

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет прикладной математики и информатики
Кафедра технологий программирования

Кражевский Алексей Игоревич
ОТЧЕТ ПО ПРОГРАММИРОВАНИЮ МОБИЛЬНЫХ И
ВСТРАИВАЕМЫХ СИСТЕМ
студента 3 курса 13а группы
Лабораторная работа №2

Преподаватель
Давидовская М.И.

Минск, 2022

Цели работы:

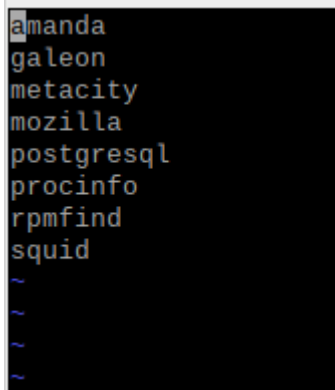
- Изучить синтаксис команд перенаправления ввода-вывода.
- Изучить утилиту `awk`, ее функциональные возможности и синтаксис.
- Изучить потоковый редактор `sed` и его возможности.
- Изучить язык сценариев `bash`

Вариант задания: 1

Задание 2.1.

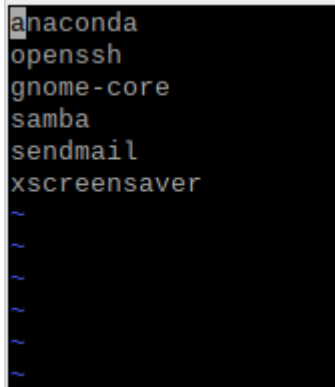
1. Создайте в текстовом редакторе два файла для дальнейшего использования в лабораторной работе.

`packages1.txt`:



```
amanda
galeon
metacity
mozilla
postgresql
procinfo
rpmfind
squid
~
~
~
```

`packages2.txt`:



```
anaconda
openssh
gnome-core
samba
sendmail
xscreensaver
~
~
~
~
~
```

5. Используя директиву `'>'`, перенаправим стандартный вывод из одного файла в другой файл

```
pi@raspberrypi:~/Desktop $ cat packages1.txt > packages1.catfile
pi@raspberrypi:~/Desktop $ cat packages1.catfile
amanda
galeon
metacity
mozilla
postgresql
procinfo
rpmfind
squid
pi@raspberrypi:~/Desktop $
```

Затем сверим исходный и полученный файлы

```
pi@raspberrypi:~/Desktop $ diff packages1.txt packages1.catfile
pi@raspberrypi:~/Desktop $ ls -l packages1*
-rw-r--r-- 1 pi pi 65 Sep 25 20:08 packages1.catfile
-rw-r--r-- 1 pi pi 65 Sep 25 20:06 packages1.txt
pi@raspberrypi:~/Desktop $
```

6. Для присоединения существующего файла к другому файлу предназначена директива '>>'

```
pi@raspberrypi:~/Desktop $ cat packages2.txt >> packages1.catfile
pi@raspberrypi:~/Desktop $ cat packages1.catfile
amanda
galeon
metacity
mozilla
postgresql
procinfo
rpmfind
squid
anaconda
openssh
gnome-core
samba
sendmail
xscreensaver
pi@raspberrypi:~/Desktop $
```

Подключиться по ssh из Raspberry PI и выполнить примеры ниже в Ubuntu

7. Если команде cat не передается аргумент и стандартный ввод перенаправлен в файл, то весь ввод с клавиатуры до нажатия клавиш <Ctrl+d> будет перенаправлен в файл

```

(krazhevskiy13@kali)-[~]
$ cat > typedin.txt
some text

(krazhevskiy13@kali)-[~]
$ ls -l typedin.txt
-rw-r--r-- 1 krazhevskiy13 krazhevskiy13 10 Sep 25 15:20 typedin.txt

(krazhevskiy13@kali)-[~]
$ cat typedin.txt
some text

(krazhevskiy13@kali)-[~]

```

8. Повторите предыдущий шаг, подставив вместо команды cat команду tr.

```

(krazhevskiy13@kali)-[~]
$ tr 'aeiou' 'AEIOU' > trfile.txt
text to insert
some text

(krazhevskiy13@kali)-[~]
$ ls -l trfile.txt
-rw-r--r-- 1 krazhevskiy13 krazhevskiy13 25 Sep 25 15:22 trfile.txt

(krazhevskiy13@kali)-[~]
$ cat trfile.txt
tExt t0 InsErt
sOmE tExt

