

MIPS 汇编器实验报告

姓名：张 彧 学号：3160105190 专业：竺可桢学院交叉创新平台

课程名称：计算机组成 指导老师：刘海凤

实验平台：python3.6 + pyQT5

实验内容：mips 汇编器，反汇编器，模拟器（调试模式）

实现功能：

1. 基础功能

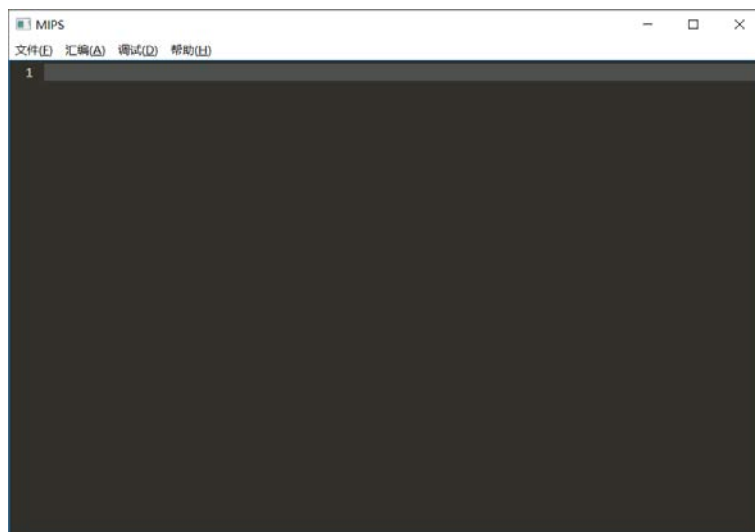
- 汇编器（支持.asm 文件和直接编辑）
- 反汇编器（支持.bin 和.coe 文件及直接编辑）

2. 拓展功能

- 伪码支持
- 图形界面（支持行号，当前行高亮，文件操作，下拉菜单，快捷键，多窗口，编辑器）
- 更多指令支持
- 调试模式（debug mode）（模拟 MIPS 程序运行，支持单步调试，一步完成，停止和重启调试，支持内存和寄存器内容显示）

