Computer Programming Homework 0

Sheng-Yi Hong

January 31, 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 structure like array. Linked list can be implemented from pointer which we have learned in Computer Programming - I.

2.1.1 Requirement

In this problem, I want you to implement Linked List of **int32_t** in C by using array. Because you havn't learn 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 requirement. There are many types of Linked List. In this problem I use the definition in C++ std::list. Not all of the function in **std::list** is required. You only have to finish ones in **list.c**. Of course, you can add some auxilary function to help you implement linked list.

After finishing the requirement, you can run **make test** to test all of your codes. After passing all of the tests, you finish this problem.

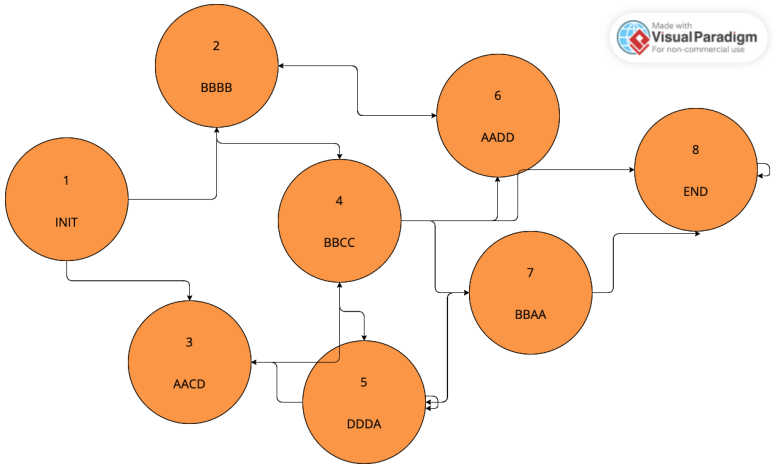
2.2 Problem 2: FSM

I think you have learned function in CP-I. Also, you have learned pointer. Pointer is variable point to some memory address. Surprisingly, function also has it's address. So pointer can also points to a function. To understand how to use function pointer, you can take a look to this website.

2.2.1 Requirement

Function pointer helps us solve many problem. In this problem, I want you to implement a FSM (Finite State Machine) where each node is a function

pointer. We have only one function called **init()** in this problem, which will return a function pointer of initial node. The prototype of node function **'void *(int)'** that has a alias called **f_type** in **fsm.h**. This function prototype require a index to indicate the next node you want to jump and the return value is the pointer to next function. Why the return value is **void *** instead of **f_type** is due to the limitation of **typedef** in C. We need to do type conversion manually after calling **f_type** object. Besides the return value, you also required to print some information on the screen when executing function in each node. When the index passed to the function is not exist. You should return the pointer to the current node. The following graph is the relationship, index and print string required. After finishing, also run **make test** to test your code. If OK output, your code is correct.



Chapter 3

Preview of C++

3.1 Problem 3

3.2 Problem 4