```

9. Команда cat принимает в качестве аргумента имя файла или стандартный ввод, перенаправленный из файла. Проверьте это при помощи следующих двух команд:

```

(krazhevskiy13@kali)-[~]
$ cat packages1.txt
amanda
galeon
metacity
mozilla
postgresql
procinfo
rpmfind
squid

(krazhevskiy13@kali)-[~]
$ cat < packages1.txt
amanda
galeon
metacity
mozilla
postgresql
procinfo
rpmfind
squid

```

10. Однако команда tr принимает ввод только из стандартного канала.

```

(krazhevskiy13@kali)-[~]
$ tr 'aeiou' 'AEIOU' < packages1.txt
AmAndA
gAlEOn
mEtAcItY
m0zIlla
p0stgrEsq1
pr0cInf0
rpmfInd
sqUIId
(krazhevskiy13@kali)-[~]

```

11. В следующем примере стандартный ввод и вывод одновременно перенаправляются

```

(krazhevskiy13@kali)-[~]
$ tr 'aeiou' 'AEIOU' < packages1.txt > packages1.trfile.txt

(krazhevskiy13@kali)-[~]
$ ls -l packages1.txt packages1.trfile.txt
-rw-r--r-- 1 krazhevskiy13 krazhevskiy13 65 Sep 25 15:28 packages1.trfile.txt
-rw-r--r-- 1 krazhevskiy13 krazhevskiy13 65 Sep 25 15:26 packages1.txt

(krazhevskiy13@kali)-[~]
$ cat packages1.trfile.txt
AmAndA
gAlEOn
mEtAcItY
m0zIlla
p0stgrEsq1
pr0cInf0
rpmfInd
sqUIId
(krazhevskiy13@kali)-[~]

```

Задание 2.2. Утилита awk примеры для изучения

1. Создаем каталог examples в домашнем каталоге пользователя.
 2. Убедитесь, что Вы находитесь в домашнем каталоге, выполнив команду
 3. Перенаправим вывод команды ls расширенном формате в файл,
 4. И выполним команду для поиска данных в созданном файле или наборе файлов. Будем искать строки, содержащие слово «bash»
- 1-4:

```
pi@raspberrypi:~$ mkdir examples
pi@raspberrypi:~$ pwd
/home/pi
pi@raspberrypi:~$ ls -la > examples/log.txt
pi@raspberrypi:~$ awk '/bash/ {print}' examples/*
-rw----- 1 pi pi 3061 Sep 25 20:30 .bash_history
-rw-r--r-- 1 pi pi 220 Jul 1 10:32 .bash_logout
-rw-r--r-- 1 pi pi 3523 Jul 1 10:32 .bashrc
pi@raspberrypi:~$
```

2. Просмотрите файл

```
(krazhevskiy13@kali)-[~/examples]
$ vim myfile.txt

(krazhevskiy13@kali)-[~/examples]
$ cat myfile.txt
This is a test.
This is the second test.
This is the third test.
This is the fourth test.

(krazhevskiy13@kali)-[~/examples]
```

2. Выполните вывод первого элемента каждой строки:

```
(krazhevskiy13@kali)-[~/examples]
$ awk '{print $1}' myfile.txt
This
This
This
This

(krazhevskiy13@kali)-[~/examples]
```

3. Выполните вывод третьего элемента каждой строки:

```
(krazhevskiy13@kali)-[~/examples]
$ awk '{print $3}' myfile.txt
a
the
the
the

(krazhevskiy13@kali)-[~/examples]
$
```

1. awk позволяет изменять регистр символов. В примере ниже необходимо ввести команду и две строки текста, одна из которых будет преобразована в строку в верхнем регистре

```
(krazhevskiy13@kali)-[~/examples]
$ awk '/foo/ { print toupper($0); }'
This line contains bar.
This line contains foo.
THIS LINE CONTAINS FOO.
^C

(krazhevskiy13@kali)-[~/examples]
$
```

2. Вывести список, состоящий из фамилий, имен и рейтинговых оценок

```
(krazhevskiy13@kali)-[~/examples]
$ vim list_students

(krazhevskiy13@kali)-[~/examples]
$ awk '{print $1,$2,$5}' list_students
Astashko Ivan 6
Byzyn Evgeniy 9
Kravchenya Leonid 4
Koshkin Vladimir 7
Lipin Fedor 4
Penov Nikolay 5
Yashin Petr 8
```

Выбрать из исходного файла list_students студентов с рейтинговой оценкой 8 и распечатать все значения полей для выбранных записей

```
(krazhevskiy13@kali)-[~/examples]
$ awk '/8/ {print $0}' list_students
Yashin Petr KB 4 8

(krazhevskiy13@kali)-[~/examples]
$
```

