

实验四报告

一、实现

- 指令选择机制：线性IR
 - 根据表11，逐条将中间代码对应到目标代码
- 寄存器分配算法：局部寄存器分配算法
 - 将代码分成基本块
 - 每个基本块内，对中间代码逐条扫描，若需要使用寄存器
 - 存在空闲寄存器，则直接分配
 - 不存在空闲寄存器，选择本块内将来用不到或最久以后才用到的变量的寄存器，将其内容写回内存
- 参数传递：
 - 参数小于等于4个，使用\$a0至\$a3这四个寄存器传递
 - 参数多于4个，前四个放在\$a0至\$a3这四个寄存器，剩下的依次压到栈里
 - 返回值放到\$v0
- 栈管理：
 - \$sp指向栈顶，\$fp指向活动记录底部
 - 寄存器保存策略：\$t0至\$t9由调用者负责保存，而\$s0~\$s8由被调用者负责保存
 - 调用者的过程调用：
 - 调用前，将保存活跃变量的所有调用者保存寄存器写入栈中，将参数传入寄存器或者栈
 - 函数调用之后，将之前保存的内容从栈中恢复
 - 被调用者的过程调用：
 - 函数开头，若函数内调用其他函数，则将\$ra压栈；若用到\$fp则将其压栈并设置好新的\$fp；将本函数内要用到的被调用者保存寄存器压栈，将形参取出
 - 函数结尾，将函数开头保存的寄存器恢复

二、编译

```
1 | Code文件夹下  make
```