CS323 - Phase 4 Report

使用方法

使用方法：在项目的根目录 Project 4文件夹中依次键入

- `make`
- `bin/splc test_4_r0#.spl`
- 进入test文件夹找到对应生成的 `test_4_r0#.s`
- `spim -file test_4_r0#.s` 来运行生成的汇编文件

设计思路

寄存器的分配方法

由于时间有限，我们使用了最简单的寄存器分配方法，即朴素寄存器分配算法。

算法思路: 将所有的变量或临时变量都放在内存里。 如此一来，每翻译一条中间代码之前我们都需要把要用到的变量先加载到寄存器中，得到该代 码的计算结果之后又需要将结果写回内存。

通过在每一次函数开始调用和函数结束调用时，分别插入预设的Calling与Return对应的MIPS32代码段，从而实现朴素寄存器分配算法。

如果时间充裕，会考虑更好的方法，如图染色算法

```
static constexpr const auto function_begin =
        R"(
    addi $sp,$sp,-32
    sw $s0,0($sp)
    sw $s1,4($sp)
    sw $s2,8($sp)
    sw $s3,12($sp)
    sw $s4,16($sp)
    sw $s5,20($sp)
    sw $s6,24($sp)
    sw $s7,28($sp))";
static constexpr const auto function_end =
        R"(
    lw $s7,28($sp)
    lw $s6,24($sp)
    lw $s5,20($sp)
    lw $s4,16($sp)
    lw $s3,12($sp)
    lw $s2,8($sp)
    lw $s1,4($sp)
```

```
        lw $s0,0($sp)
        addi $sp,$sp,32)";
```

指令选择

- 完全参考了Prof. Liu 发布文档上的映射关系实现

Table 4: An example mapping between TAC and MIPS32

three-address-code	MIPS32 instruction
x := #k	li reg(x), k
x := y	move reg(x), reg(y)
x := y + #k	addi reg(x), reg(y), k
x := y + z	add reg(x), reg(y), reg(z)
x := y - #k	addi reg(x), reg(y), -k
x := y - z	sub reg(x), reg(y), reg(z)
x := y * z	mul reg(x), reg(y), reg(z)
x := y / z	div reg(y), reg(z) mflo reg(x)
x := *y	lw reg(x), 0(reg(y))
*x := y	sw reg(y), 0(reg(x))
GOTO x	j x
x := CALL f	jal f move reg(x), $v0
RETURN x	move $v0, reg(x) jr $ra
IF x <y GOTO z	blt reg(x), reg(y), z
IF x <= y GOTO z	ble reg(x), reg(y), z
IF x >y GOTO z	bgt reg(x), reg(y), z
IF x >= y GOTO z	bge reg(x), reg(y), z
IF x != y GOTO z	bne reg(x), reg(y), z
IF x == y GOTO z	beq reg(x), reg(y), z

其他设计细节

- 通过输出重定向将标准流输出的MIPS32程序重定向到目标.s文件中
- 在Phase 3已经优化过的TAC指令的基础上生成MIPS32指令
- 在MIPS32内存分配的设计上存在优化的空间，汇编程序中存在一些无用的内存读取代码，是我们优化的方向

```
sw $t0,_v41
lw $t0,24($sp)
sw $t0,_v39
lw $t0,28($sp)
sw $t0,_v38
lw $t0,32($sp)
sw $t0,_v33
lw $t0,36($sp)
sw $t0,_v29
lw $t0,40($sp)
sw $t0,_v21
lw $t0,44($sp)
sw $t0,_v13
lw $t0,48($sp)
sw $t0,_v35
lw $t0,52($sp)
sw $t0,_v25
lw $t0,56($sp)
sw $t0,_v14
lw $t0,60($sp)
sw $t0,_v16
```

测试结果

- Case_1:
  - ```
    ➜  Phase 4 git:(main) ✗ spim -file test/test_4_r01.s
    SPIM Version 8.0 of January 8, 2010
    Copyright 1990-2010, James R. Larus.
    All Rights Reserved.
    See the file README for a full copyright notice.
    Loaded: /usr/lib/spim/exceptions.s
    10003
    10002
    30002
    10003
    20001
    20003
    10003
    ```
- Case_2
  - ```
    ➜  Phase 4 git:(main) ✗ spim -file test/test_4_r02.s
    SPIM Version 8.0 of January 8, 2010
    Copyright 1990-2010, James R. Larus.
    All Rights Reserved.
    See the file README for a full copyright notice.
    Loaded: /usr/lib/spim/exceptions.s
    Please enter an integer: 323
    1
    ```
- Case_3

```
➜  Phase 4 git:(main) ✗ spim -file test/test_4_r03.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
19
```

感谢阅读!

ref:

- 南方科技大学编译原理第四次课程项目要求报告
- 南京大学编译原理第五次课程项目要求报告
- 学长学姐在Github上的经验仓库