sed

March 9, 2017

sed 是一种流编辑器 (stream editor)，它在处理时，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。

sed 是非交互式的编辑器。它不会修改文件，除非使用 shell 重定向来保存结果。默认情况下，所有的输出行都被打印到屏幕上。

sed 编辑器逐行处理文件（或输入），并将结果发送到屏幕。具体过程如下：首先 sed 把当前正在处理的行保存在一个临时缓存区中 (也称为模式空间)，然后处理临时缓冲区中的行，完成后把该行发送到屏幕上。sed 每处理完一行就将其从临时缓冲区删除，然后将下一行读入，进行处理和显示。处理完输入文件的最后一行后，sed 便结束运行。sed 把每一行都存在临时缓冲区中，对这个副本进行编辑，所以不会修改原文件。

地址用于决定对哪些行进行编辑。地址的形式可以是数字、正则表达式、或二者的结合。如果没有指定地址，sed 将处理输入文件的所有行。

地址是一个数字，则表示行号；是 “$” 符号，则表示最后一行。例如：sed -n ‘3p’ datafile

只打印第三行

sed -n '$p' datafile

只打印最后一行

只显示指定行范围的文件内容，例如：

只查看文件的第 100 行到第 200 行

sed -n '100,200p' mysql_slow_query.log

流编辑器可以对从管道这样的标准输入接收的数据进行编辑。无需将要编辑的数据存储在磁盘上的文件中。

sed 通过对输入数据执行人任意数量用户指定的编辑操作 (命令)。sed 是基于行的，按顺序对每一行执行命令。sed 将其结果写入标准输出 (stdout)，不修改任何输入文件。

1 sed 的选项、命令、替换标记

1.1: 命令格式

sed [options] 'command' file(s)

sed [options] -f scriptfile file(s)

1.2: 选项

-e<script> 或–expression=<script>：以选项中的指定的 script 来处理输入的文本文件；直接在指令列模式上进行 sed 的动作编辑；-e 选项可以支持 sed 进行多点编辑处理，使用多个 scripts 或者 expression 时，之间使用; 分号隔开

-f<script 文件 > 或–file=<script 文件 >：以选项中的指定的 script 文件来处理输入的文本文件；直接将 sed 的动作写在一个档案内，-f filename 则可以执行

filename 内的 sed 动作；

-h 或–help：显示帮助；

-n 或–quiet 或—silent：仅显示 script 处理后的结果；在一般 sed 的用法中，所有来自 stdin 的资料一般都会被列出到屏幕上。加上-n 参数后，则只有经过 sed 处理的那一行 (或者动作) 才会被列出来；

-r：sed 的动作支援的是延伸型正规表示法的语法。(预设是基础正规表示法语法)

-V 或–version：显示版本信息。

-i：直接修改读取的档案内容，而不是由屏幕输出。

参数

文件：指定待处理的文本文件列表。

1.3: sed 命令

a\ 在当前行下面插入文本。

i\ 在当前行上面插入文本。

c\ 把选定的行改为新的文本。

d 删除，删除选择的行。

D 删除模板块的第一行。

s 替换指定字符。

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。

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。

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。

= 打印当前行号码。

把注释扩展到下一个换行符以前。

1.4: sed 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）

\1 子串匹配标记 & 已匹配字符串标记

1.5: sed 元字符集

^匹配行开始，如：/^sed/匹配所有以 sed 开头的行。

$ 匹配行结束，如：/sed$/匹配所有以 sed 结尾的行。

. 匹配一个非换行符的任意字符，如：/s.d/匹配 s 后接一个任意字符，最后是 d。

⋆ 匹配 0 个或多个字符，如：/*sed/匹配所有模板是一个或多个空格后紧跟 sed 的行。

[] 匹配一个指定范围内的字符，如/[ss]ed/匹配 sed 和 Sed。

[^] 匹配一个不在指定范围内的字符，如：/[^A − RT − Z]ed/匹配不包含 A-R 和 T-Z 的一个字母开头，紧跟 ed 的行。

\(· · · \) 匹配子串，保存匹配的字符，如 s/\(*love*\)able/\1rs，loveable 被替换成 lovers。

& 保存搜索字符用来替换其他字符，如 s/love/∗∗&∗∗/，love 这成 **love**。

\ < 匹配单词的开始，如:/\ <love/匹配包含以 love 开头的单词的行。

\ > 匹配单词的结束，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 0 的行。

x\{m, \} 重复字符 x，至少 m 次，如：/0\{5, \}/匹配至少有 5 个 0 的行。

x\{m, n\} 重复字符 x，至少 m 次，不多于 n 次，如：/0\{5, 10\}/匹配 5 ∼ 10 个 0 的行。

2 规则表达式

3 实例

3.1 替换操作：s 命令

替换文本中的字符串：

sed 's/book/books/' file

-n 选项和 p 命令一起使用表示只打印那些发生替换的行：

sed -n 's/test/TEST/p' file

直接编辑文件选项-i，会匹配 file 文件中每一行的第一个 book 替换为 books：

sed -i 's/book/books/g' file

3.2 全面替换标记 g

使用后缀/g标记会替换每一行中的所有匹配：

sed 's/book/books/g' file

当需要从第 N 处匹配开始替换时，可以使用 /Ng：

echo sksksksksksk | sed 's/sk/SK/2g'

skSKSKSKSKSK

echo sksksksksksk | sed 's/sk/SK/3g'

skskSKSKSKSK

echo sksksksksksk | sed 's/sk/SK/4g'

sksksKSKSKSK

3.3 定界符

以上命令中字符/在 sed 中作为定界符使用，也可以使用任意的定界符：

sed ‘s:test:TEXT:g’

sed ‘s|test|TEXT|g’

