CS323 Computer Phase2 Report

Team and task division:

贺晗 12012013:

Insert variables and function name into symbol table; semantic error type 1-5

郑鑫颖 11912039：

Implementation of Btree symbol table; semantic error type 6-10

张海涵 12012222：

Insert structure and array into symbol table; semantic error type 11-15

Data structure

We design an **Info** structure to store necessary information with each symbol.

```
typedef struct Info
{
    int a;//0-char/int/float 1数组 2函数 3结构体
    struct Type *type;
    struct Type *return_type;
    struct ParaList* paraList;
} Info;
```

```
typedef struct Type
{
    char* name;
    enum
    {
        PRIMITIVE,
        ARRAY,
        STRUCTURE
    } category;
    union
    {
        enum
        {
            Int,
            Float,
            Char
        } primitive;
        struct Array *array;
        struct FieldList *structure;
    };
} Type;
```

Type	related data structure

Type	related data structure
int/char/float	Type.name="int"; Type.category = PRIMITIVE
array	Type.name="array";Type.category=Array; Type.array;
struture	Type.name=name of structure; Type.structure=STRUCTURE; Type.structure
array of structure	Type.name=name of structure;Type.category=Array; Type.array;

Special Design

1: The implementation of symbol table

we use binary tree to implement out symbol table.

2: the content of symbol table

we put following into symbol table:

- varible name(int/char/float/structure)
- struture name
- function name
- function parameter name
- struture field name

3: Type of Exp

We also define type for each expression. **Every operator need to operate on the same type of data**. (int cannot be add to float)

4: check return type

we use a **bfs** algorithm to tranverse the subtree that is root at the function, when we meet return node, we check whether it is the same as defined one.