

sam的学习家园

在这里记录我的学习历程，温故知新。

[目录视图](#)[摘要视图](#)[RSS](#) [订阅](#)

个人资料



笨小鸟

访问：98878次

积分：1408

等级：

[博客专家福利](#) [有奖试读&征文——我们在互联网上奋斗的故事](#) [10月推荐文章汇总](#) [加入“技术热心人”，赢丰厚奖品](#)

pcre函数详解

分类：[C++](#)

2012-05-16 20:42

5419人阅读

[评论\(0\)](#)

[收藏](#)

[举报](#)

[正则表达式](#)

[list](#)

[buffer](#)

[perl](#)

[存储](#)

[byte](#)

PCRE是一个NFA正则引擎，不然不能提供完全与Perl一致的正则语法功能。但它同时也实现了DFA，只是满足数学意义上的正则。

PCRE提供了19个接口函数，为了简单介绍，使用PCRE内带的测试程序(pcretest.c)示例用法。

1. pcre_compile

原型：

```
#include <pcre.h>
```

排名: 第11805名

原创: 40篇 转载: 20篇

译文: 0篇 评论: 21条

文章搜索

文章分类

C++ (12)

脚本 (1)

学习总结 (12)

数据库 (7)

汇编 (2)

生活 (1)

千奇百怪的面试题 (2)

windows编程 (10)

linux (3)

感兴趣的历史 (2)

mfc (4)

web编程 (0)

工具 (1)

boost库 (1)

java (1)

文章存档

2012年10月 (1)

*pcre *pcre_compile(const char *pattern, int options, const char **errptr, int *erroffset, const unsigned char *tableptr);*

功能: 将一个正则表达式编译成一个内部表示, 在匹配多个字符串时, 可以加速匹配。其同pcre_compile2功能一样只是缺少一个参数errorcodeptr。

参数:

pattern 正则表达式

options 为0, 或者其他参数选项

errptr 出错消息

erroffset 出错位置

tableptr 指向一个字符数组的指针, 可以设置为空NULL

示例:

L1720 *re = pcre_compile((char *)p, options, &error, &erroroffset, tables);*

2. pcre_compile2

原型:

#include <pcre.h>

*pcre *pcre_compile2(const char *pattern, int options, int *errorcodeptr, const char **errptr, int *erroffset, const unsigned char *tableptr);*

功能: 将一个正则表达式编译成一个内部表示, 在匹配多个字符串时, 可以加速匹配。其同pcre_compile功能一样只是多一个参数errorcodeptr。

参数:

pattern 正则表达式

options 为0, 或者其他参数选项

errorcodeptr 存放出错码

errptr 出错消息

2012年05月 (5)

2012年04月 (7)

2012年03月 (7)

2012年02月 (1)

展开

阅读排行

JAVA_HOME环境配置

(14508)

mysql中char,varchar,tes

(11772)

C语言之断言

(6191)

vector,map,list,queue区!

(5603)

pcre函数详解

(5397)

c++子类 and 父类成员函数!

(4022)

mysql中datetime到time_

(3188)

lua学习实例,都是自己写

(2569)

CListCtrl选中某行

(2371)

结构体的构造函数

(2313)

评论排行

C语言之断言

(7)

c++子类 and 父类成员函数!

(4)

static_cast, dynamic_cast

(3)

CreateThread和_beginthread

(2)

STL各个容器性能比较

(1)

c/c++返回结构体方法和

(1)

eroffset 出错位置

tableptr 指向一个字符数组的指针, 可以设置为空NULL

3. pcre_config

原型:

#include <pcre.h>

int pcre_config(int what, void *where);

功能: 查询当前PCRE版本中使用的选项信息。

参数:

what 选项名

where 存储结果的位置

示例:

Line1312 (void)pcre_config(PCRE_CONFIG_POSIX_MALLOC_THRESHOLD, &rc);

4. pcre_copy_named_substring

原型:

#include <pcre.h>

int pcre_copy_named_substring(const pcre *code, const char *subject, int *ovector, int stringcount, const char *stringname, char *buffer, int buffersize);

功能: 根据名字获取捕获的字串。

参数:

code 成功匹配的模式

subject 匹配的串

ovector pcre_exec() 使用的偏移向量

stringcount pcre_exec()的返回值

奇怪问题绑定和监听127. (1)
 c++ new,delete,析构函数 (1)
 CListCtrl选中某行 (1)
 结构体的构造函数 (1)

推荐文章

* TF-IDF算法-Python实现（附源代码）
 * CAS原子操作实现无锁及性能分析
 * Qt on Android: 资源文件系统qrc与assets
 * Android launcher 桌面抽屉切换动画
 * 深入研究Clang（四）Clang编译器的简单分析

最新评论

奇怪问题绑定和监听127.0.0.1把j307533688: loop address与网卡没有关系，是系统的。

结构体的构造函数

Fierralin: 结构体在C语言和C++中的表现好像不一样诶，我在C语言里面定义结构体的构造函数的时候，使用gcc编译...

