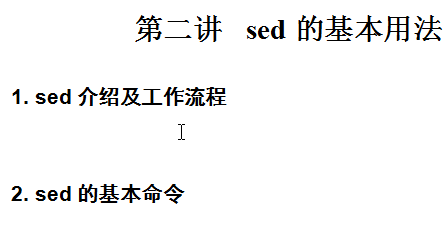
老段 Sed-awk



什么事sed

很多时候 linux下面 都需要 去编辑一些文件

所使用的编辑器 一般就是vi活着vim

vim属于屏幕编辑器

屏幕编辑器：所谓的屏幕编辑器 编辑一个文件的时候 需要把这个文件打开 这里面 存在两个问题：

(1). 因为我们需要把这个文件打开 所以 如果一个比较大的文件将会消耗很多的内存

(2). 当我们在编辑文件的时候 必须要互动 就是 用户和计算机有互动 这样的话 如果在写脚本的时候 很难调用vi的 ---- 写脚本 希望直接根据需求编辑文件 但是 vim就很难实现 这个时候 需要使用其他的编辑器

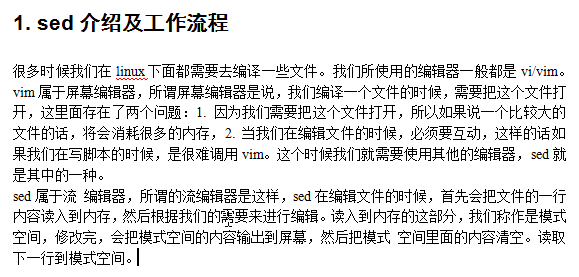
Sed就是其中的一种编辑器

Sed属于流编辑器

流编辑器：sed在编辑文件的时候 首先会把文件的一行内容读入到内存 然后 根据我们的需要进行编辑

**读入到内存的这部分 成为是模式空间**

修改完之后 会把模式空间的内容显示输出到屏幕 然后把模式空间的内容清空 读取下一行到模式空间



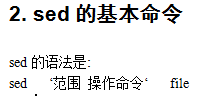
这样就会一行行读取

这样就避免了一次性读取读取整个文件

Sed的具体用法

Sed的语法：





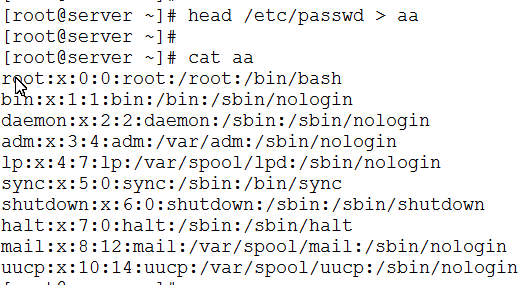
如果范围没有指定的话 则会对全文的每一行进行操作

有时候 需要指定操作范围

如何指定范围？

指定范围的方法：

1. 直接写一个数字 表示对多少行进行操作

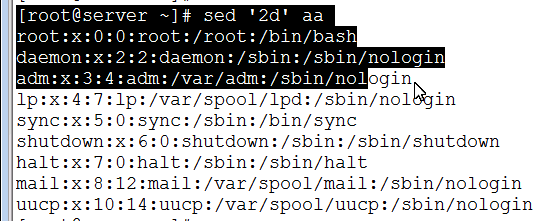


现在想删除第二行

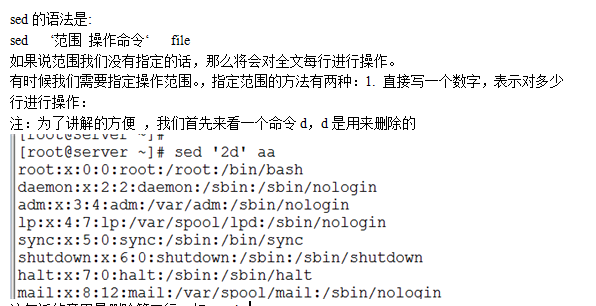