Выбрать из списка студентов, рейтинговая оценка которых 4

```
(krazhevskiy13@kali)-[~/examples]
$ awk '($5==4)' list_students
Kravchenya Leonid PM 1 4
Lipin Fedor PI 2 4

(krazhevskiy13@kali)-[~/examples]
$
```

Выбрать из списка студентов специальности КБ, 1 курса

```
(krazhevskiy13@kali) - [~/examples]
$ awk '($3=="KB")&&($4==1)' list_students
Byzyn Evgeniy KB 1 9

(krazhevskiy13@kali) - [~/examples]
$
```

Выбрать из файла со списком студентов все имена с длиной >5

```
(krazhevskiy13@kali) - [~/examples]
$ awk '{length($2)>5} {print}' list_students
Byzyn Evgeniy KB 1 9
Kravchenya Leonid PM 1 4
Koshkin Vladimir IN 1 7
Penov Nikolay PM 1 5

(krazhevskiy13@kali) - [~/examples]
$
```

Вывести данные исходного файла в виде таблицы и пронумеровать строки

```
(krazhevskiy13@kali) - [~/examples]
$ awk '{print NR, $0}' list_students
1 Astashko Ivan PI 1 6
2 Byzyn Evgeniy KB 1 9
3 Kravchenya Leonid PM 1 4
4 Koshkin Vladimir IN 1 7
5 Lipin Fedor PI 2 4
6 Penov Nikolay PM 1 5
7 Yashin Petr KB 4 8
```

Вывести все имена студентов в верхнем регистре

```
(krazhevskiy13@kali) - [~/examples]
$ awk '{print toupper($2)}' list_students
IVAN
EVGENIY
LEONID
VLADIMIR
FEDOR
NIKOLAY
PETR
```

Посчитать суммарный балл оценок студентов

```
(krazhevskiy13@kali) - [~/examples]
$ awk '{sum += $5} END {print("SUM=", sum)}' list_students
SUM= 43

(krazhevskiy13@kali) - [~/examples]
$
```

Присутствует ли фамилия Леонов в списке файла list_students?


```

(krazhevskiy13@kali)-[~/examples]
$ awk '($1 ~ /Leonov/)' list_students

(krazhevskiy13@kali)-[~/examples]
$ awk '($1 ~ /Kravchenya/)' list_students
Kravchenya Leonid PM 1 4

(krazhevskiy13@kali)-[~/examples]

```

Проверить, есть ли запись студента Липина в списке файла list_students, причем фамилия может быть написана с прописной или строчной буквы

```

(krazhevskiy13@kali)-[~/examples]
$ awk '($1 ~ /[Ll]ipin/)' list_students
Lipin Fedor PI 2 4

(krazhevskiy13@kali)-[~/examples]

```

Вывести все записи кроме студента Липин

```

(krazhevskiy13@kali)-[~/examples]
$ awk '($1 !~ /[Ll]ipin/)' list_students
Astashko Ivan PI 1 6
Byzyn Evgeniy KB 1 9
Kravchenya Leonid PM 1 4
Koshkin Vladimir IN 1 7
Penov Nikolay PM 1 5
Yashin Petr KB 4 8

(krazhevskiy13@kali)-[~/examples]

```

Выберите все записи из файла colours.csv, количество которых больше 5

```

(krazhevskiy13@kali)-[~/examples]
$ vim colours.csv

(krazhevskiy13@kali)-[~/examples]
$ awk -F"," ' $3>6 {print $1, $2}' colours.csv > output.txt

(krazhevskiy13@kali)-[~/examples]
$ cat output.txt
grape purple
apple green
potato brown

(krazhevskiy13@kali)-[~/examples]

```

Задание 2.3. Обработка строк файла с применением потокового редактора sed

2. Чтобы вывести все строки файла и продублировать строки согласно шаблону, т. е. содержащие слово book выполните команду

```
pi@raspberrypi:~$ sed '/book/ p' books
Book one.
The second book.
The second book.
The third.
This is book four.
This is book four.
Five.
This is book sixth.
This is book sixth.
This is book seven.
This is book seven.
Eighth and last.
pi@raspberrypi:~$
```

3. Чтобы выбрать определенные строки, например содержащие слово book, выполните команду

```
pi@raspberrypi:~$ sed -n '/book/ p' books
The second book.
This is book four.
This is book sixth.
This is book seven.
pi@raspberrypi:~$
```

