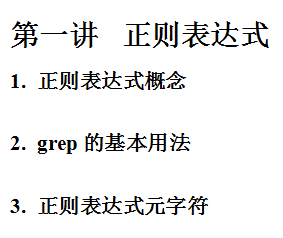
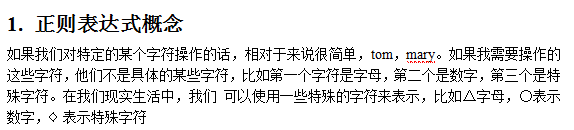
老段 Sed-awk



正则表达式 就是一堆字符组合起来 来匹配某一种模式



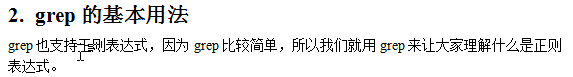
现在想用计算机语言 进行表达 这时候要使用正则表达式的元字符进行表达

我们要依赖于某一种语言

所以 我们就以grep命令来讲解正则表达式

grep也支持正则表达式

grep相对比较简单

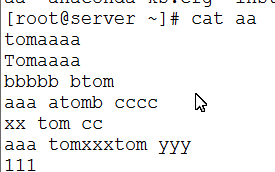


Grep是用来过滤含有特定字符的行

用法



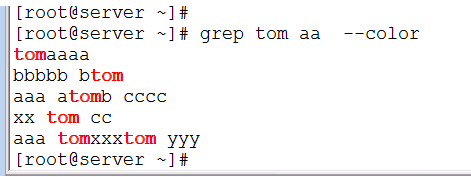
举例：

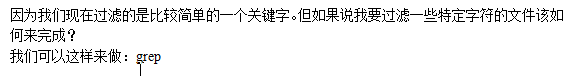




这就是 要在aa中查找含有tom的行

 --color会把颜色显示出来





使用正则表达式的元字符

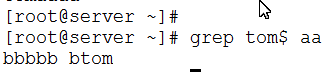


^ 表示行开头

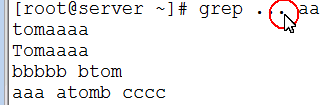


在aa中查找以tom开头的行

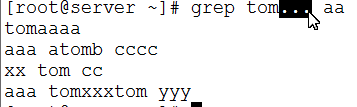
$表示行尾 一行的结束



.表示单个任意字符 匹配除了换行符之外所有的字符



查找含有三个字符的那些行



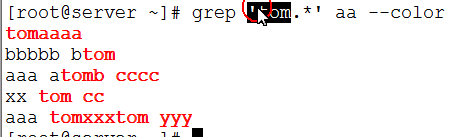
.也是可以匹配空格的

\*表示前面那个字符可以出现任意次



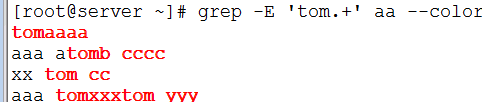
为了防止shell把这个通配符解析了

所以加上’’引起来

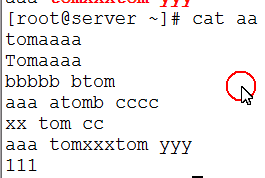


Shell中的\* 不能混淆

+ 意思是 表示前面的字符至少出现一次



源文件的内容





注意 .不匹配回车 所以 这个btom结尾的这一行 不能被过滤出来

扩展的grep的使用

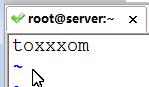


刚才讲的两个元字符 \*和+这两个元字符 都是贪婪型的

所谓的贪婪型 意思是 进行最长的匹配

比如

处理一个文件 bb



过滤一下



-E是使用扩展的grep P表示使用Perl的正则表达式



这里面过滤的是一个最长的匹配



仅仅匹配到tom是没有截止的 还是继续向后匹配

这是一种最长的匹配

这种匹配方法 叫做 贪婪匹配

有时候 并不需要这样的结果 需要最短匹配 也成为 懒惰性匹配

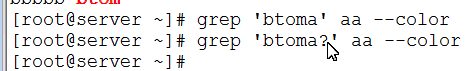
如何做一个懒惰匹配？



只要在\*后面加上?就可以了 这个就是最短匹配

除了+和\*之外 还有一个?

