

Jupyter Kernel for ASM



National Institute of Technology, Hamirpur

Term Project - 1

Group Leader: Sarita Singh (184544).

Group Members: Tamanna Rana (184536)

B. Krutik Ganpati (184537)

Nikita Mahoviya (184540)

Term Project - 1

Jupyter Kernel for ASM

Project Overview

A notebook kernel is a "computational engine" that executes the code contained in a Notebook document. Assembly languages are the most fundamental programming languages available for any processor. Our project aims to develop a kernel that can interpret assembly-level programs right from Jupyter notebook.

Literature Review

There have been numerous kernels developed in the Jupyter notebook. A kernel is basically a module that adds a new language to the Jupyter notebook world while a Jupyter notebook is an open-source web application that can be used to create and share documents that contain live code, equations, visualizations, and text. Normally, each notebook can run only one kernel and one language, but workarounds exist.

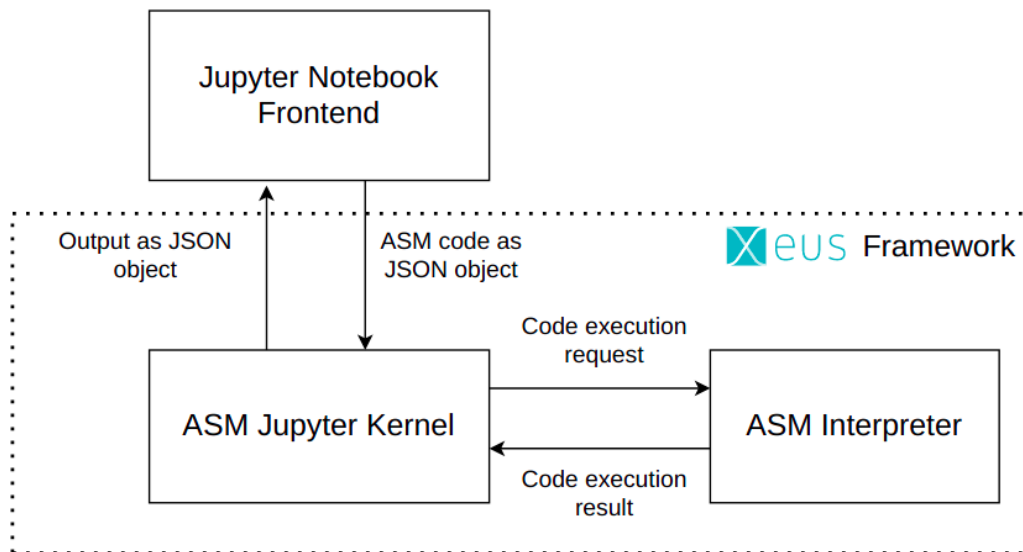
Currently, more than 100 Jupyter kernels have been developed, supporting dozens of programming languages like python, CPP, Julia, SQL, etc., but the number of kernels for interpretation of assembly-level programs is practically negligible. Assembly language is a low-level language that allows users to write a program using alphanumeric mnemonic codes, instead of numeric code for a set of instructions

Hypothesis

With the development of the Jupyter kernel to interpret assembly-level programs, we will be able to run these programs without having to master the specific tools. The intuitive graphical user interface and easy ways of saving and distributing the code will also enhance the knowledge of the beginners and help them in their learning process.

Methodology

Jupyter Notebook is used as the frontend. We will use the Xeus framework at the backend. When a user runs a command, it will be passed as a JSON object to the Jupyter kernel which in turn sends requests to the interpreter to execute the commands. Once the command is executed successfully, the result will be sent back to the Jupyter kernel who will return output as a JSON object to the Jupyter Notebook.



Tech stack

- **Assembly:** A type of low-level programming language that is intended to communicate directly with a computer's hardware.
- **C/C++:** Using C/C++ assembly program interpreter/compiler.
- **Xeus framework:** Xeus is a C++ implementation of the Jupyter kernel protocol, and a framework to create Jupyter kernels.
- **Jupyter notebook:** Using its UI to provide functionality at the client-side

Why this idea?

In our previous semesters, we went through courses dealing with 8085 microprocessor and 8086 microcontroller which required coding over software like, GNUSim8085 and arm Keil respectively. While accessing these softwares we encountered certain problems listed below::

- Compatibility issues with the different operating systems
- License required which can be either institutional or bought.
- Software size producing memory issues.

These issues made us brainstorm in a direction that could increase the ease of user experience and you won't be needing your system always for any sort of work as everything will be browser-based and you will be just a login away from your work.

References

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mckinley, *8051 Microcontroller and Embedded System*, 2nd ed, Reading, Chung Ping Young, [E-book].
Available : <https://www.pdfdrive.com/>
2. Ramesh S. Gaonkar, *Microprocessor Architecture, Programming, and Applications with the 8085*, 5th ed, Reading, Prentice Hall, [E-book].
Available : <https://www.pdfdrive.com/>
3. Jupyter-client official documentation, "Making Kernels for Jupyter" [Online].
Available: <https://jupyter-client.readthedocs.io/en/latest/kernels.html>
4. Jupyter Jupyter Kernels[Source code].
Available : <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>
5. DeepHorizons(2016) Iarm [Source code].
Available: <https://github.com/DeepHorizons/iarm>
6. Gcallah(2019) Emu86 [Source code].
Available: <https://github.com/gcallah/Emu86/tree/master/kernels>
7. NYU Tandon School of Engineering, "An easier way to learn an assembly language", December 13 2019 [Online].
Available: <https://engineering.nyu.edu/news/easier-way-learn-assembly-language>