参考 – SED 单行脚本

版权信息

1. 本文可能的出处：
   A、轩辕高端 IT 培训中心技术人员自主撰写
   B、相关书籍摘录
   C、互联网上摘录
   D、其他人员或组织赠予
2. 版权声明：

3. 原文作者：
   整理：Eric Pement pemente@northpark.edu 版本 5.5
   译者：Joe Hong hq00e@126.com
4. 原文链接
   http://sed.sourceforge.net/sed1line.txt
   http://www.pement.org/sed/sed1line.txt

一、文本间隔

```
# 在每一行后面增加一空行
sed G

# 将原来的所有空行删除并在每一行后面增加一空行。
# 这样在输出的文本中每一行后面将有且只有一空行。
sed '/^$/d;G'

# 在每一行后面增加两行空行
sed 'G;G'

# 将第一个脚本所产生的所有空行删除（即删除所有偶数行）
sed 'n;d'

# 在匹配式样“regex”的行之前插入一空行
sed '/regex/{x;p;x;}'

# 在匹配式样“regex”的行之后插入一空行
sed '/regex/G'

# 在匹配式样“regex”的行之前和之后各插入一空行
sed '/regex/{x;p;x;G;}'
```

二、编号

```
# 为文件中的每一行进行编号（简单的左对齐方式）。这里使用了“制表符”
# （tab，见本文末尾关于'\t'的用法的描述）而不是空格来对齐边缘。
sed = filename | sed 'N;s/\n/\t/'

# 对文件中的所有行编号（行号在左，文字右端对齐）。
sed = filename | sed 'N; s/^/ /; s/ *\(.\{6,\}\)\n/\1 /'

# 对文件中的所有行编号，但只显示非空白行的行号。
sed '/./=' filename | sed '/./N; s/\n/ /'

# 计算行数（模拟 "wc -l"）
sed -n '$='
```

