

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский химико-технологический университет имени Д.И.
Менделеева»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Выполнил студент группы КС-36: Золотухин А.А.

Ссылка на репозиторий: [https://github.com/
CorgiPuppy/
administration-of-information-systems-labs](https://github.com/CorgiPuppy/administration-of-information-systems-labs)

Принял: Митричев Иван Игоревич

Дата сдачи: 26.02.2025

Москва
2025

Оглавление

Описание и выполнение задачи	1
Задание 1	1
Задание 2	1
Задание 3	2
Задание 4	3
Задание 5	4
Задание 6	4
Задание 7	5
Задание 8	6
Задание 9	6
Задание 10	7
Задание 11	9
Задание 12	10
Задание 13	10
Задание 14	11
Задание 15	12
Задание 16	13
Задание 17	14
Задание 18	16
Задание 19	17
Задание 20	18
Задание 21	19

Описание и выполнение задачи

Задания 1-21 выполняются в терминале (bash) со скриншотами.

Задание 1

Создайте 3 текстовых файла с помощью команд **touch**, **>**, **nano**. В каждом файла должно быть по 2 строки текста.

```
1 #!/bin/bash
2 declare FOLDER=src/task1/target
3 declare -i LEN=3
4 declare -a FILES
5
6 for (( i=0; i<$LEN; i++ ))
7 do
8     FILES[i]=$FOLDER/file$i.txt
9 done
10
11 echo "touch ${FILES[0]} && > ${FILES[1]} && nano ${FILES[2]}"
12 touch ${FILES[0]} && > ${FILES[1]} && nano ${FILES[2]}
13
14 for (( i=0; i<$LEN; i++ ))
15 do
16     echo -e "Welcome"'\nThis is the file$i.'" > ${FILES[i]}
17
18     echo cat ${FILES[i]}
19     cat ${FILES[i]}
20 done
```

```
[corgi@corgi lab3]$ make run-task1
Running a task1..
touch src/task1/target/file0.txt && > src/task1/target/file1.txt && nano src/task1/target/file2.txt
cat src/task1/target/file0.txt
Welcome!
This is the file0.
cat src/task1/target/file1.txt
Welcome!
This is the file1.
cat src/task1/target/file2.txt
Welcome!
This is the file2.
[corgi@corgi lab3]$
```

Задание 2

Удалите один из файлов командой.

```
1 #!/bin/bash
2
3 declare FOLDER=src/task1/target
4
5 declare FILE=$FOLDER/file0.txt
6
7 if [ -f $FILE ]
8 then
9     echo ls $FOLDER
10    ls $FOLDER
```

```

11
12     echo "rm $FILE"
13     rm $FILE
14
15     echo ls $FOLDER
16     ls $FOLDER
17 else
18     echo "$FILE doesn't exist!"
19 fi

```

```

[corgi@corgi lab3]$ make run-task2
Running a task2..
ls src/task1/target
file0.txt  file1.txt  file2.txt
rm src/task1/target/file0.txt
ls src/task1/target
file1.txt  file2.txt
[corgi@corgi lab3]$ 

```

Задание 3

Передайте один из файлов во владение пользователю *root*. Поменяйте права на этот файл на 700.

```

1 #!/bin/bash
2
3 declare -f list_file
4 list_file () {
5     declare file=$1
6     echo "ls -l $file"
7     ls -l $file
8 }
9
10 declare FOLDER=src/task1/target
11 declare FILE=$FOLDER/file1.txt
12
13 if [ -f $FILE ]
14 then
15     list_file $FILE
16     echo "sudo chown root:root $FILE"
17     sudo chown root:root $FILE
18     list_file $FILE
19     echo "sudo chmod 700 $FILE"
20     sudo chmod 700 $FILE
21     list_file $FILE
22 else
23     echo "$FILE doesn't exist!"