为了讲解的方便

首先来看一个命令d

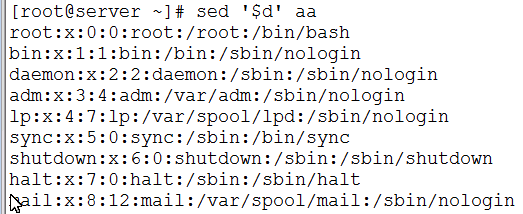
d是用来删除得



这就表示我要删除第二行



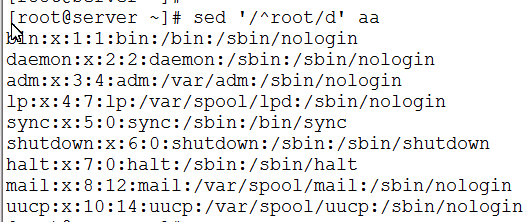
如果要删除最后一行



【解决了 比如用GP查询的结果 有一个表头 但是 结果很大 所以删除的时候 用这个sed命令操作非常方便】

2. 使用正则表达式表示范围

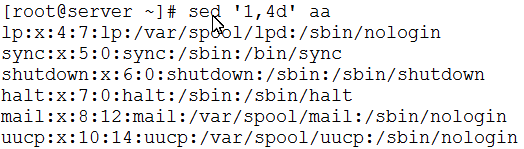
使用正则表达表达范围 必须使用//进行隔开



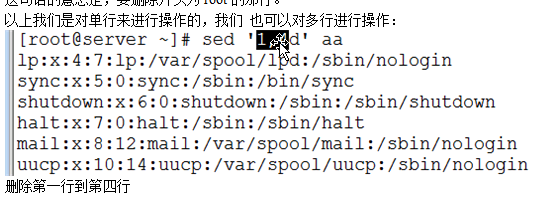


以上是对单行进行操作的

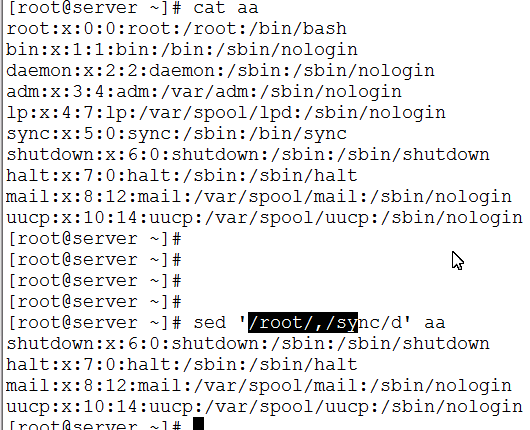
也可以对多行进行操作



这个表示要删除第1-4行的内容



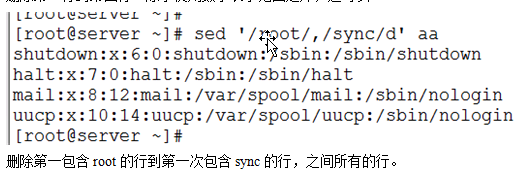
除了使用数字表示范围之外



/root/表示包含root

/sync/表示包含sync

Sed ‘/root/,/sync/d’ aa 表示删除第一个包含root到第一个sync之间所有的行



回头再看一下 sed如何工作？

Sed首先读取文件的一行 读取完一行 就操作下一行

以上是讲的范围

如果说没有指定范围 就会对全文进行操作



看到的就是把这个命令 就把aa中的所有内容全部删除了

【注意 每次操作完之后 sed就会自动打印出这个操作之后的原始文件的内容 这个原始文件在这个命令修改了 所以 展示的就是**模式空间**（源文件的内容没有被修改）的结果（实际上就是处理完之后 输出这一行 然后删除模式空间的内容 展示是处理完每一行打印的结果） 这个和vim的操作是一样的 就是 我操作完之后 就能看到结果】

