

字符串分割(C++)

经常碰到字符串分割的问题，这里总结下，也方便我以后使用。

一、用 strtok 函数进行字符串分割

原型： `char *strtok(char *str, const char *delim);`

功能：分解字符串为一组字符串。

参数说明： `str` 为要分解的字符串， `delim` 为分隔符字符串。

返回值：从 `str` 开头开始的一个个被分割的串。当没有被分割的串时则返回 `NULL`。

示例：

//借助 strtok 实现 split

```
#include <string.h>
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    char s[] = "Golden Global View,disk * desk";
```

```
    const char *d = " ,*";
```

```
    char *p;
```

```
    p = strtok(s,d);
```

```
    while(p)
```

```
    {
```

```
        printf("%s\n",p);
```

```
        p=strtok(NULL,d);
```

```
    }
```

```
    return 0;
```

```
}
```

运行效果：



```
Golden
Global
View
disk
desk

Process returned 0 (0x0)   execution time : 0.203 s
Press any key to continue.
```

二、用 STL 进行字符串的分割

涉及到 `string` 类的两个函数 `find` 和 `substr`：

1、find 函数

原型： `size_t find (const string& str, size_t pos = 0) const;`

功能：查找子字符串第一次出现的位置。

参数说明： `str` 为子字符串， `pos` 为初始查找位置。

返回值：找到的话返回第一次出现的位置，否则返回 `string::npos`

2、substr 函数

原型： `string substr (size_t pos = 0, size_t n = npos) const;`

功能：获得子字符串。

参数说明： `pos` 为起始位置（默认为 0）， `n` 为结束位置（默认为 `npos`）

返回值：子字符串

实现如下：

//字符串分割函数

```
std::vector<std::string> split(std::string str, std::string pattern)
{
    std::string::size_type pos;
    std::vector<std::string> result;
    str+=pattern; //扩展字符串以方便操作
    int size=str.size();

    for(int i=0; i<size; i++)
    {
        pos=str.find(pattern, i);
        if(pos<size)
        {
            std::string s=str.substr(i, pos-i);
            result.push_back(s);
            i=pos+pattern.size()-1;
        }
    }
    return result;
}
```

完整代码：

```
/*
    File      : split1.cpp
    Author    : Mike
    E-Mail    : Mike_Zhang@live.com
*/
#include <iostream>
#include <string>
#include <vector>

//字符串分割函数
std::vector<std::string> split(std::string str, std::string pattern)
{
    std::string::size_type pos;
    std::vector<std::string> result;
    str+=pattern; //扩展字符串以方便操作
    int size=str.size();

    for(int i=0; i<size; i++)
    {
        pos=str.find(pattern, i);
        if(pos<size)
        {
            std::string s=str.substr(i, pos-i);
            result.push_back(s);
            i=pos+pattern.size()-1;;
        }
    }
    return result;
}
```

```

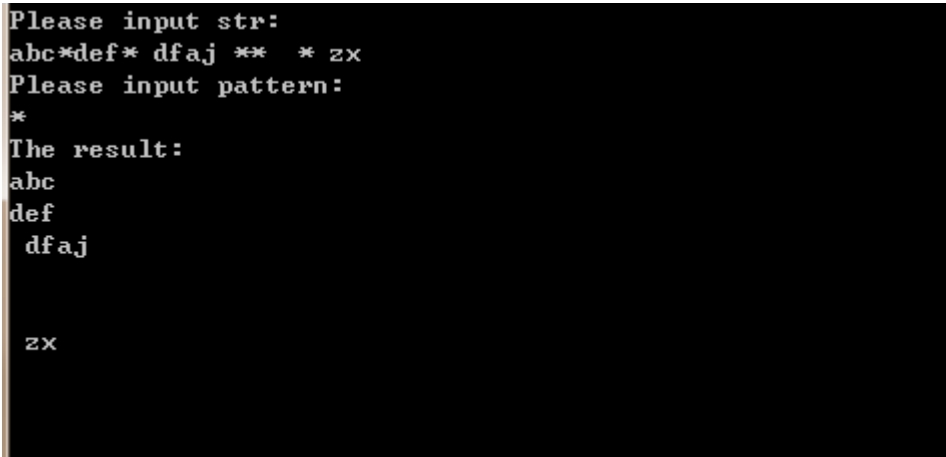
}

int main()
{
    std::string str;
    std::cout<<"Please input str:"<<std::endl;
    //std::cin>>str;
    getline(std::cin,str);
    std::string pattern;
    std::cout<<"Please input pattern:"<<std::endl;
    //std::cin>>pattern;
    getline(std::cin,pattern); //用于获取含空格的字符串
    std::vector<std::string> result=split(str,pattern);
    std::cout<<"The result:"<<std::endl;
    for(int i=0;i<result.size();i++)
    {
        std::cout<<result[i]<<std::endl;
    }

    std::cin.get();
    std::cin.get();
    return 0;
}

```

运行效果:



```

Please input str:
abc*def* dfaj ** * zx
Please input pattern:
*
The result:
abc
def
dfaj
zx

```

三、用 Boost 进行字符串的分割

用 boost 库的正则表达式实现字符串分割

实现如下:

```

std::vector<std::string> split(std::string str,std::string s)
{
    boost::regex reg(s.c_str());
    std::vector<std::string> vec;
    boost::sregex_token_iterator it(str.begin(),str.end(),reg,-1);
    boost::sregex_token_iterator end;
    while(it!=end)
    {
        vec.push_back(*it++);
    }
}

```

```
    return vec;
}
```

完整代码:

//本程序实现的是利用正则表达式对字符串实现分割

//运行环境 VC6.0 + boost 库

/*

File : split2.cpp

Author : Mike

E-Mail : Mike_Zhang@live.com

*/

#include <iostream>

#include <cassert>

#include <vector>

#include <string>

#include "boost/regex.hpp"

```
std::vector<std::string> split(std::string str,std::string s)
{
```

```
    boost::regex reg(s.c_str());
```

```
    std::vector<std::string> vec;
```

```
    boost::sregex_token_iterator it(str.begin(),str.end(),reg,-1);
```

```
    boost::sregex_token_iterator end;
```

```
    while(it!=end)
```

```
    {
```

```
        vec.push_back(*it++);
```

```
    }
```

```
    return vec;
```

```
}
```

```
int main()
```

```
{
```

```
    std::string str,s;
```

```
    str="sss/ddd/ggg/hh";
```

```
    s="/";
```

```
    std::vector<std::string> vec=split(str,s);
```

```
    for(int i=0,size=vec.size();i<size;i++)
```

```
    {
```

```
        std::cout<<vec[i]<<std::endl;
```

```
    }
```

```
    std::cin.get();
```

```
    std::cin.get();
```

```
    return 0;
```

```
}
```

运行效果:

```
sss
```

```
ddd
```

```
ggg
```

```
hh
```

```
-
```