

# CMPSC 400 Operating Systems Spring 2020

## Lab 1 Assignment:

Exploring C Programming and Writing Simple System Call Demonstration in C  
**Submit deliverables through your assignment GitHub repository bearing your name. Place source code in src/ and written work in writing/.**

## Objectives

To enhance your understanding of C programming and to explore system calls through your new-found knowledge of this programming.

## About This Lab

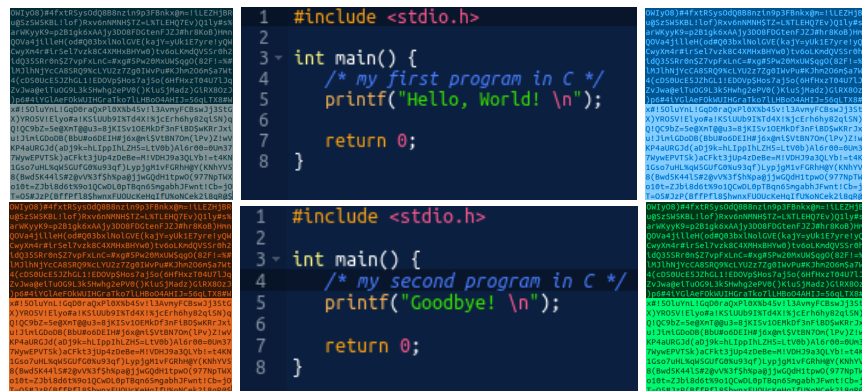


Figure 1: At the Kernel level, it is a messy place and the codes to process actions (*calls*) are not very meaningful to humans. To instruct the kernel to complete processes, programmers use libraries from C programming that handle the communication and requests. The programming code invokes a system call in the library, which then handles the communication to the kernel. Here, we note that this interaction is completed by abstraction thanks to the libraries. The human user is often totally unaware of this interaction.

In this lab, you will be writing three programs in C. Please use the **Docker** container that was given in class to write and compile your code. The objectives are the following.

- Part 1, **File part1.c**: A introduction program in C. The program must compile correctly and must complete the following tasks.
  - Ask the user for a first name
  - Ask the user for a last name

- (c) Ask the user for a first random number.
  - (d) Ask the user for a second random number.
  - (e) Print a greeting using the first and last name.
  - (f) Print the favorite numbers and their sum.
2. Part 2, **File part2.c** and, **File part3.c**: Choose two (2) basic system calls to include in source code demonstration for each.
- (a) Name and Explain your chosen *system calls*.
  - (b) Write a small program in C (for each) to demonstrate how each works.
  - (c) Note: If your code or explanation is inspired from a source other than your class textbook, please be sure to offer a complete citation of the source (i.e., website, article, book, etc.).

## GitHub Starter Link

<https://classroom.github.com/a/Yg9PgQWs>

To use this link, please follow the steps below.

- Click on the link and accept the assignment.
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab.
- Clone this repository (bearing your name) and work on the lab locally.
- As you are working on your lab, you are to commit and push regularly. You can use the following commands to add a single file, you must be in the directory where the file is located (or add the path to the file in the command):

```
- git commit <nameOfFile> -m ‘‘Your notes about commit here’’  
- git push
```

Alternatively, you can use the following commands to add multiple files from your repository:

```
- git add -A  
- git commit -m ‘‘Your notes about commit here’’  
- git push
```

## Reading Assignment

In operating systems, much of the code is written in the C programming language. In this lab, you will also be programming in C and so you will spend part of this lab to complete the parts of the C programming tutorial that you believe will be beneficial to your work, which is available at <https://www.tutorialspoint.com/cprogramming/index.htm> and shown in Figure 2.

This tutorial was chosen for its interactive enhancements where you are able to code online, however, there are other similar tutorial sites available online. If you find a better one for C programming that provides a better experience for you, then please consider letting the instructor know in your submission document. Such a tutorial and online programming tool may be chosen for use in future classes and labs.



Figure 2: The Tutorialspoint website [1] is one of many enriching resources from which you may quickly learn how to program in C. There are plenty of examples offered, in addition to associated work-spaces where code can be written and compiled online. The tutorial can be accessed from the link: <https://www.tutorialspoint.com/cprogramming/index.htm>.

Be sure to read your textbook by Tanenbaum, [2]. System calls are discussed in great detail in Chapter 1.6 and also are discussed in your slides. Examples of commonly used system calls are mentioned in both of the above resources and should be helpful to you as you complete your lab. You are also encouraged to go online to read more about system calls by conducting online searches.

## Testing Your Programs and Evaluating Your Submission

You will be using Docker in this lab. For specific commands and instructions for compiling and running your programs, as well as checking that all required files have been included, please see the README file in your lab repository.

## Required Deliverables

Submit deliverables through your assignment GitHub repository bearing your name. Place source code in `src/` and the output text file `output/` directories. To receive full credit for this assignment, all three programs must compile and run correctly and produce relevant output, and your submission must pass all GatorGrader checks.

1. Part 1, File `part1.c`: General C program conforming to the specifications above.
2. Part 2, File `part2.c`: A program in C to demonstrate the usage of a *first* chosen system call, conforming to the specifications above.
3. Part 3, File `part3.c`: A program in C to demonstrate the usage of a *second* chosen system call, conforming to the specifications above.

In adherence to the Honor Code, students should complete this assignment on an individual basis. While it is appropriate for students in this class to have high-level conversations about the assignment, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others and the student who produces assignments that are identical to, or merely variations on, someone else's work. Deliverables that are nearly identical to the work of others will be taken as evidence of violating Allegheny College's Honor Code.

## References

- [1] "Tutorialspoint," <https://www.tutorialspoint.com/cprogramming/index.htm>, accessed: 2020-01-20.
- [2] A. S. Tanenbaum and H. Bos, *Modern operating systems, Forth Edition*. Pearson, 2015.