

MIPS-sim

by 郑昱竺

简介:

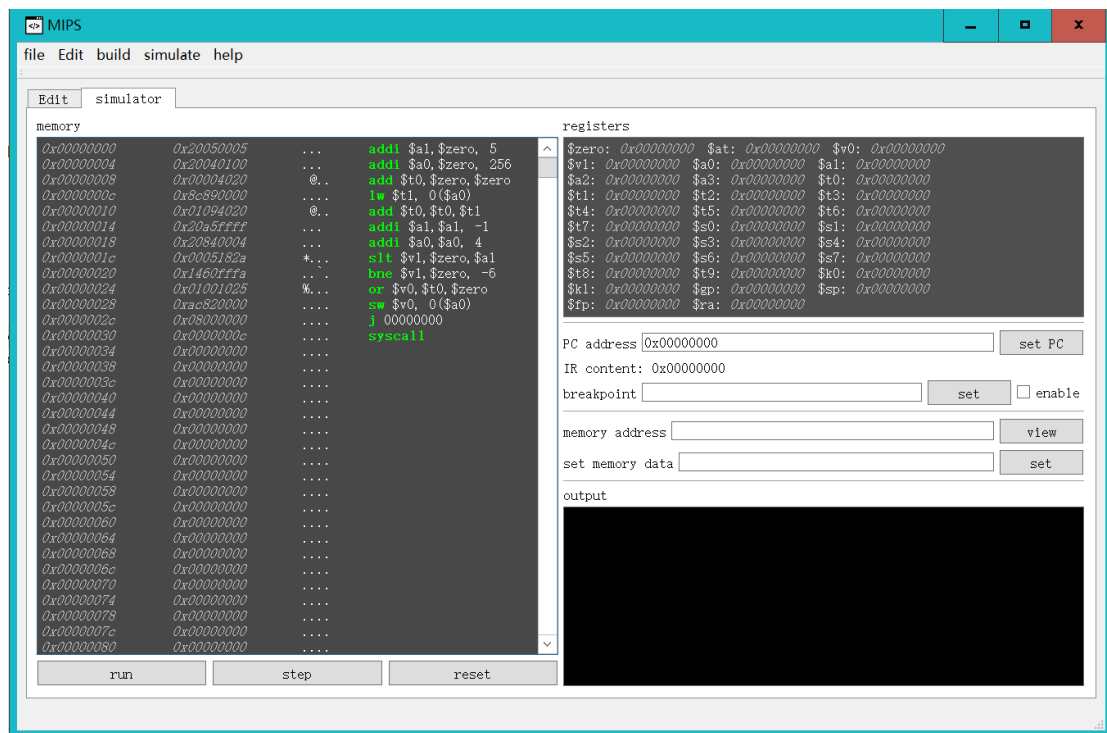
MIPS-sim 是一个基于 Qt 实现的带图形界面的 MIPS 汇编指令的编辑器、汇编器、反汇编器、模拟器。

模拟器界面:

1、模拟器界面:

左侧为 32 位内存内容、可以以 ascii 码方式或反汇编指令方式查看;

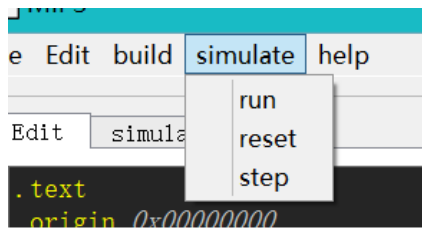
右侧为寄存器、syscall 输入输出窗口, 可通过按钮查看相应寄存器与内存的值、修改 PC 值、对应内存单元值;



可显示并修改 PC 地址、设置断点:

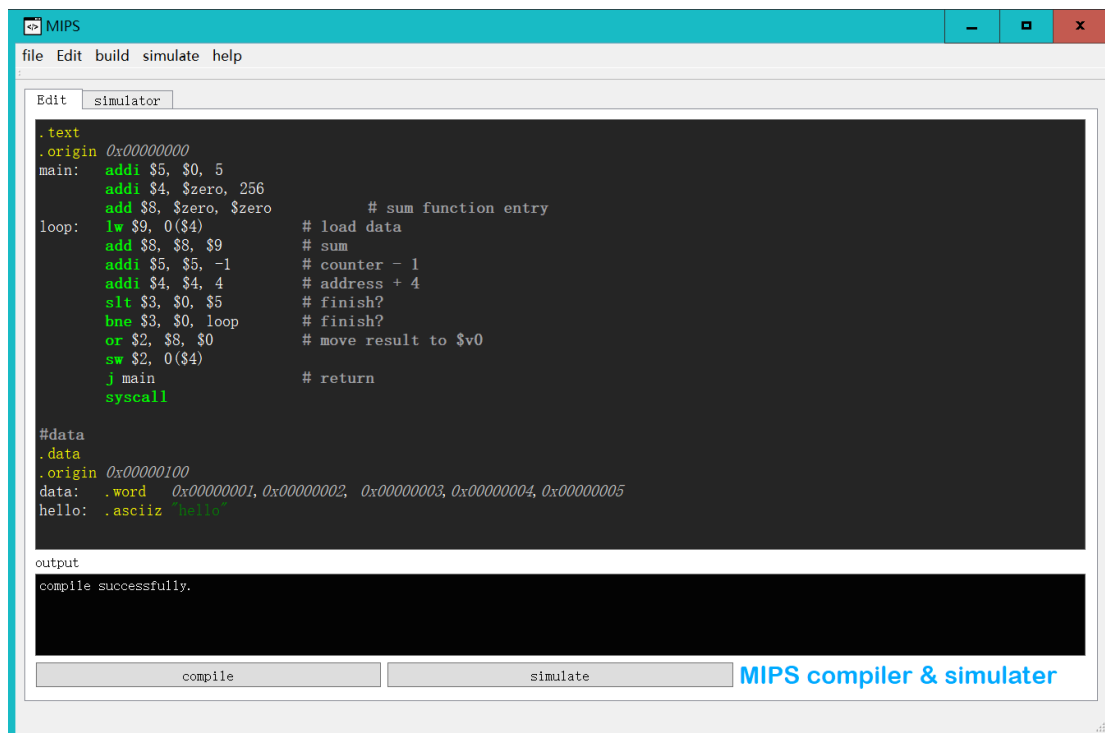
PC address	<input type="text" value="0x0000001c"/>	<input type="button" value="set PC"/>
IR content:	<input type="text" value="0x20840004"/>	
breakpoint	<input type="text"/>	<input type="button" value="set"/> <input type="checkbox"/> enable