三、文本转换和替代

```
# Unix 环境：转换 DOS 的新行符（CR/LF）为 Unix 格式。
sed 's/.$//' # 假设所有行以 CR/LF 结束
sed 's/^M$//' # 在 bash/tcsh 中，将按 Ctrl-M 改为按 Ctrl-V
sed 's/\x0D$//' # ssed、gsed 3.02.80，及更高版本

# Unix 环境：转换 Unix 的新行符（LF）为 DOS 格式。
sed "s/$/`echo -e \\\r`/" # 在 ksh 下所使用的命令
sed 's/$'"/`echo \\\r`/" # 在 bash 下所使用的命令
sed "s/$/`echo \\\r`/" # 在 zsh 下所使用的命令
sed 's/$/\r/' # gsed 3.02.80 及更高版本

# DOS 环境：转换 Unix 新行符（LF）为 DOS 格式。
sed "s/$//" # 方法 1
sed -n p # 方法 2

# DOS 环境：转换 DOS 新行符（CR/LF）为 Unix 格式。
# 下面的脚本只对 UnxUtils sed 4.0.7 及更高版本有效。要识别 UnxUtils 版本的
# sed 可以通过其特有的“--text”选项。你可以使用帮助选项（“--help”）看
# 其中有无一个“--text”项以此来判断所使用的是否是 UnxUtils 版本。其它 DOS
# 版本的的 sed 则无法进行这一转换。但可以用“tr”来实现这一转换。
sed "s/\r//" infile >outfile # UnxUtils sed v4.0.7 或更高版本
tr -d \r <infile >outfile # GNU tr 1.22 或更高版本

# 将每一行前导的“空白字符”（空格，制表符）删除
# 使之左对齐
sed 's/^[ \t]*//' # 见本文末尾关于'\t'用法的描述
```

```
# 将每一行拖尾的“空白字符”（空格，制表符）删除
sed 's/[ \t]*$//' # 见本文末尾关于'\t'用法的描述

# 将每一行中的前导和拖尾的空白字符删除
sed 's/^[ \t]*//;s/[ \t]*$//'

# 在每一行开头处插入 5 个空格（使全文向右移动 5 个字符的位置）
sed 's/^/     /'

# 以 79 个字符为宽度，将所有文本右对齐
sed -e :a -e 's/^.\{1,78\}$/ &/;ta' # 78 个字符外加最后的一个空格

# 以 79 个字符为宽度，使所有文本居中。在方法 1 中，为了让文本居中每一行的前
# 头和后头都填充了空格。 在方法 2 中，在居中文本的过程中只在文本的前面填充
# 空格，并且最终这些空格将有一半会被删除。此外每一行的后头并未填充空格。
sed -e :a -e 's/^.\{1,77\}$/ & /;ta' # 方法 1
sed -e :a -e 's/^.\{1,77\}$/ &/;ta' -e 's/\( *\)\1/\1/' # 方法 2

# 在每一行中查找字串“foo”，并将找到的“foo”替换为“bar”
sed 's/foo/bar/' # 只替换每一行中的第一个“foo”字串
sed 's/foo/bar/4' # 只替换每一行中的第四个“foo”字串
sed 's/foo/bar/g' # 将每一行中的所有“foo”都换成“bar”
sed 's/\(.*\)foo\(.*foo\)/\1bar\2/' # 替换倒数第二个“foo”
sed 's/\(.*\)foo/\1bar/' # 替换最后一个“foo”

# 只在行中出现字串“baz”的情况下将“foo”替换成“bar”
sed '/baz/s/foo/bar/g'

# 将“foo”替换成“bar”，并且只在行中未出现字串“baz”的情况下替换
sed '/baz/!s/foo/bar/g'

# 不管是“scarlet”“ruby”还是“puce”，一律换成“red”
sed 's/scarlet/red/g;s/ruby/red/g;s/puce/red/g' #对多数的 sed 都有效
gsed 's/scarlet\|ruby\|puce/red/g' # 只对 GNU sed 有效

# 倒置所有行，第一行成为最后一行，依次类推（模拟“tac”）。
# 由于某些原因，使用下面命令时 HHsed v1.5 会将文件中的空行删除
sed '1!G;h;$!d' # 方法 1
sed -n '1!G;h;$p' # 方法 2

# 将行中的字符逆序排列，第一个字成为最后一字，……（模拟“rev”）
sed '/\n/!G;s/\(.\)\(.*\n\)/&\2\1/;//D;s/.//'

# 将每两行连接成一行（类似“paste”）
sed '$!N;s/\n/ /'
```

```
# 如果当前行以反斜杠“\”结束，则将下一行并到当前行末尾
# 并去掉原来行尾的反斜杠
sed -e :a -e '/\\$/N; s/\\\n//; ta'

# 如果当前行以等号开头，将当前行并到上一行末尾
# 并以单个空格代替原来行头的“=”
sed -e :a -e '$!N;s/\n=/ /;ta' -e 'P;D'

# 为数字字串增加逗号分隔符号，将“1234567”改为“1,234,567”
gsed ':a;s/\B[0-9]\{3\}\>/,&/;ta' # GNU sed
sed -e :a -e 's/\(.*[0-9]\)\([0-9]\{3\}\)/\1,\2/;ta' # 其他 sed

# 为带有小数点和负号的数值增加逗号分隔符（GNU sed）
gsed -r ':a;s/(^|[^0-9.])([0-9]+)([0-9]{3})/\1\2,\3/g;ta'

# 在每 5 行后增加一空白行（在第 5，10，15，20，等行后增加一空白行）
gsed '0~5G' # 只对 GNU sed 有效
sed 'n;n;n;n;G;' # 其他 sed
```