C语言之断言

CAE-Prime: 学习了。

CreateThread和_beginthread区

陈小润: 顶一个，楼主讲得好详细，好具体

CListCtrl选中某行

dong4464he: 要先使用

stringname 捕获字串的名字
buffer 用来存储的缓冲区
buffersize 缓冲区大小

示例:

```
Line2730 int rc = pcre_copy_named_substring(re, (char *)bptr, use_offsets,
count, (char *)copynamesptr, copybuffer, sizeof(copybuffer));
```

5. pcre_copy_substring

原型:

```
#include <pcre.h>
```

```
int pcre_copy_substring(const char *subject, int *ovector, int stringcount, int
stringnumber, char *buffer, int buffersize);
```

功能: 根据编号获取捕获的字串。

参数:

code 成功匹配的模式
subject 匹配的串
ovector pcre_exec() 使用的偏移向量
stringcount pcre_exec()的返回值
stringnumber 捕获字串编号
buffer 用来存储的缓冲区
buffersize 缓冲区大小

示例:

```
Line2730 int rc = pcre_copy_substring((char *)bptr, use_offsets, count,
i, copybuffer, sizeof(copybuffer));
```

m_listEvent.SetFocus();

CreateThread和_beginthread区
miaoshengwu: 好文章,

C语言之断言

jeryler: 写的不错, 学习了。

C语言之断言

附加信息: 写的很浅显易懂, 比很多著作上的两个非负数之商的迭代算法例子要好多了!!!

c++子类 and 父类成员函数重名

刘尚伟: 没看明白~

C语言之断言

liuxuelinyi: 8错哈, 写得很好。

6. pcre_dfa_exec

原型:

#include <pcre.h>

*int pcre_dfa_exec(const pcre *code, const pcre_extra *extra, const char *subject, int length, int startoffset, int options, int *ovector, int oveccsize, int *workspace, int wscount);*

功能: 使用编译好的模式进行匹配, 采用的是一种非传统的方法DFA, 只是对匹配串扫描一次 (与Perl不兼容)。

参数:

code 编译好的模式

extra 指向一个pcre_extra结构体, 可以为NULL

subject 需要匹配的字符串

length 匹配的字符串长度 (Byte)

startoffset 匹配的开始位置

options 选项位

ovector 指向一个结果的整型数组

oveccsize 数组大小

workspace 一个工作区数组

wscount 数组大小

示例:

```
Line2730 count = pcre_dfa_exec(re, extra, (char *)bptr, len, start_offset,  
                                options | g_notempty, use_offsets, use_size_offsets, workspace,  
                                sizeof(workspace)/sizeof(int));
```

7. pcre_copy_substring

原型:

```
#include <pcre.h>
```

```
int pcre_exec(const pcre *code, const pcre_extra *extra, const char *subject, int length, int startoffset, int options, int *ovector, int ovecsz);
```

功能: 使用编译好的模式进行匹配, 采用与Perl相似的算法, 返回匹配串的偏移位置。。

参数:

code 编译好的模式

extra 指向一个pcre_extra结构体, 可以为NULL

subject 需要匹配的字符串

length 匹配的字符串长度 (Byte)

startoffset 匹配的开始位置

options 选项位

ovector 指向一个结果的整型数组

ovecsz 数组大小

8. pcre_free_substring

原型:

```
#include <pcre.h>
```

```
void pcre_free_substring(const char *stringptr);
```

功能: 释放pcre_get_substring()和pcre_get_named_substring()申请的内存空间。

参数:

stringptr 指向字符串的指针

示例:

```
Line2730        const char *substring;
```

```
int rc = pcre_get_substring((char *)bptr, use_offsets, count,  
                           i, &substring);
```

.....

```
pcre_free_substring(substring);
```

9. pcre_free_substring_list

原型:

```
#include <pcre.h>
```

```
void pcre_free_substring_list(const char **stringptr);
```

功能: 释放由pcre_get_substring_list申请的内存空间。

参数:

stringptr 指向字符串数组的指针

示例:

```
Line2773      const char **stringlist;
```

```
int rc = pcre_get_substring_list((char *)bptr, use_offsets, count,
```

.....

```
pcre_free_substring_list(stringlist);
```

10. pcre_fullinfo

原型:

```
#include <pcre.h>
```

```
int pcre_fullinfo(const pcre *code, const pcre_extra *extra, int what, void *where);
```

功能: 返回编译出来的模式的信息。

参数:

code 编译好的模式

extra pcre_study()的返回值, 或者NULL

what 什么信息

where 存储位置

示例:

```
Line997      if ((rc = pcre_fullinfo(re, study, option, ptr)) < 0)
fprintf(outfile, "Error %d from pcre_fullinfo(%d)/n", rc, option);
}
```

