

**Университет ИТМО, кафедра ВТ**

**Лабораторная работа №5 по “Системному  
Программному Обеспечению”**

Работу выполнил  
студент группы Р3200

**Рогов Я. С.**

Преподаватель  
**Дергачёв А.М.**

Санкт-Петербург, 2016

### Задание 1 (файл datebook):

1. Замените имя Jon на Jonathan

```
sed 's/<Jon>/Jonathan/g' datebook
```

2. Удалите первые три строки

```
sed 1,3d datebook
```

3. Выведите строки с 5-ой по 10-ю

```
sed -n 5,10p datebook
```

4. Удалите строки, содержащие Lane

```
sed /Lane/d datebook
```

5. Вывести все строки с с днем рождения в ноябре или декабре

```
sed -n '/1[12]\/[0-9]\{1,2\}\/[0-9]\{1,2\}/p' datebook
```

6. Добавить три символа \* в конец строк, начинающихся с Fred

```
sed /^Fred/s/$/***/ datebook
```

7. Замените строку, содержащую Jose на JOSE HAS RETIRED

```
sed '/Jose/c\
JOSE HAS RETIRED' datebook
```

8. Замените дату рождения Popeye на 11/14/46. При этом подразумевается, что вы не знаете значение даты, хранящейся в файле. Составьте и используйте регулярное выражение для ее поиска

```
sed '/Popeye/{ s![0-9]\{1,2\}\/[0-9]\{1,2\}\/[0-9]\{1,2\}!11\14\46!; }'
datebook
```

9. Удалите все пустые строки

```
sed '/^[[:blank:]]*$/d'
```

10. Напишите программу на языке редактора sed, которая:

1. вставляет перед первой строкой заголовок TITLE OF FILE
2. удаляет последнее поле, значение которого кратно 500
3. меняет местами имя и фамилию
4. добавляет к концу каждой строки фразу THE END

```
1i\
TITLE OF FILE
s/\(.*\)<[0-9]*[05]00>\(.*\)/\1\2/
s/^\([a-zA-Z]*\)\(.*\)\([a-zA-Z]*\)/\3\2\1/
s/$/ THE END/
```

**Задание 2:** разобрать команды и объяснить состав регулярного выражения и описать результаты вывода каждой из приведенных ниже команд.

Во избежание многократных повторений, дадим пояснение основным ключам:

**-n** – "тихий режим", т.е. подавляется стандартное поведение sed, при котором после каждого исполнения программы происходит вывод области редактирования (pattern space).

**-e [скрипт]** – позволяет явно указать скрипт и/или указать несколько.

**-f [имяфайл]** – берёт скрипт из файла с соответствующим именем.

При нескольких флагах **-e** и/или **-f** "формируется" один скрипт, которые исполняется в порядке указание аргументов.

**sed '/north/p' datafile** – при вхождении подстроки "north" в данную строку напечатать её. Т.о. sed напечатает все строки, а строки с "north" напечатает дважды

**sed -n '/north/p' datafile** – аналогично, только отключает "стандартное поведение", т.е. будут напечатаны ТОЛЬКО строки с "north".

**sed '3d' datafile** – удалить 3-ю строку, т.е. выведутся все строки, кроме 3

**sed '3,\$d' datafile** – удалить строки с 3 по последнюю. Т.е. выведутся только 1 и 2 строки.

**sed '\$d' datafile** – удалить последнюю строку, т.е. выведутся все строки, кроме последней

**sed '/north/d' datafile** – удалить строку с "north", т.е. выведутся все строки без "north"

**sed 's/west/north/g' datafile** – заменить все "west" на "north". Т.о. выведутся все строки, но с соответствующей заменой, если она возможно.

**sed -n 's/^west/north/p' datafile** – напечатать, заменив "west" в начале строки на "north"

**sed 's/[0-9][0-9]\$/&.5/' datafile** – добавить к двум цифрам в конце строки ".5.", т.е. произойдёт вывод строк с данной по возможности применённой заменой

**sed -n 's/Hemenway/Jones/gp' datafile** – заменить все "Hemenway" в строке на "Jones" и напечатать получившуюся строку.

**sed -n 's/^(Stag\)got/\1ianne/p' datafile** – заменить "Staggot" на "Stagianne" и вывести.

**sed 's#14#88#g' datafile** – заменить все "14" на "88" в строке.

**sed -n '/west/,/east/p' datafile** – напечатать все строки, начиная со строки, включающей "west", по строку, включающую "east".

