Procedure-Oriented Programming, Fall 2016

Homework Assignment #4

Due midnight Wednesday, December 14, 2016

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 still to get you acquainted with the modular design of a *large* program in a procedure-oriented programming language, C

This assignment requires that you write your own memory manager. In other words, instead of wrappers as shown below

```
1  #include <stdlib.h>
2  #include "mm.h"
3
4  void *mymalloc(size_t size)
5  {
6      return malloc(size);
7  }
8
9  void myfree(void *ptr)
10 {
11     free(ptr);
```

```
}

void *myrealloc(void *ptr, size_t size)
{
    return realloc(ptr, size);
}

void *mycalloc(size_t nmemb, size_t size)
{
    return calloc(nmemb, size);
}
```

you are writing your own memory management functions, as follows:

```
#include "mm.h"

void *mymalloc(size_t size)
{
    // your own code
}

void myfree(void *ptr)
{
    // your own code
}

void *myrealloc(void *ptr, size_t size)
{
    // your own code
}

void *mycalloc(size_t nmemb, size_t size)
{
    // your own code
}
```

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 own memory manager works for all the previous homework assignments. That is, all the previous homework assignments using your own memory manager compile and run without errors and warnings and give the same results.
- 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.