

编译原理 lab2 实验报告

211220166 王诚昊

邮箱: 211220166@smail.nju.edu.cn

使用链表来储存符号表。

链表定义和类型系统定义在 semantic.h 中。

```
> typedef struct Type_Sys...
    }T;

> typedef struct FieldList_...
    }F;

    // Node of the list
> typedef struct SymbolNode { ...
    }ListNode;
```

对于变量，存储其名称和类型。

对于函数，存储返回类型，参数个数和类型。

对于建立好的语法树，通过前序遍历扫描进行语义分析，其中主要的函数都定义在 semantic.c 中：

```
    // 给链表中插入结点
> void insertSymbol(char* name, T type, ListNode** ptail) { ...

    // 找到链表中的某个结点
> ListNode* findSymbol(char* name, ListNode* head){ ...

    // 打印错误信息
> void print_error(int error_type, int line){ ...

    // 填链表
> void fill_symboltable(struct ASTNode* astnode, ListNode** ptail){ ...

    // 查链表
> void check_symboltable(struct ASTNode* astnode, ListNode* head){ ...

    // 遍历AST进行语义分析
> void ASTtraversal(struct ASTNode* root, ListNode* head, ListNode** ptail){ ...
```

进度还是差很多，日后继续努力。