

# MIPS汇编器

- 目前做的是1.0版本，之后将继续加入更多的功能
- 实现的指令
  - 实现了对32个寄存器的使用
  - 实现了MIPS指令集中的26条R类指令
  - 实现了MIPS指令集中的20条I型指令
  - 实现了MIPS指令集中的2条J型指令
  - 实现了部分伪指令
- 程序介绍
  - 将寄存器和三类指令用结构的方式进行存储

```
struct {  
    const char *name;  
    char *addr;  
    unsigned int regNum;  
} regMap[]; // register  
  
struct {  
    const char *name;  
    char *func;  
} rMap[]; // r-Type  
  
struct {  
    const char *name;  
    char *opNum;  
} iMap[]; // i-Type  
  
struct {  
    const char *name;  
    char *opNum;  
} jMap[]; // j-Type
```

- 本程序目前为CLI形式（命令行），之后将制作图形界面进行更好地展示。实现了一条一条读取指令（这种情况不输入与标号和地址相关的指令），或者用文本形式读入指令，这两种形式的指令读取方法。

```
This is the MIPS Assembler vision 1.0.  
Please give your instruction, which will be translated into machine code.  
The result will be saved into a new file.  
Read the menu:  
0.Exit 1.Instruction 1 by 1 2.Read my file
```

- 主要实现思路：

```
int getOpCode(char * instruction); // get the operator
int getFuncCode(char * instruction); // get the function number
void getRegs_Print(const int opCode, const int funcCode, char
*operands);
char* numToBinStr(char* buf, int immed, int bits);
```

- 第一遍扫描，首先读入文本，得到每个标号所在的位置（即PC变化情况）
- 第二遍扫描，得到每一句指令所包含的operator和function，然后分多种类别查找指令包含的寄存器，最终将这一句指令的信息转化为机器码
- 将每一次转换得到的机器码读入文件，进行保存

## • 程序使用方法和示例

- 本程序有两种输入方式，

```
This is the MIPS Assembler vision 1.0.
Please give your instruction, which will be translated into machine code.
The result will be saved into a new file.
Read the menu:
0.Exit 1.Instruction 1 by 1 2.Read my file
```

分别对应以上的1和2，用户可以根据自己的需要进行选择。

程序要求的单条代码的格式是 Operator\toperands 或者 Label\tOperator\toperands 的形式

相应的解析代码为

```
sscanf(fileLine, "%s\t%s\t%s", tok0, tok1, tok2);
sscanf(fileLine, "%s\t%s", tok1, tok2);
```

- 单条指令的输入：

```
This is the MIPS Assembler vision 1.0.
Please give your instruction, which will be translated into machine code.
The result will be saved into a new file.
Read the menu:
0.Exit 1.Instruction 1 by 1 2.Read my file
1
Input one instruction:
Input "Exit" to leave.
```

这里提示用户可以选择性输入退出程序，或者输入自己需要的指令

输入指令，可得到对应的机器码，同时，该机器码会被存储到目录下的文件中

```
Input one instruction:
Input "Exit" to leave.
sw $t1,4($t0)
10101101000010010000000000000100
Please input one instruction:
```

mips\_assmbl.c  
mips\_assmbl.exe  
result.txt  
test.asm

result.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)  
10101101000010010000000000000100

- 文件类型输入

```

This is the MIPS Assembler version 1.0.
Please give your instruction, which will be translated into machine code.
The result will be saved into a new file.
Read the menu:
0.Exit 1.Instruction 1 by 1 2.Read my file
2
Please put the file under the same directory.
Input "Exit" to leave.
Input the file name:

```

用户应当将文件放在.exe的同目录下，或者输入一个绝对路径，然后可以得到该文件内所有指令的机器码，同时该机器码会被存放在同目录的文件下

```

1 lw $s4,0($s6)
2 add $t1,$s4,$t1
3 add $t1,$0,$s4
4 nor $t2,$t1,$s4
5 nor $t4,$t7,$s3
6 sll $t2,$t3,2
7 sll $t4,$t2,6
8 sll $t3,$t2,31
9 lw $s4,0($t0)
10 addi $t1,$t0,42
11 addi $t2,$t0,23
12 ori $t3,$t0,12
13 ori $t4,$t0,2
14 ori $t4,$t3,89
15 lui $s4,4533
16 lui $s2,123
17 sw $s2,0($s6)
18 sw $s4,4($s6)

```

如文件中保存如图所示的代码段，则可以得到结果

```

This is the MIPS Assembler version 1.0.
Please give your instruction, which will be translated into machine code.
The result will be saved into a new file.
Read the menu:
0.Exit 1.Instruction 1 by 1 2.Read my file
2
Please put the file under the same directory.
Input "Exit" to leave.
Input the file name:
test.asm
10001110110101000000000000000000
00000010100010010100100000100000
00000001000010000100100000100000
00000001001101000101000000100111
00000001111100110110000000100111
00000001011000000101000010000000
00000001010000000110000110000000
00000001010000000101111110000000
10001101000101000000000000000000
001000010000100100000000000101010
00100001000010100000000000010111
00110101000010110000000000001100
00110101000011000000000000000010
00110101011011000000000001011001
00111100000101000001000110110101
0011110000010010000000000111011
10101110110100100000000000000000
10101110110101000000000000000100
Bye! 请按任意键继续. . .

```

同时在结果文件中可以得到

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)