本文实例汇总了C++常用字符串分割方法，分享给大家供大家参考。具体分析如下：

我们在编程的时候经常会碰到字符串分割的问题，这里总结下，也方便我们以后查询使用。

**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。

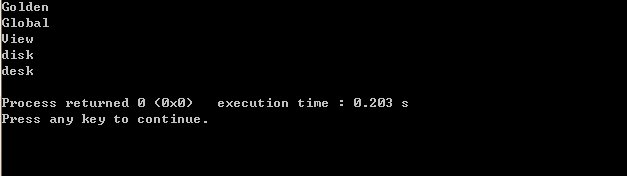
其它：strtok函数线程不安全，可以使用strtok\_r替代。

示例：

[?](http://www.jb51.net/article/55954.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | //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;  } |

运行效果如下图所示：

[](http://files.jb51.net/file_images/article/201410/2014108101416933.png?201498101535)

**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：

[?](http://www.jb51.net/article/55954.htm)

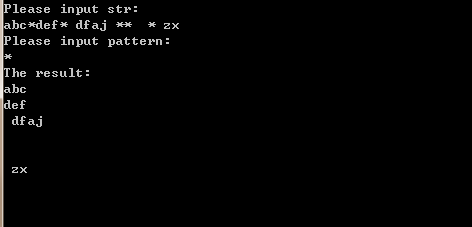
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | //字符串分割函数  std::vector<std::string> split(std::string str,std::string pattern)  {  std::string::size\_type pos;  std::vector<std::string> result;  str+=pattern;//扩展字符串以方便操作  int size=str.size();  for(int i=0; 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：

[?](http://www.jb51.net/article/55954.htm)

|  |  |
| --- | --- |
| 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  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | /\*  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;  } |

运行效果如下图所示：



**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：

[?](http://www.jb51.net/article/55954.htm)

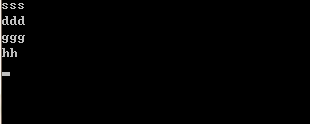
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | std::vector<std::string> split(std::string str,std::string s)  {  boost::regex reg(s.c\_str());  std::vector<std::string> vec;  boost::sregex\_token\_iterator it(str.begin(),str.end(),reg,-1);  boost::sregex\_token\_iterator end;  while(it!=end)  {  vec.push\_back(\*it++);  }  return vec;  } |

完整代码：

[?](http://www.jb51.net/article/55954.htm)

|  |  |
| --- | --- |
| 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  31  32  33  34  35  36  37  38  39 | //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;  } |

运行效果如下图所示：



**补充：**

最近发现boost里面有自带的split的函数，如果用boost的话，还是直接用split的好，这里就不多说了，代码如下：

[?](http://www.jb51.net/article/55954.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | #include <iostream>  #include <string>  #include <vector>  #include <boost/algorithm/string/classification.hpp>  #include <boost/algorithm/string/split.hpp>  using namespace std;  int main()  {  string s = "sss/ddd,ggg";  vector<string> vStr;  boost::split( vStr, s, boost::is\_any\_of( ",/" ), boost::token\_compress\_on );  for( vector<string>::iterator it = vStr.begin(); it != vStr.end(); ++ it )  cout << \*it << endl;  return 0;  } |

希望本文所述对大家的C++程序设计有所帮助。