11. pcre_get_named_substring

原型:

#include <pcre.h>

*int pcre_get_named_substring(const pcre *code, const char *subject, int *ovector, int stringcount, const char *stringname, const char **stringptr);*

功能: 根据编号获取捕获的字串。

参数:

code 成功匹配的模式

subject 匹配的串

ovector pcre_exec() 使用的偏移向量

stringcount pcre_exec()的返回值

stringname 捕获字串的名字

stringptr 存放结果的字符串指针

示例:

```
Line2759      const char *substring;
int rc = pcre_get_named_substring(re, (char *)bptr, use_offsets,
count, (char *)getnamesptr, &substring);
```


12. pcre_get_stringnumber

原型:

```
#include <pcre.h>
```

```
int pcre_get_stringnumber(const pcre *code, const char *name);
```

功能: 根据命名捕获的名字获取对应的编号。

参数:

code 成功匹配的模式

name 捕获名字

13. pcre_get_substring

原型:

```
#include <pcre.h>
```

```
int pcre_get_substring(const char *subject, int *ovector, int stringcount, int  
stringnumber, const char **stringptr);
```

功能: 获取匹配的子串。

参数:

subject 成功匹配的串

ovector pcre_exec() 使用的偏移向量

stringcount pcre_exec()的返回值

stringnumber 获取的字符串编号

stringptr 字符串指针

14. pcre_get_substring_list

原型:

```
#include <pcre.h>
```

```
int pcre_get_substring_list(const char *subject, int *ovector, int stringcount, const char ***listptr);
```

功能：获取匹配的所有子串。

参数：

subject 成功匹配的串

ovector pcre_exec() 使用的偏移向量

stringcount pcre_exec()的返回值

listptr 字符串列表的指针

15. pcre_info

原型：

```
#include <pcre.h>
```

```
int pcre_info(const pcre *code, int *optptr, int *firstcharptr);
```

已过时，使用pcre_fullinfo替代。

16. pcre_maketables

原型：

```
#include <pcre.h>
```

```
const unsigned char *pcre_maketables(void);
```

功能：生成一个字符表，表中每一个元素的值不大于256，可以用它传给pcre_compile()替换掉内建的字符表。

参数：

示例：

```
Line2759 tables = pcre_maketables();
```

17. pcre_refcount

原型:

```
#include <pcre.h>
```

```
int pcre_refcount(pcre *code, int adjust);
```

功能: 编译模式的引用计数。

参数:

code 已编译的模式

adjust 调整的引用计数值

18. pcre_study

原型:

```
#include <pcre.h>
```

```
pcre_extra *pcre_study(const pcre *code, int options, const char **errptr);
```

功能: 对编译的模式进行学习, 提取可以加速匹配过程的信息。

参数:

code 已编译的模式

options 选项

errptr 出错消息

示例:

```
Line1797 extra = pcre_study(re, study_options, &error);
```

19. pcre_version

原型:

```
#include <pcre.h>
```

```
char *pcre_version(void);
```

功能：返回PCRE的版本信息。

参数：

示例：

```
Line1384 if (!quiet) fprintf(outfile, "PCRE version %s/n/n", pcre_version());
```

上一篇 [windos下编译正则表达式库pcre](#)

下一篇 [regex 库](#)

顶

0

踩

0

主题推荐

[正则表达式](#)

[存储](#)

[正则](#)

[结构](#)

[内存](#)

猜你在找

[C语言库函数access的使用](#)

[nginx module 开发谈\(1\)](#)

[python读取指定字节及位置的文件内容](#)

[keil MDK编译器警告和错误详解](#)

[网络性能测试和标准](#)

[Nginx + Lua + 共享内存实现动态查询（简单例子）](#)

[zlog使用手册](#)

[Oracle数据库表命名规则及验证中文表名的方法](#)

[黑马程序员.Android攻城狮.JAVA基础.1.2.基础语法](#)

[编解码学习笔记（十一）：Flash Video系列](#)

- | | | | | |
|--------------------------|---------------------------|----------------------------|----------------------------|-----------------------------|
| 1 c语言 | 5 上海面包加盟 | 9 微量元素测试仪 | 13 北京会计网首页 | 17 眼科医院挂号 |
| 2 c9 | 6 全球拍软件下载 | 10 六安装修网 | 14 农村平房设计 | 18 2014港澳台联 |
| 3 北京婚姻中介 | 7 电液推杆 | 11 红宝石蛋糕加盟 | 15 微型电动球阀 | 19 酋长咖啡 |
| 4 c++ | 8 产妇恢复加盟 | 12 带浴缸的酒店 | 16 怎样加盟面包 | 20 拉皮修复 |

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点,不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker
OpenStack VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC
WAP jQuery BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML
LBS Unity Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra
CloudStack FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App
SpringSide Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP
HBase Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 