**sed -n '5,/^northeast/p' datafile** – напечатать все строки, начиная с 5 строки по строку, включающую "northeast" в начале строки.

**sed '/west/,/east/s/\$/\*\*WAKA\*\*/' datafile** – напечатать все строки, добавив "\*\*\*WAKA\*\*" в конец тех, которые находятся между строк, включающих "west" и "east" (включительно)

**sed -e '1,3d' -e 's/Hemenway/Jones/' datafile** – удаляет строки с 1 по 3 и заменяет все "Hemenway" на "Jones"

**sed '/Suan/r newfile' datafile** – вставляет содержимое файла newfile после строки, содержащей "Suan". Т.о. напечатаются все строки с соответствующей возможной вставкой.

**sed -n '/north/w newfile' datafile** – записать все строки, содержащие "north" в файл newfile.

**sed '/^north /a\**  
**--->THE NORTH SALES DISTRICT HAS MOVED<---** ' datafile – ко всем строкам, начинающимся с "north<пробел>" строку "--->...<---".

**sed '/eastern/i\**  
**NEW ENGLAND REGION\**  
**-----** ' datafile – ПЕРЕД всем строками, включающими "eastern" добавить строки "NEW..." и "----...".

**sed '/eastern/c\**  
**THE EASTERN REGION HAS BEEN TEMPORARILY CLOSED** ' datafile – заменить все строки, включающие "eastern" на строку "THE EASTERN..."

**sed '/eastern/{ n; s/AM/Archie;/ }' datafile** – найти строки, включающие "eastern", вывести их, если вывод неподавлен, и заменить "AM" в СЛЕДУЮЩЕЙ строке на "Archie".

**sed '3,4y/abcdefghijklmnopqrstuvwxyz/ABCDEFGHIJKLMNOPQRSTUVWXYZ/' datafile** – перевести строки 3 и 4 в верхний регистр

**sed '1q' datafile** – закончить выполнение скрипта на первой строке

**sed '/Lewis/{ s/Lewis/Joseph;q; }' datafile** – заменить первое "Lewis" на "Joseph" и выйти.

**sed -e '/northeast/h' -e '\$G' datafile** – добавить последнюю строку, включающую "northeast" после последней строки файла.

**sed -e '/WE/{h; d; }' -e '/CT/{G; }' datafile** – вырезать строку (удалить и сохранить), содержащую "WE", добавить к строке, содержащей "CT" последнюю подобную строку (с "WE").

**sed -e '/northeast/h' -e '\$g' datafile** – заменить последнюю строку файла на последнюю строку, включающую "northeast".

**sed -e '/WE/{h; d; }' -e '/CT/{g; }' datafile** – заменить все строки, содержащие "CT" на последнюю предшествующую строку, содержащую "WE". Строки с "WE" вырезаются.

**sed -e '/Patricia/h' -e '/Margot/x' datafile** – программа выполняет следующие действия: при строке с "Patricia" происходит сохранение данной строки в область удержания (hold space/hold buffer), а при строке с "Margot" меняются местами данная строка, и строка в области удержания. Т.о. на место строки с "Margot" будет вставлена последняя предшествующая строка с "Patricia" или "Margot". Последняя строка с "Margot", по-сути, удалится.

**sed -n '/sentimental/p' datafile** – напечатать только строки с "sentimental".

**sed '1,1d' datafile > newfile** – записать содержимое файла datafile, начиная со второй строки, в файл newfile. Аналогично 'tail +2 datafile > newfile'

**sed '/[Dd]aniel/d' datafile** – удаляет строки с "Daniel" и/или "daniel"

**sed -n '16,20p' datafile** – напечатать строки с 16-й по 20-ю.

**sed '1,10s/Montana/MT/g' datafile** – заменить все "Montana" на строках с 1 по 10 на "MT"

**sed '/March/!d' datafile** – удалить все строки, кроме строк с "March"

**sed '/report/s/5/8/' datafile** – во всех строках, содержащих "report" заменить "5" на "8".

**sed 's/....//' datafile** – если строка длиннее 4 символов, то первые 4.

**sed 's/...\$//' datafile** – если строка длиннее 3, то удалить последние 3 символа

**sed '/east/,/west/s/North/South/' datafile** – заменить "North" на "South" на строках со строки, содержащей "eest" по строку, содержащую "west"

**sed -n '/Time off/w timefile' datafile** – записать строки с "Time off" в файл timefile

**sed 's/([Oo]ccur\)\ence/\1rence/' datafile** – исправить орфографическую ошибку в слове "occurrence" игнорируя регистр первой буквы.

