

# My Little MIPS Assembler

---

by Manqing ZHU

This assembler currently doesn't have the ability to deal with exceptions caused by weird input. Please be careful.

## Content

- **develop platform**
- **what's in the package**
- **function**
- **how to use it**
- **reference**
- **declaration**

## 1. Develop Platform

This application is developed on the macOS, using C++ and the Qt framework. It's supported by Qt 5.8 and C++ 11 standard library. **Unfortunately, it only runs on the macOS.** I'm sorry about it. (´\_`」 ∠)\_

## 2. What's in the Package

- **ReadMe.pdf**  
The document you are reading now
- **Assembler.app**  
The executable program
- **SourceCode.zip**  
It contains the source code of the Assembler program
- **images**  
It contains the resource of icons
- **TestSample**  
It contains a ".asm" file, a ".coe" file and a ".bin" file, which can be used as sample input files.

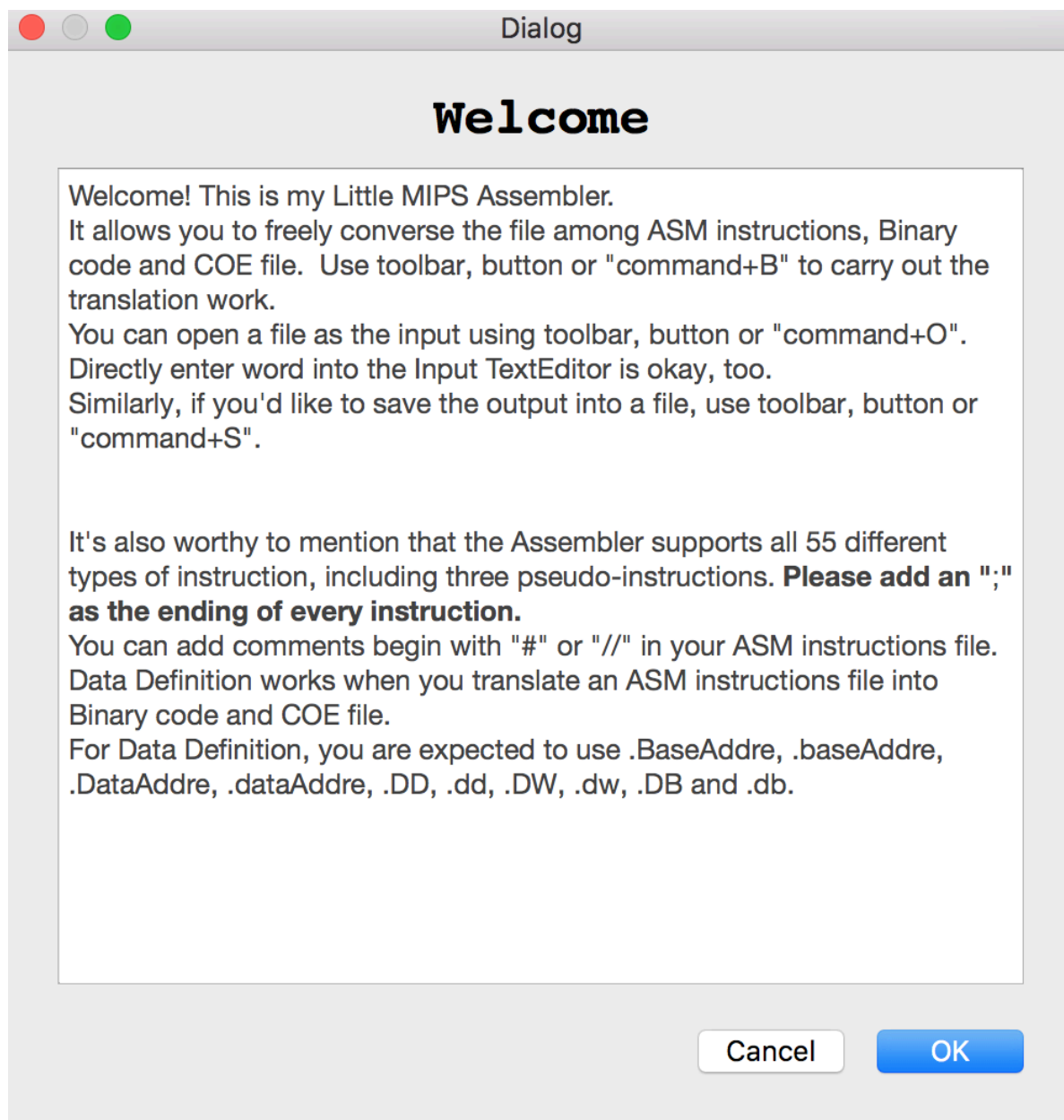
## 3. Function

- ☒ translate MIPS instructions into Binary code
- ☒ translate Binary code into MIPS instructions

