

# 编译原理实验4机器代码生成实验报告

201840009田永上

## 实验内容

将实验三中得到的中间代码经过与具体体系结构相关的指令选择、寄存器选择以及栈管理之后，转换为MIPS32汇编代码

实现：遍历每一行中间代码，然后按代码类型去生成机器代码，采用朴素寄存器分配方法，虽然不高效但是很简单。

## 核心代码设计

```
struct Varoff_  
{  
    char name[50];  
    int reg_num;  
    int offset;  
    Operand op;  
};  
struct Offlist_  
{  
    Varoff var;  
    Offlist next;  
};  
struct Register_  
{  
    bool if_used;  
    char *name;  
    Varoff var;  
}; // registers
```

总共额外设计了三个结构体辅助产生机器代码，register类用于存储寄存器信息，varoff类用于存储变量地址偏移量，使用哪个寄存器以及对应operand等信息。Offlist是一个链表，主要用来查找有没有和即将分配的内容相同的已分配内容，避免重复分配带来复杂操作，再预处理的过程中该链表可以记录预分配offset值，提前设计好空间分配。

关于地址，我仿照x86地址管理方式进行函数调用和栈帧管理，也是比较顺利。

关于每一行中间代码如何生成对应机器代码，我对照实验3讲义中比作样例的中间代码和实验4讲义中的机器代码，再结合一个比较简单的对照的例子提取出了各种类型中间代码对应生成的机器代码，某些需要特殊变量的，放置在全局，然后在需要使用的几个代码中进行维护即可。

## 遇到的困难

包括但不限于实验3的遗留问题，以及对地址偏移量没理解清楚，导致地址有重合等，debug过程相比于实验3要轻松太多了。

qtspim非常难用，一不小心把console关闭后甚至不知道在哪里开启，运行一次代码后又得重启qtspim才能运行另外的汇编代码。

