

Memory Snippets

时间如同白驹过隙，转瞬即逝。无穷VS有限。总有一天，我会第一。

管理 订阅 联系 新随笔

C语言函数sscanf()的用法

在我的学习过程中，从文件读取数据是一件很麻烦的事，所幸有sscanf()函数。

C语言函数sscanf()的用法

sscanf() - 从一个字符串中读进与指定格式相符的数据。

函数原型：

int sscanf( string str, string fmt, mixed var1, mixed var2 ... );

int scanf( const char \*format [,argument]... );

说明：

sscanf与scanf类似，都是用于输入的，只是后者以屏幕(stdin)为输入源，前者以固定字符串为输入源。

其中的format可以是一个或多个 { %[\*] [width] [{h | l | I64 | L}]type | ' ' | '\t' | '\n' | 非%符号 }

注：

- 1、 \* 亦可用于格式中, (即 %\*d 和 %\*s) 加了星号 (\*) 表示跳过此数据不读入. (也就是不把此数据读入参数中)
- 2、 {a|b|c}表示a,b,c中选一, [d],表示可以有d也可以没有d。
- 3、 width表示读取宽度。
- 4、 {h | l | I64 | L}:参数的size,通常h表示单字节size, l表示2字节 size,L表示4字节size(double例外),I64表示8字节size。

5、 type :这就很多了，就是%s,%d之类。

6、特别的： %\* [width] [{h | l | I64 | L}]type 表示满足该条件的被过滤掉，不会向目标参数中写入值  
支持集合操作：

%[a-z] 表示匹配a到z中任意字符，贪婪性(尽可能多的匹配)

%[aB] 匹配a、B、'中一员，贪婪性

%[^a] 匹配非a的任意字符，贪婪性

注意：在读入的字符串是空字符串时， sscanf函数并不改变待读入到的字符串的值。

例子：

1. 常见用法。

char buf[512] = ;  
sscanf("123456 ", "%s", buf);  
printf("%s\n", buf);  
结果为： 123456

2. 取指定长度的字符串。如在下例中，取最大长度为4字节的字符串。

sscanf("123456 ", "%4s", buf);  
printf("%s\n", buf);  
结果为： 1234

3. 取到指定字符为止的字符串。如在下例中，取遇到空格为止字符串。

sscanf("123456 abcdef", "%[^ ]", buf);  
printf("%s\n", buf);  
结果为： 123456

4. 取仅包含指定字符集的字符串。如在下例中，取仅包含1到9和小写字母的字符串。

sscanf("123456abcdefBCDEF", "%[1-9a-z]", buf);  
printf("%s\n", buf);  
结果为： 123456abcdef

5. 取到指定字符集为止的字符串。如在下例中，取遇到大写字母为止的字符串。

sscanf("123456abcdefBCDEF", "%[^A-Z]", buf);  
printf("%s\n", buf);  
结果为： 123456abcdef

6、给定一个字符串iios/12DDWDFf@122，获取 / 和 @ 之间的字符串，先将 "iios/"过滤掉，再将非'@'的一串内容送到buf中

sscanf("iios/12DDWDFf@122", "%\*[^/]/%[^@]", buf);  
printf("%s\n", buf);  
结果为： 12DDWDFf

7、给定一个字符串""hello, world"，仅保留world。（注意： "， "之后有一空格）

sscanf("hello, world", "%\*s%s", buf);

公告

昵称： Walter L  
园龄： 8年2个月  
粉丝： 18  
关注： 0  
+加关注

常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

随笔档案(23)

2012年10月 (1)  
2010年12月 (1)  
2010年6月 (2)  
2010年5月 (1)  
2009年12月 (1)  
2009年11月 (6)  
2009年10月 (1)  
2009年7月 (1)  
2009年6月 (4)  
2009年5月 (1)  
2009年3月 (1)  
2008年12月 (3)

```
printf("%s\n", buf);
结果为: world
%s表示第一个匹配到的%s被过滤掉, 即hello被过滤了
如果没有空格则结果为NULL。
sscanf的功能很类似于正则表达式, 但却没有正则表达式强大,所以如果对于比较复杂的字符串处理,建议使用正则表达式。
//-----
sscanf,表示从字符串中格式化输入
上面表示从str中, 输入数字给x, 就是32700
久以前, 我以为c没有自己的split string函数, 后来我发现了sscanf; 一直以来, 我以为sscanf只能以空格来界定字符串, 现在我发现我错了。
sscanf是一个运行时函数, 原形很简单:
int sscanf(
const char *buffer,
const char *format [,
argument ] ...
);
它强大的功能体现在对format的支持上。
我以前用它来分隔类似这样的字符串2006:03:18:
int a, b, c;
sscanf("2006:03:18", "%d:%d:%d", a, b, c);
以及2006:03:18 - 2006:04:18:
char sztime1[16] = "", sztime2[16] = "";
sscanf("2006:03:18 - 2006:04:18", "%s - %s", sztime1, sztime2);
但是后来, 我需要处理2006:03:18-2006:04:18
仅仅是取消了`-`两边的空格, 却打破了%s对字符串的界定。
我需要重新设计一个函数来处理这样的情况? 这并不复杂, 但是, 为了使所有的代码都有统一的风格, 我需要改动很多地方, 把已有的sscanf替换成我自己的分割函数。我以为我肯定需要这样做, 并伴随着对sscanf的强烈不满而入睡: 一觉醒来, 发现其实不必。
format-type中有%[]这样的type field。如果读取的字符串, 不是以空格来分隔的话, 就可以使用%[]。
%[]类似于一个正则表达式。[a-z]表示读取a-z的所有字符, [^a-z]表示读取除a-z以外的所有字符。
所以那个问题也就迎刃而解了:
sscanf("2006:03:18 - 2006:04:18", "%[0-9,:]- %[0-9,:]", sztime1, sztime2);
```

分类: [编程语言](#), [不得不转的好贴](#)

好文要顶

关注我

收藏该文







Walter L

关注 - 0

粉丝 - 18

6

0

[+加关注](#)

« 上一篇: [C中的volatile关键字用法 \(转\)](#)  
» 下一篇: [new的六种重载形式 \(转\)](#)

posted @ 2009-11-28 17:16 Walter L 阅读(88350) 评论(2) 编辑 收藏

评论列表

- #1楼 2015-02-03 14:16 icebrain

sscanf("2006:03:18", "%d:%d:%d", a, b, c);有误吧, 应该是sscanf("2006:03:18", "%d:%d:%d", &a, &b, &c);

支持(2) 反对(0)
- #2楼 2015-12-24 19:32 gjbmxy

读取的字符串有空格的话%[空格]就是了

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库  
【福利】微软Azure给博客园的你专属双重好礼  
【推荐】融云发布 App 社交化白皮书 IM 提升活跃超 8 倍  
【推荐】BPM免费下载