awk系列1

awk系列1主要是介绍如何使用awk和正则表达式过滤文本或文件中的字符串。

**什么是正则表达式**

正则表达式可以定义为代表若干个字符序列的字符串，可以是先一条命令过滤一个文本的输出、编辑文本或配置文件的一部分等。

**正则表达式的特点**

正则表达式由以下内容组合而成：

* 普通字符：例如空格、下划线、A-Z、a-z、0-9
* 可以扩展为普通字符的**元字符**，它们包括：
  + (.)：它匹配除了换行符外的任何单个字符
  + (\*)：它匹配零个或多个在其之前紧挨着的字符
  + [character(s)]：它匹配任何由其中的字符/字符集指定的字符，你可以使用连字符(-)代表字符区间，例如[a-f]、[1-5]等；
  + ^：它可以匹配文件中一行的开头
  + $：它可以匹配文件中一行的结尾
  + \：这是一个转义字符

**awk**

awk的语法一般如下：

* awk "script" filename

这里的 'script' 是由一个awk可以理解并应用于filename的命令集合。

它可以通过读取文件中的给定行，复制该行的内容并在该行上执行脚本的方式工作。这个过程会在该文件中的所有行上重复。

这个脚本的'script'中的内容的格式是：'/pattern/ action'，其中pattern是一个正则表达式，而action是当awk在该行中找到此模式时应当执行的动作。

**awk实战**

* 下面这个例子可以打印文件/etc/hosts中的所有行，因为没有指定任何的模式。

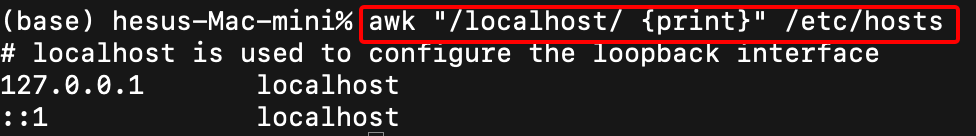


等价于：



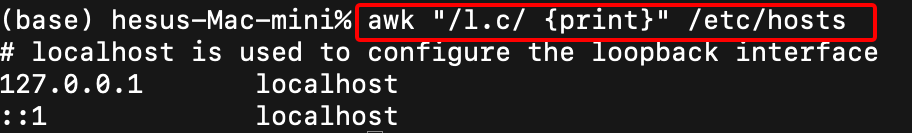
**结合模式使用awk**

在下面的例子中，指定了模式localhost，因此awk将匹配文件/etc/hosts中有localhost的那些行。



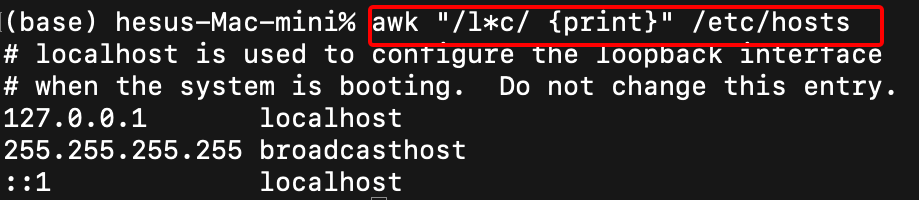
**在awk模式中使用通配符(.)**

.表示匹配任意的**一个**字符，因此下面的正则表达式其实表示将匹配**l一个字符c，**比如将匹配：loc、localhost、localnet等字符。

****

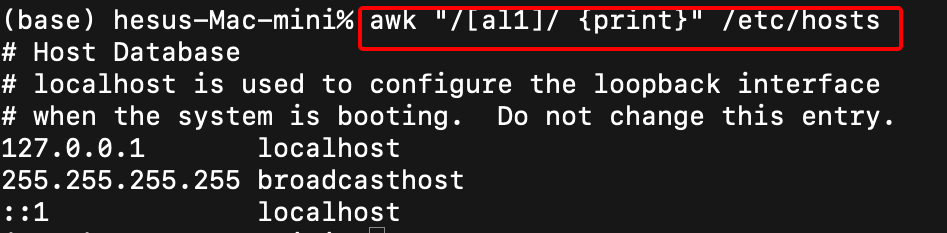
**在awk模式中使用字符(\*)**

\*表示匹配零个或任意个的任意字符，因此下面的正则表达式将匹配l和c之间有0个或任意多个字符的字符串，比如：localhost、localnet、lines、capable等字符串。



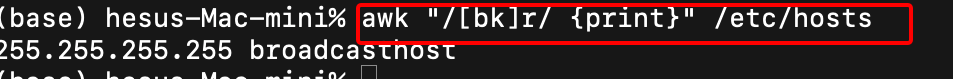
**结合集合[character(s)]使用awk**

比如集合[al1]，那么awk将匹配文件/etc/hosts中所有包含字符a或l或1的字符串。



**使用awk打印文件中匹配的字符**

下一个例子匹配以b或k开头，然后后面跟着一个r的字符串：

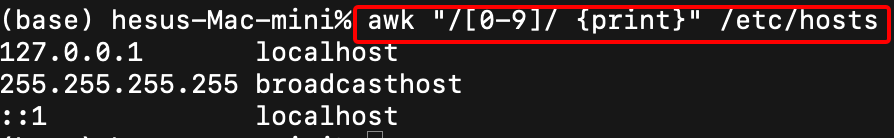


**以范围的方式指定字符**

awk所能理解的字符：

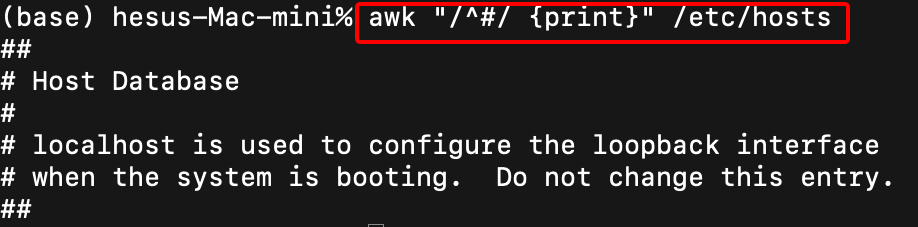
* [0-9]代表一个单独的字符
* [a-z]代表一个单独的小写字母
* [A-Z]代表一个单独的大写字母
* [a-zA-Z]代表一个单独的字母
* [a-zA-Z0-9]代表一个单独的字母或者数字

让我们看下面的例子，它实现了打印文件中有数字的行：



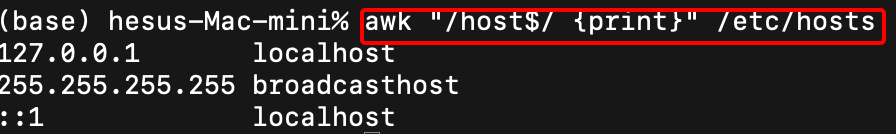
**结合元字符(^)使用awk**

在下面的例子中，它匹配所有以给定模式开头的行：



**结合元字符($)使用awk**

$将匹配以给定模式结尾的行：



**结合转义字符(\)使用awk**

它将允许你将该转义字符后面的字符作为文字，即理解其字面的意思。

比如我想要匹配具有"$25.0"的一行，那么我应该使用下面的命令：

* awk "/\$25.0/ {print}" /etc/hosts