```

```
[corgi@corgi lab3]$ make run-task3
Running a task3..
ls -l src/task1/target/file1.txt
-rw-r--r-- 1 corgi corgi 28 Feb 22 20:25 src/task1/target/file1.txt
sudo chown root:root src/task1/target/file1.txt
[sudo] password for corgi:
ls -l src/task1/target/file1.txt
-rw-r--r-- 1 root root 28 Feb 22 20:25 src/task1/target/file1.txt
sudo chmod 700 src/task1/target/file1.txt
ls -l src/task1/target/file1.txt
-rwx----- 1 root root 28 Feb 22 20:25 src/task1/target/file1.txt
[corgi@corgi lab3]$ █
```

Задание 4

Попытайтесь из-под вашего обычного пользователя удалить предыдущий файл, на вопрос отвечайте «Нет». Попытайтесь с опцией **-f**.

```
1 #!/bin/bash
2
3 declare FOLDER=src/task1/target
4 declare FILE=$FOLDER/file1.txt
5
6 if [ -f $FILE ]
7 then
8     echo "rm $FILE"
9     rm $FILE
10    echo "ls $FOLDER"
11    ls $FOLDER
12
13    if [ -f $FILE ]
14    then
15        echo "rm -f $FILE"
16        rm -f $FILE
17        echo "ls $FOLDER"
18        ls $FOLDER
19    fi
20 else
21     echo "$FILE doesn't exist!"
22 fi
```

```
[corgi@corgi lab3]$ make run-task4
Running a task4..
rm src/task1/target/file1.txt
rm: remove write-protected regular file 'src/task1/target/file1.txt'? no
ls src/task1/target
file1.txt  file2.txt
rm -f src/task1/target/file1.txt
ls src/task1/target
file2.txt
[corgi@corgi lab3]$ █
```

Задание 5

Переименуйте третий файл в *djan.conf*.

```
1 #!/bin/bash
2
3 declare FOLDER=src/task1/target
4 declare FILE=$FOLDER/file2.txt
5 declare TARGET_FILE=djan.conf
6
7 if [ -f $FILE ]
8 then
9     echo "ls $FOLDER"
10    ls $FOLDER
11
12    mv $FILE $FOLDER/$TARGET_FILE
13
14    echo "ls $FOLDER"
15    ls $FOLDER
16 else
17     echo "$FILE doesn't exist!"
18 fi
```

```
[corgi@corgi lab3]$ make run-task5
Running a task5..
ls src/task1/target
file2.txt
ls src/task1/target
djan.conf
[corgi@corgi lab3]$
```

Задание 6

Переместите *djan.conf* в */usr/share*. Добейтесь этого.

```
1 #!/bin/bash
2
3 declare FOLDER=src/task1/target
4 declare FILE_NAME=djan.conf
5 declare FILE=$FOLDER/$FILE_NAME
6
7 declare TARGET_FOLDER=/usr/share
8
9 if [ -f $FILE ]
10 then
11     echo "mv $FILE $TARGET_FOLDER"
12     mv $FILE $TARGET_FOLDER
13
```

```

14 echo "sudo mv $FILE $TARGET_FOLDER"
15 sudo mv $FILE $TARGET_FOLDER
16
17 echo "sudo find $TARGET_FOLDER -name "$FILE_NAME""
18 sudo find $TARGET_FOLDER -name "$FILE_NAME"
19 else
20 echo "$FILE doesn't exist!"
21 fi

```

```

[corgi@corgi lab3]$ make run-task6
Running a task6..
mv src/task1/target/djan.conf /usr/share
mv: cannot move 'src/task1/target/djan.conf' to '/usr/share/djan.conf': Permission denied
sudo mv src/task1/target/djan.conf /usr/share
[sudo] password for corgi:
sudo find /usr/share -name djan.conf
/usr/share/djan.conf
[corgi@corgi lab3]$ █