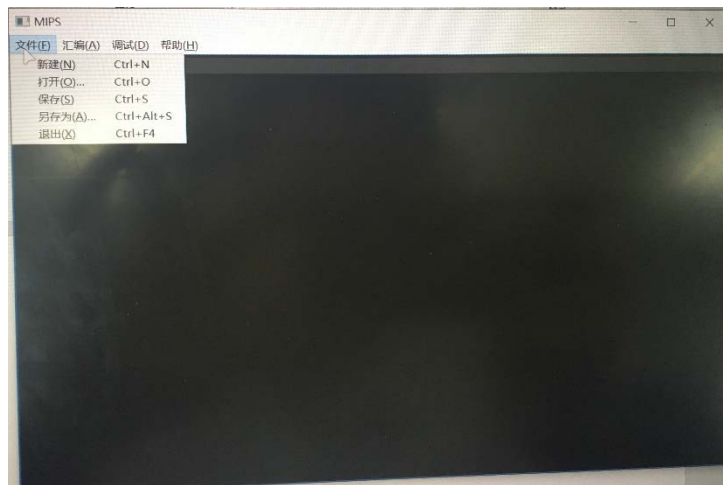
功能展示：

1. 运行主程序初始界面

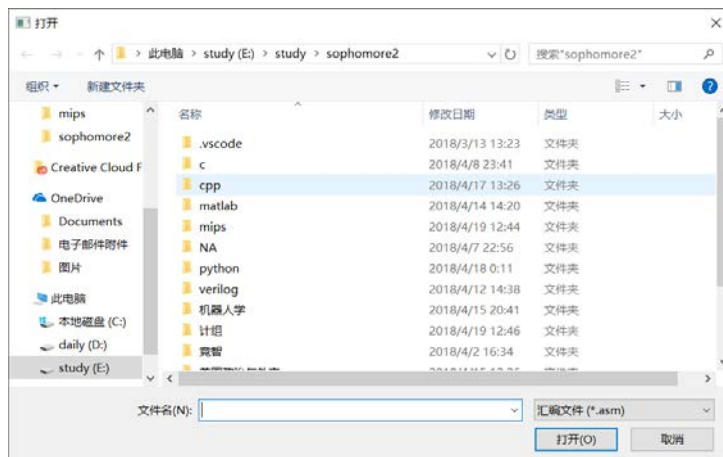


2. 下拉菜单展示，快捷键支持

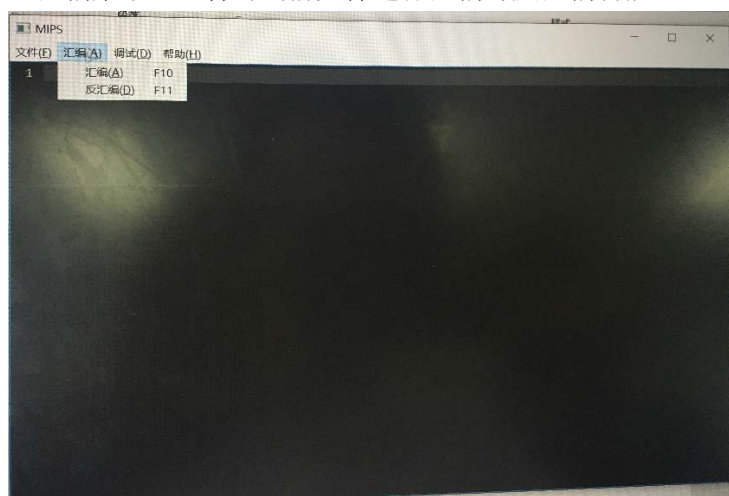
- 文件菜单，包含新建，打开，保存，另存为，和退出功能



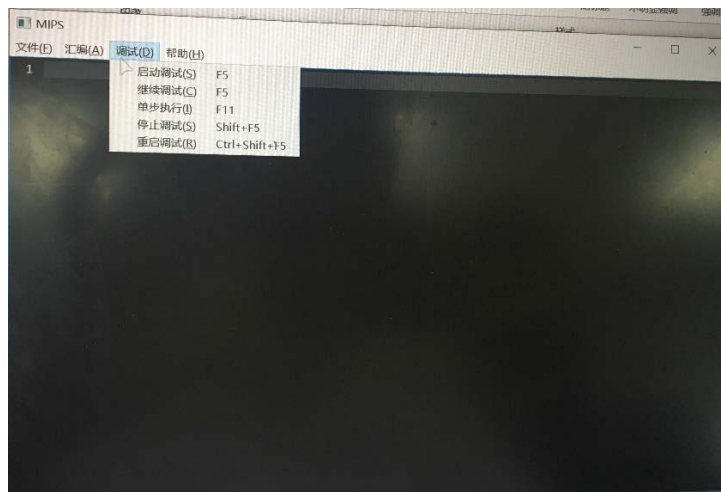
- 点击新建，则会清空当前屏幕
- 点击打开，则会进入文件选择界面



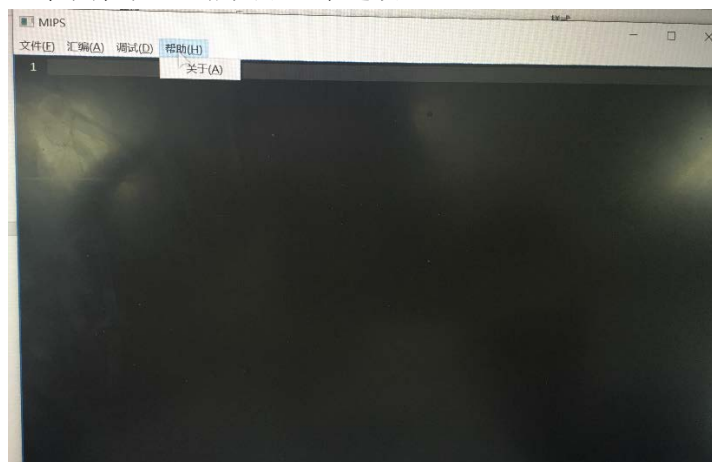
- 点击保存，则会将文件存至当前路径。
- 点击另存为，则会同样弹出文件操作界面进行保存
- 点击退出，则会退出当前程序
- 汇编菜单，包含对当前文件进行汇编与反汇编功能



