

ИТМО
Факультет ПИиКТ
Системное программное обеспечение

Лабораторная работа №5

Нестеров Дали Константинович
Группа Р3202

Задание 1

1. Замените имя Jon на Jonathan
`sed 's/Jon/Jonathan/' 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 днем рождения в ноябре или декабре
`sed -nE '/: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/$/****/' datebook`
7. Замените строку, содержащую Jose на JOSE HAS RETIRED
`sed '/Jose/ s/.*/JOSE HAS RETIRED/' datebook`
8. Замените дату рождения Popeye на 11/14/46. При этом подразумевается, что вы не знаете значение даты, хранящейся в файле. Составьте и используйте регулярное выражение для ее поиска.
`sed '/^Popeye/ s![0-9]\\{1,2\\}/[0-9]\\{1,2\\}/[0-9]\\{1,2\\}!11/14/46!' datebook`
9. Удалите все пустые строки
`sed '/^$/d' datebook`
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 THE END
`sed '1 i\\TITLE OF FILE;

s/:[0-9]*[50]00//;

s/^\\([A-Za-Z]*\\) \\([A-Za-z]*\\)/\\2 \\1/;

s/$/ THE END/' datebook`

Задание 2

1. `sed '/north/p' datafile`
Выводит все строки из файла datafile, дублируя строки, в которых содержится north. По умолчанию sed обрабатывает каждую строку и выводит их по мере обработки в независимости от того совпадает ли она с регулярным выражением. Команда p вдобавок к этому выводит все строки, совпадающие с регулярным выражением.
2. `sed -n '/north/p' datafile`

Выводит все строки, содержащие north. Ключ -n используется для подавления стандартного вывода.

3. `sed '3d' datafile`
Удаляет третью строку. Команда d используется для удаления строк.
4. `sed '3,$d' datafile`
Удаляет строки начиная с третьей и до конца файла.
5. `sed '$d' datafile`
Удаляет последнюю строку.
6. `sed '/north/d' datafile`
Удаляет строки, содержащие north.
7. `sed 's/west/north/g' datafile`
Заменяет все вхождения west на north. Флаг g используется для замены всех вхождений в строке, а не только первого.
8. `sed -n 's/^west/north/p' datafile`
Подставляет north вместо west в строках, которые начинаются с west и выводит только эти строки.
9. `sed 's/[0-9][0-9]$/&.5/' datafile`
Добавляет .5 к концу строк, оканчивающихся на две любые цифры. & соответствует \0, то есть всей строке, которая подошла под регулярное выражение, и используется, чтобы добавлять новый текст к строке.
10. `sed -n 's/Hemenway/Jones/gp' datafile`
Заменяет все вхождения Hemenway на Jones и выводит только измененные строки.
11. `sed -n 's/\(Stag\)got/\lianne/p' datafile`
Заменяет первое вхождение слова Staggot на Staggiane в каждой строке и выводит измененные строки. Скобки используются для группировки, а \1 подставляет текст, попавшийся в первую группу.
12. `sed 's#14#88#g' datafile`
Заменяет все вхождения 14 на 88. В конструкции s/a/b/ можно использовать любые свои разделители, например s#a#b# или s!a!b!. Запрещены только обратный слэш и символ новой строки.
13. `sed -n '/west/,/east/p' datafile`
Выводит все строки, находящиеся между двух строк, содержащих west и east соответственно.
14. `sed -n '5,/^\northeast/p' datafile`
Выводит все строки, начиная с пятой строки до строки, начинающейся с northeast. (Если такой строки нет, то с пятой строки и до конца файла.)

15. `sed '/west/,/east/s/$/**WAKA**/' datafile`

Для каждой строки, в диапазоне от строки, содержащей west, до строки, содержащей east, происходит добавление ****WAKA**** в конец строки

16. `sed -e '1,3d' -e 's/Hemenway/Jones/' datafile`

Ключ -e служит для объединения нескольких правил обработки. Таким образом, сначала удалятся строки с первой по третью, а затем первые вхождения Hemenway в каждой строке заменятся на Jones.

17. `sed '/Suan/r newfile' datafile`

Команда r служит для добавления текста из файлов. После каждой строки, содержащей Suan добавится текст из файла newfile.