```

Задание 7

Сделайте *djan.conf* скрытым, изменив имя.

```

1 #!/bin/bash
2
3 declare FOLDER=/usr/share
4 declare FILE_NAME=djan.conf
5 declare FILE=$FOLDER/$FILE_NAME
6
7 if [ -f $FILE ]
8 then
9 echo "sudo find $FOLDER -name "$FILE_NAME""
10 sudo find $FOLDER -name "$FILE_NAME"
11
12 echo "sudo mv $FILE $FOLDER/.$FILE_NAME"
13 sudo mv $FILE $FOLDER/.$FILE_NAME
14
15 echo "sudo find $FOLDER -name ".$FILE_NAME""
16 sudo find $FOLDER -name ".$FILE_NAME"
17 else
18 echo "$FILE doesn't exist!"
19 fi

```

```

[corgi@corgi lab3]$ make run-task7
Running a task7..
sudo find /usr/share -name djan.conf
/usr/share/djan.conf
sudo mv /usr/share/djan.conf /usr/share/.djan.conf
sudo find /usr/share -name .djan.conf
/usr/share/.djan.conf
[corgi@corgi lab3]$ █

```

Задание 8

Перейдите в домашнюю папку. С помощью **ls -la** обнаружьте предыдущий файл, не переходя в директорию */usr/share*.

```
1 #!/bin/bash
2
3 declare CURRENT_DIR=$PWD
4 declare TARGET_FOLDER=/usr/share
5 declare FILE_NAME=.djan.conf
6 declare FILE=$TARGET_FOLDER/$FILE_NAME
7
8 if [ -f $FILE ]
9 then
10     echo "cd ~"
11     cd ~
12
13     echo "ls -la $TARGET_FOLDER/$FILE_NAME"
14     ls -la $TARGET_FOLDER/$FILE_NAME
15
16     cd $CURRENT_DIR
17 else
18     echo "$FILE doesn't exists!"
19 fi
```

```
[corgi@corgi lab3]$ make run-task8
Running a task8..
cd ~
ls -la /usr/share/.djan.conf
-rw-r--r-- 1 corgi corgi 28 Feb 22 22:01 /usr/share/.djan.conf
[corgi@corgi lab3]$ █
```

Задание 9

Изучите, что делает команда **cd -**.

```
1 #!/bin/bash
2
3 declare -i max_cd=4
4
5 for (( i=0; i<$max_cd; i++ ))
6 do
7     echo "cd -"
8     cd -
9 done
```



```
[corgi@corgi lab3]$ make run-task9
Running a task9..
cd -
/home/corgi
cd -
/home/corgi/git-repositories/administration-of-information-systems-labs/lab3
cd -
/home/corgi
cd -
/home/corgi/git-repositories/administration-of-information-systems-labs/lab3
[corgi@corgi lab3]$ █
```

Задание 10

Изучите, что делает нажатие **TAB**, стрелочки вверх и вниз на клавиатуре, а также нажатие **Ctrl-r** и ввод текста (введите **ls**). Возьмите на вооружение.

TAB используется для автодополнения команд или имён файлов или директорий. Например, я прописываю команду **ls** и после пробела нажимаю на **TAB**, но ничего не происходит, поскольку нет одного единственного автодополнения. После второго нажатия на **TAB** терминал покажет список всех доступных вариантов.

```
[corgi@corgi lab3]$ ls █

[corgi@corgi lab3]$ ls
assets/    Makefile  report/   src/
[corgi@corgi lab3]$ ls █
```

Если я начну вводить имя файла или директории и нажму **TAB**, терминал попытается дополнить его автоматически, поскольку существует только одно совпадение.

```
[corgi@corgi lab3]$ ls
assets/    Makefile  report/   src/
[corgi@corgi lab3]$ ls re█

[corgi@corgi lab3]$ ls
assets/    Makefile  report/   src/
[corgi@corgi lab3]$ ls report/█
```

