

#ВСТАВКА ПУСТЫХ СТРОК:

вставить пустые строки после каждой строки

sed G

вставить пустые строки после каждой строки - если пустые строки есть то заменить их одной

sed '/^\$/d;G'

вставить 2 пустые строки после каждой строки

sed 'G;G'

удалить каждую вторую строку

sed 'n;d'

вставить пустую строку перед каждой строкой с regex

sed '/regex/{x;p;x;}'

вставить пустую строку после каждой строкой с regex

sed '/regex/G'

вставить пустую строку перед и после каждой строкой с regex

sed '/regex/{x;p;x;G;}'**#НУМЕРАЦИЯ:**

выводит файл с нумерацией строк

sed = filename | sed 'N;s/\n/\t/'

выводит файл с нумерацией строк - номера выравнены по правой границе

sed = filename | sed 'N; s/^/ /; s/ *\(.{6,}\)\n/\1 /'

выводит файл с нумерацией строк - если строка пустая номер опускается

sed './.=' filename | sed './.N; s/\n/ /'

число строк (аналог wc -l)

sed -n '\$='**#ЗАМЕНА И ПРЕОБРАЗОВАНИЕ ТЕКСТА:**

замена CR/LF на LF (DOS -> Unix)

sed 's/.\$//'

Предполагается, что все строки заканчиваются CR/LF

sed 's/^M\$//'

в bash/tcsh, нажать Ctrl-V потом Ctrl-M

sed 's/\x0D\$//'

gsed 3.02.80, но верхние варианты легче

замена LF на CR/LF (Unix -> DOS)

sed "s/\$/\`echo -e \\r`/"

в ksh

sed 's/\$/"`echo \\r`/"

в bash

sed "s/\$/\`echo \\r`/"

в zsh

sed 's/\$/\r/'

gsed 3.02.80

```

# в ДОСе, замена LF на CR/LF (Unix -> DOS)
sed "s/$//" # вариант 1
sed -n p # вариант 2

# в ДОСе, замена CR/LF на LF (DOS -> Unix) , возможно только с UnxUtils sed >=
4.0.7
# с ДОС-версией sed надо использовать tr
sed "s/\r//" infile >outfile # UnxUtils sed >= 4.0.7
tr -d \r <infile >outfile # GNU tr >= 1.22

# стереть все начальные отступы
sed 's/^[ \t]*//' # См. примечание о '\t' в конце

# стереть все завершающие строку пробелы и табуляции
sed 's/[ \t]*$//' # См. примечание о '\t' в конце

# стереть и начальные и конечные пробелы и табуляции
sed 's/^[ \t]*//;s/[ \t]*$//'

# вставить в начало всех строк 5 пробелов
sed 's/^/ /'

# выравнивание текста по правому краю на 79 столбце
sed -e :a -e 's/^\.\{1,78\}$/ &;ta' # ставим на 78 + 1

# выравнивание по центру (ширина 79 столбцов) в 1 варианте строки дополняются
пробелами и сзади, во 2 нет
sed -e :a -e 's/^\.\{1,77\}$/ & /;ta' # вариант 1
sed -e :a -e 's/^\.\{1,77\}$/ &;ta' -e 's/\( *\)\1/\1/' # вариант 2

# замена foo на bar в каждой строке
sed 's/foo/bar/' # замена только первого вхождения в строке
sed 's/foo/bar/4' # замена 4 первых вхождений в каждой строке
sed 's/foo/bar/g' # замена всех вхождений
sed 's/(.*)foo\(.foo\)/\1bar\2/' # замена предпоследнего вхождения
sed 's/(.*)foo/\1bar/' # замена только последнего

# замена foo на bar только если строка содержит baz
sed '/baz/s/foo/bar/g'

# замена foo на bar только если строка НЕ содержит baz
sed '/baz!/s/foo/bar/g'

# замена scarlet или ruby или puce на red
sed 's/scarlet/red/g;s/ruby/red/g;s/puce/red/g' # большинство вариантов
seds
sed 's/scarlet|ruby|puce/red/g' # только для GNU sed

# вывести текст с последней по первую строку (эмуляция tac)
# фиче-бага в HHsed v1.5 - пустые строки будут удалены
sed '1!G;h;$!d' # вариант 1