4. Чтобы вывести часть файла, например, строки с 2 по 5, выполните команду

```
pi@raspberrypi:~$ sed -n '2,5 p' books
The second book.
The third.
This is book four.
Five.
pi@raspberrypi:~$
```

5. Для выполнения более сложных и длинных инструкций можем использовать файл программы для sed. Выполним команду из примера 4, указав параметры в файле records

```
(krazhevskiy13@kali)-[~]
$ vim records

(krazhevskiy13@kali)-[~]
$ sed -n -f records books
The second book.
The third.
This is book four.
Five.

(krazhevskiy13@kali)-[~]
```

6. В данном примере выбираем строку 3 и используем инструкцию Добавить, чтобы добавить разделитель строк и текст «My favorite book.» к третьей строке

```
(krazhevskiy13@kali)-[~]
$ vim appends

(krazhevskiy13@kali)-[~]
$ sed -f appends books
Book one.
The second book.
The third.
My favourite book.
This is book four.
Five.
This is book sixth.
This is book seven.
Eighth and last.

(krazhevskiy13@kali)-[~]
$
```

7. В данном примере требуется вставить разделитель строк и текст «SKARBONKA.» перед строками, в которых содержится слово «This»

```
(krazhevskiy13@kali)-[~]
$ vim insert

(krazhevskiy13@kali)-[~]
$ sed -f insert books
Book one.
The second book.
The third.
SKARBONKA.
This is book four.
Five.
SKARBONKA.
This is book sixth.
SKARBONKA.
This is book seven.
Eighth and last.
```

8. В следующем примере выполняется замена слова book на novel

```
(krazhevskiy13@kali)-[~]  
$ sed -n 's/book/novel/ p' books  
The second novel.  
This is novel four.  
This is novel sixth.  
This is novel seven.  
(krazhevskiy13@kali)-[~]
```

Задание 2.4. Исполнительная среда оболочки Bash

Выполните следующие команды, которые используют переменные `a`, `full_name` и `short_name` командной оболочки Bash

```
pi@raspberrypi:~$ a=879
pi@raspberrypi:~$ echo "The value of \"a\" is $a."
The value of "a" is 879.
pi@raspberrypi:~$ export full_name="John Lee"
pi@raspberrypi:~$ echo $full_name
John Lee
pi@raspberrypi:~$ export short_name="John"
pi@raspberrypi:~$ export short_name
pi@raspberrypi:~$ echo $ short_name
$ short_name
pi@raspberrypi:~$
```

Выполните следующие команды, которые иллюстрируют операции над переменными

```
pi@raspberrypi:~$ export foo=""
pi@raspberrypi:~$ echo ${foo:-one}
one
pi@raspberrypi:~$ echo $foo
pi@raspberrypi:~$ echo ${foo:=one}
one
pi@raspberrypi:~$ export foo="this is a test"
pi@raspberrypi:~$ echo $foo
this is a test
pi@raspberrypi:~$ echo ${foo:+bar}
bar
pi@raspberrypi:~$
```

4. Выполнить следующие команды, которые создают переменные массивы.

```
(krazhevskiy13@kali) - [~]
$ ARR[1]=one

(krazhevskiy13@kali) - [~]
$ ARR[2]=two

(krazhevskiy13@kali) - [~]
$ ARR[3]=three

(krazhevskiy13@kali) - [~]
$ ARR1=(zero one two three)

(krazhevskiy13@kali) - [~]
$
```

5. Выполнить следующую команду, которая выводит на консоль значения элементов массива

```
(krazhevskiy13@kali)-[~]
$ echo ${ARR[0]} ${ARR[1]}
one
(krazhevskiy13@kali)-[~]
```

6. Выполните следующие команды для удаления элемента ARR[1] массива ARR и для удаления всего массива ARR:

```
(krazhevskiy13@kali)-[~]
$ unset ARR[1]

(krazhevskiy13@kali)-[~]
$ unset ARR

(krazhevskiy13@kali)-[~]
```

7. Создайте псевдоним для команды clear, выполнив следующую команду

```
(krazhevskiy13@kali)-[~]
$ alias c='clear'
```

Просмотрите созданные псевдонимы

```
(krazhevskiy13@kali)-[~]
$ alias
c=clear
diff='diff --color=auto'
egrep='egrep --color=auto'
fgrep='fgrep --color=auto'
grep='grep --color=auto'
history='history 0'
ip='ip --color=auto'
l='ls -CF'
la='ls -A'
ll='ls -l'
ls='ls --color=auto'
which-command=whence

(krazhevskiy13@kali)-[~]
$
```

