

实验三报告

一、实现

- 使用双向链表实现线性IR，并添加了尾指针
- 翻译过程将中间代码存储到链表中，最后在打印
- 要求3.1：对结构体类型变量的处理
 - 需要考虑结构体参数地址的计算：

```
1 InterCodes TransExpStruct(node_t *node, Operand place, Type *ret){
2     InterCodes code1 = TransExp(CHILD(1, node), place, ret);
3     FieldList field = field_find(CHILD(3, node)->str, (*ret)-
>u.structure);
4     Operand c1 = operand_malloc(CONSTANT_O, field->offset);
5     InterCodes code2 = gen_arith_code(place, place, c1, ADD);
6     *ret = field->type;
7     return MergeCodes(code1, code2);
8 }
```

- 要求3.2：对数组变量的处理
 - 需要考虑高维数组地址的计算：

```
1 InterCodes TransExpArray(node_t *node, Operand place, Type *ret){
2     InterCodes code1 = TransExp(CHILD(1, node), place, ret);
3     Operand t1 = new_temp();
4     InterCodes code2 = TransExp(CHILD(3, node), t1, NULL);
5     *ret = (*ret)->u.array.elem;
6     InterCodes code3 = gen_arith_code(t1, t1,
operand_malloc(CONSTANT_O, type_size(*ret)), MUL);
7     InterCodes code4 = gen_arith_code(place, place, t1, ADD);
8
9     code1 = MergeCodes(code1, code2);
10    code1 = MergeCodes(code1, code3);
11    return MergeCodes(code1, code4);
12 }
```

- 基本表达式、语句、函数调用的翻译：与实验二类似，借助SDT，为不同的目标设计不同翻译模式

二、编译

```
1 | Code文件夹下  make
```