四、选择性地显示特定行

```
# 显示文件中的前 10 行（模拟“head”的行为）
sed 10q

# 显示文件中的第一行（模拟“head -1”命令）
sed q

# 显示文件中的最后 10 行（模拟“tail”）
sed -e :a -e '$q;N;11,$D;ba'

# 显示文件中的最后 2 行（模拟“tail -2”命令）
sed '$!N;$!D'

# 显示文件中的最后一行（模拟“tail -1”）
sed '$!d' # 方法 1
sed -n '$p' # 方法 2

# 显示文件中的倒数第二行
sed -e '$!{h;d;}' -e x # 当文件中只有一行时，输入空行
sed -e '1{$q;}' -e '$!{h;d;}' -e x # 当文件中只有一行时，显示该行
sed -e '1{$d;}' -e '$!{h;d;}' -e x # 当文件中只有一行时，不输出

# 只显示匹配正则表达式的行（模拟“grep”）
sed -n '/regexp/p' # 方法 1
sed '/regexp/!d' # 方法 2
```

```
# 只显示“不”匹配正则表达式的行（模拟“grep -v”）
sed -n '/regexp/!p' # 方法 1，与前面的命令相对应
sed '/regexp/d' # 方法 2，类似的语法

# 查找“regexp”并将匹配行的上一行显示出来，但并不显示匹配行
sed -n '/regexp/{g;1!p;};h'

# 查找“regexp”并将匹配行的下一行显示出来，但并不显示匹配行
sed -n '/regexp/{n;p;}'

# 显示包含“regexp”的行及其前后行，并在第一行之前加上“regexp”所
# 在行的行号（类似“grep -A1 -B1”）
sed -n -e '/regexp/{=;x;1!p;g;$!N;p;D;}' -e h

# 显示包含“AAA”、“BBB”或“CCC”的行（任意次序）
sed '/AAA/!d; /BBB/!d; /CCC/!d' # 字串的次序不影响结果

# 显示包含“AAA”、“BBB”和“CCC”的行（固定次序）
sed '/AAA.*BBB.*CCC/!d'

# 显示包含“AAA”“BBB”或“CCC”的行（模拟“egrep”）
sed -e '/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d # 多数 sed
gsed '/AAA\|BBB\|CCC/!d' # 对 GNU sed 有效

# 显示包含“AAA”的段落（段落间以空行分隔）
# HHsed v1.5 必须在“x;”后加入“G;”，接下来的 3 个脚本都是这样
sed -e '/./{H;$!d;}' -e 'x;/AAA/!d;'

# 显示包含“AAA”“BBB”和“CCC”三个字串的段落（任意次序）
sed -e '/./{H;$!d;}' -e 'x;/AAA/!d;/BBB/!d;/CCC/!d'

# 显示包含“AAA”、“BBB”、“CCC”三者中任一字串的段落（任意次序）
sed -e '/./{H;$!d;}' -e 'x;/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d
gsed '/./{H;$!d;};x;/AAA\|BBB\|CCC/b;d' # 只对 GNU sed 有效

# 显示包含 65 个或以上字符的行
sed -n '/^.\{65\}/p'

# 显示包含 65 个以下字符的行
sed -n '/^.\{65\}/!p' # 方法 1，与上面的脚本相对应
sed '/^.\{65\}/d' # 方法 2，更简便一点的方法

# 显示部分文本——从包含正则表达式的行开始到最后一行结束
sed -n '/regexp/,$p'
```

```
# 显示部分文本——指定行号范围（从第 8 至第 12 行，含 8 和 12 行）
sed -n '8,12p' # 方法 1
sed '8,12!d' # 方法 2

# 显示第 52 行
sed -n '52p' # 方法 1
sed '52!d' # 方法 2
sed '52q;d' # 方法 3，处理大文件时更有效率

# 从第 3 行开始，每 7 行显示一次
gsed -n '3~7p' # 只对 GNU sed 有效
sed -n '3,${p;n;n;n;n;n;n;}' # 其他 sed

# 显示两个正则表达式之间的文本（包含）
sed -n '/Iowa/,/Montana/p' # 区分大小写方式
```

