Milestone 01

Lars Pastewka

Table of contents

Setting up the build environment

Learning goals

The student will…

- …be able to compile a C++ project with CMake.
- …be able to specify build types of a CMake project.

Introduction

In this first milestone we will set up a build environment and make sure that you can compile C++ code on your computer. This will be the starting point for the developments in the following milestones.

Setting up your system

The starting point of your project is setting up a proper build environment. This means, you need to install and test all tools necessary for the project. In particular, you will need:

- A C++ compiler
- CMake (at least version 3.21) for our build environment

We provide installation instructions for Ubuntu installations. If you have a Windows machine, we recommend to use the Windows Subsystem for Linux (WSL). Documentation on how to install WSL on Windows can be found here. The following instructions also apply for Ubuntu installed within WSL.

Once you have these things set up, open a command shell and type

```
sudo apt-get update
sudo apt-get install gcc g++ make cmake gdb valgrind openmpi-bin openmpi-common libopenmpi-d
```

The OpenMPI packages are not needed until milestone 6, where we parallelize the code for distributed-memory machines, but it is convenient to install them right away. Note that if you are on a different system than Ubuntu, these commands may differ. On Ubuntu/Debian it is `apt` or `apt-get` but on Fedora/CentOS/RHEL the package manager command is `dnf`. On macOS it depends on which package manager you have installed, the most popular one is Homebrew which provides the `brew` command for package installation. The names of the packages will also vary between these systems.

We recommend using a development environment for developing code. We ourselves use CLion or Visual Studio Code. Free educational licenses for CLion can be obtained here. Visual Studio Code is free to use without an explicit license. CLion and Visual Studio Code are available for all of the above platforms and can on Windows be configured to use WSL.

- Documentation on CLion and WSL can be found here.
- Documentation on Visual Studio Code and WSL can be found here.

Creating an empty repository

The first thing you need to do is to set up your build environment. We have done this for you and provide a template repository here: https://github.com/imtek-simulation/cmake-skeleton/. We will now be walking you through the process of obtaining this skeleton repository from the command line. You can also carry out the whole process within CLion.

On GitHub, you can simply create a new repository from our template. Navigate to the link above and click on "Use this template". You will be asked for a new name of the repository: Let's call this yalb (Yet Another Lattice Boltzmann code). Since my username is "pastewka", the repository now resides under

```
https://github.com/pastewka/yalb
```

You can now check this repository out, i.e. copy it to your local machine. On the shell, type

```
git clone git@github.com:pastewka/yalb.git
```

(and replace "pastewka" and "yalb" with whatever is appropriate for you). The code now resides in the subdirectory "yalb". Note that if you do not want to work on github, you can also directly check out the template repository.

Our template repository has existing CMake files. Those CMake files are set up to automatically download the libraries

- Kokkos for parallel hardware abstraction and
- Googletest as a testing framework.

We will work with these libraries throughout this class.

> **Note**
>
> If you are unfamiliar with the unix shell, we recommend this tutorial. We also strongly recommend that you use version control (git). If you are unfamiliar with git and want to learn more, see this tutorial.

Compiling the code

At the root of your repository is a README file with the compilation instructions. It also describes in details how the repository is laid out and where you should add new files.

For now, follow the instructions to compile the code, run the first milestone executable and run the tests. Please refer to this README when you wish to add new files and push them to GitHub.

Task summary

This milestone requires the following tasks:

- Create a repository for your code development from our template.
- Carefully read the `README.md` file in your fresh code repository.
- Compile the template, run first milestone executable and tests, for the following configuration options:
  - `Debug` build
  - `Release` build

We provide the following files for you:

- Skeleton repository at https://github.com/imtek-simulation/cmake-skeleton/