Step 3 Report

刘轩奇 2018011025

2020年10月5日

1 工作内容

本人选择不使用辅助工具 ANTLR 因此自己实现了 lexer 和 parser。

1.1 文件说明

`montLexer.h/cpp` 词法分析器；

`montParser.h/cpp` 语法分析器；

`montConceiver.h/cpp` 产生中间代码；

`montAssembler.h/cpp` 从中间代码产生汇编代码；

`montLog.h` 记录编译错误信息；

`montCompiler.cpp` MiniDecaf 编译器，包含主函数，编译成功返回 0 否则返回 -1，并将错误信息输出到 `std::cerr`。

1.2 本步骤完成的工作

1 词法分析 添加了五种 Token，类型分别为 `Plus` 加号， `Asterisk` 星号即乘号， `LSlash` 左斜杠即除号和 `Percent` 百分号即取模符号。

2 句法分析 添加了 `additive`, `multiplicative`, `primary` 类型的树节点。其生成逻辑按照指示书上给出的指导生成。即 `additive` 不断生成新的 `multiplicative` 直到不再出现加减号； `multiplicative` 不断生成 `unary` 直到不再出现乘除模号； `primary` 检测括号和单个值。

3 产生中间代码 按照指导书上的说明，将对应符号分别生成对应的中间代码：`ADD`, `SUB`, `MUL`, `DIV`, `REM`。

4 产生汇编代码 按照指导书的说明，将对应中间代码转换为对应汇编语句，使用已有的汇编器编译得到的结果在一些操作上并不能看出使用了哪些汇编命令，例如乘法的汇编是调用了一个内置函数。于是参考了C++参考实现中的代码，了解到这五种运算对应的操作实际上就是五种汇编命令 `add`, `sub`, `mul`, `div`, `rem`。

2 思考题

1 请给出将寄存器 `t0` 中的数值压入栈中所需的 riscv 汇编指令序列；请给出将栈顶的数值弹出到寄存器 `t0` 中所需的 riscv 汇编指令序列。

答 `addi sp, sp, -4; sw t0, 0(sp);`

2 语义规范中规定“除以零、模零都是未定义行为”，但是即使除法的右操作数不是 0，仍然可能存在未定义行为。请问这时除法的左操作数和右操作数分别是什么？请将这时除法的左操作数和右操作数填入下面的代码中，分别在你的电脑（请标明你的电脑的架构，比如 x86-64 或 ARM）中和 RISCV-32 的 qemu 模拟器中编译运行下面的代码，并给出运行结果。（编译时请不要开启任何编译优化）

```
#include <stdio.h>
int main() {
  int a = Left operand;
  int b = Right operand;
  printf("%d\n", a / b);
  return 0;
}
```

答 只要使得算术溢出即可。取 `a = 0x80000000; b = -1;` 在本机（x86-64 Windows 10）上运行，进程将阻塞，不会给出任何输出。在 QEMU 上执行的结果是 `0x80000000 = -2147483647`。