D是一个命令

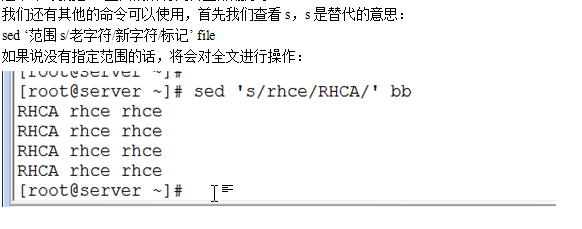
还有其他命令

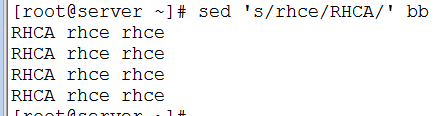
命令s 是替换的意思

/老字符/新字符/标记

这个是在使用替换的时候 的一种用法

如果没有指定范围的话 将会对全文进行操作

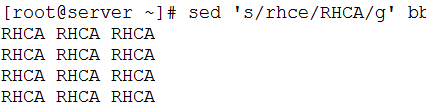




这就是把全文的rhce全部换成RHCA

如果想替换所有的 或者个别的

可以使用 标记 进行处理

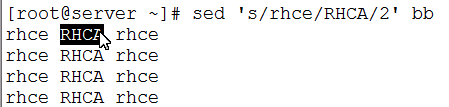


最后面加上g表示 对每行所有的都要做替换

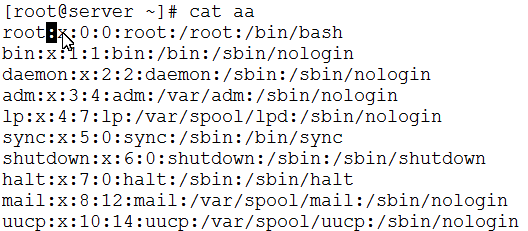
如果没有g的话 只会替换第一个

如果替换第二个 或者第三个

我就是在标记的位置写上一个数字



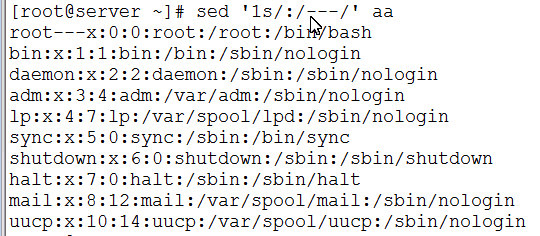
这样就会替换第二个关键字

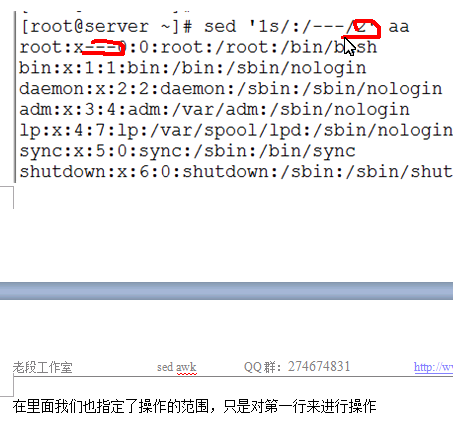


Aa中有很多：

想把第一行的第二个:换成-

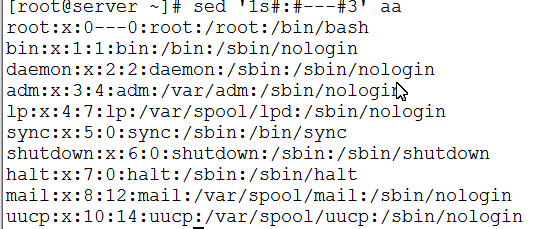
第一行操作 所以 范围写成1





我们在执行替换操作的时候 不一定非要使用/来隔开 可以使用任意字符

比如：



用#也是可以的

这个仅仅在s操作中生效

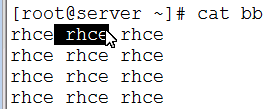


在sed用法部分

老字符部分是支持正则表达式的

但是 新字符的位置是不支持正则表达式的

【所以 范围 替换的老字符都可以使用正则表达式】

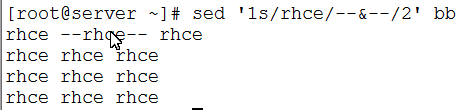


