编译原理 stage-5 实验报告

计算机系 2019011312 姚建竹

完成的工作

前端

ply_parser.py

1. 增加了dimension的处理，记录数组的各个维度值
2. 增加了数组声明的处理，使用dimension记录的信息，获得数组的类型ArrayType，并赋给declaration
3. 在postfix中加入数组的处理

tree.py

在postfix里面，注明一个postfix是否是数组，以及数组的类型

namer.py

对于postfix，分别处理ident和exprList。同时也要在这里检查数组的维度个数与postfix的维度个数是否一致
在namer中可以标记一个identifier是否是全局变量。

tacinstr.py

1. 加入指令Alloc，包含dst和分配的字节数
2. 加入指令Store，包含src, dst, offset和symbol，其中src和dst其实都应该属于src类别的Temp或寄存器

tacgen.py

1. 在visitAssignment里面处理数组的赋值情况
2. 在visitPostfix中处理数组的情况。对于数组的偏移，我维护了两个list，分别是数组每个维度的大小，和每个索引的代表的Temp。首先根据不同的Temp生成对应的子偏移量，然后再将其相加即可。然后通过获得数组的symbol的基地址，加上偏移量，就获得数组元素的目标地址。

3. 对于数组访问是否合法需要进行判断

后端

bruteregalloc.py

在处理不同的bb时，由于要将其他bb使用的变量存到栈里，同时也要在栈上给局部数组提供空间，所以要记录一共给局部声明的数组提供了多少空间。然后在riscvsubemitter存变量的时候使用到这个偏移量。

riscvasmemitter.py

加入store, alloc两个指令的处理

riscv.py

在Binary时，由于equ, geq, neq, leq, neq, lor需要同时使用到两个寄存器的值，所以需要将其中一个寄存器暂存到栈上，使用完再reload回去。

遇到的问题

1. 我发现自己的全局变量处理的有问题，无法赋值，能通过stage-4的测试却不能通过stage-5的测试，所以进行了修改
2. 数组的偏移量我忘记使用4字节为单位

思考题

C 语言规范规定，允许局部变量是可变长度的数组（Variable Length Array，VLA），在我们的实验中为了简化，选择不支持它。请你简要回答，如果我们决定支持一维的可变长度的数组(即允许类似 int n = 5; int a[n]; 这种，但仍然不允许类似 int n = ...; int m = ...; int a[n][m]; 这种)，而且要求数组仍然保存在栈上（即不允许用堆上的动态内存申请，如malloc等来实现它），应该在现有的实现基础上做出那些改动?

1. 增加一种类型，可变长度数组的类型，保存了类型和个数。既然只需要支持一维可变长度的数组，就不考虑递归定义了
2. 在declaration的时候，先visit可变长度数组类型的expression，计算那个Temp的值应该是多少，然后生成TAC的时候就直接生成类似于_T1 = ALLOC _T0这样的指令。
3. 数组的长度是n，所以在每次访问postfix的时候还要检查数组访问的长度是否匹配。
4. 在生成目标代码时，对于上面那个alloc指令，假设t0分给了_T0，t1分给了_T1，那么可以先将sp - t0分给t0，然后将t0分给t1和sp。然后再进行栈偏移的种种保存工作，即可

以实现。