定界符出现在样式内部时，需要进行转义：

sed ‘s/\/bin/\/usr\/local\/bin/g’

3.4 删除操作：d 命令

删除空白行：

sed ‘/ˆ$/d’ file

删除文件的第 2 行：

sed ‘2d’ file

删除文件的第 2 行到末尾所有行：

sed ‘2, $d’ file

删除文件最后一行：

sed ‘$d’ file

删除文件中所有开头是 test 的行：sed ‘/ˆ test/’d file

3.5 已匹配字符串标记 &

正则表达式 \ w\+ 匹配每一个单词，使用 [&] 替换它，& 对应于之前所匹配到的单词：

echo this is a test line | sed ‘s/\w\+/[&]/g’

[this] [is] [a] [test] [line]

所有以 192.168.0.1 开头的行都会被替换成它自已加 localhost：

sed ‘s/ˆ192.168.0.1/&localhost/’ file

192.168.0.1localhost

3.6 子串匹配标记 \1

匹配给定样式的其中一部分：

echo this is digit 7 in a number | sed ‘s/digit \([0 − 9]\)/\1/’

this is 7 in a number

命令中 digit 7，被替换成了 7。样式匹配到的子串是 7，\(· · · \) 用于匹配子串，对于匹配到的第一个子串就标记为 \1，依此类推匹配到的第二个结果就是 \2，例如：

echo aaa BBB | sed ‘s/\([a − z]\ + \)\([A − Z]\ + \)/\2\1/’

BBB aaa

love 被标记为 1，所有 loveable 会被替换成 lovers，并打印出来：

sed -n ‘s/\(love\)able/\1rs/p’ file

3.7 组合多个表达式

sed ‘表达式’ | sed ‘表达式’

等价于：

sed ‘表达式; 表达式’

3.8 引用

sed 表达式可以使用单引号来引用，但是如果表达式内部包含变量字符串，就需要使用双引号。test=hello

echo hello WORLD | sed “s/$test/HELLO”

HELLO WORLD

3.9 选定行的范围：，（逗号）

所有在模板 test 和 check 所确定的范围内的行都被打印：sed -n ’/test/,/check/p’ file

打印从第 5 行开始到第一个包含以 test 开始的行之间的所有行：

sed -n ‘5,/^test/p’ file

对于模板 test 和 west 之间的行，每行的末尾用字符串 aaa bbb 替换：

sed ‘/test/,/west/s/$/aaa bbb/’ file

3.10 多点编辑：e 命令

-e 选项允许在同一行里执行多条命令：

sed -e ‘1,5d’ -e ’s/test/check/’ file

上面 sed 表达式的第一条命令删除 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。

和 -e 等价的命令是 –expression：

sed –expression=’s/test/check/’ –expression=’/love/d’ file

3.11 从文件读入：r 命令

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：

sed '/test/r file' filename

3.12 写入文件：w 命令

在 example 中所有包含 test 的行都被写入 file 里：

sed -n '/test/w file' example

3.13 追加（行下）：a\ 命令

将 this is a test line 追加到以 test 开头的行后面：

sed '/^test/a\this is a test line' file

在 test.conf 文件第 2 行之后插入

this is a test line：sed -i '2a\this is a test line' test.conf

3.14 插入（行上）：i\ 命令

将 this is a test line 追加到以 test 开头的行前面：

sed '/^test/i\this is a test line' file

在 test.conf 文件第 5 行之前插入 this is a test line：

sed -i '5i\ this is a test line' test.conf

3.15 下一个：n 命令

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：

sed '/test/{ n; s/aa/bb/; }' file

3.16 变形：y 命令

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：

sed '1,10y/abcde/ABCDE/' file

3.17 退出：q 命令

打印完第 10 行后，退出 sed

sed '10q' file

3.18 保持和获取：h 命令和 G 命令

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。

sed -e '/test/h' -e '$G' file

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，任何包含 test 的行都被复

制并追加到该文件的末尾。

3.19 保持和互换：h 命令和 x 命令

互换模式空间和保持缓冲区的内容。也就是把包含 test 与 check 的行互换：

sed -e '/test/h' -e '/check/x' file

3.20 脚本 scriptfile

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。

sed [options] -f scriptfile file(s)

3.21 打印奇数行或偶数行

方法 1：

sed -n 'p;n' test.txt # 奇数行

sed -n 'n;p' test.txt # 偶数行

方法 2：

sed -n '1 ~ 2p' test.txt # 奇数行

sed -n '2 ~ 2p' test.txt # 偶数行

3.22 打印匹配字符串的下一行

grep -A 1 SCC URFILE

sed -n '/SCC/{n;p}' URFILE

```
awk '/SCC/{getline; print}' URFILE
```