

## 简单的程序诠释C++ STL算法系列之十三：copy

前面十二个算法所展现的都属于非变易算法（Non-mutating algorithms）系列，现在我们来看看变易算法。所谓变易算法(Mutating algorithms)就是一组能够修改容器元素数据的模板函数，可进行序列数据的复制，变换等。

我们现在来看看第一个变易算法：元素复制算法copy。该算法主要用于容器之间元素的拷贝，即将迭代器区间[first, last)的元素复制到由复制目标result给定的区间[result, result+(last-first))中。下面我们来看看它的函数原型：

函数原形：

```
template<class InputIterator, class OutputIterator>
OutputIterator copy(
    InputIterator _First,
    InputIterator _Last,
    OutputIterator _DestBeg
);
```

参数

**\_First, \_Last**

指出被复制的元素的区间范围[\_First, \_Last)。

**\_DestBeg**

指出复制到的目标区间起始位置

返回值

返回一个迭代器，指出已被复制元素区间的最后一个位置

程序示例：

首先我们来一个简单的示例，定义一个简单的整形数组myints，将其所有元素复制到容器myvector中，并将数组向左移动一位。

```

/*****
 * Copyright (C) Jerry Jiang
 *
 * File Name   : copy01.cpp
 * Author      : Jerry Jiang
 * Create Time : 2012-3-20 22:44:28
 * Mail       : jbiaojerry@gmail.com
 * Blog       : http://blog.csdn.net/jerryjbiao
 *
 * Description : 简单的程序诠释C++ STL算法系列之十三
 *              变易算法 : 元素复制copy
 *
 *****/

#include <iostream>
#include <algorithm>
#include <vector>

using namespace std;

int main ()
{
    int myints[] = {10, 20, 30, 40, 50, 60, 70};
    vector<int> myvector;
    vector<int>::iterator it;

    myvector.resize(7);    // 为容器myvector分配空间

    //copy用法一:
    //将数组myints中的七个元素复制到myvector容器中
    copy ( myints, myints+7, myvector.begin() );

    cout << "myvector contains: ";
    for ( it = myvector.begin(); it != myvector.end(); ++it )
    {
        cout << " " << *it;
    }
    cout << endl;

    //copy用法二:
    //将数组myints中的元素向左移动一位
    copy(myints + 1, myints + 7, myints);

    cout << "myints contains: ";
    for ( size_t i = 0; i < 7; ++i )
    {
        cout << " " << myints[i];
    }
    cout << endl;

    return 0;
}

```

从上例中我们看出copy算法可以很简单地将一个容器里面的元素复制至另一个目标容器中，上例中代码特别要注意一点就是**myvector.resize(7)**;这行代码，在这里一定要先为**vector**分配空间，否则程序会崩，这是初学者经常犯的一个错误。其实copy函数最大的威力是结合标准输入输出迭代器的时候，我们通过下面这个示例就可以看出它的威力了。

```

/*****
 * Copyright (C) Jerry Jiang
 *
 * File Name   : copy2.cpp
 * Author      : Jerry Jiang
 * Create Time : 2012-3-20 23:25:29
 * Mail        : jbiaojerry@gmail.com
 * Blog         : http://blog.csdn.net/jerryjbiao
 *
 * Description : 简单的程序诠释C++ STL算法系列之十三
 *               变易算法 : 元素复制copy
 *
 *****/

#include <iostream>
#include <algorithm>
#include <vector>
#include <iterator>
#include <string>

using namespace std;

int main ()
{
    typedef vector<int> IntVector;
    typedef istream_iterator<int> IstreamItr;
    typedef ostream_iterator<int> OstreamItr;
    typedef back_insert_iterator< IntVector > BackInsItr;

    IntVector myvector;

    // 从标准输入设备读入整数
    // 直到输入的是非整型数据为止 请输入整数序列, 按任意非数字键并回车结束输入
    cout << "Please input element: " << endl;
    copy(IstreamItr(cin), IstreamItr(), BackInsItr(myvector));

    //输出容器里的所有元素, 元素之间用空格隔开
    cout << "Output : " << endl;
    copy(myvector.begin(), myvector.end(), OstreamItr(cout, " "));
    cout << endl;

    return 0;
}

```

呵呵，现在知道copy的神奇所在吧，^\_^

\*\*\*\*\*

**C++经典书目索引及资源下载: <http://blog.csdn.net/jerryjbiao/article/details/7358796>**

\*\*\*\*\*

阅读更多      登录后自动展开