五、选择性地删除特定行

```
# 显示通篇文档，除了两个正则表达式之间的内容
sed '/Iowa/,/Montana/d'

# 删除文件中相邻的重复行（模拟“uniq”）
# 只保留重复行中的第一行，其他行删除
sed '$!N; /^\(.*\)\n\1$/!P; D'

# 删除文件中的重复行，不管有无相邻。注意 hold space 所能支持的缓存
# 大小，或者使用 GNU sed。
sed -n 'G; s/\n/&&/; /^\([ -~]*\n\).*\n\1/d; s/\n//; h; P'

# 删除除重复行外的所有行（模拟“uniq -d”）
sed '$!N; s/^\(.*\)\n\1$/\1/; t; D'

# 删除文件中开头的 10 行
sed '1,10d'

# 删除文件中的最后一行
sed '$d'

# 删除文件中的最后两行
sed 'N;$!P;$!D;$d'

# 删除文件中的最后 10 行
sed -e :a -e '$d;N;2,10ba' -e 'P;D' # 方法 1
sed -n -e :a -e '1,10!{P;N;D;};N;ba' # 方法 2
```

```
# 删除 8 的倍数行
gsed '0~8d' # 只对 GNU sed 有效
sed 'n;n;n;n;n;n;n;d;' # 其他 sed

# 删除匹配式样的行
sed '/pattern/d' # 删除含 pattern 的行。当然 pattern 可以换成任何有效的正则表达式

# 删除文件中的所有空行（与“grep '.' ”效果相同）
sed '/^$/d' # 方法 1
sed '/./!d' # 方法 2

# 只保留多个相邻空行的第一行。并且删除文件顶部和尾部的空行。
# （模拟“cat -s”）
sed '/./,/^$/!d' #方法 1，删除文件顶部的空行，允许尾部保留一空行
sed '/^$/N;/\n$/D' #方法 2，允许顶部保留一空行，尾部不留空行

# 只保留多个相邻空行的前两行。
sed '/^$/N;/\n$/N;//D'

# 删除文件顶部的所有空行
sed '/./,$!d'

# 删除文件尾部的所有空行
sed -e :a -e '/^\n*$/{$d;N;ba' -e '}' # 对所有 sed 有效
sed -e :a -e '/^\n*$/N;/\n$/ba' # 同上，但只对 gsed 3.02.*有效

# 删除每个段落的最后一行
sed -n '/^$/{p;h;};/./{x;/./p;}'
```

六、特殊应用

```
# 移除手册页（man page）中的 nroff 标记。在 Unix System V 或 bash shell 下使
# 用'echo'命令时可能需要加上 -e 选项。
sed "s/.`echo \\\b`//g" # 外层的双引号是必须的（Unix 环境）
sed 's/.^H//g' # 在 bash 或 tcsh 中，按 Ctrl-V 再按 Ctrl-H
sed 's/.\x08//g' # sed 1.5，GNU sed，ssed 所使用的十六进制的表示方法

# 提取新闻组或 e-mail 的邮件头
sed '/^$/q' # 删除第一行空行后的所有内容

# 提取新闻组或 e-mail 的正文部分
sed '1,/^$/d' # 删除第一行空行之前的所有内容
```

```
# 从邮件头提取“Subject”（标题栏字段），并移除开头的“Subject:”字样
sed '/^Subject: */!d; s///;q'

# 从邮件头获得回复地址
sed '/^Reply-To:/q; /^From:/h; /./d;g;q'

# 获取邮件地址。在上一个脚本所产生的那一行邮件头的基础上进一步的将非电邮
# 地址的部分剔除。（见上一脚本）
sed 's/ *(.*)//; s/>.*//; s/.*[:<] *//'

# 在每一行开头加上一个尖括号和空格（引用信息）
sed 's/^/> /'

# 将每一行开头处的尖括号和空格删除（解除引用）
sed 's/^> //'

# 移除大部分的 HTML 标签（包括跨行标签）
sed -e :a -e 's/<[^>]*>//g;/</N;//ba'

# 将分成多卷的 uuencode 文件解码。移除文件头信息，只保留 uuencode 编码部分。
# 文件必须以特定顺序传给 sed。下面第一种版本的脚本可以直接在命令行下输入；
# 第二种版本则可以放入一个带执行权限的 shell 脚本中。（由 Rahul Dhesi 的一
# 个脚本修改而来。）
sed '/^end/,/^begin/d' file1 file2 ... fileX | uudecode # vers. 1
sed '/^end/,/^begin/d' "$@" | uudecode # vers. 2

# 将文件中的段落以字母顺序排序。段落间以（一行或多行）空行分隔。GNU sed 使用
# 字元“\v”来表示垂直制表符，这里用它来作为换行符的占位符——当然你也可以
# 用其他未在文件中使用的字符来代替它。
sed '/./{H;d;};x;s/\n/={NL}=/g' file | sort | sed
'1s/={NL}=//;s/={NL}=/\n/g'
gsed '/./{H;d};x;y/\n/\v/' file | sort | sed '1s/\v//;y/\v/\n/'

# 分别压缩每个.TXT 文件，压缩后删除原来的文件并将压缩后的.ZIP 文件
# 命名为与原来相同的名字（只是扩展名不同）。（DOS 环境：“dir /b”
# 显示不带路径的文件名）。
echo @echo off >zipup.bat
dir /b *.txt | sed "s/^\(.*\)\.TXT/pkzip -mo \1 \1.TXT/" >>zipup.bat
```