

map 和 unordered_map以char * 为key_xuh5156的博客-CSDN博客

成就一亿技术人!

map 和 unordered_map以char *为key

使用map或者unordered_map进行字符串查找一般都是用std::string类型作为key，但是std::string的效率实在太低，不得不进行优化，尝试使用char*作key来查找。

一、map以char*为key

默认的map<char*,int>的key是指针，比较的是指针值的大小，不能进行字符串的匹配。

查看map的模板定义：(<http://www.cplusplus.com/reference/map/map/?kw=map>)

```
template < class Key,                //map::key_type
          class T,                  //map::mapped_type
          class Compare = less<Key>, // map::key_compare
          class Alloc = allocator<pair<const Key,T>> > // map::allocator_type
> class map;
```

因此我们可以编写字符串的比较类来实现map以char*为key，代码如下：

```
struct Cmp
{
    //重载运算符
    bool operator()(const char * str1,constchar * str2)
    {
        return strcmp(str1,str2) < 0;
    }
};

map<char*,int,Cmp> mapStr;
```

在mapStr中插入char*就可以进行字符串的查找操作了：

切记，如果是局部指针变量，可能会出现随机结果，因此建议使用char*ss = new char[N]，通过new来进行指针的空间申请。

二、unordered_map以char*为key

unordered_map是C++11中的hash表，因为hash_map已经存在，因此搞了个怪名字。

unordered_map模板定义

(http://www.cplusplus.com/reference/unordered_map/unordered_map/?kw=unordered_map)：

```
template < class Key,                // unordered_map::key_type
          class T,                  // unordered_map::mapped_type
          class Hash = hash<Key>,   // unordered_map::hasher
          class Pred = equal_to<Key>, // unordered_map::key_equal
          class Alloc = allocator< pair<const Key,T> > > // unordered_map::allocator_type
> class unordered_map;
```

unordered_map需要重新写hash函数和比较函数，代码如下：

```
//hash函数

struct Hash_Func
{
    //BKDR hash algorithm, 有关字符串hash函数，可以去找找资料看看
    int operator()(PSA_CHAR * str)const
    {
        int seed = 131;//31 131 1313 13131131313 etc//
        int hash = 0;
        while(*str)
        {
            hash = (hash * seed) + (*str);
            str++;
        }
        return hash & (0x7FFFFFFF);
    }
};

//比较函数

struct Cmp
{
    bool operator()(const PSA_CHAR *str1,const PSA_CHAR * str2)const
```

```
{  
    return strcmp(str1,str2) == 0;  
}  
};  
  
unordered_map<char*,int, Hash_Func ,Cmp> hashStr;
```

编译unordered_map是需要加上编译选项 -std=c++0x。

重写hash函数和比较函数的目的应该是只有hash值相同并且字符串也相同才能判断两个字符串是一样的，如果只是hash值相同并不能判断字符串相同。

切记：一定要注意局部指针变量的问题，使用new运算符对指针申请空间，在使用结束遍历map或者unordered_map释放空间。