模拟菜单 (和按钮相同的功能):



2、编辑器/编译界面

上方为代码编辑器，下方为编译输出结果，对相应错误有错误详细信息以及行号提示；
编辑器可自行设置字体、字号等；



编译错误输出如下（例）：

a.

```

:  lw $9, 0($4)
   ad $8, $8, $9
   addi $5, $5, -1
  
```

output

```

ERROR: text segment line number 7
      message instruction not support
      compile terminate.
  
```

b.

```

      addi $1, $1, 1
      slt $3, $0, $5
      bne $3, $0, loo
      or $2, $8, $0
  
```

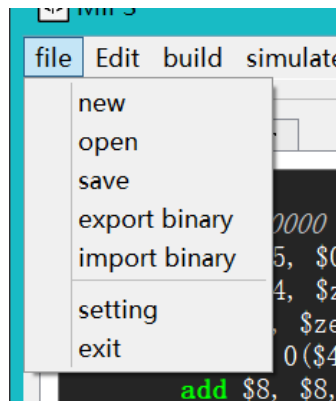
```
output
ERROR: text segment line number 11
message label not found
compile terminate.
```

3、文件菜单：

新建、打开、保存代码文件；

支持 .asm 汇编文件汇编为 .bin 文件并导出；

支持 .bin 文件反汇编为 .asm 文件，支持导入加载 .bin 文件并执行：



实现的特性：

- 支持语法高亮；
- 支持部分伪指令；
- 支持 .asm 汇编文件汇编为 .bin 文件；
- 支持 .bin 文件反汇编为 .asm 文件，支持加载 .bin 文件并执行；
- 支持模拟运行机器码，支持模拟终端输入输出
- 支持简单的调试功能：单步运行、连续运行、设置断点、查看寄存器与内存的值、修改 PC 值、对应内存单元值

指令集

参考《ZPC 之 MIPS 指令集 2019》

实现的部分指令：

- R 指令：

add slt sltu and or xor nor sllv srlv srav mul mfhi mflo mtlo subu

- I 指令：

slti sltiu addi addiu andi ori xori sub sw sh lw lh lhu bne beq bgez bgtz blez bltz lb lhu lui

- J 指令：

j jal

- syscall 功能

模拟

MIPS-sim 具有简单的模拟与调试功能。可以通过 `syscall` 指令向终端输出信息，或从终端读入用户输入信息。在编辑器输入代码后可以使用 "simulate" 按键进行编译和将机器码加载到内存，可以通过 `step` 按键单步执行内存中的指令、或设置断点进行连续执行，代码将会执行至断点处停止。

项目代码结构：

1、文件夹：

- project: Qt 项目源代码文件
- test: 测试用例
- document: 文档
- build: 生成的二进制文件

运行时需要将 `code.txt` 代码配置文档放在程序运行目录中。

2、源代码：

由三个部分组成，其中模拟器类和汇编、反汇编 c 语言库可复用

a. C 语言实现的将 MIPS 汇编指令转换为机器码，或进行反汇编

公共头文件 `compiler.h`:

`compile.c` 实现了将输入的源代码文件转换为二进制数组；

`decompiler.c` 将一条机器码反汇编为 MIPS 汇编指令

`singleCompiler.c` 将一条 MIPS 汇编指令编译为机器码（可单独调用）

函数接口：

将输入的源代码文件转换为 32 位二进制数组；

```
int compileText(char* text);
```

二进制数组声明为全局变量，分为数据段和代码段，类型定义如下：

```
struct segment{
    int length;
    int origin;
    code* codes;
};
```

```
struct binarys {
    struct segment text;
    struct segment data;
};
```

将一条机器码反汇编为 MIPS 汇编指令：

```
code instToBinary(char* instruction);
```

返回错误信息：

```
char* getErrorMessage();
```

将一条机器码反汇编为 MIPS 汇编指令

```
int decompileCode(code c,char* buffer);
```

b. c++ 实现的模拟器类：

simulator.h

simulator.cpp

c. Qt 窗口类：

mainwindow.cpp

mainwindow.h

highlighter.cpp 代码高亮类

highlighter.h

main.h （作为程序入口文件）