Стрелочки вверх и вниз позволяют перемещаться по истории команд в командной строке. Например, нажатие стрелки вверх дважды покажет предпоследнюю введённую команду.

```
[corgi@corgi lab3]$ ls report
report.pdf  report.tex
[corgi@corgi lab3]$ cd report
[corgi@corgi report]$ cd ../
[corgi@corgi lab3]$
```

```
[corgi@corgi lab3]$ ls report
report.pdf  report.tex
[corgi@corgi lab3]$ cd report
[corgi@corgi report]$ cd ../
[corgi@corgi lab3]$ cd report
```

А стрелка вниз - следующую после команды, к которой пришли стрелкой вверх. Это удобно для быстрого повторного ввода команд без необходимости их повторного набора.

```
[corgi@corgi lab3]$ ls report
report.pdf  report.tex
[corgi@corgi lab3]$ cd report
[corgi@corgi report]$ cd ../
[corgi@corgi lab3]$ cd ../
```

Ctrl+R позволяет открыть режим поиска по истории команд. Например, после ввода текста, терминал будет показывать последние команды, которые содержат введённый текст, что позволяет быстро находить и повторно использовать ранее введенные команды.

```
[corgi@corgi lab3]$ ls report
report.pdf  report.tex
[corgi@corgi lab3]$ cd report
[corgi@corgi report]$ cd ../
(reverse-i-search)`ls': ls report
```

```
[corgi@corgi lab3]$ ls report
report.pdf  report.tex
[corgi@corgi lab3]$ cd report
[corgi@corgi report]$ cd ../
(reverse-i-search)`cd .': cd ../
```

Ввод текста, например, **ls** позволяет отображать список файлов и директорий в текущем каталоге. После ввода команды терминал выведет результат на экран.

```
[corgi@corgi lab3]$ ls
assets  Makefile  report  src
[corgi@corgi lab3]$
```

Задание 11

Изучите назначение файла */etc/sudoers*. Используя *visudo*, добавьте в файл *sudoers* строку, которая позволяет пользователю *student* выполнять команду **pacman** (работа с ПО).

```
##
## User privilege specification
##
root ALL=(ALL:ALL) ALL
student ALL=(ALL:ALL) /usr/bin/pacman

[corgi@corgi lab3]$ make run-task11
Running a task11..
sudo visudo
[sudo] password for corgi:
[corgi@corgi lab3]$
```

Задание 12

Добавьте пользователя *student*, группа *student* в операционную систему одной командой (читать **man useradd** или **adduser**). Зайдите под его учётной записью. Установите программу **gnuplot**.

```
1 #!/bin/bash
2
3 declare USER=student
4 declare BASH=/bin/bash
5
6 echo "sudo useradd --create-home --shell $BASH --user-group $USER"
7 sudo useradd --create-home --shell $BASH --user-group $USER
8
9 echo "sudo passwd $USER"
10 sudo passwd $USER
11
12 echo "sudo --user $USER"
13 sudo --user $USER bash -c '
14     echo "sudo pacman -S gnuplot"
15     sudo pacman -S gnuplot
16 ,
```

```
[corgi@corgi lab3]$ make run-task12
Running a task12..
sudo useradd --create-home --shell /bin/bash --user-group student
[sudo] password for corgi:
sudo passwd student
New password:
Retype new password:
passwd: password updated successfully
sudo --user student
sudo pacman -S gnuplot
[sudo] password for student:
resolving dependencies...
looking for conflicting packages...

Packages (1) gnuplot-6.0.2-1

Total Installed Size: 3.56 MiB

:: Proceed with installation? [Y/n] Y
(1/1) checking keys in keyring [#####] 100%
(1/1) checking package integrity [#####] 100%
(1/1) loading package files [#####] 100%
(1/1) checking for file conflicts [#####] 100%
(1/1) checking available disk space [#####] 100%
:: Processing package changes...
(1/1) installing gnuplot [#####] 100%
:: Running post-transaction hooks...
(1/3) Arming ConditionNeedsUpdate...
(2/3) Updating TeXLive filename database...
(3/3) Updating ConTeXt database...
[corgi@corgi lab3]$
```