8. Откройте файл .bashrc в текстовом редакторе, например vim

```
krazhevskiy13@kali: ~  
File Edit Tabs Help  
# ~/.bashrc: executed by bash(1) for non-login shells.  
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)  
# for examples  
  
# If not running interactively, don't do anything  
case $- in  
  *i*) ;;  
  *) return;;  
esac  
  
# don't put duplicate lines or lines starting with space in the  
# See bash(1) for more options  
HISTCONTROL=ignoreboth  
  
# append to the history file, don't overwrite it  
shopt -s histappend  
  
# for setting history length see HISTSIZE and HISTFILESIZE  
HISTSIZE=1000  
HISTFILESIZE=2000  
  
# check the window size after each command and, if necessary,  
# update the values of LINES and COLUMNS.  
".bashrc" 156L, 5551B
```

```
alias l='ls -CF'  
alias c='clear'
```

9. Чтобы проверить сохранение псевдонима, сначала выйдите из командной оболочки, а затем снова войдите в неё под именем student и выполните команду

```
(krazhevskiy13@kali)-[~]  
$ alias  
c=clear  
diff='diff --color=auto'  
egrep='egrep --color=auto'  
fgrep='fgrep --color=auto'  
grep='grep --color=auto'  
history='history 0'  
ip='ip --color=auto'  
l='ls -CF'  
la='ls -A'  
ll='ls -l'  
ls='ls --color=auto'  
which-command=whence
```

10. Отобразите в терминале текущее значение вашей строки приглашения, выполнив команду

```
(krazhevskiy13@kali)-[~]  
$ echo $PS1  
%F{%(#.blue.green)}└─${debian_chroot:+($debian_chroot)}$  
ame $VIRTUAL_ENV)}-(%B%F{%(#.red.blue)}%n⊕%m%b%F{%(#.blu  
}%{6~.%-1~/.../%4~.%5~}%b%F{%(#.blue.green)}}]  
└─%B{#.%F{red}#.%F{blue}$}%b%F{reset}
```

11. Измените вашу строку приглашения, выполнив следующую команду

```
(krazhevskiy13@kali)-[~]
$ PS1='Linux -> '
Linux -> 
```

12. Восстановите традиционную строку приглашения, выполнив следующую команду

```
(krazhevskiy13@kali)-[~]
$ PS1='\h $ '
\h $ 
```

2.4.2. Примеры для изучения: Исполнение команд

1. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором фигурных скобок

```
(krazhevskiy13@kali)-[~]
$ echo sp{el,il,al}l
spell spill spall
(krazhevskiy13@kali)-[~]
$ 
```

2. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором символа тильда на домашний каталог пользователя

```
(krazhevskiy13@kali)-[~]
$ cat ~/message.txt
"~" is not a valid identifier
```

3. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором переменной

```
(krazhevskiy13@kali)-[~]
$ echo $SHELL
/usr/bin/zsh
```

4. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором команд

```
(krazhevskiy13@kali)-[~]
$ echo $(date)
Mon Sep 26 03:59:20 AM EDT 2022
```

5. Объявите переменные X и Y и задайте им некоторые значения. Выполните следующую команду, которая иллюстрирует расширение командным интерпретатором арифметического выражения


```
(krazhevskiy13@kali)-[~]
$ X=5

(krazhevskiy13@kali)-[~]
$ Y=6
```

```
(krazhevskiy13@kali)-[~]
$ echo Area: $[ $X * $Y ]
Area: 30
```

6. Выполните следующую команду, которая иллюстрирует запрещение расширения символа, который следует за символом обратный слэш

```
(krazhevskiy13@kali)-[~]
$ echo Your cost: \5,00
Your cost: 5,00
```

7. Выполните следующую команду, которая иллюстрирует запрещение расширения символов, заключенных в одинарные кавычки

```
(krazhevskiy13@kali)-[~]
$ echo '$date'
$date

(krazhevskiy13@kali)-[~]
$ date
Mon Sep 26 04:01:37 AM EDT 2022
```

2.4.3. Примеры для изучения: Сценарии Bash

Пример 1

Script:

```
File Edit Tabs Help
#!/bin/bash
# This script displays some information about your environment
echo "Hello."
echo "The date and time are $(date)"
echo "Your working directory is: $(pwd)"
```

```
(krazhevskiy13@kali)-[~/task2.4.3]
$ chmod u+x ascript.sh

(krazhevskiy13@kali)-[~/task2.4.3]
$ ./ascript.sh
Hello.
The date and time are Mon Sep 26 04:07:10 AM EDT 2022
Your working directory is: /home/krazhevskiy13/task2.4.3
```

