Prof. Dr. Burkhard Stiller, University of Zurich, Binzmühlestrasse 14, CH-8050 Zürich
Telephone: +41 44 635 6710, Fax: +41 44 635 6809, stiller@ifi.uzh.ch
Bruno Rodrigues, Telephone: +41 44 635 4381, rodirgues@ifi.uzh.ch
Sina Rafati, Telephone: +41 44 635 6702, rafati@ifi.uzh.ch

Exercises in SSDS

Introductory Exercise

Hand out date: Feb 19, 2020
Submission until and discussion: Feb 20, 2020 08:15

Introduction to Operating Systems and C Programming Language

An Operating System (OS) is a collection of software that manages computer hardware resources and provides common services for high-level application software to interact with the hardware. It is a vital component in a computer system responsible for basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directiores on disk, and controlling peripheral devices such as printers.

Processors run binary Machine Code. Assembly is the language that represents Machine Code. OSs requirements became too arduous to implement purely in Assembly, therefore using a high-level programming language can make the OS development task much easier. The most popular language for writting OSs is C, which is used to perform basic functionalities (Kernel) of Linux, Microsft Windows and macOS. C was initially used for system development work, it was widely adoped as a system development language because it produces code that runs nearly as fast as code written in assembly language.

In this course we will use Ubuntu as OS configured in a VM. Therefore, this exercise has the objective to prepare the baseline environment to perform the SSDS exercises and, also to introduce basic concepts of the C programming language.

Submission Guidelines

- Submitted code must compile without errors.
- Zip your code using the name pattern: ssds_ex**+exercise+number_initials**. For instance, the file of Exercise 01 for Bruno Rodrigue would be named: sssds_ex01_BR.zip
- Submit the exercise to the reseach assistants e-mail addresses.

1) Windows

Alternative #1: download and install Dev-Cpp:
https://sourceforge.net/projects/orwelldevcpp/

Alternative #2: use a VM with ubuntu

1. Download Virtualbox.
   https://www.virtualbox.org/wiki/Downloads
2. Download the Ubuntu 18.04 LTS image and configure it on VirtualBox.
   https://www.ubuntu.com/download/desktop
3. GCC is installed by default. Read/use the tutorial on how to compile C programs.
   http://pages.cs.wisc.edu/~beechung/ref/gcc-intro.html

2) Linux

Most distros already have “gcc” built-in. Check whether it is installed by typing in your terminal:

```
$ which gcc
```

To compile a C code, you can typing in the terminal:

```
$ gcc hello.c
```

If the program was compiled without errors or warnings, you will have a new file in the directory called ***a.out***. To run the file, go to the same directory and type in the terminal:

```
$ ./a.out
```

Furthermore, if you want to specify the output, it is possible by doing the following:

```
$ gcc hello.c -o hello
$ ./hello
“Hello World”
```

3) The first program, hello.c

The first has the objective to get you familiar with the development environment, the C syntax and the GCC compiler. Therefore, in this program you shold produce a “Hello World” message and wait for keyboard entry to terminate the execution. Save your code as “hello.c”.

Note that the important is not to note the result of the code itself (*i.e.,* a simple “Hello World), but how the high-level calls (system calls) are handled by the operating system, how an output is made, among other examples.