2023-2024 Project 3 - Virtual Processor - Call for Tender

The goal of the project is to create a virtual processor and an interpreter for running assembly code on that processor.

The project will be developed in plain, portable, C language without the use of any external library beside C standard libraries. We recommend you use gcc as a compiler and Visual Studio Code as IDE.

First, you need to invent a minimal assembly language for your processor. Your instruction set will contain at least the following instructions:

- Storing an immediate value into a register.
- Copying the value of a register into another register.
- Reading the value of the memory at the address contained by a register and storing it into another register.
- Storing the value of a register into memory at the address contained by another register.
- Comparing the content of registers.
- Jumping unconditionally to a label.
- Jumping conditionally to a label.
- Calling a subroutine.
- Returning from a subroutine.
- The 4 basic arithmetic operations: addition, subtraction, multiplication, and division.
- The 4 basic logical operations: OR, AND, XOR, and NOT.

Your assembly language should be fully described in your ***functional*** specification.

Then you need to write a C program that can read a text file containing a program written in your assembly dialect and run it. The C program also needs to check that the assembly program is semantically valid and detect syntactical errors.

In order to see that the assembly program is actually running, implement a virtual system call for displaying text in a virtual terminal, that can be accessed from the assembly code. You could also display the content of registers and have a built-in debugger.

The way the C program is intended to work should be described in your technical specification.

Finally, you need to write small assembly programs conceptually similar to unit tests to prove that everything is working as expected.