

# L5 目标代码生成 实验报告

181250015 陈彦泽

## 1 实现功能

将L4生成的中间代码转为目标代码，并在SPIM Silumator中运行

主要分为三个模块

- 指令选择
- 寄存器分配
- 栈管理

## 2 实现方法

### 2.1 指令选择

在L4中，中间代码结构保存在线性链表中，因此只需要根据实验讲义上的规则，逐条将中间代码对应到目标代码上即可

### 2.2 寄存器分配

采用**朴素寄存器分配算法**：最简单，也最低效，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

在MIPS中，除了load/store型指令之外，其余指令的所有操作数都必须来自寄存器而不是内存，因此，需要解决立即数的乘除法问题，解决思路为：先将立即数装进寄存器中（这里选择了s2作为寄存器），然后再正常乘除法

### 2.3 栈管理

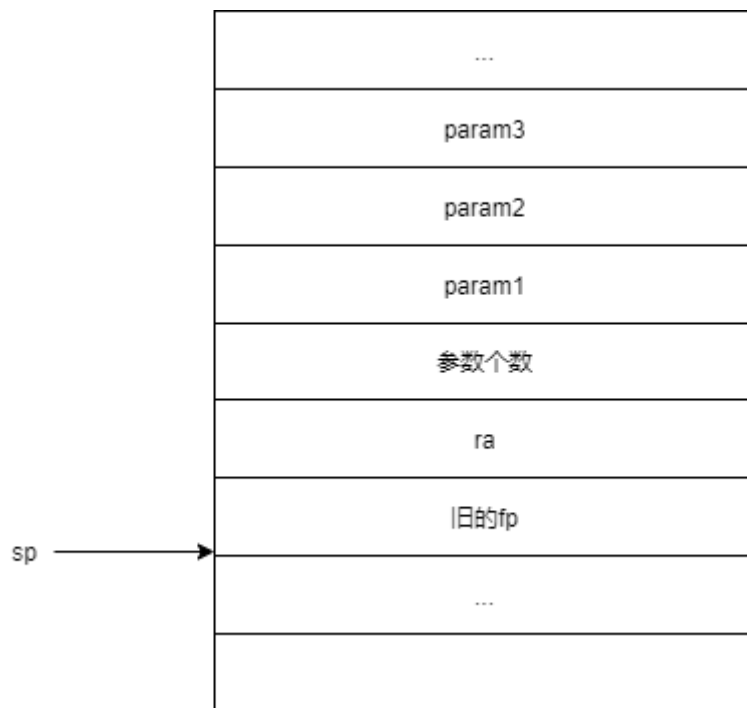
主要解决的是函数调用和数组空间分配的问题

所以变量都作为栈中临时变量处理，在Var中加入了spOffset确定相对于fp的偏移；

- **函数调用**

为了简化实现，并没有采用实验手册中的规则，而是将所有的参数都通过栈来传递

设计栈结构如下



- 调用时:

ARG、压入参数个数、保存ra、保存旧的fp

设新fp=sp、sp=sp-100

- 返回时:

取旧的fp、取旧的ra、sp+=100、取参数个数、sp+=参数个数\*4

- 数组空间分配**

假设数组长度为3，栈分配结构如下