```

```

sed -n '!G;h;$p' # вариант 2

# напечатать справа налево (эмуляция rev)
sed '/\n!/G;s/(.\\)\(.*\n\\)/&2\1/;/D;s/.//'

# сливает четные и нечетные строки (как paste)
sed '$!N;s/\n/ /'

# если строка оканчивается на \ добавляет к ней следующую
sed -e :a -e '/\n$/N; s/\n\n//; ta'

# если строка начинается на "=", то к ней добавляется предыдущая строка, "=" заменяется
на пробел
sed -e :a -e '$!N;s/\n=/ /;ta' -e 'P;D'

# добавляет разделитель разрядов к числам ("1234567" -> "1,234,567")
sed ':a;s/\B[0-9]\{3\}\>/,/;&;ta' # для GNU sed
sed -e :a -e 's/(.*[0-9])\([0-9]\{3\}\)/\1,\2/;ta' # для других версий sed

# добавляет разделитель разрядов к числам с точкой и минусом (для GNU sed)
# -58869087097.657898790 -> -58,869,087,097.657898790
sed ':a;s/(\^[^0-9.])([0-9]\+)([0-9]\{3\}\)/\1\2,\3/g;ta'

# добавляет пустую строку через каждые 5 строк (после 5,10,15,20 и т.д. строк)
sed '0~5G' # для GNU sed
sed 'n;n;n;n;G;' # для других версий sed

#ВЫБОРОЧНАЯ ПЕЧАТЬ СТРОК:

# печать первых 10 строк (эмуляция head)
sed 10q

# печать первой строки (head -1)
sed q

# печать последних 10 строк (эмуляция tail)
sed -e :a -e '$q;N;11,$D;ba'

# печать последних 2 строк (эмуляция tail -2)
sed '$!N;$!D'

# печать последней строки (эмуляция tail -1)
sed '$!d' # вариант 1
sed -n '$p' # вариант 2

# печать строк совпадающих с регулярным выражением (эмуляция grep)
sed -n '/regex/p' # вариант 1
sed '/regex/!d' # вариант 2

# печать строк НЕ совпадающих с регулярным выражением (эмуляция grep -v)
sed -n '/regex/!p' # вариант 1

```

```

sed '/regexp/d' # вариант 2

# напечатать строки, которые стоят ПЕРЕД строками с регулярным выражением regexp
sed -n '/regexp/{g;1!p;};h'

# напечатать строки, которые стоят ПОСЛЕ строк с регулярным выражением regexp
sed -n '/regexp/{n;p;}'

# печать по 1 строке, окружающих строку с регулярным выражением, саму строку,
# а также номер строки (похоже на grep -A1 -B1)
sed -n -e '/regexp/{=;x;1!p;g;${N;p;D;}}' -e h

# печать строк которые содержат и AAA и BBB и CCC (в любом порядке)
sed '/AAA/!d; /BBB/!d; /CCC/!d'

# печать строк которые содержат и AAA и BBB и CCC (в таком порядке)
sed '/AAA.*BBB.*CCC/!d'

# печать строк которые содержат или AAA или BBB или CCC (эмуляция egrep)
sed -e '/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d # большинство версий sed
sed '/AAA\|BBB\|CCC/!d' # только GNU sed

# печать абзаца если он содержит AAA (абзацы должны быть разделены пустыми линиями)
# с GNUsed v1.5 нужно вставлять 'G;' после 'x;' в следующих 3 примерах
sed -e '/./{H;$!d;}' -e 'x;/AAA/!d;'

# печать абзаца если он содержит AAA и BBB и CCC
sed -e '/./{H;$!d;}' -e 'x;/AAA/!d;/BBB/!d;/CCC/!d'

# печать абзаца если он содержит AAA или BBB или CCC
sed -e '/./{H;$!d;}' -e 'x;/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d
sed '/./{H;$!d;};x;/AAA\|BBB\|CCC/b;d' # только GNU sed

# печать только строк, которые длинее 65 символов
sed -n '/^.\{65\}/p'

# печать только строк, которые короче 65 символов
sed -n '/^.\{65\}/!p' # вариант 1
sed '/^.\{65\}/d' # вариант 2

# печать части файла от regexp до конца
sed -n '/regexp/, $p'

# печать с 8 по 12 строку
sed -n '8,12p' # вариант 1
sed '8,12!d' # вариант 2

# печать 52 строки
sed -n '52p' # вариант 1
sed '52!d' # вариант 2
sed '52q;d' # вариант 3, эффективно на большом файле

