

191098328 计算机科学与技术系 张世茂

邮箱: 191098328@smail.nju.edu.cn

在本次实验中,我在之前三次实验中进行词法分析、语法分析、语义分析和已经生成的中间代码程序的基础上进一步实现翻译方案,整个程序实现了将C—源代码程序按照一定的规则转化为线性中间代码,并将中间代码进一步转换为在SPIM Simulator上可运行的MIPS32指令序列。最终的指令序列输出到指定的.s文件中,并可以在SPIM Simulator中运行得到正确的结果。本次提交实现了全部的实验要求,且本次实验不包含选做内容。

由于在前三次实验中已经可以实现将C—源代码转换为对应的固定形式的中间代码,因此本次实验的主要任务其实就是将中间代码转换为MIPS32指令。而实验三中生成的中间代码是采用线性链表存储,则只需要通过对链表进行线性遍历,并对遍历过程中遇到的每一条中间代码采用特定的模式进行翻译即可。在本次实验中我采用的方法是通过条件分支语句对实验三中定义的每种类型的中间代码分别定义翻译模式,并逐句进行独立的翻译。将翻译得到的指令序列同样采用链表进行存储,在遍历完成后同时也完成了代码的翻译,然后将代码进行输出。

本次实验中在分配变量的存储时,优先使用空闲的空寄存器,实验中的寄存器采用结构体类型的变量进行表示,包含占用信息used、数据类型信息、数据为VARIABLE或TEMP类型时需要存储的编号信息。当存储器占满后,则需要额外考虑数据存储时向栈中的溢出情况,我是用一个独立的数据结构,即创建了一个新的结构体类型并使用该类型的链表存储变量及其在栈中相对于栈底指针\$fp的偏移量的对应关系。因为本次实验中约定不出现全局变量的使用,这就意味着实际的代码是以定义的函数为一个个模块的,且不同函数之间不会跨函数地使用同一个变量名指向的变量,这为存储变量的偏移量提供了方便,只需要考虑在当前函数的活动记录中的局部偏移量即可。实验中我设置了一个名为LocalOffset的全局变量进行当前活动记录中偏移量的记录,这一变量每当遇到中间代码中代表函数定义的语句时就进行一次初始化。在遇到main函数时,因为将\$fp和\$sp初始设置为一样的值,因此LocalOffset的初始值也设为0;在遇到其他函数时,由于涉及到调用者的活动记录保存,因此此时\$sp所指示的栈位置相对于\$fp所指示位置的偏移量为-8,则LocalOffset的初始值设为-8。在接下来每遇到一个变量时先查找是否已存储,若未存储则优先分配空寄存器,若没有空寄存器则在栈中分配空间,并对\$sp和LocalOffset进行同步更新。

还有一个特殊的地方在于,由于用于函数传递参数的寄存器只有四个,因此在参数多于四个时需要向栈中进行存放,但由于ARG和PARAM的顺序是相反的,因此存储时按照一个

顺序存储，取参数时就要按一个相反的方向来取。同时，由于可能有部分函数参数存储在栈里，参数数量多于四个时要么先向栈里存，要么先从栈里取，因此要对参数数量大于4和不大于4的情况进行分别处理。考虑到ARG语句和PARAM语句都是连续出现的，因此这就要求一次性用While语句统计处理完所有的相关语句。

数组类型需要在指令序列的开头.data部分进行预分配空间，这可以通过对中间代码进行遍历实现，每当遇到DEC的中间代码语句，就根据DEC里存储的名称和大小信息增加相应的空间分配的语句。这一操作在对指令序列进行初始化时进行。

本次实验为方便实现，对一些需要重复进行的操作进行了有益的封装。同时，由于本次实验需要建立在之前三次实验的基础上进行，之前几次代码实现的相对比较正确，且提供了很多可供使用的接口方法，相关数据结构里比较完整地存储了需要用到的相关数值，为本次实验的实验提供了不少方便。总的来看，这次实验算得上是四次实验里最水到渠成的一次，整个编译的流程也打通明晰了。

本次实验是《编译原理》课程的最后一次实验，实现了一个完整的可以将C—源程序翻译为MIPS32指令序列的程序。真正从零开始实现一个能跑起来的编译器，虽然简陋但工程量并不小，也是一件很有成就感的事。课上学的很多知识听起来容易，真正做起来才发现有很多工程上的细节需要考虑，同时在做实验的过程中，通过反复揣摩查看相关知识，相较于原先片段式的学习对整个编译器工作的流程有了一个更加整体全面的认识。最后也感谢老师和助教学长学姐们在设计和批改实验时付出的辛苦。

编译运行该程序的方法为：

1. 将文件解压后进入到Code文件夹下；
2. 在终端中输入make clean命令并回车，防止有冗余残留的多余文件；
3. 在终端中输入make命令并回车，对程序进行编译；
4. 生成了最终的可执行文件；
5. 在Linux命令行下运行./可执行文件名 源代码文件名.cmm xxx.s即可将指定的源代码文件中的C—源代码对应的MIPS32指令序列写入当前目录下名为xxx.s的文件中