**[Linux的简单shell脚本中修改文件操作](http://www.cnblogs.com/hnrainll/archive/2012/01/08/2316622.html)**

理解linux sed命令(2010-02-27 18:21:20)

1. Sed简介

sed 是一种在线编辑器，它一次处理一行内容。处理时，把当 前处理的行存储在临时缓冲区中，称为“模式空间”（pattern space），接着用sed命 令处理缓冲区中的内容，处理完成后，把缓冲区的内容送往屏幕。接着处理下一行，这样不断重复，直到文件末尾。文件内容并没有改变，除非你使用重定向存储输 出。Sed主要用来自动编辑一个或多个文件；简化对文件的反复操作；编写转换程序等。

2. 定址

可以通过定址来定位你所希望编辑的行，该地址用数字构成，用逗号分隔的两个行数表示以这两行为起止的行的范围（包括行数表示的那两行）。如1，3表示1，2，3行，美元符号($)表示最后一行。范围可以通过数据，正则表达式或者二者结合的方式确定。

3. Sed命令格式

调用sed命令有两种形式：

sed [options] 'command' file(s)

sed [options] -f scriptfile file(s)

4.常见的Sed命令

a

在当前行后面加入一行文本。

b lable

分支到脚本中带有标记的地方，如果分支不存在则分支到脚本的末尾。

c

用新的文本改变本行的文本。

d

从模板块（Pattern space）位置删除行。

D

删除模板块的第一行。

i

在当前行上面插入文本。

h

拷贝模板块的内容到内存中的缓冲区。

H

追加模板块的内容到内存中的缓冲区

g

获得内存缓冲区的内容，并替代当前模板块中的文本。

G

获得内存缓冲区的内容，并追加到当前模板块文本的后面。

l

列表不能打印字符的清单。

n

读取下一个输入行，用下一个命令处理新的行而不是用第一个命令。

N

追加下一个输入行到模板块后面并在二者间嵌入一个新行，改变当前行号码。

p

打印模板块的行。

P（大写）

打印模板块的第一行。

q

退出Sed。

r file

从file中读行。

t label

if分支，从最后一行开始，条件一旦满足或者T，t命令，将导致分支到带有标号的命令处，或者到脚本的末尾。

T label

错误分支，从最后一行开始，一旦发生错误或者T，t命令，将导致分支到带有标号的命令处，或者到脚本的末尾。

w file

写并追加模板块到file末尾。

W file

写并追加模板块的第一行到file末尾。

!

表示后面的命令对所有没有被选定的行发生作用。

s/re/string

用string替换正则表达式re。

=

打印当前行号码。

#

把注释扩展到下一个换行符以前。

以下的是替换标记

g表示行内全面替换。

p表示打印行。

w表示把行写入一个文件。

x表示互换模板块中的文本和缓冲区中的文本。

y表示把一个字符翻译为另外的字符（但是不用于正则表达式）

5. 命令选项

-e command, --expression\_r=command

允许多台编辑。

-h, --help

打印帮助，并显示bug列表的地址。

-n, --quiet, --silent

取消默认输出。

-f, --filer=script-file

引导sed脚本文件名。

-V, --version

打印版本和版权信息。

6. 元字符集

^

锚定行的开始 如：/^sed/匹配所有以sed开头的行。

$

锚定行的结束 如：/sed$/匹配所有以sed结尾的行。

.

匹配一个非换行符的字符 如：/s.d/匹配s后接一个任意字符，然后是d。

\*

匹配零或多个字符 如：，love这成\*\*love\*\*。

<

锚定单词的开始，如:/<love/匹配包含以love开头的单词的行。

>

锚定单词的结束，如/love>/匹配包含以love结尾的单词的行。

x{m}

重复字符x，m次，如：/0{5}/匹配包含5个o的行。

x{m,}

重复字符x,至少m次，如：/o{5,}/匹配至少有5个o的行。

x{m,n}

重复字符x，至少m次，不多于n次，如：/o{5,10}/匹配5--10个o的行。

7. 举例说明

删除：d命令

$ sed '2d' example-----删除example文件的第二行。

$ sed '2,$d' example-----删除example文件的第二行到末尾所有行。

$ sed '$d' example-----删除example文件的最后一行。

$ sed '/test/'d example-----删除example文件所有包含test的行。

替换：s命令

$ sed 's/test/mytest/g' example-----在整行范围内把test替换为mytest。如果没有g标记，则只有每行第一个匹配的test被替换成mytest。

$ sed -n 's/^test/mytest/p' example-----(-n)选项和p标志一起使用表示只打印那些发生替换的行。也就是说，如果某一行开头的test被替换成mytest，就打印它。

$ sed 's/^192.168.0.1/&localhost/' example-----&符号表示替换换字符串中被找到的部份。所有以192.168.0.1开头的行都会被替换成它自已加 localhost，变成192.168.0.1localhost。

$ sed -n 's/(love)able/1rs/p' example-----love被标记为1，所有loveable会被替换成lovers，而且替换的行会被打印出来。

$ sed 's#10#100#g' example-----不论什么字符，紧跟着s命令的都被认为是新的分隔符，所以，“#”在这里是分隔符，代替了默认的“/”分隔符。表示把所有10替换成100。