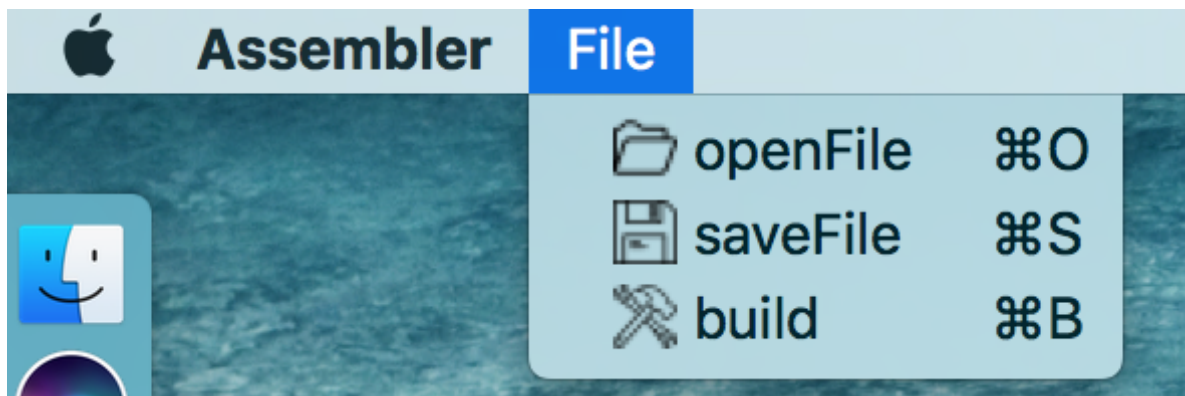
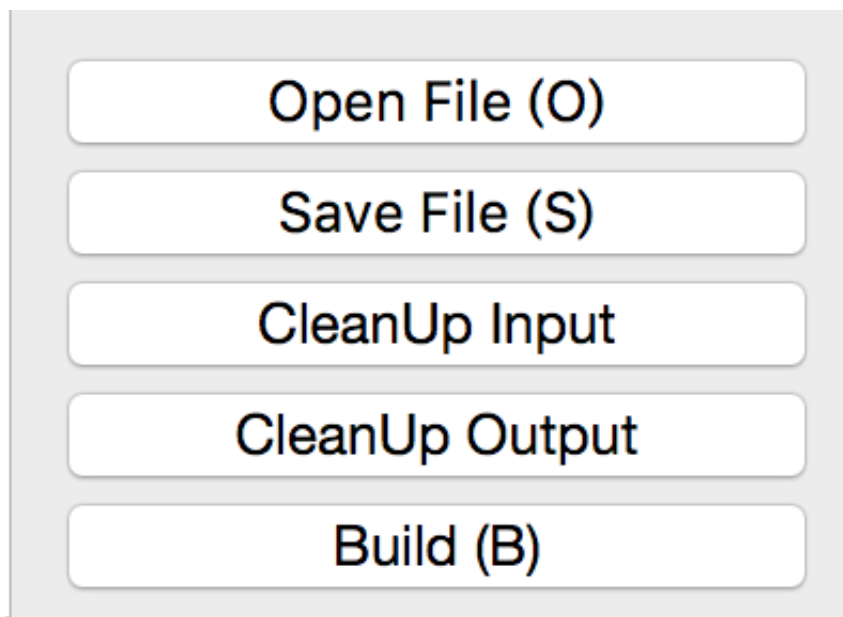
- ☒ translate MIPS instructions into a ".coe" file
- ☒ translate a ".coe" file into MIPS instructions
- ☒ converse between the ".coe" file and Binary code
- ☒ support 55 MIPS instructions, including 3 pseudo-instructions
- ☒ support Data Definition in the ".asm" file
- ☒ a friendly GUI supporting text editing and file operating

#### 4. How to use It

- After clicking the icon of application, you will see a welcome window first. There is a brief user manual for the application. **You will not enter into the main window unless you click the "ok" button in the bottom.**



- In the main window, you can assembly a program or disassembly one. Opening a file and Writing in the "Input" text editor are both acceptable. Short cut, tool bar and button can do the same work.



- Let's talk about the **input format**.
  - For the ".asm" file, a ";" is expected to be found as the ending of each instruction.

```
lw $a1, ($v1);
add $a1, $a1, $a1;
```

- For the ".bin" file, every line is assumed as one completed instruction.

```
00111100000010001000000000000000
00100000000101000000000000111111
```

- For the ".coe" file, the beginning of file should be these two sentence.

```
memory_initialization_radix=16;
memory_initialization_vector=
```

- the comments should begin with `//` or `#`.

- the identifiers used for Data Definition are `DataAddre`, `dataAddre`, `BaseAddre`, `baseAddre`, `.DD`, `.dd`, `.DW`, `.dw`, `DB` and `db`.
- before clicking the "Build" button, you are supposed to make the input type and output type explicit. Otherwise, you will get a warning.

**Input Type**

☒ asm ☐ bin ☐ coe

**Output Type**

☐ asm ☒ bin ☐ coe



- You can save the output result as an individual file. The operation is same as that of opening a file.
- It may be also worthy to mention that you can adjust the size of windows and text editors if you like.

## 5. Reference

[1] Qt Documentation, <http://doc.qt.io/>

[2] David A. Patterson, John L. Hennessy, "Computer Organization and Design: The Hardware and Software Interface", *China Machine Press*, 2013

[3] Markdown Reference, <http://support.typora.io/Markdown-Reference/>

## **6. Declaration**

***I hereby declare that all the work done in this homework is of my independent effort.***