18. `sed -n '/north/w newfile' datafile`

Команда w служит для записи строк в файл. В newfile запишутся все строки, содержащие north, а на стандартный вывод ничего подаваться не будет.

19. `sed '/^north /a\
--->THE NORTH SALES DISTRICT HAS MOVED<---' datafile`

Добавляет строку --->THE NORTH SALES DISTRICT HAS MOVED<--- после строк, начинающихся с north .

20. `sed '/eastern/i\
NEW ENGLAND REGION\
-----' datafile`

Добавляет строки

NEW ENGLAND REGION

Перед строками, содержащими eastern. \ здесь используется для переноса строки.

21. `sed '/eastern/c\
THE EASTERN REGION HAS BEEN TEMPORARILY CLOSED' datafile`

Заменяет строки, содержащие eastern на THE EASTERN REGION HAS BEEN TEMPORARILY CLOSED.

22. `sed '/eastern/{ n; s/AM/Archie/; }' datafile`

Находит строки, содержащие eastern и заменяет в строках, следующих непосредственно за ними первое вхождение AM на Archie. { } используются для применения нескольких правил к одному условию. Команда n применяет дальнейшие команды к следующей строке.

23. `sed '2,4y/abcdefghijklmnopqrstuvwxyz/ABCDEFGHIJKLMNOPQRSTUVWXYZ/' datafile`

Команда y осуществляет замену каждого исходного символа на символ из второго аргумента с соответствующей позиции. В результате в строках со второй по четвертую все буквы станут заглавными.

24. `sed '1q' datafile`

Команда q завершает выполнение после определенной строки. В результате будет выведена первая строка файла.

25. `sed '/Lewis/{ s/Lewis/Joseph/;q; }' datafile`
В первой строке, содержащей Lewis, это слово заменится на Joseph и выполнение прекратится (больше строк выведено не будет).
26. `sed -e '/northeast/h' -e '$G' datafile`
h сохраняет строку в буфер hold space, а G добавляет строку из этого буфера. В данном случае будет сохранена последняя строка, содержащая northeast и вставлена в конец файла
27. `sed -e '/WE/{h; d; }' -e '/CT/{G; }' datafile`
Строки, содержащие WE помещаются в буфер и удаляются из вывода. После строк, содержащих CT, добавляется текущая строка из буфера
28. `sed -e '/northeast/h' -e '$g' datafile`
g похож на G, только он не добавляет, а заменяет строку. В итоге будет сохранена последняя строка, содержащая northeast и вставлена вместо последней строки файла.
29. `sed -e '/WE/{h; d; }' -e '/CT/{g; }' datafile`
Строки, содержащие WE помещаются в буфер и удаляются из вывода. После строк, содержащих CT, добавляется текущая строка из буфера
30. `sed -e '/Patricia/h' -e '/Margot/x' datafile`
Команда x меняет местами строку из файла и строку из буфера. Таким образом строка
31. `sed -n '/sentimental/p' datafile`
Выводит все строки, содержащие sentimental.
32. `sed '0,6d' datafile > newfile`
Удаляет строки с нулевой по шестую и результат записывает в newfile. Команда не срабатывает, т. к. отсчет строк начинается с единицы.
33. `sed '/[Dd]aniel/d' datafile`
Удаляет строки, содержащие Daniel или daniel.
34. `sed -n '19,20p' datafile`
Выводит 19 и 20 строку.
35. `sed '1,10s/Montana/MT/g' datafile`
Заменяет все вхождения Montana на MT в строках с первой по десятую.
36. `sed '/March/!d' datafile`
Удаляет все строки, кроме тех, где есть слово March.
37. `sed '/report/s/5/8/' datafile`
Заменяет первое вхождение цифры 5 на 8 в строках, где есть слово report.
38. `sed 's/...../' datafile`

В каждой строке, состоящей из 4 и больше символов, удаляются первые 4 символа.

39. `sed 's/...$//' datafile`

В каждой строке, состоящей из 3 и больше символов, удаляются последние три символа.

40. `sed '/east/,/west/s/North/South/' datafile`

В строках, между строками с подстроками east и west, первый North заменяется на South.

41. `sed -n '/Time off/w timefile' datafile`

Строки, содержащие Time off записываются в timefile.