表示他前面的字符可以有 也可以没有【这个不同于\*后面的?】





从上图可以看出来 ?前面的a可以出现 也可以没有 都能匹配出来

**注意 grep不支持 + ?这个元字符 如果要使用 只能使用扩展的grep**

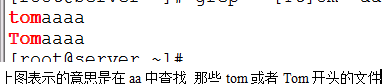
使用egrep或者 grep –E这个命令

[]的意思是 匹配一个字符

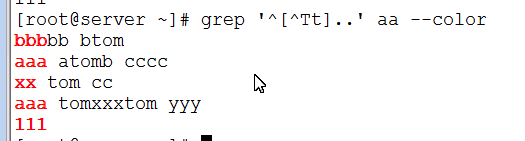
出现在[]里面的字符都是或的关系

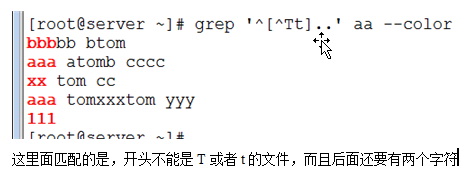


上图表示的意思意思 是 在aa中查找Tom或者tom开头的行



^如果出现在[]外面的话 表示开头的意思 如果出现在[]中表示否定的意思





刚才是给大家讲了^和$ 匹配是开头是行的开头 和 行的结尾

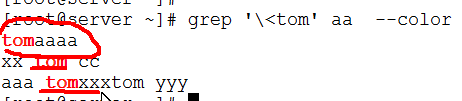
如果想匹配单词的开头 使用 \<

aa里面的内容



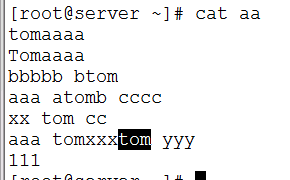
这个里面有很多是以tom开头 但是不是行的开头

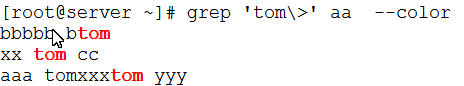
这样 过滤出来的单词的开头是tom开头的

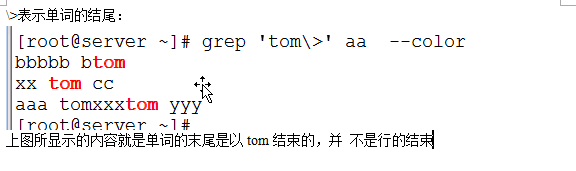


这里面过滤的不是行的开头 是单词的开头

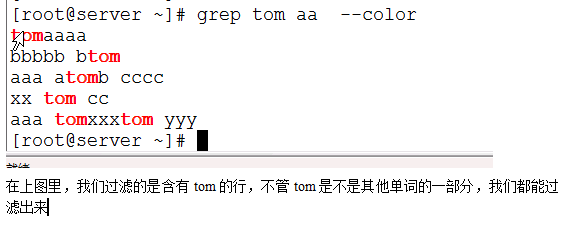
单次的结尾 \>表示单词的结尾







应该说 并不一定是行的结束



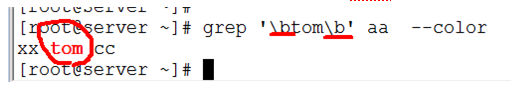
这个是不是单词的一部分 仅仅含有tom

很多时候 就想找tom这个单次 不是含有tom的单词

做法\b



哟了\b 就可以定位一个单词



在正则表达式中 可以做标签 可以在匹配模式中 给某一部分 做一个标签 这样 在后面 就可以直接引用它

如果我们想对某一部分作标签的话 只要使用\(\) 就可以了 --- 实际上就是分组的意思

如果想引用的话 就可以使用\n ---- n是一个数字



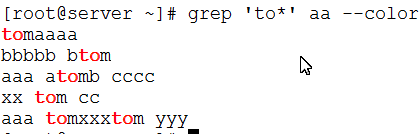
这样做了一个标签 因为是第一个()括起来的 可以使用\1进行引用



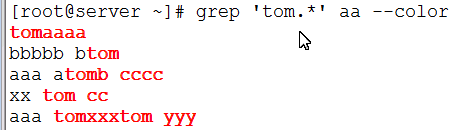
这个 grep ‘\(tom\)…\1’ aa –color就相当于 grep ‘tom…tom’ 这样 要写两次tom 所以 写成一次 分组 然后引用这个组中的内容 这样 可以简化命令的书写

刚才有一个\* ?还有+

其中 \*和+会匹配无穷个字符 有时候 对要匹配的字符做一些限制



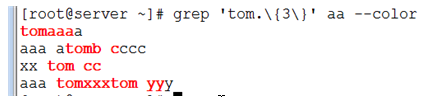
也就是o可以出现任意次



要匹配tom后面 一个字符 或者多个字符 或者不跟

Tom后面有字符 可以有无穷多个 因为\*表示任意个数

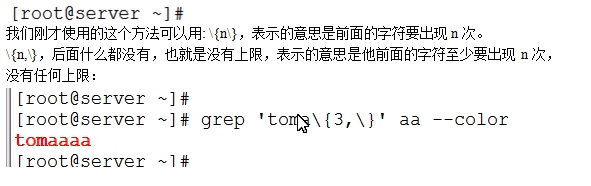
希望匹配tom后面有3个字符

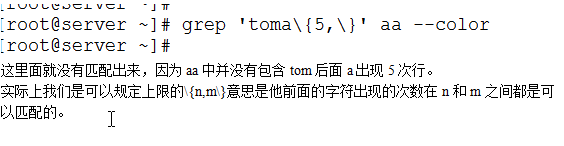






刚才使用的这个方法 可以用\{n\}来表示





使用grep是时候 刚才就是一个中括号 匹配的就是一个字符

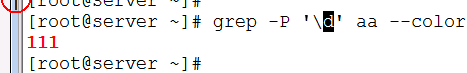
如果相匹配数字 可以使用[0-9]



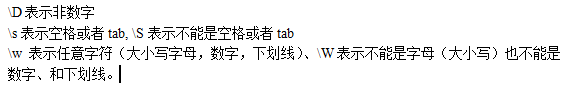
可以使用0-9来表示数字



\d是不支持的 所以 用Perl表达式



\D表示非数字



正则表达式 也支持 国际字符

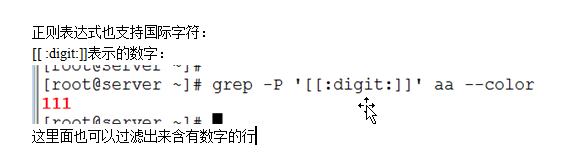
常见的国际字符

表示一个数字

刚才表示一个数字的另一种写法



这也可以过滤出含有数字的行



如果表示字母的话

可以使用 [[:alpha:]]

如果单独表示大写字母的话 [[:upper]]

