

编译原理部分实验报告

郭昉泽 马强 王倩

实验目标

修改Decaf/Mind，使之能够编译出在MIPS32指令系统的一个子集上以及 μ core系统下成功运行的程序。

实验环境

支持MIPS32指令系统子集的流水CPU，主频可达到50MHz

能够在上述CPU上运行的操作系统 μ core

Decaf/Mind编译器源码

实现方法

指令集的修改

将add、sub、neg三条指令替换为addu、subu和negu，这样做的后果是编译得到的程序不会因为溢出而引发异常，但在正常运行情况下不会有任何影响。

将mul指令替换为连续的两条mult指令和mflo指令，修改方法涉及翻译、TAC代码生成、基本块划分、寄存器分配和汇编代码生成五个阶段。首先将Translator类中的genMul函数中将mul指令替换为两条指令，其次在Tac类中的Kind枚举类型中加入MULT和MFLO，在TAC类中加入成员函数genMult及genMflo，在BasicBlock类computeDefAndLiveUse函数中加入对MULT和MFLO的处理，在RegisterAllocator类中加入对MULT和MFLO的处理，最后再Mips类genAsmForBB中加入对MULT和MFLO指令的生成。以上基础修改都可借鉴其他指令的处理方法。

由于我们的CPU并不支持div、rem指令且没有独立的除法器，所以此处采用的方法是这两条指令翻译为库函数调用，然后在库函数中使用其他指令以及加减交替法获得商和余数。方法是在TransPass2类visitBinary函数中将DIV和REM分别翻译成函数调用，而在Intrinsic类中加入对decaf_div和decaf_rem函数的调用。

函数调用方式的修改

由于我们的 μ core系统默认传参方式是通过寄存器传参，而Decaf/Mind默认通过栈传递参数。我们的解决办法是在原来的每个库函数嵌套在对应的一个新的函数中，该函数的任务是将参数从栈里面读出来然后依次放入a0-a3寄存器当中，再调用原来的对应库函数即可。

新增库函数

使用C语言编写了PrintString、PrintInt、PrintBool、ReadLine、ReadInteger、Halt、Allocate、Div和Rem九个库函数，按照上一个项目中的方法，只需将decaf生成的汇编程序与上面两个文件进行链接即可得到对应的二进制可执行文件。

改变程序的退出方式

μ core操作系统默认程序退出时应返回0，反则视作出错处理。而Decaf的main函数是void类型，没有返回值。我们的处理方法是在main函数推出前加一句"move v0, \$0"语句。具体实现起来是在汇编代码生成部分Mips类的emitTrace函数中增加一个判断，如果当前退出函数名为main则在退出恢复栈帧指针时增加上述语句即可。

实验结果

修改后的Decaf/Mind编译器能够成功编译运行三个测试程序math、fibonacci和blackjack

注：具体修改详见decaf_modified文件夹下