现在想把第一行的rhce换成



&标示的是前面的关键字

是对前面的一个引用



从这里面可以看到 &在替换部分是有意义的 代表的前面的关键字

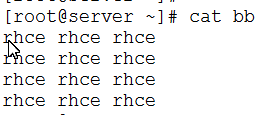
除了& 还有\ 表示去掉特殊字符的特殊意义

替换字符部分 有一个&和\可以使用

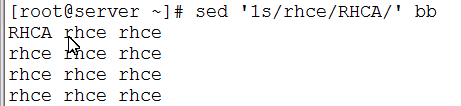
刚才讲了两个命令 s和d

如果说 对特定的返回做多个操作的话

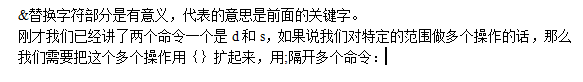
需要把这多个操作用{}括起来

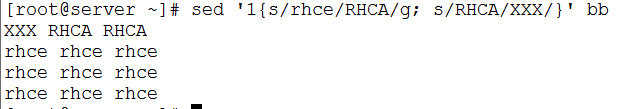


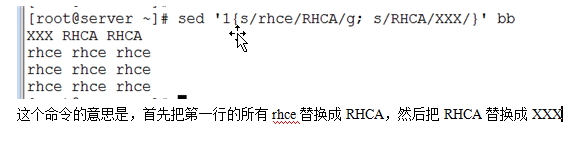
现在需要把第一行的rhce换成RHCA



现在 对第一行做多个操作 用分号隔开多个命令



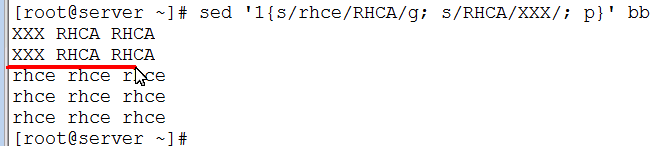




从这些操作中 可以看到sed读取文件的一行 然后进行操作

操作完成之后 会把结果显示在屏幕中

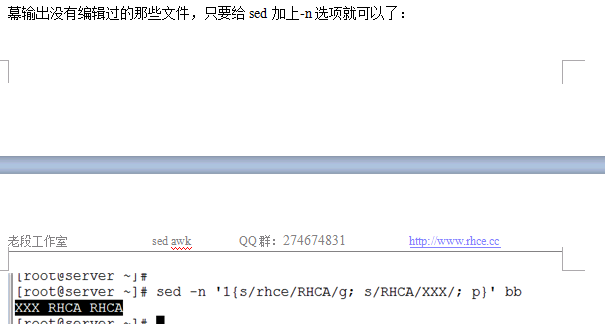
如果说我们想查看模式空间里面的内容 可以使用p命令



第一行是正常的显示以外 第二行就是就是显示模式空间的内容

如果我们不想在屏幕输出没有编辑过的那些文件 只要在 –a



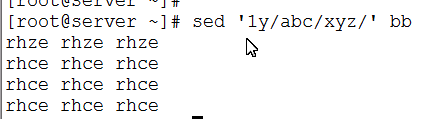


刚才讲解是替换一个单词

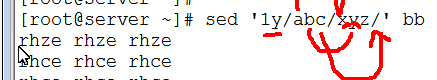
如果说 想替换单个字符的话 可以使用y命令

举例

Y也是替换



含义就是



