Advanced Procedure-Oriented Programming, Fall 2013

Homework Assignment #1

Due midnight Monday, October 21, 2013

Instructions

1. If any question is unclear, please ask for a clarification.
2. You are required to do all the homework assignments on Linux. To ensure that your C program is also a C++ program, you are required to use both gcc and g++ version 4 or later to compile your program.
3. You are encouraged to make sure that you program can be compiled by MFC, but it is not required.
4. You are required to give your TA a demo of your program. Make sure that your program can compile and run on the server machine, which will be used for the demo.
5. For the program that you write, you are required to include a Makefile. Otherwise, the grade for your program will be zero.
6. Unless stated otherwise, you are required to work on the homework assignment individually.
7. No late homework will be accepted.

Programming Project

The purpose of this homework assignment is to get you acquainted with the GNU make and the modular design of a program in a procedure-oriented programming language, C.

The requirement for this assignment is simple and is given as follows:

- First, you are required to write the following four modules:
  1. a "list" module that implements a doubly linked list,
  2. a "stack" module that builds on the list module,
  3. a "queue" module that builds on the list module, and
  4. a "mm" module that acts as the memory manager by wrapping up the functions `malloc`, `calloc`, `realloc`, and `free` as defined in the standard library. One way to wrap up these functions is to add a prefix to the name of these functions so that `malloc` is named `mymalloc`, `calloc` is named `mycalloc`, and so on.

Taking into account the interface and implementation, you are supposed to have at least the following eight files: `list.h`, `list.c`, `stack.h`, `stack.c`, `queue.h`, `queue.c`, `mm.h`, and `mm.c`. Moreover, it is up to you to define the interface of each module, and to hide as much as possible the implementation of each module.

- Then, you are required to write drivers to test the stack and queue modules that you design and implement. You may name them `main_stack.c` and `main_queue.c` and assume that the input is a list of integers each of which is on a line by itself, and so is the output.
- Finally, you are required to write a Makefile—which contains at least three targets: **all**, **dep**, and **clean**—to manage the project.

As far as this homework assignment is concerned, you may assume that all you need is a stack of type integer or a queue of type integer. However, you are encouraged to prepare for the changes that may require that several stacks or queues of different types be needed at the same time.

Grading Policy

The grading policy for this assignment is as follows:

- This assignment accounts for 10 points to your final grade.
- Make sure that a **Makefile**, which contains at least three targets—**all**, **dep**, and **clean**—is provided. Otherwise, the grade for your program will be zero.
- 8 points if your program compiles and runs without errors and warnings.
- 2 points if the program is properly modularized and well structured.

Gentle Reminder

1. If you have never had experience on using Linux, start earlier. It may take you quite a while to get used to it.
2. If you have never had Linux installed on your system, it is time to get it installed.