**sed -n l datafile** – напечатать файл в "недвусмысленном" виде: непечатаемые символы печатаются в виде восьмеричных кодов.

**Задание 3:** Проанализировать программы на языке sed и описать их поведение.

```
1./Lewis/a\
Lewis is the TOP Salesperson for April!!\
Lewis is moving to the southern district next month.\
CONGRATULATIONS!
/Margot/c\
*****\
MARGOT HAS RETIRED\
*****
1i\
EMPLOYEE DATABASE\
-----
$d
1. Добавляет в начало заголовков
2. Добавляет поздравления после строки с "Lewis"
3. Заменяет запись с "Margot" на уведомление об уходе на пенсию
4. Удаляет последнюю строку
```

```
2./western/, /southeast/{
/^ *$/d
/Suan/{ h; d; }
}
/Ann/g
s/TB \(Savage\) /Thomas \1/
1. На диапазоне строк, со строки с "western" по строку с "southeast"...
    1. Удаляет все пустые (состоящие из пробелов) строки
    2. Вырезает и записывает в область удержания строку с "Suan"
2. Заменяет строку с "Ann" на строку из области удержания, т.е. строку с "Suan" либо пустую.
3. Заменяет "TB Savage" на "Thomas Savage"
```

```

3. echo '!qqqq#qqqqq#q#qq#qqqqqqq#qqq@' |
gsed -n '
1p;
:b;
s/!(q*)#\(\1[^\@]*\)@/!\2@\1#/;
tx;
s/!(^[#]#)\(/\1!/;
:x;
/![^#]*@/be;
s/!(^[#]**)#\([^\@]*\)@/!\2#\1/;
bb;
:e;
s/!(^[^\@]**)@.*\/\1/;
p'

```

! - курсор начала

@ - курсор конца

# - разделитель q-строк

На ввод подаётся массив из q-строк вида "@qqq#qq#qq!" в соответствии с обозначениями выше.

### Регулярные выражения:

```
:b 1. s/!(q*)#\(\1[^\@]*\)@/!\2@\1#/;
```

Если второй элемент большей первого, то "удалить" первый элемент

(переместить за @-курсor)

Если замена была, то перейти к (x)

```
2. s/!(^[#]#)\(/\1!/;
```

Иначе если первый элемент - "q", то "удалить его" (переместить перед !-курсorом)

```
:x 3. /![^#]*@/be;
```

Если между курсорами находится только один элемент массива, то перейти в

конец программы (e)

```
4. s/!(^[#]**)#\([^\@]*\)@/!\2#\1/;
```

Поменять первый и второй элемент местами

Перейти к (b)

```
:e 5. s/!(^[^\@]**)@.*\/\1/;
```

Заменить весь массив первым элементом массива

```
4. echo QQQWWW |
```

```
gsed -n 's/^\(.*\)$/x\1=;/;:t;s/xQ\((Q*)\)\(W*)\=(.*\)/Qx\1\2\3\2/;tt;p'
```

Производит "умножение" Q-подстроки на W-подстроки:

Вывод: <Q-подстрока>x<W-подстрока>=<W-подстрока length(Q) раз>

```
5. echo wwwwww | gsed -n 's/(w*)\1\1(w*)/\1,\2/p'
```

Деление w-строки на 3 с остатком. Т.е. пусть дана w-строка s. Тогда вывод будет следующим:

<length(s)//3 символов "w">,<length(s)%3>

```
6. s/([0-9])\1/g
```

```
s/0//g; s/1/i/g; s/2/ii/g; s/3/iii/g; s/4/iiii/g; s/5/iiiii/g; s/6/iiiiii/g;
```

```
s/7/iiiiiii/g; s/8/iiiiiiii/g; s/9/iiiiiiiiii/g
```

```
:ten
```

```
s/i / iiiiiiiiii/g
```

```
t ten
```

```
s/ //g
```

```
s/+//
```

```

:sub
s/i-i/-/g
t sub
s/-$/

/\//b divid

s/\*/m/
s/^mi*$//
s/^i*m$//
s/^i+/i/
:mul
s/mi$//
s/+\(i*\)mi\ \(i*\)/\1+\1m\2/
t mul
s/+//

b result

:divid
s/\//7/
s/^7i*$0/
/^0$/b quit
s/^i*7$/недопустимая операция/
/недопустимая операция/b quit
#делимое+делитель+уменьшаемое+вычитаемое+счетчик(частное)
s/\(i*\)7\ \(i*\)/\1+\2+\1-\2+/
:div
:sb1
s/+\(i*\)i-i\ \(i*\)/+/\1-\2+/
t sb1
s/\(i*\)+\ \(i*\)+-\ \(i*\)+/\2+ -\3+/
t end
s/\(i*\)+\ \(i*\)+-+ /null/
t end
s/\(i*\)+\ \(i*\)+\ \(i*\)-+\ \(i*\)/\1+\2+\3-\2+i\4/
b div

:end
s/\(i*\)+-\ \(i*\)/\1m\1-\2/
:sb2
s/\(i*\)i-i\ \(i*\)/\1-\2/
t sb2
s/-//
s/\(i*\)m\ \(i*\)+\ \(i*\)/\3 \
остаток\:\2 \/\1 /
t res1

s/null\ \(i*\)/i\1/

:result
s/iiiiiiiiii/ /g
s/ \([0-9]*\) $/ 0\1/g
s/iiiiiiiiii/9/g;s/iiiiiiiiii/8/g;s/iiiiiiiiii/7/g; s/iiiiiiii/6/g;
s/iiiiii/5/g; s/iiii/4/g; s/iii/3/g;s/ii/2/g; s/i/1/g;
s/ /i/g
t result

:res1
s/iiiiiiiiii/_/g
s/_\([0-9]*\) /_0\1/g
s/iiiiiiiiii/9/g;s/iiiiiiiiii/8/g;s/iiiiiiiiii/7/g; s/iiiiiiii/6/g;
s/iiiiii/5/g; s/iiii/4/g; s/iii/3/g;s/ii/2/g; s/i/1/g;
s/_/i/g

```

```
t res1
s/ //
s/^\$/0/
:quit
```

Разберём программу по блокам кода:

1. Перевод числа n в форму <n символов "i">
2. Сложение: Убрать все "+" (т.к. сложение чисел в i-форме - просто их конкатенация)
3. Вычитание:
  - Заменять все "i-i" на "-" (итеративное вычитание по единице), пока находятся замены
  - Убрать - в конце строки
4. Если найден "/", перейти на Деление
5. Умножение:
  - Заменить первое \* на m
  - Убрать вместе с числом, если перед выражением ...или после
  - Добавить "+" в начало (курсор)
  - Произвести умножение (по схеме, обратной в sed4)
6. Деление:
  - Заменить / на 7
  - При делении 0 на число, заменить на 0 и перейти к завершению программы
  - При делении на 0 выдать ошибку и закончить выполнение
  - Заменить строку строку "делимое7делитель" на "делимое+делитель+уменьшаемое+вычитаемое+счетчик(частное)"
  - Метка div
  - Произвести вычитание вычитаемого из уменьшаемого (итеративное по единице)
  - Если вычитаемое осталось, то перейти в конец процедуры (деление закончено)
  - Иначе заменить строку на null (деление без остатка) и перейти в конец процедуры
  - Иначе обновить делитель и увеличить счётчик на 1
  - Перейти к div
  - Конец:
  - Посчитать остаток: продублировать делитель и вычесть из него оставшееся вычитаемое
  - Убрать "-"
  - Произвести замену на выражение вида "<частное>\n остаток: <остаток>/<делитель>"
  - Перейти к обратной замене
  - Если в строке есть null, то заменить строку частным.
  - Произвести обратную замену i-строк на числа
  - Убрать пробелы
  - В случае пустой строки добавить "0"

