Stockholms Tekniska Institut

IOT23

Lab 1 – C Programming
2024-02-20

Introduction

Welcome to the first lab in the course. In this lab, you will learn the low-level basics of the C programming language. More specifically, after finishing this lab, you will be able to:

1. Construct shorter C programs that include for-loops, if-statements, arrays, and pointers.
2. Understand memory layout and how it interacts with program.
3. Utilization of Github to properly work in a project group environment.

Preparations

Make sure to have gotten a grasp of pointers, malloc, memory layout (stacks, heaps), generic C programming (Datatypes, arrays, while loops, for loops).

Some refreshment of generic mathematics is also recommended for the final task, if needed there is a code skeleton provided for it.

Remember to also make sure you have a good understanding of how datatypes are stored in the memory and how memory addresses work!

Help for the Lab will be provided with the last lesson this Friday and possibly after lunch on normal lessons.

Lab Environment

Github Desktop or Terminal if you prefer that.

Free IDE of choice, VSCode is recommended.

MingW as C/C++ compiler.

The provided .zip file skeletons.

Tutorials are provided in the lecture slides but can also be found via quick googling.

For compiling in MingW: open a terminal shell and type.

gcc <filename.c> -o <output name>

Reading Guidelines

Go through material noted in the introduction, most of it is covered by the first three lectures and the websites referenced in the slides.

Github grading

Because the tasks differ from each other, you do not need to setup skeleton code for the main branch.

- Atomic commit system (Each commit may only change one thing at a time, the change must be stated in the comment.
- Main Branch should not be used except for merging. In detail, you should, when finished with the task in the branch you are in, push the finished code towards the main branch.
- Branches must be utilized to split up the workload, it's recommended each problem is its own branch (Do not use the main file for each problem, otherwise they will merge with the main branch file).
- When Merging / Pull request, make sure that everything works, the other partner should conduct a review before pushing to the main branch.
- Github Desktop or Terminal should be used, using the website UI is not allowed.

Plagiarism

Plagiarism will be reported, and thus copying others or using AI generated tools is not allowed. It is fine to take inspiration from the internet and other areas but remember that you must explain your code when presenting.

Assignments

Assignment 1: Warmup, Prime Printing

The program prints number 0 three times.

Task: Your task is to implement the body of function `is_prime()`, so that it returns value 1 if parameter `n` is a prime number, and 0 if it is not. The function must behave correctly for all positive integer numbers. The function should not use any data structures, only simple loops, and conditional `if`-statements.

Assignment 2: Malloc and Pointers

Your task is to implement the function initializeArrayWithInput() in Malloc.c. The point of this task is to understand how memory layout works and how one can iterate through the memory address to access the next array elements without ever creating or accessing an array. When properly implemented and choosing a array size of 10, the following result should come out.

Note that using array indexing is not allowed!

```
PS C:\Users\mertd\Downloads\Lab1> gcc malloc.c
PS C:\Users\mertd\Downloads\Lab1> ./a
Ange storleken pa arrayen: 10
Ange startvardet for heltalen: 1
Arrayelement och adresser:
Varde: 1, Adress: 00000172CA1BDFD0
Varde: 2, Adress: 00000172CA1BDFD4
Varde: 3, Adress: 00000172CA1BDFD8
Varde: 4, Adress: 00000172CA1BDFDC
Varde: 5, Adress: 00000172CA1BDFE0
Varde: 6, Adress: 00000172CA1BDFE4
Varde: 7, Adress: 00000172CA1BDFE8
Varde: 8, Adress: 00000172CA1BDFEC
Varde: 9, Adress: 00000172CA1BDFF0
Varde: 10, Adress: 00000172CA1BDFF4
PS C:\Users\mertd\Downloads\Lab1>
```

Note that the address can vary and will be random with each run as memory is freed when the program exits!

Questions for assignment 2

Before presenting, you should prepare the answers to the following questions. You will need to be able to answer these questions to pass the assignment.

- In a normal memory structure, where is the data stored? Which section of the memory? (Take note on how Malloc works)
- Why does the Adress increment by four each iteration?
- What does each Adress contain?

Assignment 3: Malloc and Pointers Continued

Your task is to expand further and implement the function performOperations() in Malloc.c. The point of this task is to understand how data can be changed by direct memory access. When properly implemented and choosing an array size of 10, the following result should come out.

Note that using array indexing is not allowed!

```
PS C:\Users\mertd\Downloads\Lab1> gcc malloc.c -o test
PS C:\Users\mertd\Downloads\Lab1> ./test
Ange storleken pa arrayen: 10
Arrayelement och adresser:
Varde: 1, Adress: 000001C6F216DEE0
Varde: 3, Adress: 000001C6F216DEE4
Varde: 5, Adress: 000001C6F216DEE8
Varde: 7, Adress: 000001C6F216DEEC
Varde: 9, Adress: 000001C6F216DEF0
Varde: 11, Adress: 000001C6F216DEF4
Varde: 13, Adress: 000001C6F216DEF8
Varde: 15, Adress: 000001C6F216DEFC
Varde: 17, Adress: 000001C6F216DF00
Varde: 19, Adress: 000001C6F216DF04
```

Note that the address can vary and will be random with each run as memory is freed when the program exits!

Questions for assignment 3

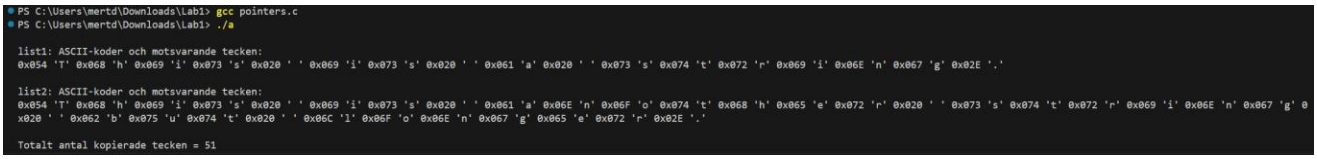
Before presenting, you should prepare the answers to the following questions. You will need to be able to answer these questions to pass the assignment.

- How do you iterate through the elements? Why?
- Why does this work similarly to an array?
- Why is there 16 characters in the Adress?
- If you wanted to change the incrementation, what would you do?

Assignment 4: Pointers

The purpose of this assignment is to learn how to use pointers to iterate through arrays, in this case a char string array. Solve the copyStringToAsciiValues(). The code should print out something like below.

Note that using array indexing is not allowed!



Questions for assignment 4

Before presenting, you should prepare the answers to the following questions. You will need to be able to answer these questions to pass the assignment.

- How do you iterate through the char elements? Why?
- Why does this work similarly to an array?
- Why do you need chars to form a string?
- Why is the hexadecimal converter function needed? What values does char contain?

Grading

All tasks need to be solved to pass; exceptions can be made with smaller errors if the others are passed. For VG all tasks need to be solved, explained in detail, commented on in detail and the code structure being easy to follow. The ruleset of GitHub must also be followed with proper showing of working in a group.