

# Step 10 Report

刘轩奇 2018011025

2020年10月8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 没有改变。

2 句法分析 添加或修改了以下的 AST 节点：

- \* `function` 内容可以包括函数和全局变量声明两种。

- \* `globdecl` 表示全局变量声明，可以有默认初始值或指定初始值的形式。

3 产生中间代码 在此阶段，遍历语法树过程中记录所定义的全局变量，同时查看是否与已经定义的全局变量或函数重名。每次遇到表达式中的变量调用，首先查看栈中是否包含该变量，若不包含则在全局变量中查找该变量。另外添加了指导书中所指示的一种中间代码，即 `GLOBADDR`（指导书上为 `GLOBALADDR`）。

4 产生汇编代码 除了读取中间代码的执行代码段输出到 `.text` 以外，还要读取语义分析中所存储的全局变量，输出到 `.data`，`.bss` 段中。`GLOBADDR` 所生成的代码与实验指导书上相同，不再赘述。

## 2 思考题

1 请给出将全局变量 `a` 的值读到寄存器 `t0` 所需的 `riscv` 指令序列。

答  $\ln t_0, a$