目录

编译原理实验 3 报告--石卓凡 120L021011

A） 程序实现了哪些功能？简要说明如何实现这些功能。

1） 将 C--源代码翻译为中间代码

设置全局变量表，全局标签编号

通过全局变量表，在每定义一个变量或者声明一个参数时候，利用全局变量编号 globalVarId 记录并分配唯一的变量编号，再并记录在全局变量表 variables

维护全局标签编号 globalLabelId 用于后续分配唯一的标签编号

与下文的两条函数配合使用

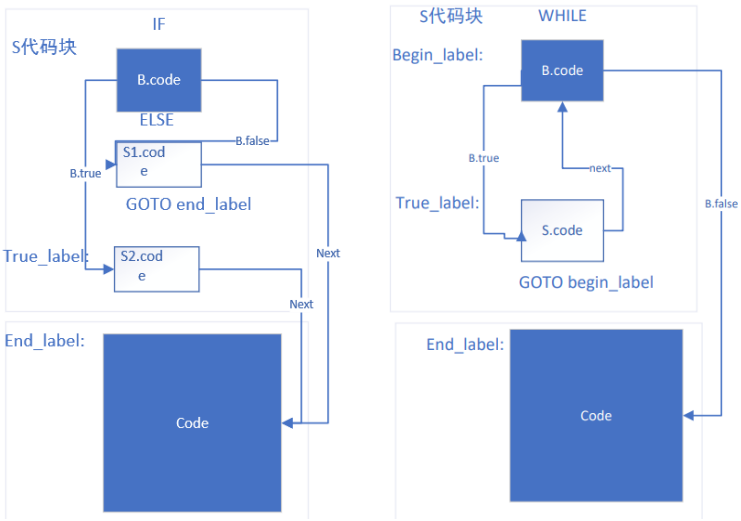
CHAR* GETTRANSLATE(INT VAR,INT K)

根据 var 下标,返回对应的字符串内容,比如说 var=3 代表着常数 2,返回为#2;变量 x 应该返回 t1 这种东西,即临时变量或者常量的编号翻译为字符串

INT GETVARIABLEBYNAME(CHAR* NAME)

根据 variableNames 在变量表里查 variables 中的变量编号

对 IF/WHILE 的处理



If 与 while 处理示意图

IF 的处理，void handleIfElse(Node *tree);

1. If B 为真 GOTO true_label:
    a) S1 语句(若有 else，则把这一句迭代分析语句)
2. GOTO end_label
3. True_label:
    a) S2 语句
4. End_label:
    a) 下一段语句

WHILE 的处理，void handleWhile(Node *tree);

1. Begin_babel:
    a) If B 为真 GOTO true_label:
2. GOTO end_label
3. True_label:
    a) S 语句
    b) Goto begin_label
4. End_label:
    a) 下一段语句

对 INT 一维数组的处理

int handleIntArray(Node *tree, int isAarrayOnLeft):

以 a[0] = 0 为例，通过 tree->child->child 取出对应的数组名字 a,

通过数组名字 a,获取数组 a[]基地址,再通过 tree->child->next 获取 a[0]中的 0

计算出偏移地址与实际地址。

根据节点的 isArrayOnleft 属性，来判断数组元素在左侧或是右侧，依次来判断是否需要取值

对函数的处理

int handleFunc(const char *funcname, const int *arglist, int ptr)

处理函数调用,处理 read 与 write 以及其他函数

1. 首先特殊处理 read 与 write
   a) 如果发现 funcname == read 或者 write，直接根据函数特性继续进行输出
2. 随后在调用一个函数之前，先通过 arg 传入所有参数，
3. 然后通过使用 CALL 调用该函数并存储返回值

对表达式的编译（部分）

int CalcExp(Node* tree,int isAarrayOnLeft)，对于非条件的 Exp 生成计算代码并返回临时变量在符号表中的编号，而 isAarrayOnLeft=1 则表示数组，返回的是数组元素地址

1. EXP->ID:
   a) 对扫描到的 ID，比如 a，直接通过全局变量表，返回 ti 的 i
2. EXT->LP EXP RP
   a) 递归处理中间的 EXP,CalcEXP(tree->child->next,0)
3. EXP->ID LP ARGS RP
   a) 这里需要处理函数
   b) 先从 tree->child 取出 functionName，再 tree->child->next->next 拿出参数 args
   c) 根据 Args → Exp COMMA Args | Exp ,递归处理其中的所有参数 args，直到结束
   d) 调用函数处理 handleFunc(funcname, arglist, ptr)
4. Exp->MINUS Exp
   a) 通过 handleNegativeNum(tree)处理负数，简单的根据-1 翻译为 ti = #0 - 1
5. Exp->Exp STAR Exp； Exp->Exp DIV Exp； Exp->Exp PLUS Exp； Exp->Exp MINUS Exp
   a) 通过 tree->subtype 来获取操作符（*,/,+）
   b) 对第一个和第二个操作数进行 CalcEXP()；来进行处理
   c) 输出本次表达式"t%d := %s %c %s\n"

B）程序应该如何被编译？

编译命令为

./make.sh

运行测试样例命令

1. chmod +x parser
2. ./parser 1.cmm result1.ir（以 1.cmm 输出到 result1.ir 为例）