Пример 2

```
(krazhevskiy13@kali) - [~/task2.4.3]
$ vim script2.sh

(krazhevskiy13@kali) - [~/task2.4.3]
$ chmod u+x script2.sh

(krazhevskiy13@kali) - [~/task2.4.3]
$ ./script2.sh
Enter your name and press [ENTER]: Alex
Your name is: Alex
```

Пример 3

```
(krazhevskiy13@kali) - [~/task2.4.3]
$ printf "%d\n" 5
5

(krazhevskiy13@kali) - [~/task2.4.3]
$ printf "Hello, $USER.\n\n"
Hello, krazhevskiy13.

(krazhevskiy13@kali) - [~/task2.4.3]
$ distance=15

(krazhevskiy13@kali) - [~/task2.4.3]
$ printf "Distance is %5d Miles\n" $distance
Distance is    15 Miles

(krazhevskiy13@kali) - [~/task2.4.3]
$ printf "%d\n" 0xF
15

(krazhevskiy13@kali) - [~/task2.4.3]
$ printf "0x%X\n" 15
0xF
```

Пример 4

```
(krazhevskiy13@kali) - [~/task2.4.3]
$ vim example.awk

(krazhevskiy13@kali) - [~/task2.4.3]
$ chmod u+x example.awk

(krazhevskiy13@kali) - [~/task2.4.3]
$ ./example.awk colours.csv
2: banana,yellow,6
3: strawberry,red,3
4: grape,purple,10
5: apple,green,8
6: plum,purple,2
7: kiwi,brown,4
8: potato,brown,9
9: pineapple,yellow,5
```

Пример 5

```

(krazhevskiy13@kali) - [~/task2.4.3]
$ vim script5.sh

(krazhevskiy13@kali) - [~/task2.4.3]
$ chmod u+x script5.sh

(krazhevskiy13@kali) - [~/task2.4.3]
$ ./script5.sh
false

```

Пример 6

```

(krazhevskiy13@kali) - [~/task2.4.3]
$ vim script6.sh

(krazhevskiy13@kali) - [~/task2.4.3]
$ chmod u+x script6.sh

(krazhevskiy13@kali) - [~/task2.4.3]
$ ./script6.sh
Which Linux distribution do you know?

ubuntu
I know ubuntu.

```

Пример 7

```

(krazhevskiy13@kali) - [~/task2.4.3]
$ bash script7.sh
1      iteration
2      iteration
3      iteration
echo Manufacturer
      is: Apple Sony Panasonic "Hewlett Packard" Nokia
infinite loops [CTRL+C to
      stop]
infinite loops [CTRL+C to
      stop]

```

Пример 8

```

(krazhevskiy13@kali) - [~/task2.4.3]
$ vim script8.sh

(krazhevskiy13@kali) - [~/task2.4.3]
$ ./script8.sh
While
x = 1
x = 2
x = 3
x = 4

Until
x = 5
x = 4

```

Пример 9

```

(krazhevskiy13@kali) - [~/task2.4.3]
$ vim script9.sh

(krazhevskiy13@kali) - [~/task2.4.3]
$ chmod u+x script9.sh

(krazhevskiy13@kali) - [~/task2.4.3]
$ ./script9.sh
Iteration #1
Iteration #2
To skip iteration No. 3

Iteration #4
Iteration #5

```

Пример 10

```

(krazhevskiy13@kali) - [~/task2.4.3]
$ vim script10.sh

(krazhevskiy13@kali) - [~/task2.4.3]
$ ./script10.sh
Iteration #1
Iteration #2
Iteration #3

To break loop after 3 iterations

(krazhevskiy13@kali) - [~/task2.4.3]
$ █

```

Пример 11

```

(krazhevskiy13@kali)-[~/task2.4.3]
$ ./script11.sh arg1 2 decimal
My
name: ./script11.sh
Total number of arguments: 3
Arg 1: arg1
Arg 2: 2
Arg 3: decimal

(krazhevskiy13@kali)-[~/task2.4.3]
$

```

Пример 12

```

(krazhevskiy13@kali)-[~/task2.4.3]
$ ./script12.sh 1 2 3 4 ds lk 5 dfv
1
2
3
4
ds
lk
5
dfv