因为是对第一行的a->x b->y c->z

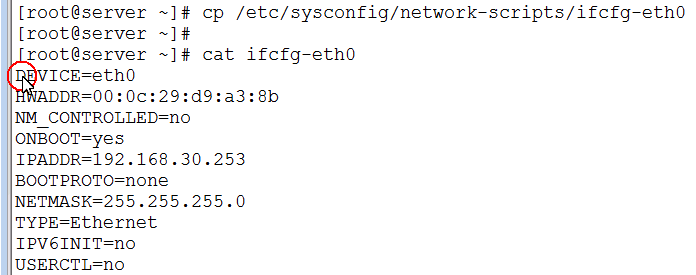
做替换的时候 不是吧abc整体替换成xyz

以上讲了3个命令 d s y

还有其他简单的命令

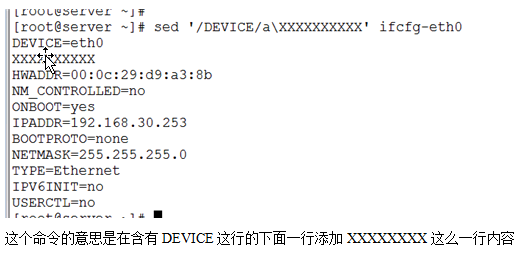
a\的意思是追加 在关键词后面一行加上特定的一行内容

很多时候对网卡进行操作



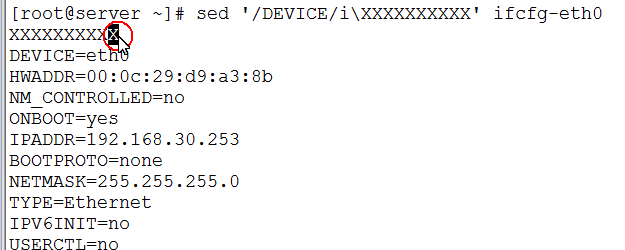
现在想在DEVICE=eth0下面先添加上XXX

这个是对包含/DEVICE/的这一行进行操作



可以在上一行产生内容

\i是爱关键词的上一行产生内容



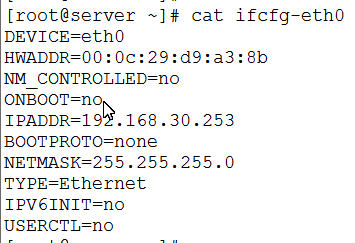
Append --- a\

Insert ---- i\

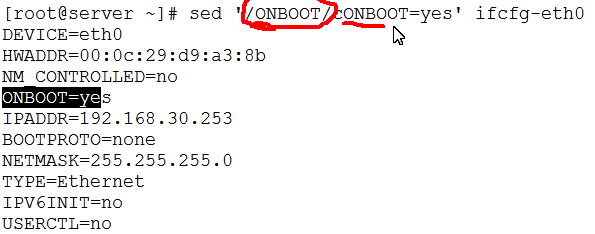
无论是a\还是i\都是在原有的一行上面或者下面添加一行

如果想修改内容 可以使用c

直接举例

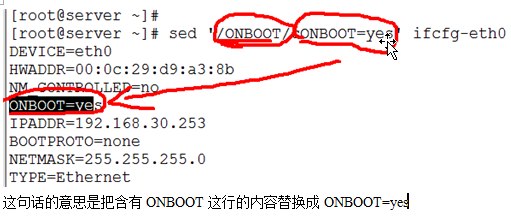


希望把脚本中的ONBOOT修改成yes



把包含某一行的整行的内容进行替换

这句话的意思是 把含有ONBOOT的内容 替换成 替换成ONBOOT=yes



**以上是sed的基本的命令**

**所做的所有的操作都是在模式空间中进行的**

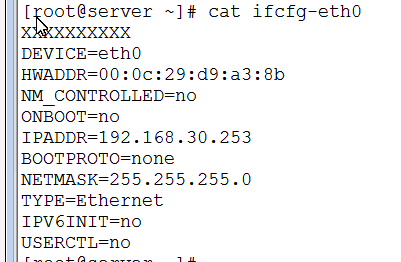
**也就是在内存中操作的**

**并没有改变源文件的内容**

如果要改变源文件的内容

加上-i之后 屏幕上面没有做任何输出

因为这样操作的话 直接改变了源文件的内容



刚才讲的是sed的基本的用法 sed的基本流程