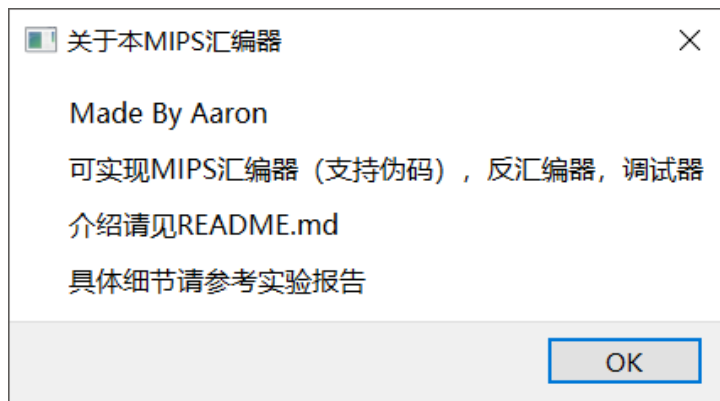
- 调试菜单，包含启动调试，继续调试，单步执行，停止调试，以及重启调试功能



- 帮助菜单，包括关于一个选项



- 点击关于会打开一个小说明界面



3. 汇编器功能展示

- 在编辑器屏幕输入代码或者打开.asm 文件之后，点击汇编，若代码正确，则可正确汇编，并将结果写入同名.coe 文件中，并显示在右侧结果栏，其中左边数字为指令地址，右边数字为指令内容，都采用 16 进制表示
- 支持 18 条基本指令及部分拓展指令（如 xor、blt 等）
- 支持用伪指令调整基地址
- 支持基本伪码
- 支持字母大小写
- 支持 '\\' 和 '#' 格式注释

The screenshot shows the MIPS IDE with the following assembly code on the left:

```

1 #baseAddr: 00000000
2 l:
3 add $t8,$s3,$v0;
4 sub $s1,$t1,$t2;
5 addi $t0,$zero,1000;
6 sw $t0,50($t3);
7 lw $s7,50($t3);
8 and $t1,$t2,$t3;
9 or $a1,$a2,$a3;
10 move $t1,$t0;
11 sll $t0,$t0,5;
12 srl $s5,$s7,5;
13 beq $t0,$s6,1;
14 bne $t2,$s7,p;
15 slt $s0,$s1,$s2;
16 nor $t1,$t0,$t3;
17 bge $t1,$t0,1;
18 blt $s3,$s2,p;
19
20 p:
21 slt $gp,$sp,$fp;
22 xor $t0,$t1,$t0;
23 jal 1000;
24 j 2000;
25 jr $ra;
26
27 Exit:

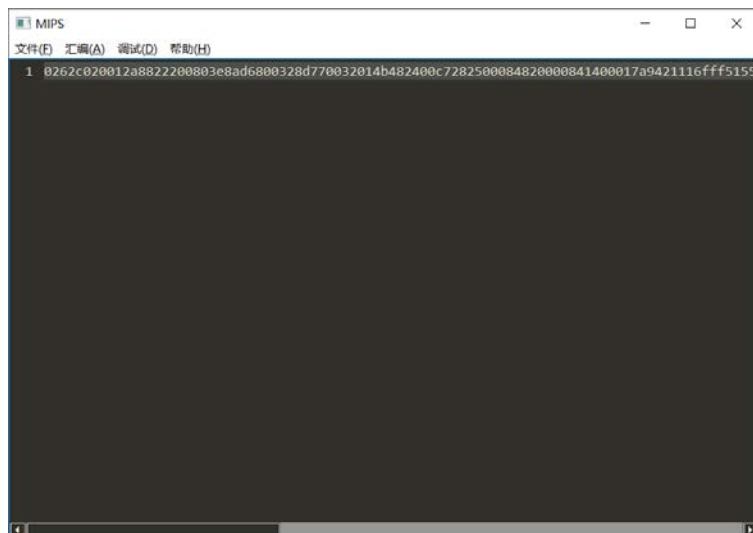
```

On the right, the '运行结果/内存' (Run Results/Memory) window shows a memory dump:

00000000	0262c020
00000004	012a8822
00000008	200003e8
0000000c	ad680032
00000010	8d770032
00000014	014b4824
00000018	00c72825
0000001c	00004820
00000020	00004140
00000024	0017a942
00000028	1116fff5
0000002c	15570004
00000030	0232802a
00000034	010b4827
00000038	01280823
0000003c	0421fff1
00000040	02720823
00000044	0420ffff
00000048	03bee02a
0000004c	01284026
00000050	0c0003e8
00000054	080007d0
00000058	03e00008

