


## 导航

博客园  
首页  
新随笔  
联系  
订阅   
管理

< 2020年9月 >						
日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

## 公告

昵称: tomcao  
园龄: 4年6个月  
粉丝: 0  
关注: 15  
[+加关注](#)

## 搜索

<input type="text"/>	<input type="button" value="找找看"/>
<input type="text"/>	<input type="button" value="谷歌搜索"/>

## 常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

## 我的标签

mysql 无法登录 初始密码(1)  
二维指针(1)

## 随笔分类

boost(1)  
C++基础(5)  
lua(2)  
mysql(2)

## 随笔档案

2017年4月(2)

## 正确使用stl map的erase方法

先声明：下面的文章是针对windows的用法，因为std::map的erase函数的windows的实现版本是返回一个std::map的迭代器，但是STL标准里面的该函数的返回值确是：

map.erase有3个重载：

```
void erase ( iterator position );  
size_type erase ( const key_type& x );  
void erase ( iterator first, iterator last );
```

。

所以下面的代码中的最后一个例子仅仅可以在windows下的map下运行。

STL的map表里有一个erase方法用来从一个map中删除掉指令的节点

eg1:

```
map<string,string> mapTest;  
typedef map<string,string>::iterator ITER;
```

```
ITER iter=mapTest.find(key);  
mapTest.erase(iter);
```

像上面这样只是删除单个节点,map的形为不会出现任务问题,但是当在一个循环里用的时候,往往会被误用,那是因为使用者没有正确理解iterator的概念.

像下面这样的例子就是错误的写法,  
eg2:

```
for(ITER iter=mapTest.begin();iter!=mapTest.end();++iter)  
{  
    cout<<iter->first<<":"<<iter->second<<endl;  
    mapTest.erase(iter);  
}
```

2017年2月(5)  
2016年10月(3)

## 阅读排行榜

1. 正确使用stl map的erase方法(7293)
2. c++遍历lua中的table(2289)
3. 删除map容器中指定的元素(2091)
4. C++二维指针(1050)
5. C++调用lua中的函数(707)

这是一种错误的写法,会导致程序行为不可知.究其原因是map是关联容器,对于关联容器来说,如果某一个元素已经被删除,那么其对应的迭代器就失效了,不应该再被使用;否则会导致程序无定义的行为。

可以用以下方法解决这问题:

正确的写法

- 1.使用删除之前的迭代器定位下一个元素。STL建议的使用方式

```
for(ITER iter=mapTest.begin();iter!=mapTest.end();)
{
    cout<<iter->first<<":"<<iter->second<<endl;
    mapTest.erase(iter++);
}
```

2. erase() 成员函数返回下一个元素的迭代器

```
for(ITER iter=mapTest.begin();iter!=mapTest.end();)
{
    cout<<iter->first<<":"<<iter->second<<endl;
    iter=mapTest.erase(iter);
}
```

分类: [C++基础](#)

好文要顶

关注我

收藏该文



tomcao

关注 - 15

粉丝 - 0

+加关注

0

0

» 下一篇: [operator++\(\)和operator++\(int\)的区别](#)

posted on 2016-10-29 17:43 tomcao 阅读(7294) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问](#) [网站](#) [首页](#)。

【推荐】超50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库