   Alternatively, if your OS is 32-bit:

   ```
   $ pacman -S --needed mingw-w64-i686-gcc make
   ```

3. You should now be able to use GCC in several ways:

- Click on the Start menu and launch "MSYS2 MinGW-W64 64-bit" (or "MSYS2 MinGW-W64 32-bit" if you installed the 32-bit version). This opens up a new MSYS2 Bash shell. Check that the following commands return some version information:

  ```
  $ gcc --version
  $ make --version
  ```

- Alternatively, if you prefer to use the Command Prompt instead of the MSYS2 Bash shell:

  ```
  C:\>path c:\msys64\usr\bin;c:\msys64\mingw64\bin;%PATH%
  C:\>gcc --version
  C:\>make --version
  ```

  Note that to avoid having to set the path every time you launch Command Prompt, you can add `c:\msys64\usr\bin` and `c:\msys64\mingw64\bin` to the `Path` environment variable in the "System Properties" control panel.

  Finally, note that if you installed MSYS2 using Chocolatey, the path to MSYS2 is most likely `c:\tools\msys64` instead of `c:\msys64`, so you will need to set the path accordingly.

Remark: MSYS2/Bash emulates a Linux/Unix-style file system where a Windows-style path of the form

```
C:\Users\Username\My Documents\
```

is expressed as

```
/c/Users/Username/My Documents/
```

in MSYS2. Note that the Linux/Unix-style path uses a `/` (slash) instead of a \(backslash) as path separator. To change the directory in the Windows Command Prompt, a path with a space must be enclosed by quotation marks:

```
C:\>cd "C:\Users\Username\My Documents\"
```

In Bash, spaces in a path must be prepended by a backslash:

```
$ cd /c/Users/Username/My\ Documents/
```

Finally, note that you can open Windows applications directly from an MSYS2 terminal. For example, if the Atom editor is located in `C:\Program Files\atom.exe`, you can start it from the terminal as follows:

```
$ /c/Program\ Files/atom.exe
```

Windows Subsystem for Linux

Recent versions of Windows 10 provide a simple way to install and run a Linux environment directly on Windows. If your version of Windows supports this and you prefer this option, you can install a Linux environment by following this installation guide. If you do not already have a favorite Linux distribution, we recommend installing either Debian GNU/Linux or Ubuntu. Afterwards you can install the GNU Compiler Collection by following the instructions for Linux.

Remark: The Windows Subsystem for Linux provides a Linux/Unix-style file system where a Windows-style path of the form

```
C:\Users\Username\My Documents\
```

is expressed as

```
/mnt/c/Users/Username/My Documents/
```

in the Windows Subsystem for Linux. Note also that you can start Windows applications (such as an editor) directly from the terminal. For example, if the Atom editor is located in `C:\Program Files\atom.exe`, you can start it from the terminal as follows:

```
$ /mnt/c/Program\ Files/atom.exe
```

Microsoft Visual C++

If you are already familiar with Microsoft's *Visual Studio* development tools, you may prefer to use Microsoft's Visual C++ compiler. Note that we can only provide very limited support, so this option is not recommended if you are not already familiar with Visual Studio.

DTU students have access to a number of Microsoft software products through the Microsoft DreamSpark program; go to DTU Software Downloads to download Visual Studio 2017 and obtain a license. Official installation instructions can be found here, and information about how to use the compiler on the command line can be found here.

After installing and setting up MS Visual Studio 2017, launch "x64 Native Tools Command Prompt for VS 2017" via the Start menu. Verify that the C compiler and a tool called Nmake are installed:

```
C:\>cl
C:\>nmake
```