Задание 13

Под учётной записью *student* создайте командой папку *ПАПКА1* в директории */tmp*.

```
1 #!/bin/bash
2
3 declare USER=student
4 declare FOLDER_NAME=ПАПКА1
5 declare FOLDER=/tmp
6 declare TARGET_FOLDER=$FOLDER/$FOLDER_NAME
7
8 echo "sudo --user $USER"
9 sudo --user $USER bash << EOF
```

```

10 echo "mkdir "$TARGET_FOLDER"
11 mkdir $TARGET_FOLDER
12
13 echo "ls -l $FOLDER"
14 ls -l $FOLDER
15 EOF

```

```

[corgi@corgi lab3]$ make run-task13
Running a task13..
sudo --user student
[sudo] password for corgi:
mkdir /tmp/PAPKA1
ls -l /tmp
total 0
drwxr-xr-x 2 student student 40 Feb 23 17:44 PAPKA1
drwx----- 2 corgi corgi 60 Feb 23 12:27 ssh-XXXXXX5ImSzt
drwx----- 3 root root 60 Feb 23 12:27 systemd-private-4f3a20d9f8334df69bd56a02f5f6ad01-dbus-broker
.service-mqYMyZ
drwx----- 3 root root 60 Feb 23 12:27 systemd-private-4f3a20d9f8334df69bd56a02f5f6ad01-httpd.servi
ce-hAEa2s
drwx----- 3 root root 60 Feb 23 12:27 systemd-private-4f3a20d9f8334df69bd56a02f5f6ad01-mariadb.ser
vice-Rms3pu
drwx----- 3 root root 60 Feb 23 12:27 systemd-private-4f3a20d9f8334df69bd56a02f5f6ad01-polkit.serv
ice-NUsLvH
drwx----- 3 root root 60 Feb 23 12:27 systemd-private-4f3a20d9f8334df69bd56a02f5f6ad01-systemd-log
ind.service-PeAEYO
drwx----- 3 root root 60 Feb 23 12:27 systemd-private-4f3a20d9f8334df69bd56a02f5f6ad01-upower.serv
ice-hecv2l
drwx----- 2 corgi corgi 60 Feb 23 15:42 tmux-1000
[corgi@corgi lab3]$

```

Задание 14

Создайте под учётной записью *root* одной командой вложенную папку */tmp/PAPKA2/PAPKA3*. В ней создайте одной командой файл, содержащий десять последних строк файла */etc/hosts*. Выйдите из-под рута.

```

1 #!/bin/bash
2
3 declare USER=root
4 declare FOLDER_NAME=PAPKA2/PAPKA3
5 declare FOLDER=/tmp
6 declare TARGET_FOLDER=$FOLDER/$FOLDER_NAME
7 declare FILE_NAME=file.txt
8 declare FROM_FOLDER=/etc/hosts
9 declare TARGET_FILE=$TARGET_FOLDER/$FILE_NAME
10 declare -i LINES=10
11
12 echo "sudo --user $USER"
13 sudo --user $USER bash << EOF
14   echo "mkdir -p $TARGET_FOLDER"
15   mkdir -p $TARGET_FOLDER
16
17   echo "tail --lines=$LINES $FROM_FOLDER > $TARGET_FILE"
18   tail --lines=$LINES $FROM_FOLDER > $TARGET_FILE
19
20   echo "cat $TARGET_FILE"
21   cat $TARGET_FILE
22
23   echo "exit"
24   exit
25 EOF

```

```

[corgi@corgi lab3]$ make run-task14
Running a task14..
sudo --user root
mkdir -p /tmp/PAPKA2/PAPKA3
tail --lines=10 /etc/hosts > /tmp/PAPKA2/PAPKA3/file.txt
cat /tmp/PAPKA2/PAPKA3/file.txt
# Static table lookup for hostnames.
# See hosts(5) for details.

127.0.0.1      localhost
::1           localhost
127.0.0.1      corgi.localdomain    corgi
exit
[corgi@corgi lab3]$

