

# lab4实验报告

151242060 银加

## 实验环境及编译方法

---

我使用的是MacOS系统，我在实验的根目录下创建了Makefile文件。在测试时可以在根目录下执行 `make all` 即可完成编译并且生成parser语法分析程序。需要注意的是，与实验指导稍有区别的是在mac上 `-lfl` 需要替换成 `-ll` 才能编译成功，所以在测试时可能需要根据相应操作系统对Makefile进行修改。

进行测试时，可以执行 `make test1`，`make test2` 对测试样例分别进行测试。所有样例均来自实验指导。如果需要添加新的测试样例，可以在根目录下test文件夹中添加新的样例，然后在Makefile中添加相应语句或直接执行即可。

## 实现功能说明

---

我实现了本次实验的所有功能，并且生成的目标代码可以在SPIM Simulator上正确运行。

### 文件结构

在lab2的基础上新增了以下两个文件：

- `objcode.h`，`objcode.c` 主要实现了与目标代码生成相关的函数。
- 其他文件除了 `main.c` 均没有做修改。

### 主要功能实现方法

- **指令选择：**  
因为我在实现中间代码生成时使用的是线性IR，所以这里要做的就是逐条翻译，详细翻译方案实验指导中已经给出，这里不再赘述。
- **寄存器选择：**  
我使用的局部寄存器分配方案是如果存在空闲寄存器则选择空闲寄存器，否则替换掉最“老”的寄存器，即从存入新的值到当前经历时间最长的寄存器，具体实现方式是每个寄存器保存一个 `old` 属性（寄存器数据结构为 `_reg_t`，在 `objcode.h` 中），更新方式是寄存器中的值被替换或从没存放值到存放值时 `old` 属性清零，其他情况下随着扫描进行 `old` 值增加，需要替换时找到 `old` 值最大的即可。

## 实验总结

---

本次实验主要实现了目标代码生成，将课堂中学到的关于指令选择和寄存器选择的方法付诸实际，并且了解

了关于MIPS32指令的相关知识，收获颇丰。