```

```

# печать каждой 7 строки начиная с 3
sed -n '3~7p' # только GNU sed
sed -n '3,${p;n;n;n;n;n;n;}' # другие версии sed

# печать файла со строки с Iowa по строку с Montana
sed -n '/Iowa/,/Montana/p' # case sensitive

#ВЫБОРОЧНОЕ УДАЛЕНИЕ СТРОК:

# печать файла кроме части со строки с Iowa по строку с Montana
sed '/Iowa/,/Montana/d'

# удаление повторяющихся строк, идущих друг за другом (аналог uniq)
sed '$!N; /^\(.*\)\n\1$/!P; D'

# удаление повторяющихся строк, НЕ идущих друг за другом (возможно переполнение
буфера, используйте GNU sed)
sed -n 'G; s/\n/&&; /^\([ -~]*\n\).*\n\1/d; s/\n//; h; P'

# удаление всех строк, кроме повторяющихся (аналог uniq -d)
sed '$!N; s/^\(.*\)\n\1$/\1/; t; D'

# удаление первых 10 строк файла
sed '1,10d'

# удаление последней строки файла
sed '$d'

# удаление последних 2 строк файла
sed 'N;$!P;$!D;$d'

# удаление последних 10 строк файла
sed -e :a -e '$d;N;2,10ba' -e 'P;D' # вариант 1
sed -n -e :a -e '1,10!{P;N;D;}N;ba' # вариант 2

# удаление каждой 8 строки
sed '0~8d' # GNU sed only
sed 'n;n;n;n;n;n;n;d;' # other seds

# удаление всех пустых строк (похоже на grep '.' )
sed '/^$/d' # вариант 1
sed '/./!d' # вариант 2

# замена всех повторяющихся пустых строк из файла на одну,
# также удаляет все пустые строки в начале и конце файла (аналог cat -s)
sed '/./,/^$/!d' # вариант 1, оставляет 0 пустых строк вверху, 1 в конце
sed '/^$/N;/\n$/D' # вариант 2, оставляет 1 пустую строку вверху, 0 в конце

# замена всех повторяющихся пустых строк из файла на две
sed '/^$/N;/\n$/N;//D'