```

Пример 13

```

(krazhevskiy13@kali)-[~/task2.4.3]
$ function mcd() { mkdir $1 && cd $1; }

(krazhevskiy13@kali)-[~/task2.4.3]
$ mcd temp

(krazhevskiy13@kali)-[~/task2.4.3/temp]
$

```

Задание 2.5. Для самостоятельной работы

Вариант 1

Задача 1. Перенаправьте вывод команды `ls`, просматривающей содержимое текущего каталога в файл `file_list.txt`. В выводе `ls` отображать все файлы, включая скрытые, inode (индексный дескриптор) файла и размер в килобайтах. Просмотрите содержимое файла `file_list.txt`.

```
pi@raspberrypi:~$ ls -a -l -s > file_list.txt
pi@raspberrypi:~$ cat file_list.txt
total 96
4 drwxr-xr-x 17 pi pi 4096 Sep 26 10:26 .
4 drwxr-xr-x 3 root root 4096 Sep 3 16:29 ..
4 -rw----- 1 pi pi 3795 Sep 26 09:53 .bash_history
4 -rw-r--r-- 1 pi pi 220 Jul 1 10:32 .bash_logout
4 -rw-r--r-- 1 pi pi 3523 Jul 1 10:32 .bashrc
4 drwxr-xr-x 2 pi pi 4096 Sep 3 16:17 Bookshelf
4 drwxr-xr-x 8 pi pi 4096 Sep 3 16:30 .cache
4 drwx----- 14 pi pi 4096 Sep 25 22:36 .config
4 drwxr-xr-x 3 pi pi 4096 Sep 25 21:40 Desktop
4 drwxr-xr-x 2 pi pi 4096 Sep 3 16:29 Documents
4 drwxr-xr-x 2 pi pi 4096 Sep 3 16:29 Downloads
0 -rw-r--r-- 1 pi pi 0 Sep 26 10:26 file_list.txt
```

Задание 2. Напишите AWK-программу для вывода четных строк списка автомобилей из файла `cars.txt`, в котором буквами в верхнем регистре будет обозначен только Модель, а производитель с первой заглавной буквы. Последующие поля каждой строки должны появляться в том виде, в котором они сохранены в файле `cars`.

```
pi@raspberrypi:~$ awk 'NR%2==0 {print NR, toupper(substr($1,1,1)) substr($1,2), toupper($2), $3, $4, $5}' cars.txt
2 Chevy MALIBU 1999 60 3000
4 Volvo S80 1998 102 9850
6 Chevy MALIBU 2000 50 3500
8 Honda ACCORD 2001 30 6000
10 Toyota RAV4 2002 180 750
12 Ford EXPLOR 2003 25 9500
pi@raspberrypi:~$
```

Задание 3. Напишите `sed`-команду, копирующую файл на стандартный вывод, удаляя при этом все строки, начинающиеся со слова «Yesterday» или «TODAY» в указанном написании. Проверить для случаев: а) наличие слов в начале строки, середине и в конце и в разных регистрах.
file:

```
pi@raspberrypi:~$ cat file.txt
First line.
Yesterday line.
line Yesterday.
TODAY line.
line TODAY.
line yesterday.
Today line.
yesterday line.
pi@raspberrypi:~$
```

Script:

```
pi@raspberrypi:~$ sed '/^TODAY\|^Yesterday/ d' file
First line.
line Yesterday.
line TODAY.
line yesterday.
Today line.
yesterday line.
pi@raspberrypi:~$
```

Задача 4. Напишите скрипт, который создает файл с заданным именем и устанавливает для него заданную маску разрешений на доступ. После завершения работы скрипта, маска разрешений на доступ должна быть установлена в значение, используемое по умолчанию. Использовать команду `umask`

raspberrypi:

```
pi@raspberrypi:~$ ./script.sh secondfile 000
access set successfully!
pi@raspberrypi:~$ ./script.sh thirdfile 777
access set successfully!
pi@raspberrypi:~$
```

```
#!/bin/bash

if (umask $2 | touch $1) then
echo "access set successfully!"
else
echo "$1: error changing ecess: $2"
exit 1
fi
umask 123
```

kali:

```
(krazhevskiy13@kali)-[~]
$ if [ ! -z "SSTV" ]; then echo 'Screen session'; fi
Screen session
```

```
(krazhevskiy13@kali)-[~]
$ ./script.sh file 000
access set successfully!
```

Задача 5. Написать скрипт для поиска файлов заданного размера в заданном каталоге и во всех его подкаталогах (имя каталога задаётся пользователем в качестве третьего аргумента командной строки). Диапазон (мин.– макс.) размеров файлов задаётся пользователем в качестве первого и второго аргумента командной строки. Проверить работу программы для каталога /usr и диапазона (мин.– макс.) 1000 1510.

