Assembler README

Overview

This project is a two-pass SIC/XE assembler that converts assembly `.asm` files into object `.obj` and `lst` files for my systems programming class. It consists of multiple scripts to handle the assembly process efficiently.

The `Modules` folder contains all the classes, and modules that are called by the programs in the assignment folders.

Requirements

- **Python 3.x.**
  - Preferably above 3.10

Usage

1. Running with `.asm` Files

If you have an assembly `.asm` file, use the ComboP1P2.py script to perform both Pass 1 and Pass 2.

Command:

```
python ComboP1P2.py <filename>
```

Example:

```
python ComboP1P2.py Htest3.asm
```

This command processes `Htest3.asm`, runs Pass 1 to generate an intermediate `.int` file, and then runs Pass 2 to produce the final `Htest3.obj` file.

2. Running with Preprocessed `.int` Files

If you already have an intermediate `.int` file from Pass 1 (johnA4P1.py), you can directly run Pass 2 using the johnA4P2.py script.

Command:

```
python johnA4P2.py <intermediate_file>
```

Example:

```
python johnA4P2.py Htest1.int
```

This command processes `Htest1.int` and generates the corresponding `Htest1.obj` file.

Scripts Overview

- **ComboP1P2.py:** Runs both Pass 1 and Pass 2 sequentially on a given `.asm` file.
- **johnA4P2.py:** Executes only Pass 2 on a preprocessed `.int` file.
- **AssemblerPass1.py:** Handles the first pass of the assembler, building the symbol table and computing addresses.
- **AssemblerPass2.py:** Manages the second pass, generating object codes and writing the final `.obj` file.

Notes

- Ensure that all required modules are correctly placed in the Modules directory.

- Logs and error messages are handled by the ErrorLogHandler. Check logs for troubleshooting.

Example Workflow

1. **Assemble `.asm` to `.obj`:**

   ```
   python ComboP1P2.py MyProgram.asm
   ```

   - Processes `MyProgram.asm`
   - Generates `MyProgram.int` (intermediate file)
   - Produces `MyProgram.obj` (object file)
   - Generated `MyProgram.lst`

2. **Run Pass 2 on Existing `.int` File:**

   ```
   python johnA4P2.py MyProgram.int
   ```

   - Processes `MyProgram.int`
   - Generates `MyProgram.obj`
   - Generated `MyProgram.lst`