

C语言正则表达式 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/647/2021_2022_C_E8_AF_AD_E8_A8_80_E6_AD_A3_c97_647470.htm

如果用户熟悉Linux下的sed、awk、grep或vi，那么对正则表达式这一概念肯定不会陌生。由于它可以极大地简化处理字符串时的复杂度，因此现在已经在许多Linux实用工具中得到了应用。千万不要以为正则表达式只是Perl、Python、Bash等脚本语言的专利，作为C语言程序员，用户同样可以在自己的程序中运用正则表达式。标准的C和C++都不支持正则表达式，但有一些函数库可以辅助C/C++程序员完成这一功能，其中最著名的当数Philip Hazel的Perl-Compatible Regular Expression库，许多Linux发行版本都带有这个函数库。编译正则表达式为了提高效率，在将一个字符串与正则表达式进行比较之前，首先要

用regcomp()函数对它进行编译，将其转化为regex_t结构：int regcomp(regex_t *preg, const char *regex, int cflags)。参数regex是一个字符串，它代表将要被编译的正则表达式；参数preg指向一个声明为regex_t的数据结构，用来保存编译结果；参数cflags决定了正则表达式该如何被处理的细节。如果函数regcomp()执行成功，并且编译结果被正确填充到preg中后，函数将返回0，任何其它的返回结果都代表有某种错误产生。

匹配正则表达式一旦用regcomp()函数成功地编译了正则表达式，接下来就可以调用regexec()函数完成模式匹配：int regexec(const regex_t *preg, const char *string, size_t nmatch, regmatch_t pmatch[], int eflags)。typedef struct { regoff_t rm_so. regoff_t rm_eo. } regmatch_t。参数preg指向编译后的正则

表达式，参数string是将要进行匹配的字符串，而参数nmatch和pmatch则用于把匹配结果返回给调用程序，最后一个参数eflags决定了匹配的细节。在调用函数regexec()进行模式匹配的过程中，可能在字符串string中会有多处与给定的正则表达式相匹配，参数pmatch就是用来保存这些匹配位置的，而参数nmatch则告诉函数regexec()最多可以把多少个匹配结果填充到pmatch数组中。当regexec()函数成功返回时，从string pmatch[0].rm_so到string pmatch[0].rm_eo是第一个匹配的字符串，而从string pmatch[1].rm_so到string pmatch[1].rm_eo，则是第二个匹配的字符串，依此类推。释放正则表达式 无论什么时候，当不再需要已经编译过的正则表达式时，都应该调用函数regfree()将其释放，以免产生内存泄漏。 void regfree(regex_t *preg). 函数regfree()不会返回任何结果，它仅接收一个指向regex_t数据类型的指针，这是之前调用regcomp()函数所得到的编译结果。如果在程序中针对同一个regex_t结构调用了多次regcomp()函数，POSIX标准并没有规定是否每次都必须调用regfree() 函数进行释放，但建议每次调用regcomp()函数对正则表达式进行编译后都调用一次regfree()函数，以尽早释放占用的存储空间。报告错误信息 如果调用函数regcomp()或regexec()得到的是一个非0的返回值，则表明在对正则表达式的处理过程中出现了某种错误，此时可以通过调用函数regerror()得到详细的错误信息。 size_t regerror(int errcode, const regex_t *preg, char *errbuf, size_t errbuf_size). 参数errcode是来自函数regcomp()或regexec()的错误代码，而参数preg则是由函数regcomp()得到的编译结果，其目的是把格式化消息所必须的上下文提供给regerror()函数。在执行函

数regerror()时，将按照参数errbuf_size指明的最大字节数，在errbuf缓冲区中填入格式化后的错误信息，同时返回错误信息的长度。应用正则表达式 最后给出一个具体的实例，介绍如何在C语言程序中处理正则表达式。

```
#include #include  
#include /* 取子串的函数 */ static char* substr(const char*str,  
unsigned start, unsigned end) { unsigned n = end - start. static char  
stbuf[256]. strncpy(stbuf, str start, n). stbuf[n] = 0. return stbuf. }
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com