42. `sed 's/\([Oo]ccur\)ence/\1rence/' datafile`

Заменяет Occurence и occurrence на Occurrence и occurrence соответственно.

43. `sed -n l datafile`

Выводит содержимое файла и непечатные символы.

Задание 3

```
$ cat sed1
/Lewis/a\
Lewis is the TOP Salesperson for April!!\
Lewis is moving to the southern district next month.\
CONGRATULATIONS!
/Margot/c\
*****\
MARGOT HAS RETIRED\
*****
li\
EMPLOYEE DATABASE\
-----
$d
```

После строк, содержащих Lewis вставляет текст:

Lewis is the TOP Salesperson for April!!

Lewis is moving to the southern district next month.

CONGRATULATIONS!

Строки, содержащие Margot, заменяются на текст:

MARGOT HAS RETIRED

Перед первой строкой вставляет текст:

EMPLOYEE DATABASE

Последняя строка удаляется

```
$ cat sed2
/western/, /southeast/{
/^ */d
/Suan/{ h; d; }
}
/Ann/g
s/TB \ (Savage\)/Thomas \1/
```

Для каждой строки, находящейся между строк, содержащих western и southeast соответственно выполняется следующие два действия: 1) Пустые строки и строки, содержащие только пробелы, удаляются. 2) Строки, содержащие Suan помещаются в буфер и удаляются. Строки, содержащие Ann, заменяются на строки из буфера. Строки, содержащие TB Savage, заменяются на Thomas Savage.

Задание 4

```
war=peace      ; sleep=cat ; ask=man ;
freedom=slavery ; whole=tee ; or=more ;
ignorance=strength ; life=pain ; die=cat ;
echo "Don't worry! " ; dd if=/dev/urandom \
bs=17 count=1 2>/dev/null |openssl base64 |
less|$sleep | more|tee|$or|$die | gsed ':s;
s/\(^|\n\)\|(\^[^n]\)\|(\^[^n]*\)$/\1\2\n\3/ ;
ts'|gsed -r -n 'x;s/^.*/iiiiiii/x; :s;N;
x;s/^i(i*)$/\1/x;ts;s/\n//g; s/^. {3}/! ?+ /;
y! ? !+ ! l A l !; s/^(. {3}). (....) . /\1 \2 /;
s/ ./ wi /; s/i. /i /; h; s/. (. {2}) . * /\1 /; G;
s/^(.*) \n (.... {3}) (..) (.) ($) /\2 \1 \4 /; x;
s/^.*/+123Mec /; x; :t; N; 12 {s/(. {13}) . /\1 /};
11 {s/.$/b! /}; s/.. (b) (!) /\1 e \2 \2 \2 /; :r;
s/! //; tr; x; s/^.//; x; tt; s/(.) . {2} $ /\1 \n /;
s/\n//g; 17 {s/[a-zA-Z0-9=] {4} $/#+?=/;
y.=?.ie.; x; s!^.*$! ?>d!; x; :f;
s/(.) i ([^i]*) $/i \1 \2 /; x; th; :h; ss.ss;x; tf;
s/e(i) .../e \ln /; s/i [^l] /f&/}; s$ \ $ $ . $; p' & &
$ask https://vk.com/id248059105 with love \
2>&1 |tail -1 ;kill -STOP $$
```

На первой строке объявляются переменные.

Выводится строка Don't worry!

Копируется один блок в 17 байт из /dev/urandom с перенаправлением ошибок в /dev/null, затем они конвертируются в еще более случайные символы с помощью openssl.

Набор команд `less | cat | more | tee | more | cat` не несет никакого полезного смысла и оставит те же 17 байт.

`gsed ':s; s/\(^|\n\)\([\^\\n]*\)$/\1\2\n\3/ ; ts'` – в этой команде : используется для объявления метки, а `t` для перехода к этой метке в случае, если предыдущая замена прошла успешно. В результате получится строка, где каждый символ на отдельной строке.

`s/^. {3}/! ? + /; y! ? \! + ! l A l !;` заменяет первые три символа на All

`s/^(. {3}). (....) ./\1 \2 /; s/ ./ wi/;` напишет wi после All

`s/.. (b) (!) /\1e\2\2\2/;` напишет be

Затем напишется слово fine

В результате программа выведет:

«Don't worry!

All will be fine.

No manual entry for love.»

И процесс завершится.