选定行的范围：逗号

$ sed -n '/test/,/check/p' example-----所有在模板test和check所确定的范围内的行都被打印。

$ sed -n '5,/^test/p' example-----打印从第五行开始到第一个包含以test开始的行之间的所有行。

$ sed '/test/,/check/s/$/sed test/' example-----对于模板test和west之间的行，每行的末尾用字符串sed test替换。

多点编辑：e命令

$ sed -e '1,5d' -e 's/test/check/' example-----(-e)选项允许在同一行里执行多条命令。如例子所示，第一条命令删除1至5行，第二条命令用check替换test。命令的执行顺序对结果有影响。如果两个命令都是替换命令，那么第一个替换命令将影响第二个替换命令的结果。

$ sed --expression='s/test/check/' --expression='/love/d' example-----一个比-e更好的命令是--expression。它能给sed表达式赋值。

从文件读入：r命令

$ sed '/test/r file' example-----file里的内容被读进来，显示在与test匹配的行后面，如果匹配多行，则file的内容将显示在所有匹配行的下面。

写入文件：w命令

$ sed -n '/test/w file' example-----在example中所有包含test的行都被写入file里。

追加命令：a命令

$ sed '/^test/a\this is a example' example-----'this is a example'被追加到以test开头的行后面，sed要求命令a后面有一个反斜杠。

插入：i命令

sed '/test/i\abcde' example-------abcde被插入到包含test的行的前面。

下一个：n命令

$ sed '/test/{ n; s/aa/bb/; }' example-----如果test被匹配，则移动到匹配行的下一行，替换这一行的aa，变为bb，并打印该行，然后继续。

变形：y命令

$ sed '1,10y/abcde/ABCDE/' example-----把1--10行内所有abcde转变为大写，注意，正则表达式元字符不能使用这个命令。也就是说，被替换的字符串要与替换的字符串的长度相同。

退出：q命令

$ sed '10q' example-----打印完第10行后，退出sed。

保持和获取：h命令和G命令

$ sed -e '/test/h' -e '$G‘ example----- 在sed处理文件的时候，每一行都被保存在一个叫模式空间的临时缓冲区中，除非行被删除或者输出被取消，否则所有被处理的行都将打印在屏幕上。接着模式空 间被清空，并存入新的一行等待处理。在这个例子里，匹配test的行被找到后，将存入模式空间，h命令将其复制并存入一个称为保持缓存区的特殊缓冲区内。 第二条语句的意思是，当到达最后一行后，G命令取出保持缓冲区的行，然后把它放回模式空间中，且追加到现在已经存在于模式空间中的行的末尾。在这个例子中 就是追加到最后一行。简单来说，若使用的是H命令，则任何包含test的行都被复制并追加到该文件的末尾，若是h命令，则只有最后一个包含test的行被 复制并追加到该文件的末尾。

保持和互换：h命令和x命令

$ sed -e '/test/h' -e '/check/x' example -----互换模式空间和保持缓冲区的内容。也就是用前面包含test的行去替换包含check的行，而并不是互换两行。

7. 脚本

Sed脚本是一个sed的命令清单，启动Sed时以-f选项引导脚本文件名。Sed对于脚本中输入的命令非常挑剔，在命令的末尾不能有任何空白或文本，如果在一行中有多个命令，要用分号分隔。以#开头的行为注释行，且不能跨行。

# [通过shell脚本替换属性文件中的某行记录](http://blog.csdn.net/wangmuming/article/details/19605281)

假设有如下属性文件 demo.properties

user.name=test

user.password=123456

...............................

需求：先需要通过shell 脚本将 user.name 和 user.password 的value值替换为实际需要的用户名和密码，

将可以通过如下方式实现：

sed -i "s#^user.name=.\*#user.name=用户名#g"  path/demo.properties

sed -i "s#^user.password=.\*#user.password=密码#g"  path/demo.properties

这样demo属性文件中的内容即被替换掉了。

# [通过shell脚本修改xml文件中的某行记录](http://blog.csdn.net/wangmuming/article/details/19606593)

通过shell脚本修改xml文件中的某行记录

跟之前一样，假设有如下的xml文件：

web.xml中有如下内容：

**[html]** [view plaincopy](http://blog.csdn.net/wangmuming/article/details/19606593)[在CODE上查看代码片](https://code.csdn.net/snippets/197524)

1. **<param-name>**cookieName**</param-name>**
2. **<param-value>**sessionName**</param-value>**

需求：现在需要利用shell脚本替换掉cookieName对应的value值，

利用shell实现方式如下：

shell脚本部分：

declare -i Dline

getline()   
   {  
grep -n "cookieName" ${DOMAIN\_HOME}/portal/ROOT/WEB-INF/web.xml | head -1 | cut -d ":" -f 1;  
   }  
  
  
getlinenum()   
{  
awk "BEGIN{a=`getline`;b="1";c=(a+b);print c}";  
}

Dline=`getlinenum`;  
echo "line is ${Dline}";  
sed -i "${Dline},${Dline}s/.\*/                  \<param-value\>替换的value值\<\/param-value\>/g" path/web.xml

说明：这个跟之前的替换属性文件的区别在于，这个是精确定位到行的，然后替换正行code。

           这个xml和properties 中都是默认需要替换的内容是唯一的。