

D:/H-Drive/public_html/91.204/91.204-2012-13f/204-lecs/code/BoostRegexTests/regex_split_example_2.cpp

```
1  /*
2  *
3  * Copyright (c) 1998-2002
4  * John Maddock
5  *
6  * Use, modification and distribution are subject to the
7  * Boost Software License, Version 1.0. (See accompanying file
8  * LICENSE_1_0.txt or copy at http://www.boost.org/LICENSE_1_0.txt)
9  *
10 */
11
12 /*
13 *   LOCATION:    see http://www.boost.org for most recent version.
14 *   FILE        regex_split_example_2.cpp
15 *   VERSION      see <boost/version.hpp>
16 *   DESCRIPTION: regex_split example: spit out linked URLs.
17 */
18
19
20 #include <list>
21 #include <fstream>
22 #include <iostream>
23 #include <iterator>
24 #include <boost/regex.hpp>
25
26 boost::regex e("<\s*A\s+[>]*href\s*=\s*\"([^\"]*)\">",
27               boost::regex::normal | boost::regexbase::icase);
28
29 void load_file(std::string& s, std::istream& is)
30 {
31     s.erase();
32     if(is.bad()) return;
33     //
34     // attempt to grow string buffer to match file size,
35     // this doesn't always work...
36     s.reserve(static_cast<std::string::size_type>(is.rdbuf()->in_avail()));
37     char c;
38     while(is.get(c))
39     {
40         // use logarithmic growth strategy, in case
41         // in_avail (above) returned zero:
42         if(s.capacity() == s.size())
43             s.reserve(s.capacity() * 3);
44         s.append(1, c);
45     }
46 }
47
```

D:/H-Drive/public_html/91.204/91.204-2012-13f/204-lecs/code/BoostRegexTests/regex_split_example_2.cpp

D:/H-Drive/public_html/91.204/91.204-2012-13f/204-lecs/code/BoostRegexTests/regex_split_example_2.cpp

```
48 int main(int argc, char** argv)
49 {
50     std::string s;
51     std::list<std::string> l;
52     int i;
53     for(i = 1; i < argc; ++i)
54     {
55         std::cout << "=====" << std::endl;
56         std::cout << "Finding URLs in file:  " << argv[i] << std::endl;
57         std::cout << "=====" << std::endl;
58         s.erase();
59         std::ifstream is(argv[i]);
60         load_file(s, is);
61         is.close();
62         boost::regex_split(std::back_inserter(l), s, e);
63         while(l.size())
64         {
65             s = *(l.begin());
66             l.pop_front();
67             std::cout << s << std::endl;
68         }
69     }
70     //
71     // alternative method:
72     // split one match at a time and output direct to
73     // cout via ostream_iterator<std::string>....
74     //
75     for(i = 1; i < argc; ++i)
76     {
77         std::cout << "Findings URL's in " << argv[i] << ":" << std::endl;
78         s.erase();
79         std::ifstream is(argv[i]);
80         load_file(s, is);
81         is.close();
82         while(boost::regex_split(std::ostream_iterator<std::string>(std::cout
83     }
84
85     return 0;
86 }
```

```
), s, e, boost::match_default, 1)) std::cout << std::endl;
```