Lab #2 - More C: Formats and Pointers

C W Liew

September 2, 2019

Goals

The goals of this lab are to start learning how to program in C using different formats for output, pointers and multiple files.

Learning C

There are many textbooks and online tutorials for learning C. Please pick something that you are comfortable with.

In Lab 1, you were given an example program that consisted of multiple files. This week an example of a C program that outputs data in different formats (hex, decimal) is shown in `ex1.c`. The program shows you how an example of how pointers are declared and used. Your assignment this week will build on that.

Assignment

Your assignment for this lab consists of completing the following steps (all to be done in terminal):

1. use *ssh* to login to the remote computer and account that have been created for you. All of the subsequent parts are to be done on the remote computer.
2. create a directory lab02. The rest of this assignment should be done in the lab02 directory.

Program Description

Write a program that has the following parts (each part is in its own file so you will have multiple .c and .h files).

1. a function *bool palindrome(char* x)* returns **true** if the string (char array) is a palindrome. Uses pointer notation only. There are three cases you need to think about - (1) the char array has an even number of characters, (2) the char array has an odd number of characters, (3) the string is empty.
2. a function *unsigned equals(int* x, int* y)* that is passed two int pointers. The function returns 0 if the two ints do not match, -1 (0xffffffff)if they match but they are different locations, and the address if they are both pointing to the same address.

3. a function *int equals2(int** x, int** y, int size)* that is passed two arrays of int* (x and y) and size is the number of elements in each array. The function returns -1 if **ALL** the int* in the two arrays are the same and the index of the first occurrence where they are different. The function uses the *unsigned equals(int* x, int* y)* function to determine equality of pointers.

4. a function *bool find(int x, int* p, int size)* that is passed an int (x) and an array of int. The function checks each of the int in p, and returns true if ALL of the contents is equal to x and returns false otherwise. Note that you have to check that the int* pointer is valid, i.e., does not have a value of 0, before comparing the contents.

5. a function *initData()* that will:

   (a) prompt the user for a string (array of char) and saves the data in a global variable

   (b) create an array of int* (i.e., int**) of size 5 called intptrArray (global variable). The function will then allocate space for an int (holds one int) for each one of the int* and initialize the value to a random int (use *rand()*).

6. a function *testProgram()* that will perform the following tests and prints the corresponding results:

   (a) testPalindrome(): calls palindrome() on the user string and prints whether it is a palindrome.

   (b) testEquals(): calls equals(int*, int*) with:

   - equals(intptrArray[0], intptrArray[0])
   - equals(intptrArray[0], 0)
   - equals(intptrArray[1], intptrArray[2])

   (c) testFind(): calls find(int, int**) with:

   - find(*(intptrArray[0]), intptrArray, 5)
   - find(*(intptrArray[3]), intptrArray, 5)
   - sets intptrArray[2] to 0 - i.e., zeros out the pointer, then find(*(intptrArray[4]), intptrArray, 5). After the test, reallocates space for intptrArray[2] and reinitializes the value so that it can be used for other tests

   (d) testEquals2(): calls equals2(int*, int*, int) with:

- equals2(intptrArray, intptrArray, 5)
- equals2(&intptrArray[1], intptrArray, 4)

7. a main function in the file lab02.c that will call *initData()* and then *testProgram()*.