```

```

# удаление пустых строк в начале файла
sed '/./,$!d'

# удаление пустых строк в конце файла
sed -e :a -e '/^\n*$/{$d;N;ba' -e '}' # все варианты sed
sed -e :a -e '/^\n*$/N;\n$/ba' # все кроме gsed 3.02*

# удаление последней строки каждого абзаца
sed -n '/^$/ {p;h;};/./ {x;/./p;}'

# СПЕЦИАЛЬНОЕ ПРИМЕНЕНИЕ:

# удаление спецсимволов из ман-страниц (echo в баше и в некоторых Unix System V
нуждается в ключе -e)
sed "s/.\`echo \\b`//g" # двойные кавычки нужны в Unix окружении
sed 's/.\^H//g' # в bash/tcsh, нажать Ctrl-V и затем Ctrl-H
sed 's/.\x08//g' # hex-выражение для sed v1.5

# отделить заголовок Usenet/e-mail сообщения
sed '/^$/q' # удалить все после первой пустой строки

# отделить тело Usenet/e-mail сообщения
sed '1,/^\$/d' # удалить все до первой пустой строки

# вывести тему письма, удалив "Subject: "
sed '/^Subject: */!d; s///;q'

# вывести обратный адрес письма
sed '/^Reply-To:/q; /^From:/h; ./d;g;q'

# извлечение адреса электронной почты (смотри предыдущий пример)
sed 's/ *(.*)//; s/>.*//; s/.*[:<] *//'

# добавить "> " в каждую строку (цитата в письме)
sed 's/^\> /'

# убрать "> " из каждой строки (отменить цитату)
sed 's/^\> //'

# удалить большинство HTML - тэгов (с многострочными лажа)
sed -e :a -e 's/<[^>]*>//g;/</N;//ba'

# извлечение закодированного бинарника из нескольких файлов
sed '/^end/,/^begin/d' file1 file2 ... fileX | uudecode # версия 1
sed '/^end/,/^begin/d' "$@" | uudecode # версия 2

# архивирование в zip каждого txt файла в папке с удаление исходного
echo @echo off >zipup.bat
dir /b *.txt | sed "s/^\(.*\)\\.TXT/pkzip -mo \1 \1.TXT/" >>zipup.bat

```

*#ИСПОЛЬЗОВАНИЕ: Sed применяет команду(ы) к каждой строке на своем входе
#а затем выводит результат на стандартный вывод*

```
cat filename | sed '10q'      # использование ввода через пайп
sed '10q' filename          # то-же но без cat
sed '10q' filename > newfile # вывод -> в файл newfile
```

*#КАВЫЧКИ: Приведенные примеры используют одинарные кавычки ('...')
#вместо двойных ("...") для экранирования команд, так sed
#используется на Unix платформах. Одинарные кавычки не дают шеллу
#интерпретировать символ доллара (\$) и обратных кавычек (`...`).
#Пользователям "csh" и производных также надо экранировать восклицательный
#знак (!) обратным слэшем (\!) даже в случай использования одинарных
#кавычек. Версии sed для ДОСа всегда используют двойные кавычки вместо
#одинарных.*

*#ИСПОЛЬЗОВАНИЕ '\t' В SED СКРИПТАХ: Большинство версий sed не принимают
#символ табуляции (0x09) '\t' в командах, поэтому надо вместо него нажимать
#кнопку TAB при наборе. '\t' поддерживается как регулярное выражение в
#awk, perl, HHsed, sedmod, и в GNU sed v3.02.80.*

*#ВЕРСИИ SED: В разных версиях sed синтаксис команд может различаться
#большинство версий не поддерживают метки (:name) или инструкции b, t
#версия GNU sed имеет более краткий синтаксис, например:*

```
sed -e '/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d
```

#для GNU sed уменьшится до:

```
sed '/AAA/b;/BBB/b;/CCC/b;d'      # или даже
sed '/AAA\|BBB\|CCC/b;d'
```

*#Кроме того, помните что многие версии sed понимают команду вида
#"/one/ s/RE1/RE2/", но не понимают "/one/! s/RE1/RE2/", которая
#содержит пробел перед 's'. Надо исключить пробел при наборе команды.*

*#ОПТИМИЗАЦИЯ СКОРОСТИ: Если скорость выполнения недовлетворительна, то
#выражения могут быть выполнены гораздо быстрее, если до "s/.../.../"
#сделать поиск по файлу*

```
sed 's/foo/bar/g' filename      # Стандартная команда замены
sed '/foo/ s/foo/bar/g' filename # выполняется гораздо быстрее
sed '/foo/ s//bar/g' filename   # сокращенное выражение
```

*#Если выбор или удаление производится только в части файла, то команда
#выход (q) ускорит выполнение на больших файлах*

```
sed -n '45,50p' filename        # печать строк 45-50 файла
sed -n '51q;45,50p' filename    # то-же, но быстрее
```

From:

<https://docs.infomir.com.ua/> -

Permanent link:

https://docs.infomir.com.ua/doku.php?id=knowledge_base:sed_examples

Last update: **2021/12/15 14:31**

