

# lab4实验报告

---

这次的内容比前几次lab简单一些，我们需要新实现一个模块，根据上一次生成的中间代码IR，进行翻译，输出生成的MIPS32机器代码，写入指定文件

定义数组 `varOffset` 用于存储变量的偏移量， `Regs` 用于记录寄存器的占用情况

首先进行一些初始化，打开输出文件，并写入机器代码的头部

```
init varOffset and Regs
fopen
fprintf(f, ".data ... .text ... read: ... write: ...")
while(p){...}
fclose
```

接着遍历中间代码链表，根据不同的中间代码类型调用相应的处理函数

```
while (p){
    switch (p->code.kind){
        case FUNCTION:
            ...
        case LABEL:
            ...
        case GOTO:
            ...
        case ...
            ...
    }
    p = p-> next
}
```

主要部分就是一直在分情况讨论

注意在处理函数调用和返回时，需要计算函数内所有变量和临时变量的偏移量，并调整栈指针

在处理 `read` 和 `write` 函数时，需要使用系统调用，确保调用号和参数正确