```

Задание 15

Под учётной записью *student* создайте в *PAPKA1* файл *4!!4.txt*, содержащий последние десять строк файла */.bashrc*. Проверьте, что файл с именем *4!!4.txt* корректно создан - другие имена не принимаются. Добавьте в файл *4!!4.txt*, не затирая прошлые 10 строк, ещё первые 10 строк файла */.bashrc*.

```

1 #!/bin/bash
2
3 declare USER=student
4 declare FILE_NAME="4!!4.txt"
5 declare DIR=/tmp
6 declare FOLDER_NAME=PAPKA1
7 declare TARGET_FOLDER=$DIR/$FOLDER_NAME
8 declare TARGET_FILE=$TARGET_FOLDER/$FILE_NAME
9 declare BASHRC=~/.bashrc
10 declare -i LINES=10
11
12 echo "sudo --user $USER"
13 sudo --user $USER bash << EOF
14     echo "tail --lines=$LINES $BASHRC > $TARGET_FILE"
15     tail --lines=$LINES $BASHRC > $TARGET_FILE
16
17     echo "cat $TARGET_FILE"
18     cat $TARGET_FILE
19
20     echo "find $TARGET_FOLDER -name \"$FILE_NAME\""
21     find $TARGET_FOLDER -name "$FILE_NAME"
22
23     echo "head --lines=$LINES $BASHRC >> $TARGET_FILE"
24     head --lines=$LINES $BASHRC >> $TARGET_FILE
25
26     echo "cat $TARGET_FILE"
27     cat $TARGET_FILE
28 EOF

```

```

[corgi@corgi lab3]$ make run-task15
Running a task15..
sudo --user student
[sudo] password for corgi:
tail --lines=10 ~/.bashrc > /tmp/PAPKA1/4!!4.txt
cat /tmp/PAPKA1/4!!4.txt
#
# ~/.bashrc
#

# If not running interactively, don't do anything
[[ $- != *i* ]] && return

alias ls='ls --color=auto'
alias grep='grep --color=auto'
PS1='[\u@\h \W]\$ '
find /tmp/PAPKA1 -name 4!!4.txt
/tmp/PAPKA1/4!!4.txt
tail --lines=10 ~/.bashrc >> /tmp/PAPKA1/4!!4.txt
cat /tmp/PAPKA1/4!!4.txt
#
# ~/.bashrc
#

# If not running interactively, don't do anything
[[ $- != *i* ]] && return

alias ls='ls --color=auto'
alias grep='grep --color=auto'
PS1='[\u@\h \W]\$ '
#
# ~/.bashrc
#

# If not running interactively, don't do anything
[[ $- != *i* ]] && return

alias ls='ls --color=auto'
alias grep='grep --color=auto'
PS1='[\u@\h \W]\$ '
[corgi@corgi lab3]$ █

```

Задание 16

Под учётной записью *student* попробуйте удалить обе папки командой **rm** без использования **опций** команды. Если не получилось, попробуйте удалить их по отдельности. Удалось? Если нет, то укажите причины.

```

1 #!/bin/bash
2
3 declare USER=student
4
5 echo "sudo --user $USER"
6 sudo --user $USER bash << EOF

```

```
[corgi@corgi lab3]$ make run-task16
Running a task16..
sudo --user student
[sudo] password for corgi:
find 2>/dev/null /tmp -name PAPKA1
/tmp/PAPKA1
find 2>/dev/null /tmp -name PAPKA2
/tmp/PAPKA2
rm /tmp/PAPKA1 /tmp/PAPKA2
rm: cannot remove '/tmp/PAPKA2': Is a directory
rm: cannot remove '/tmp/PAPKA2': Is a directory
rm /tmp/PAPKA1
rm: cannot remove '/tmp/PAPKA1': Is a directory
rm /tmp/PAPKA2
rm: cannot remove '/tmp/PAPKA2': Is a directory
make: *** [Makefile:17: run-task16] Error 1
[corgi@corgi lab3]$
```

