CS 251: Lab 05: [Code Warrior] Makefiles, cmake, compilation and linking

- Handed out: 21/Aug Due: 24/Aug Mon 11pm
- Please write (only if true) the honor code. If you used any source (person or thing) explicitly state it. You can find the honor code on the web page.

Overview

In this lab we will learn about "Makefiles, compiling and linking to libraries in Linux.

A large project has many components, possibly written by different group members, or even by people on the Internet. To produce the final executable, it is unnecessary to recompile code that has already been previously compiled into an intermediate stage. This was valuable when compilers were slow, but even today, it is very relevant (especially when we have optimizing compilers). A file that tells the compiler what to compile when things change (i.e., when development is ongoing), and what not to, is called a Makefile.

The process of producing an executable involves intermediate files called "object files and libraries. The final executable involves complicated shared libraries in todays run time environment.

We will also learn about `cmake`, a tool that helps ease the process of writing makefiles. It is used in conjunction with native build environments such as make, Apple's Xcode, and Microsoft's Visual Studio.

The Tasks

1. Makefiles
   (a) We will be working with the `cs251_base` folder that contains the Box2D project.
   (b) Rename the file `Makefile` to `Makefile.orig`. Open the file and study it.
   (c) We will now make our own `makefile`. Name the new file as `makefile_gXX`, where `XX` is the group number. If your group number is a single digit, prepend a zero. What's the difference between `makefile`, and `Makefile`?
   (d) Add a target called `dirs` which makes 2 folders, `mybins` and `myobjs` inside the base folder. If they are already there, then running the target should have no effect.
   (e) Add a target called `b2dsetup` which checks if Box2D is installed in `external/src`, and if it is not installed, automatically installs it from the tar file.
   Note: The makefile should be doing this, you will not manually `cd` and do it.
   (f) Add a target called `setup` which sets up Box2D and creates the two folders mentioned in point 3 above.
   (g) Add a target that compiles the source files into object files and places them in the `myobjs` folder created in step (c).
   (h) Add a target called `exe` which takes the object files in the `myobjs` folder and creates an exe inside the `mybins` folder. The executable must be called `cs251_exe_gXX`, where `XX` is defined similar to point (b).
   (i) Add a target called `exe_opt` which makes an optimized version of the executable using the `-O3` flag for `g++/gcc`. Name the executable `cs251_exe_gXX_opt`. Note that the object files created should also be optimized. Name them `<orig obj name>_opt.o`. Do not create another target for making the optimized object files.
   (j) Add a target called `clean` which deletes the object files and the executables.
   (k) Add a target called `distclean` which deletes the object files, the executables, the folders and the box2D installation.
   (l) Add a target called `makesubmission`, which creates a tar.gz file of only the `src` folder and the makefile. the tar.gz file should be called `lab05_gXX.tar.gz`, where `XX` is defined as in point (b).
   (m) To submit: In each of the steps above, there is a learning objective. Write one line on each of the steps above in `make.txt` as to what your observations were in each step. Why are you doing what you are doing?

2. +cmake

   (a) We will create a new project using cmake.

   (b) We have provided with you a project with following directory structure:

```
square_and_add
       |- src
       |    |- <Some source files>
       |- libsrc
       |    |- include
       |        |- <library header files>
       |    |- src
       |        |- <library source files>
       |- build
       |- libs
            |- <installed libraries>
```

   (c) You have to use cmake to generate a makefile in the build directory with the following targets

      i. no_lib_exec: This target will generate an executable for the given source files in the build folder called sqnadd_no_lib.

      ii. build_static_lib: This target should create a static library called static_library.a in the folder build from the library sources.

      iii. static_lib_exec: This target creates an executable called sqnadd_stat_lib in the build folder by using the static library created in the previous question.

      iv. build_dynamic_lib: This target should create a dynamic library called dynamic_library.so in the folder build from the library sources.

      v. dynamic_lib_exec: This target creates an executable called sqnadd_dyn_lib in the build folder which uses the dynamic library created by the previous target.

      vi. install: This target will locally install the two libraries in the folder libs

   (d) Compare the sizes of the three executables you built and write down your observations in the cmakeSubmit.txt file.

What to Submit

Submit the makefile you wrote for task 1, the cmake configuration file you wrote for task 2 and the relevant text files in a folder. Don't forget your honour code. The folder and its compressed version should both be named lab05_groupXY_final. Hence, you submit a tar.gz named lab05_group07_final.tar.gz if your group number is 7.

How we will score you

1. Task 1: 50 marks
2. Task 2: 20 marks