Computer Programming Homework 0

Sheng-Yi Hong

February 8, 2023

Contents

Chapter 1

Requirement

I recommend to use any Unix-Like OS (e.g. FreeBSD, MacOS, Linux) to finish this homework. In windows, you can use either Virtual Machine or WSL to install Linux over Windows.

This homework require **google test** package. Google test is a unit test library on C/C++. It helps us build test system over er this homework so that you can check if you write the correct code. It is avaliable on most of the operating system. If you use Ubuntu Linux, you can use command **sudo apt install libgtest-dev -y** to install **google test** package. Other system you need to find by yourself.

Chapter 2

Review of C

2.1 Problem 1: Linked List

Linked list is widely used in operating system kernel due to the high performance on insert and delete elements compare with other data structures like array. Linked list can be implemented from pointer, which we have learned in Computer Programming - I.

2.1.1 Requirement

In this problem, You're asked to implement Linked List of **int32_t** in C by using array. Because you haven't learned structure, I want you to use array to emulate the linking relationship. You have to modify all of the functions in **list.c** to comply with the requirements. There are many types of Linked List. In this problem I use the definition in C++ std::list. Not all of the functions in **std::list** are required. You only have to finish ones in **list.c**. Of course, you can add some auxilary functions to help you implement linked list.

After accomplishing these requirements, you can run **make test** to test all of your codes. And once you passed all these tests, you finished this problem.

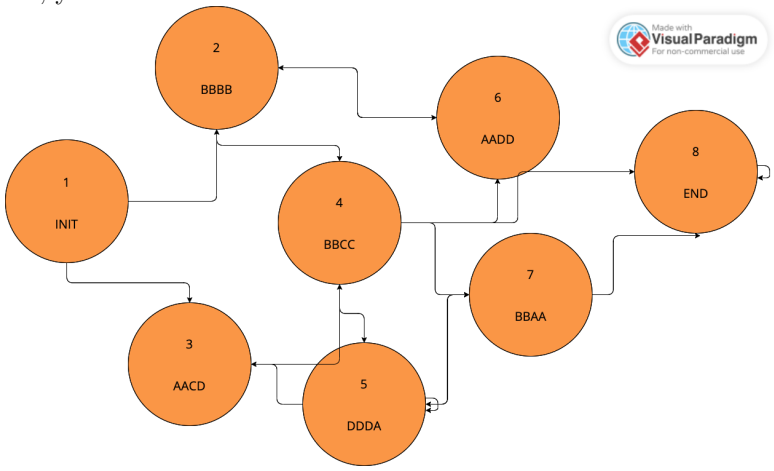
2.2 Problem 2: FSM

I think you have learned Function in CP-I. Also, you have learned Pointer: a variable point to some memory address. Surprisingly, function also has its address. So pointer can also point to a function. To understand how to use function pointer, you can take a look at this website here.

2.2.1 Requirement

Function pointer helps us solve many problems. In this problem, You're asked to implement a FSM (Finite State Machine) where each node is a function

pointer. We have to provide one function called **init()** and several functions with nodes in this problem, which will return a function pointer of the initial node. The prototype of node function **'void *(int)'** has an alias called **f_type** in **fsm.h**. This function prototype requires an index to indicate the next node you want to jump, and the return value will be the pointer to the next function. The reason why the return value is **void *** instead of **f_type** is the limitation of **typedef** in C. We need to do type conversion manually after calling **f_type** objects. Besides from the return value, you are also required to print some information show in the middle of node in the under graph(e.g. INIT, BBBB) on the screen when executing the function in each node. Once the index passed to the function does not exist anymore, you should return the pointer to the current node. The following graph is the relationship(Arrow), index(First line in the circle), and the required print string (Second line in the circle). After finishing it, make sure to run **make test** to test your code. If the output string is **OK**, your code is correct.



Chapter 3

Preview of C++

3.1 Problem 3

3.2 Problem 4