Задание 17

Изучите файл `./bashrc`. Увеличьте максимальную длину файла с историй и максимальное количество строк, выводимых командой **history** в 10 раз. Сделайте это на постоянной основе (не только, пока запущен данный терминал) - используйте **export**.

```
1 #!/bin/bash
2
3 declare BASHRC_FILE=~/.bashrc
4 declare HISTSIZE_VALUE=$(grep -E "^HISTSIZE=" $BASHRC_FILE | cut -d '=' -f 2)
5 declare HISTFILESIZE_VALUE=$(grep -E "^HISTFILESIZE=" $BASHRC_FILE | cut -d '=' -f 2)
6
7 if [[ -n $HISTSIZE_VALUE ]]
8 then
9     declare HISTSIZE=$(expr 10 \* $HISTSIZE_VALUE)
10 else
11     declare HISTSIZE=1000
12 fi
13
14 if [[ -n $HISTFILESIZE_VALUE ]]
15 then
16     declare HISTFILESIZE=$(expr 10 \* $HISTFILESIZE_VALUE)
17 else
```



```

18 declare HISTFILESIZE=1000
19 fi
20
21 if grep -q "^HISTSIZE=" $BASHRC_FILE
22 then
23     echo "sed -i "s/^HISTSIZE=.* /HISTSIZE=$HISTSIZE/" $BASHRC_FILE"
24     sed -i "s/^HISTSIZE=.* /HISTSIZE=$HISTSIZE/" $BASHRC_FILE
25 else
26     echo "echo "HISTSIZE=$HISTSIZE" >> $BASHRC_FILE"
27     echo "HISTSIZE=$HISTSIZE" >> $BASHRC_FILE
28 fi
29
30 if grep -q "^HISTFILESIZE=" $BASHRC_FILE
31 then
32     echo "sed -i "s/^HISTFILESIZE=.* /HISTFILESIZE=$HISTFILESIZE/" $BASHRC_FILE"
33     sed -i "s/^HISTFILESIZE=.* /HISTFILESIZE=$HISTFILESIZE/" $BASHRC_FILE
34 else
35     echo "echo "HISTFILESIZE=$HISTFILESIZE" >> $BASHRC_FILE"
36     echo "HISTFILESIZE=$HISTFILESIZE" >> $BASHRC_FILE
37 fi
38
39 if grep -q "^export HISTSIZE=" $BASHRC_FILE
40 then
41     echo "sed -i "s/^export HISTSIZE=.* /export HISTSIZE=$HISTSIZE/" $BASHRC_FILE"
42     sed -i "s/^export HISTSIZE=.* /export HISTSIZE=$HISTSIZE/" $BASHRC_FILE
43 else
44     echo "echo "export HISTSIZE=$HISTSIZE" >> $BASHRC_FILE"
45     echo "export HISTSIZE=$HISTSIZE" >> $BASHRC_FILE
46 fi
47
48 if grep -q "^export HISTFILESIZE=" $BASHRC_FILE
49 then
50     echo "sed -i "s/^export HISTFILESIZE=.* /export HISTFILESIZE=$HISTFILESIZE/"
51     $BASHRC_FILE"
52     sed -i "s/^export HISTFILESIZE=.* /export HISTFILESIZE=$HISTFILESIZE/" $BASHRC_FILE
53 else
54     echo "echo "export HISTFILESIZE=$HISTFILESIZE" >> $BASHRC_FILE"
55     echo "export HISTFILESIZE=$HISTFILESIZE" >> $BASHRC_FILE
56 fi
57
58 echo "source $BASHRC_FILE"
59 source $BASHRC_FILE
60
61 echo "tail -4 $BASHRC_FILE"
62 tail -4 $BASHRC_FILE

