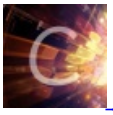
编译原理实验 之 符号表

原创

置顶 ibmmicrosoft 于 2012-05-29 00:22:40 发布 6413 收藏 12
分类专栏： c++语言 文章标签： search insert 编译器 table struct 工作

本文链接：https://blog.csdn.net/ibmmicrosoft/article/details/7611198



这是本系列第一篇文章, 也是编译开始阶段的准备工作.

编译器首先都是进行的词法分析,而词法分析的基础就是符号表.

把不同类型的数据和其操作,属性都先预留下来,后面词法分析的时候可以用到.

这次的实验只是很简单的模拟而已,所以代码比较简单,编译实验真正的是在词法分析之后开始的.

先贴代码,有用的mark下就好.

```
#include <iostream>

#define MAX_SIZE  100


using namespace std;


int myHash(const char * expr)                    //哈希函数
{
 unsigned long h =0;
 while(*expr)
 {
   h = (h<<4)+*expr++;
   unsigned long g = h&0xF0000000L;
   if(g)h^=g>>24;
   h &= ~g;
 }
 return h%10;
}



struct Arra                                      //数组类型
{
public:
```

```
public:
 //类型，个数，地址，维数
 int num;
 int colum;
 int type;
    long addr;
 const char * nam;
 Arra * next;
 bool arra_insert(Arra& a);                  //插入
 bool arra_insertHelp(Arra& a);     //插入帮助
    bool arra_search(Arra& a);   //查找
} *Alist;
Arra Arra_Hash_Table[MAX_SIZE];

bool Arra::arra_search(Arra& a)
{
    int h = myHash(a.nam);
 if(Arra_Hash_Table[h].nam-a.nam == 0)
 {
 cout<<"数组 "<<a.nam<<"  已存在"<<endl;
 return 1;
 }
 else
 {
 cout<<"数组 "<<a.nam<<"  不存在 "<<endl;
 return 0;
 }
}

bool Arra::arra_insert(Arra& a)
{
 if(arra_search(a))
 {
  printf("是否覆盖   Y/N  ?\n");
     char an ;
  cin>>an;
  if(an == 'Y'||'y')arra_insertHelp(a);
  else return 0;
 }
 else
 arra_insertHelp(a);
 return 1;
}

bool Arra::arra_insertHelp(Arra& a)
{
   int h = myHash(a.nam);
   Arra_Hash_Table[h] = a;
   cout<<"插入成功!"<<endl;
   return true;
}
```

每种类型的数据都有自己的基本操作,我们的c++编译器其实也是这样做的,

比如函数,数组,变量,等等,都是有自己的属性的,只不过是我们看不到而已.

下次的就是词法分析过程了,其实挺简单的,只不过是代码比较长,希望对感兴趣的童鞋有帮助吧~

有啥意见的可以和我交流~