VG101 — Intoduction to Computers & Programming

Lab 4

Instructor: Manuel Charlemagne

TA: Yihao Liu – UM-JI (Summer 2018)

Goals of the lab

- Design and build a project
- Build a static/shared library
- Documentation of C/C++

1 Introduction

Krystor and Frank are busy with many many exams recently so that they have no time to introduce the background of their situation this time.

2 Working Flow

2.1 A simple CLI program

Krystor is going to build a project according to the ctemplate. It execute some specific functions according to the CLI options, so first he wanted to understand it.

Simon explained

Command-line interface

A command-line interface or command language interpreter (CLI), also known as command-line user interface, console user interface and character user interface (CUI), is a means of interacting with a computer program where the user (or client) issues commands to the program in the form of successive lines of text (command lines). A program which handles the interface is called a command language interpreter or shell (computing).

Here is an example of a c program with command line options.

```c
int main(int argc, char *argv[]) {
    return 0;
}
```

Krystor's goal is to print each of the arguments passed to the c program line by line to the standard output (`stdout`). You can find more details about the problem on JOJ.

Note: Instead of iterating the arguments directly, we usually use `getopt` to parse the arguments, reference: `http://www.gnu.org/software/libc/manual/html_node/Getopt.html`. You can try it later.

2.2 CMake Project (for CLion users)

Frank is interested in some modern project building technologies, such as CMake. He downloaded the cmaketemplate and wanted to make some modifications on it.
The `CMakeLists.txt` contains all of the configuration of a project.

```cmake
cmake_minimum_required(VERSION 2.7)

project(hwx)

# Set the C standard to C11
set(CMAKE_C_STANDARD 11)

# Open all warnings and consider warnings as errors
set(CMAKE_C_FLAGS "-Wall -Werror")

# Add executable for different exercises
add_executable(ex1 ex1.c)
add_executable(ex2 ex2.c)
add_executable(ex3 ex3.c)
add_executable(ex4 ex4.c)
```

You need to help him complete these teaks:

i) Understand each command in CMakeLists.txt

ii) Add an executable `ex5`, with a file `ex5.c`

iii) Add files `ex5_other.c` and `ex5_other.h` to `ex5`

iv) Create a folder for each executable and move the corresponding file(s) into the folder.

2.3 Static and Shared Library

Library

In computer science, a library is a collection of non-volatile resources used by computer programs, often for software development. These may include configuration data, documentation, help data, message templates, pre-written code and subroutines, classes, values or type specifications.

For a large project, it may contain millions of lines of code. It will take several hours to build only part of the whole project. So it is very important to separate different parts of the project into libraries.

A static library or statically-linked library is a set of routines, external functions and variables which are resolved in a caller at compile-time and copied into a target application by a compiler, linker, or binder, producing an object file and a stand-alone executable.

A shared library or shared object is a file that is intended to be shared by executable files and further shared object files. Modules used by a program are loaded from individual shared objects into memory at load time or run time, rather than being copied by a linker when it creates a single monolithic executable file for the program.

Now you are supposed to build a static and a shared library and link the library to your executable.

For CMake users, you will find the command `add_library` and `target_link_libraries` useful. You can find more information on CMake Documentation.

You can use `ex5_other.c` and `ex5_other.h` to build the library and `ex5.c` and and `ex5_other.h` to build the executable. Then link the library to the executable.

2.4 C/C++ Documentation

Simon recommended you to check these references regularly.

- C Reference
- C++ Reference

Simon had a small test for you to examine whether you are able to use these references. He will give you a `time_t` returned by `time(NULL)` and you should print the date and time in the correct format. You may use the function `strftime` and UTC time.

Input: 1529351300
Output: 2018-06-18 19:48:20

Test your program on JOJ.