

lab3实验报告

201180087 陈泰霖

201180087@smail.nju.edu.cn

任务号: 2

实现功能

支持生成IR, 完成选作1.

如何编译

用提供的makefile, 使用 `$ make` 指令即可编译。

实现中的细节

采用模块化设计, 完全不依赖于lab2中的内容, 但也因此需要额外的符号表等来记录变量名, 数组元素大小, 结构体大小及其域的偏移。

按照顺序将变量重命名为 `vi`, 临时变量命名为 `ti`, 标签命名为 `li`, 除main以外函数命名为 `fi`。变量表按原名存着变量别名, 数组元素大小表按照别名记录对应元素大小, 结构体按照结构体名记录其大小, 域按其域名记录其偏移。

ir的存法

采用双向链表的方式, 按照树形结构, 每一个子树根的ir代码对应其所有儿子的ir代码的拼接。这样我从root节点递归的去翻译整棵树, 就能在root节点上得到整个程序的ir代码。

数组变量以及结构体的存法

由于DEC x命令只是声明变量x占据的空间, 而不会将其所占空间的首地址存在x中。因此我们要为数组x或结构体变量x申请空间时, 先为临时变量t申请空间, 然后将t的地址赋给x。这样就得到了符合我理解中的C语言语义。

结构体参数

实际上时选做2中出现的问题, 并且当直接传一个结构体变量时, 由于我们变量中存的是地址, 我们直接按照其他值传递的方式传递就是引用传递。但当我们要传的参数是一个结构体数组中的一员时, 由于之前在翻译表达式时, 如果不考虑选做, 我们会把数组表达式对应地址的值做拷贝, 即 `t1=*t2`, 因为数组永远是一维的简单数组; 而考虑选作1后, 数组的成员可能是结构体, 这时候我们就如果再直接解引用, 就会在传递结构体参数时出错 (在别的地方由于结构体不会直接出现再等号两边, 所以不会出错)。为此我们在做数组解引用时, 若数组为结构体数组 (判断一下数组元素大小), 则我们不直接解引用, 而是将地址直接传递, 即修改为 `t1 = t2`。

手册外的翻译模式

实际上手册未提到的产生式的翻译模式大多都不产生ir代码。但其大多也都各有作用，如Dec系列可能会产生申请空间的代码，Specifier会为Dec提供类型大小，还会为声明的结构体和其域保存信息。

相较于上个实验的进步

上个实验写的一坨屎（指写法），写了一个超长的switch来进行检查。这次吸取教训，选择将其分开成为各个函数，使得基本上大部分语法单元都有其对应的翻译函数，并且各类初始化ir节点，获取新的临时变量名等操作都写成独立的函数。这样虽然函数个数大幅增加，但平均长度大幅降低，其可读性和可维护性大大提升。

检测错误

因为会对另外选做进行检查，而对于这部分我们不应该产生ir文件。我在翻译时，会考虑一些不符合的状况，出现这些情况时，不会停止翻译，但最后的结果不会输出到文件中。如果在考虑的状况外的时候，很有可能会发生段错误，这时也不会产生文件。