4. 反汇编器功能展示

- 从文件打开.coe 文件



- 使用反汇编操作，结果将输出到同名.asm 文件，并输出在右侧结果栏内，左边为行号，右边为代码内容，经检查和原结果一致

The screenshot shows the MIPS IDE with the disassembled assembly code on the left:

```

1 0262c020012a8822200803e8ad6800328d770032014b482400c728250008
0 add $t8,$s3,$v0
1 sub $s1,$t1,$t2
2 addi $t0,$zero,1000
3 sw $t0,50($t3)
4 lw $s7,50($t3)
5 and $t1,$t2,$t3
6 or $a1,$a2,$a3
7 add $t1,$zero,$t0
8 sll $t0,$t0,5
9 srl $s5,$s7,5
10 beq $t0,$s6,-11
11 bne $t2,$s7,4
12 slt $s0,$s1,$s2
13 nor $t1,$t0,$t3
14 subu $at,$t1,$t0
15 bgez $at,-15
16 subu $at,$s3,$s2
17 bgez $at,-1
18 slt $gp,$sp,$fp
19 xor $t0,$t1,$t0
20 jal 1000
21 j 2000
22 jr $ra

```

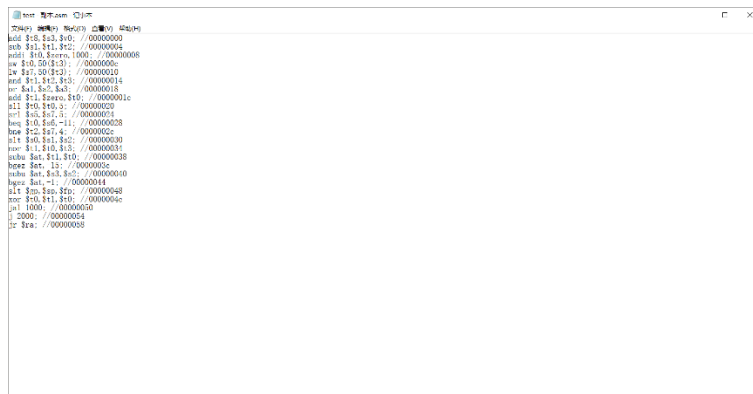
On the right, the '运行结果/内存' (Run Results/Memory) window shows the original assembly code:

```

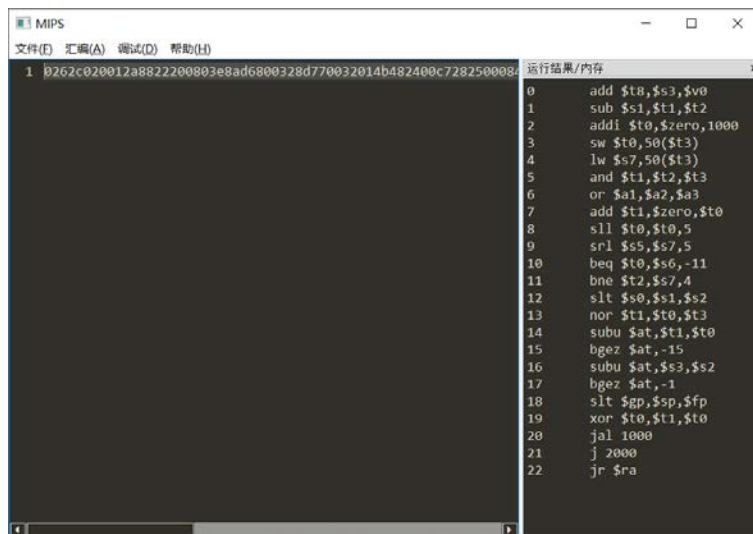
0 add $t8,$s3,$v0
1 sub $s1,$t1,$t2
2 addi $t0,$zero,1000
3 sw $t0,50($t3)
4 lw $s7,50($t3)
5 and $t1,$t2,$t3
6 or $a1,$a2,$a3
7 add $t1,$zero,$t0
8 sll $t0,$t0,5
9 srl $s5,$s7,5
10 beq $t0,$s6,-11
11 bne $t2,$s7,4
12 slt $s0,$s1,$s2
13 nor $t1,$t0,$t3
14 subu $at,$t1,$t0
15 bgez $at,-15
16 subu $at,$s3,$s2
17 bgez $at,-1
18 slt $gp,$sp,$fp
19 xor $t0,$t1,$t0
20 jal 1000
21 j 2000
22 jr $ra

