编译原理实验 4

孙伟杰 191250127

2022 年 6 月 9 日

1 项目进度

完成了所有文档中的必做和选做要求

2 使用方法

使用方式即在 Code/路径下在 terminal 中输入 make parser，然后./parser + 需要运行的文件名 + 输出汇编代码的文件名即可

3 实现方法

临时变量的地址采用最朴素的分配方式，在中间代码中中每有一个变量就在栈中为其留出位置。在遇到 IR 中的函数定义时，先建立类似于语法分析中的符号表，遍历代码记录每个出现的变量并且给其中每个变量分配距离栈的 offset，直到计算出函数的栈大小，函数内的临时变量在保留好 $fp 和 $ra 的地址的情况下从-12$(fp) 开始依次向小地址部分延展

采用了朴素寄存器分配方法，总共用了 t_0 到 t_7，8 个寄存器，在每次写入寄存器的操作后都会将寄存器的内容 spill 到内存中，这样就不用考虑保存寄存器的问题

在实现传参和函数调用时，将 $ra 和 $fp 都设成是被调用者保存，在每次调用前压栈的顺序是，先将函数参数按照顺序压栈，在函数开始时将 $ra

和 $fp 压栈，在 return 时将 $fp 和 $ra 的值恢复。
但是由于 write 和 read 函数并没有进行 $ra 和 $fp 的保存，所以在对 write 和 read 函数进行调用时，我选择将 $ra,$fp 在调用前压栈然后再调用后由调用者恢复

4 参考资料

感谢 wzj 同学提供的测试代码