Http://www.tp-link.com.cn

TP-LINK®

新员工分享交流活动

SMB Switch

深圳市普联技术有限公司
SHENZHEN TP-LINK TECHNOLOGIES CO., LTD.

正则表达式

英文：**regular expressions**

简写：**regex**或者**regexp**

例如：**GNU Regex Library**正则表达式库

```
……
#include <sys/types.h>
#include <regex.h>
……
```

Regex 语法和风格（1）

- **POSIX Basic Regular Expressions**

  元字符（metacharacter)：. * ^ [ ]等

  示例：

  Pattern: ^[hc]at

  matches "*hat*" and "*cat*", but only at the beginning of the string or line.

  So not “that”.

  Pattern:[^b]at

  matches all strings matched by .at except "*bat*".

  Pattern: \[.*\]

  matches characters surrounded by “[” and “]” since the brackets are escaped, for example: “*[]*” 、“*[a]*”、“[ab]”.

Regex 语法和风格（2）

- **POSIX Extended Regular Expressions**

  增加了元字符：？ +等

  示例：

  Pattern: [hc]+at

  matches "hat", "cat", "hhat", "chat", "hcat", "ccchat", and so on, but not "at".

  Pattern:[hc]?at

  matches "hat", "cat", and "at".

  Pattern: cat|dog

  matches "cat" or "dog".

Regex 语法和风格（3）

- **Perl-derivative regular expressions**

  Java, JavaScript, Python, Ruby,Microsoft's .NET Framework all use regular expression syntax similar to Perl's.

  Boost and PHP support multiple regular expression flavors.

- \ always escapes a non-alphanumeric character. （ *character classes* ）
  \s匹配任意的空白符 [[:space:]] [\t\r\n\v\f]
- Perl has the concept of lazy quantification

  Pattern: \[.*\] \[[^\]]*\]

  String: [ab]c]

  Result: [ab]? [ab]c]?

GNU Regex Library

包含头文件

```
#include <sys/types.h>
#include <regex.h>
```

函数

int regcomp (……)	编译正则表达式pattern
int regexec (……)	执行匹配
void regfree (……)	释放编译好的正则表达式
size_t regerror (……)	获得错误码

子正则表达式：

(\d{1,3}\.){3}\d{1,3}

简单的IP地址匹配表达式，其中(\d{1,3}\.)是一个子正则表达式。

习题中使用到的表达式

pattern:　　^[[:space:]]*#include[[:space:]]+".+"[[:space:]]*$

解析：^[[:space:]]*表示以空白符开头，出现的次数是零个或者多个；#include表示紧跟#include字符串；[[:space:]]+表示#include字符之后是一个或者多个空白符；然后是双引号字符”；.+表示一个或者多个任意除换行符以外的字符；然后是双引号字符”；最后，[[:space:]]*$表示以零个或者多个空白符到行尾。

pattern:　　"[^;]*/*[^;]*"

解析：双引号字符”开始，[^;]*表示不包括分号的零个或者多个字符，然后紧跟/*字符串，然后再是[^;]*，不包括分号的零个或者多个字符，最后以双引号结束。这是用来判断一个/*字符是为字符串字符，还是注释标示的正则表达式，如果匹配，则表示该/*字符是字符串，如果不匹配，则为注释开始标示。（这是其中的一种判定方式，能够判定绝大部分字符串形式的/*符号）

应用举例

- 删除空行 ^[\t]*\n
  演示：EditPlus
  使用替换工具，勾选使用正则表达式选项

- 替换带有半角括号的多行
- 删除每一行行尾的指定字符

Regex & globbing

通配符

多用在文件名上，在LINUX下，命令find ls cp等

正则表达式：

多用在文本过滤工具中，是针对字符串的处理，在LINUX下，命令grep sed等

没有提及的知识

- 零宽断言
- 负向零宽断言
- 处理选项
- 平衡组/递归匹配

……

《精通正则表达式（第三版）》Jeffrey E.F.Friedl著 电子工业出版社

others

Know everything from something

Know something from everything

……