Script:

```
#!/bin/bash
# 1-min_size, 2-max_size, 3-catalog

for i in $(find "$3" -size "+$1c" -size "-$2c" -type f)
do
    ls -lh "$i" | awk '{print $5, $9}'
done
```

Tests:

raspberrypi:

```
pi@raspberrypi:~ $ ./script5.sh 100 1000 /home/pi/Desktop/lab2
112 /home/pi/Desktop/lab2/task2.5/2.5.3/file
303 /home/pi/Desktop/lab2/task2.5/2.5.2/cars.txt
137 /home/pi/Desktop/lab2/task2.5/2.5.4/script.sh
121 /home/pi/Desktop/lab2/task2.1/packages1.catfile

pi@raspberrypi:~ $ ./script5.sh 1000 1510 /usr
1.4K /usr/lib/cryptsetup/scripts/decrypt_opensc
1.1K /usr/lib/cryptsetup/scripts/decrypt_derived
1.1K /usr/lib/cryptsetup/checks/blkid
1.1K /usr/lib/tmpfiles.d/journal-nocow.conf
1022 /usr/lib/mime/mailcap
1001 /usr/lib/kernel/install.d/85-initrd.install
1.3K /usr/lib/jvm/java-11-openjdk-i386/lib/server/Xusage.txt
1.3K /usr/lib/jvm/java-11-openjdk-i386/lib/client/Xusage.txt
```

kali:

```
(krazhevskiy13@kali)-[~]
$ ./script5.sh 1000 1510 /usr
1.2K /usr/share/X11/locale/zh_TW.UTF-8/XLC_LOCALE
1.1K /usr/share/X11/locale/zh_HK.big5hkscs/XLC_LOCALE
1.4K /usr/share/X11/locale/zh_CN.gb18030/XLC_LOCALE
1.1K /usr/share/X11/locale/zh_HK.big5/XLC_LOCALE
1007 /usr/share/X11/locale/th_TH.UTF-8/XLC_LOCALE
1.4K /usr/share/X11/bindings/intergraph
1.4K /usr/share/X11/bindings/doubleclick
```


Задача 6. Написать скрипт, который генерирует уникальное название для каждого созданного файла, и выводит на экран список из 10 случайных имен. Имя файла должно состоять из префикса „fileN“, где N принимает значения от 1 до 10, и суффикса, генерируемого случайно, например утилитой mcookie. Например, результат вывода может быть таким:

Имя файла1 = "file1.7a3e710af8c0f94798a666313fd502a1"

Имя файла2 = "file2.885fd9f36289275d0af397dc1cded003"

script:

```
#!/bin/bash

for (( c=1; c<=10; c++ ))
do
a=$(mcookie)
filename="file$c.$a"
touch "$filename"
echo "$filename"
done
```

result:

```
pi@raspberrypi:~$ ./script6.sh
file1.8246321480fc70ea23613e1f338a0155
file2.892e32bb59ad145416a971aba807f751
file3.a22acb83c167a5eec14fa2ac91ab298d
file4.82303aad57acbd52c3fa6cc6296b150
file5.0582359d89965dd36fa77624a602e6a3
file6.1b6ed8c471627c0ce97ec0397f28fdc3
file7.1b78886b33e4f8cf939c35eac773872f
file8.6390613be51bc904134fc5dc7764da0c
file9.64231563d468e901a5e1d49818ddbfe8
file10.17ebde5da4cc4fea3baceb51919e229b
pi@raspberrypi:~$ ls
Bookshelf
Desktop
Documents
Downloads
file10.17ebde5da4cc4fea3baceb51919e229b
file1.8246321480fc70ea23613e1f338a0155
file2.892e32bb59ad145416a971aba807f751
file3.a22acb83c167a5eec14fa2ac91ab298d
file4.82303aad57acbd52c3fa6cc6296b150
file5.0582359d89965dd36fa77624a602e6a3
file6.1b6ed8c471627c0ce97ec0397f28fdc3
file7.1b78886b33e4f8cf939c35eac773872f
file8.6390613be51bc904134fc5dc7764da0c
file9.64231563d468e901a5e1d49818ddbfe8
krazhevskiy
Music
Pictures
Public
script6.sh
task2_5_6Krazhevskiy13r
Templates
timelog2_5_6Krazhevskiy13r
Videos
pi@raspberrypi:~$
```