**Т.о.** программа представляет собой калькулятор со следующими особенностями:

- Приоритет операций в порядке убывания: +, -, /, \*
- В выражении разрешено только одно деление, либо одно умножение. Иначе будет "мусор" на выводе.

```
7.
#!/opt/sfw/bin/gsed -nf
s/$/@/
tx
:x
:y
s/0@/@ /
```

```

ty
s/1@/@ | /
ty
s/2@/@ || /
ty
s/3@/@ ||| /
ty
s/4@/@ |||| /
ty
s/5@/@ ||||| /
ty
s/6@/@ ||||| /
ty
s/7@/@ ||||| /
ty
s/8@/@ ||||| /
ty
s/9@/@ ||||| /
s/@ *//
p
:z
s/| / ||||| /
tz
s/ *//
p

```

Здесь используется конструкция "shebang"/"shabang", которая позволяет исполнять запускать программу, записанную в конструкции и передавать данный файл как аргумент. Т.о., в данном примере при запуске ./sed7 (при выставленные правах запуска) будет запущено "/opt/sfw/bin/gsed -nf sed7", т.е. GNU-реализация sed с отключением автоматического вывода и указанием файла как источника команд.

Сама программа принимает число, сначала выводит число поразрядно в форме количества вертикальных черт "|" (123 = " | | | "; команды по первую "р"), а затем всё число в такой форме (123 – 123 вертикальных линии; команды от первого "р" по второе "р"). Первое действие выполняется простой заменой цифры n на "<n раз '|'> (в начале пробел), в результате чего разряды разделены пробелом, а второе: заменой, начиная со старших разрядов, " | " на "<10 раз ' | ' > (пробелы в конце и начале соответственно). **Т.е.**, по сути, выполняется итеративное добавление 10 единиц к младшему разряду и вычитание 1 из старшего.

```

8. $ cat sed8
#!/opt/sfw/bin/gsed -rf
/^0-9]/ d
:d
s/9\(_*\)$/_\1/
td
s/^\(_*\)$/_\1\1/; tn
s/8\(_*\)$/_\1\1/; tn
s/7\(_*\)$/_\1\1/; tn
s/6\(_*\)$/_\1\1/; tn
s/5\(_*\)$/_\1\1/; tn
s/4\(_*\)$/_\1\1/; tn
s/3\(_*\)$/_\1\1/; tn
s/2\(_*\)$/_\1\1/; tn
s/1\(_*\)$/_\1\1/; tn
s/0\(_*\)$/_\1\1/; tn
:n
y/_/0/

```

Удаляет первую последовательность нецифр

Заменяет все последние 9 с конца на \_ (если после них идут только \_)



Вставляет 1 в начало при отсутствии цифры

В порядке убывания цифр от 8 до 0 пытается заменить крайнюю правую. В случае удачи переходит к последней команде

Заменить все \_ на 0.

Т.о. эта программа инкрементирует число, поданное на ввод, игнорируя любой мусор, который подали вместе с числом.

```
9. echo '\//\//\//\' | sed -e 's/\\/a/g' -e 's/\\/\\/g' -e 's/a/\\/g'
```

Происходит 3 глобальных замены: "\" на "a", "/" на "\", "a" на "/". **Т.о.** программа выполняет зеркальное отражение слешей: меняет обычные на обратные и наоборот.

**10.**

```
s/^1@i@g
s/^2@ii@g
s/^3@iii@g
s/^4@iiii@g
s/^5@iiiii@g
s/^6@iiiii@g
s/^7@iiiii@g
s/^8@iiiii@g
s/^9@iiiii@g
s/$/@/g
s/(i*\)@/1@1/g
:a ta
:x
s/@ii/@/g
tx
/@i/ {s/i@i//g;}
s/@//g
s/^$/0/g
s/^i$/1/g
s/^ii$/2/g
s/^iii$/3/g
s/^iiii$/4/g
s/^iiiii$/5/g
s/^iiiii$/6/g
s/^iiiii$/7/g
s/^iiiii$/8/g
```

Глобально заменяет число в начале на соответствующую i-представление

Вставляет в конец всех строк @ (курсор конца)

Глобально дублирует i-строку за курсор

Делит с остатком на 2 (результат находится за курсором)

Вычитает остаток из числа

Переводит обратно в нумерическую форму

**Т.о.** программа выводит наименьшее ближайшее чётное число для цифры, поданной на ввод

**Вывод:** в ходе выполнения данной лабораторной работы я изучил обработчик текста sed, а также узнал интересные способы его использования (нахождение самого длинного элемента массива, арифметика и калькулятор)