

编译原理实验 Lab2-1190201303-王艺丹

实验完成情况综述

完成所有必做内容以及选做所有内容：测试用例1-23均通过

完成C--语言文法的全部词义分析

完成C--语言文法作用域的限制

实验编译方法

文件组成结构

```
└── Lab2-1190201303-王艺丹
    ├── code
    │   ├── lexical.l
    │   ├── main.c
    │   ├── makefile
    │   ├── syntax.y
    │   ├── enum.h
    │   ├── TreeNode.c
    │   ├── TreeNode.h
    │   ├── semantic.h
    │   └── semantic.c
    ├── test
    │   ├── 1.txt
    │   ├── .....
    │   └── 23.txt
    ├── Lab2-Report.pdf
    └── README.md
```

编译方法

在code目录下使用make命令编译所有文件

在code目录下使用make test命令进行语义分析，对测试样例进行测试

实验完成功能及亮点

语义分析

在semantic.h文件中：

- 使用哈希table与十字链表的方式结合维护全局符号栈构建符号表，实现了确定函数作用域的功能
- 相关代码如下：

```
1. typedef struct type {
2.     Kind kind;
3.     union {
4.         // 基本类型
5.         BaseType basic;
6.         // 数组类型信息包括元素类型与数组大小构成
7.         struct {
8.             pType elem;
9.             int size;
10.        } array;
```

```

11.      // 结构体类型信息是一个链表
12.      struct {
13.          char* structName;
14.          pFieldList field;
15.      } structure;
16.
17.      struct {
18.          int argc;
19.          pFieldList argv;
20.          pType returnType;
21.      } function;
22.  } u;
23. } Type;
24.
25. typedef struct fieldList {
26.     char* name;          // 域的名字
27.     pType type;          // 域的类型
28.     pFieldList tail;     // 下一个域
29. } FieldList;
30.
31. typedef struct tableItem {
32.     int symbolDepth;
33.     pFieldList field;
34.     pItem nextSymbol;    // 栈中同等深度的符号
35.     pItem nextHash;     // 处理 hash 冲突
36. } TableItem;
37.
38. typedef struct hashTable {
39.     pItem* hashArray;    // 散列表
40. } HashTable;
41.
42. typedef struct stack {
43.     pItem* stackArray;
44.     int curStackDepth;   // 记录当前元素在栈中的深度
45. } Stack;

```

- ✧ 此外为了区分函数符号以及结构体的dec和def，在讲义中type的基础上进行了修改，存储了结构体和函数的信息，如果是结构体的def，结构体名会存储于上一层的FieldList中，type中的structname为null，如果为dec的话，上一层的FieldList中会存储变量的名字，而此处的type中会存储变量结构体类型的名字。对于匿名结构体，因为词法分析中规定了id不能为纯数字，所以此处用数字编号作为匿名结构体的名字，可以避免与其他符号的命名冲突，数字编号通过在table结构体中记录匿名结构体的个数来获得编号。

相关代码如下：

```

1. typedef struct table { // 十字链表与 open hashing
2.     pHash hash;
3.     pStack stack;
4.     // 未命名结构体，词法分析中 id 不能为纯数字，所以将未命名结构体 id 暂时定义为 int，同时可以处理冲突
5.     int unNamedStructNum;
6. } Table;

```

- ✧ 语义分析实现

语义分析中，进入ExtDef词法单元才意味着程序需要处理当前输入符号，时必定是后，所以程序遍历语法树，遇到ExtDef节点便从该节点开始进行语义分析，然后对ExtDef展开，得到的所有非终结符节点进行处理对符号表进行操作以及查找语义错误

相关代码如下：

```

1. void Traversal(pNode node) {
2.     if (node == NULL) return;
3.     //处理符号一定进入“ExtDef”
4.     if (!strcmp(node->name, "ExtDef")) ExtDef(node);
5.     //递归进行语义分析
6.     Traversal(node->child);
7.     Traversal(node->next);
8. }

```

实验过程中遇到的问题：

当采取：

bison -d syntax.y

flex lexical.l

gcc main.c syntax.tab.c TreeNode.c semantic.c lex.yy.c -lfl -o parser

的命令编译文件时，出现

warning: implicit declaration of function ‘client_tcpinit’ [-Wimplicit-function-declaration]

将编译方式改为静态链接，将所有.c文件编译为.o文件再统一链接生成parser

测试结果：

```

sxj19910136@ubuntu-hitics:~/Desktop/wyd/code$ make test
./parser ../test/1.txt
Error type 1 at Line 4: Undefined variable "j".
./parser ../test/2.txt
Error type 2 at Line 4: Undefined function "inc".
./parser ../test/3.txt
Error type 3 at Line 4: Redefined variable "i".
./parser ../test/4.txt
Error type 4 at Line 6: Redefined function "func".
./parser ../test/5.txt
Error type 5 at Line 4: Type mismatched for assignment.
./parser ../test/6.txt
Error type 6 at Line 4: The left-hand side of an assignment must be a variable.
./parser ../test/7.txt
Error type 7 at Line 4: Type mismatched for operands.
./parser ../test/8.txt
Error type 8 at Line 4: Type mismatched for return.
./parser ../test/9.txt
Error type 9 at Line 8: too many arguments to function "func", except 1 args.
./parser ../test/10.txt
Error type 10 at Line 4: "i" is not an array.
./parser ../test/11.txt
Error type 11 at Line 4: "i" is not a function.
./parser ../test/12.txt
Error type 12 at Line 4: "1.5" is not an integer.
./parser ../test/13.txt
Error type 13 at Line 9: Illegal use of ".".
./parser ../test/14.txt
Error type 14 at Line 9: NONEXISTFIELD.
./parser ../test/15.txt
Error type 15 at Line 4: Redefined field "x".
./parser ../test/16.txt
Error type 16 at Line 6: Duplicated name "Position".
./parser ../test/17.txt
Error type 17 at Line 3: Undefined structure "Position".
./parser ../test/18.txt
Error type B at line 1: syntax error.
./parser ../test/19.txt
Error type B at line 6: syntax error.
./parser ../test/20.txt
./parser ../test/21.txt
Error type 3 at Line 10: Redefined variable "i".
./parser ../test/22.txt
Error type 5 at Line 17: Type mismatched for assignment.
./parser ../test/23.txt
Error type 5 at Line 16: Type mismatched for assignment.

```