```

- 查看输出的.asm 文件，包含每一行地址的注释

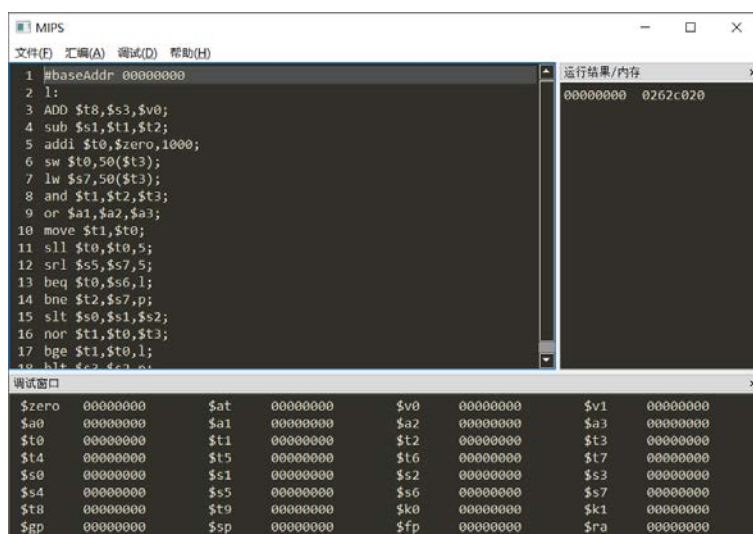


• 最终实现反汇编功能展示

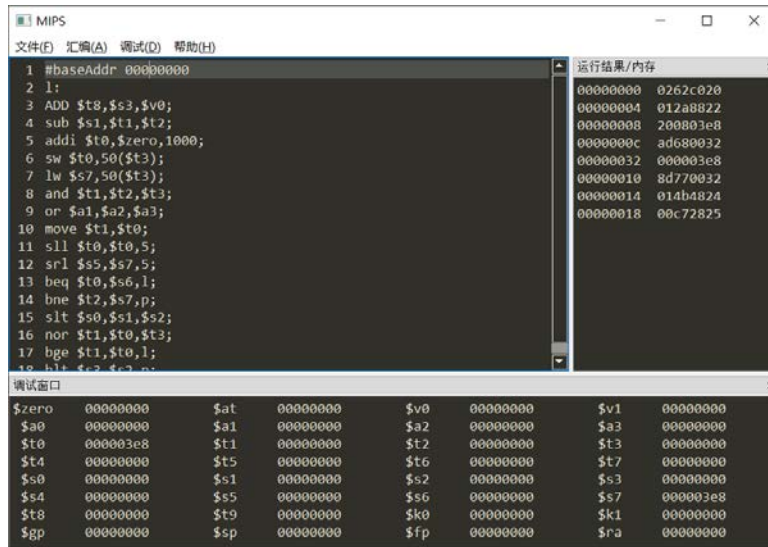


5. 调试模式功能展示

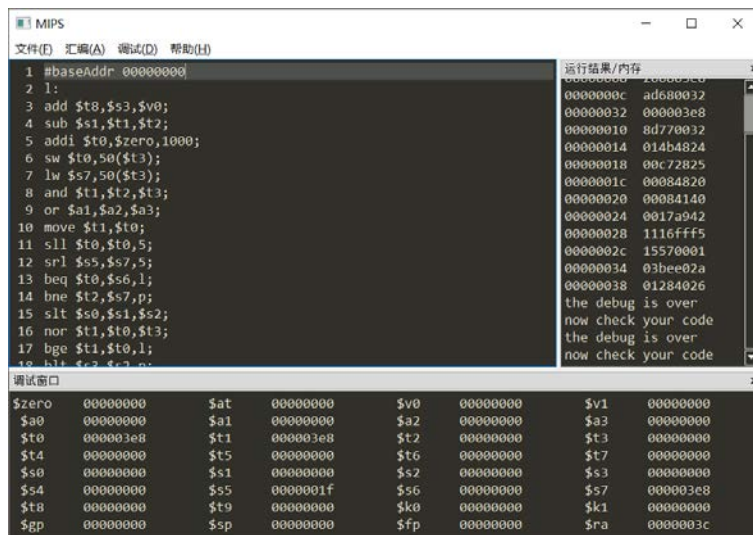
- 开始调试，则进行第一步操作，并在右侧展示内存中的值（包括写入的值和指令值）
- 支持跳转等大部分汇编实现指令
- 支持部分伪码模拟操作
- 支持内存写入和读出



- 点击单步调试，则会进入下一步操作，并继续显示，可以看到寄存器和内存内值的改变



- 点击停止调试，则调试界面隐藏，记录清空
- 点击重启则会重新开始调试
- 点击继续调试则会直接进行到最后一步。并显示调试已经完成信息在右侧窗口中



- 最终实现调试功能（debug mode）展示

