

简单的程序诠释C++ STL算法系列之八：mismatch

C++STL的非变异算法（Non-mutating algorithms）是一组不破坏操作数据的模板函数，用来对序列数据进行逐个处理、元素查找、子序列搜索、统计和匹配。

mismatch算法是比较两个序列，找出首个不匹配元素的位置。它有如下两个函数原型，找出迭代器区间[first1, last1) 上第一个元素 *i，它和迭代器区间[first2, first2 + (last1 - first1))上的元素*(first2 + (i - first1))不相等（或者不满足二元谓词binary_pred条件）。通过匹配对象pair返回这两个元素的迭代器，指示不匹配元素位置。

函数原型：

```
template<class InputIterator1, class InputIterator2>
pair<InputIterator1, InputIterator2> mismatch(
    InputIterator1 _First1,
    InputIterator1 _Last1,
    InputIterator2 _First2
);

template<class InputIterator1, class InputIterator2, class BinaryPredicate>
pair<InputIterator1, InputIterator2> mismatch(
    InputIterator1 _First1,
    InputIterator1 _Last1,
    InputIterator2 _First2
    BinaryPredicate _Comp
);
```

示例代码：

```

/*****
 * Copyright (C) Jerry Jiang
 * File Name   : mismatch.cpp
 * Author      : Jerry Jiang
 * Create Time : 2011-10-9 21:16:53
 * Mail       : jbiaojerry@gmail.com
 * Blog       : http://blog.csdn.net/jerryjbiao
 * Description : 简单的程序诠释C++ STL算法系列之八
 *             非变易算法 : 元素不匹配查找mismatch
 *****/

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

using namespace std;

bool strEqual(const char* s1, const char* s2)
{
    return strcmp(s1, s2) == 0 ? true : false;
}

typedef vector<int>::iterator ivecIter;

int main()
{
    vector<int> ivec1, ivec2;
    ivec1.push_back(2);
    ivec1.push_back(0);
    ivec1.push_back(1);
    ivec1.push_back(4);

    ivec2.push_back(2);
    ivec2.push_back(0);
    ivec2.push_back(1);
    ivec2.push_back(7);

    pair<ivecIter, ivecIter> retCode;
    retCode = mismatch(ivec1.begin(), ivec1.end(), ivec2.begin());
    if (retCode.first == ivec1.end() && retCode.second == ivec2.end() /* ivec2.begin() */)
    {
        cout << "ivec1 和 ivec2完全相同" << endl;
    }
    else
    {
        cout << "ivec1 和 ivec2 不相同, 不匹配的元素为: \n"
              << *retCode.first << endl
              << *retCode.second << endl;
    }

    char* str1[] = {"apple", "pear", "watermelon", "banana", "grape"};
    char* str2[] = {"apple", "pears", "watermelons", "banana", "grape"};

    pair<char**, char**> retCode2 = mismatch(str1, str1+5, str2, strEqual);
    if (retCode2.first == str1+5 && retCode2.second == str2+5)
    {
        cout << "str1 和 str2 完全相同" << endl;
    }
    else
    {
        cout << "str1 和 str2 不相同, 不匹配的字符串为: " << endl
              << str1[retCode2.first - str1] << endl
              << str2[retCode2.second - str2] << endl;
    }

    return 0;
}

```

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