LEGv8 Assembler

Author: Gabriel Hofer

Instructor: Dr. Karlsson

Course: CENG-325 CPU & GPU Organization and Architecture

Date: November 16, 2020

Description

This program is an assembler for the LEGv8 instruction set.

Algorithms & Libraries

- The re (regular expression) library was used for parsing instructions
- The json library is used for loading objects from json files into dictionaries to make code more modular.
- The sys library is used to access command line arguments (sys.argv).

Program Structure

There are two JSON files that are passed as command line arguments to the main program file named Assembler.py. Assembler.py loads the contents of data.json and b_cond.json into python dictionaries.

How to Run the Program

Run with the python3 command on the command line:

```
$ python3 Assembler.py data.json b_cond.json [optional_input_file]
```

If no input file is given, the program reads instructions line-by-line on the command line. If an input file is given, instructions are read line-by-line from the input file.

The program ends when the string ‘quit’ is read from standard input on a single line.

Testing and Verification

I Ran all of the examples to make sure that the output from asm.py was the same as the output from the examples on the writeup.

Deliverable

I submitted a compressed file containing the following:

1. asm.py
2. data.json
3. b_cond.json

Grading & Notes

Grade for a B. Also, I didn’t implement the LDA pseudo-instruction. Finally, instructions should be provided in all caps because I didn’t convert instructions to uppercase.