```

```

[corgi@corgi lab3]$ make run-task17
Running a task17..
echo HISTSIZE=1000 >> /home/corgi/.bashrc
echo HISTFILESIZE=1000 >> /home/corgi/.bashrc
echo export HISTSIZE=1000 >> /home/corgi/.bashrc
echo export HISTFILESIZE=1000 >> /home/corgi/.bashrc
source /home/corgi/.bashrc
tail -4 /home/corgi/.bashrc
HISTSIZE=1000
HISTFILESIZE=1000
export HISTSIZE=1000
export HISTFILESIZE=1000
[corgi@corgi lab3]$ make run-task17
Running a task17..
sed -i s/^HISTSIZE=.* /HISTSIZE=10000/ /home/corgi/.bashrc
sed -i s/^HISTFILESIZE=.* /HISTFILESIZE=10000/ /home/corgi/.bashrc
sed -i s/^export HISTSIZE=.* /export HISTSIZE=10000/ /home/corgi/.bashrc
sed -i s/^export HISTFILESIZE=.* /export HISTFILESIZE=10000/ /home/corgi/.bashrc
source /home/corgi/.bashrc
tail -4 /home/corgi/.bashrc
HISTSIZE=10000
HISTFILESIZE=10000
export HISTSIZE=10000
export HISTFILESIZE=10000
[corgi@corgi lab3]$ █

```

Задание 18

Выведите содержимое домашней папки в длинном формате вместе со скрытыми файлами сортированной по времени, первыми идёт самые старые файлы и папки. Задание выполнить одной командой **ls** и её опциями, не использовать команду **sort**.

```

1 #!/bin/bash
2
3 echo "ls -l --all -t -r $HOME"
4 ls -la --all -t -r $HOME

```

```

[corgi@corgi lab3]$ make run-task18
Running a task18..
ls -l --all -t -r /home/corgi
total 200
-rw-r--r--  1 corgi corgi    57 Sep 24 22:46 .bash_profile
-rw-r--r--  1 corgi corgi   21 Sep 24 22:46 .bash_logout
drwxr-xr-x  2 corgi corgi 4096 Feb 15 16:11 Videos
drwxr-xr-x  2 corgi corgi 4096 Feb 15 16:11 Templates
drwxr-xr-x  2 corgi corgi 4096 Feb 15 16:11 Public
drwxr-xr-x  2 corgi corgi 4096 Feb 15 16:11 Music
drwxr-xr-x  2 corgi corgi 4096 Feb 15 16:11 Documents
drwxr-xr-x  2 corgi corgi 4096 Feb 15 16:11 Desktop
-rw-r--r--  1 corgi corgi   23 Feb 15 16:14 .dmrc
drwxr-xr-x  3 corgi corgi 4096 Feb 15 16:14 .local
-rw-----  1 corgi corgi    0 Feb 15 16:14 .ICEauthority
drwx-----  3 corgi corgi 4096 Feb 15 16:14 .gnupg
drwx-----  3 corgi corgi 4096 Feb 15 16:14 .pki
drwxr-xr-x 19 corgi corgi 4096 Feb 15 20:47 OpenBLAS
drwxr-xr-x 10 corgi corgi 4096 Feb 15 20:48 .cache
drwxr-xr-x  4 corgi corgi 4096 Feb 15 21:06 opt
drwxr-xr-x  2 corgi corgi 4096 Feb 15 21:10 Downloads
drwxr-xr-x 11 corgi corgi 4096 Feb 15 21:14 hpl
-rw-r--r--  1 root  root   375 Feb 15 22:32 .vimrc
-rw-r--r--  1 corgi corgi   132 Feb 15 22:41 .gitconfig
drwx-----  2 corgi corgi 4096 Feb 16 17:22 .ssh
drwx-----  9 corgi corgi 4096 Feb 19 08:49 .config
drwx-----  3 corgi corgi 4096 Feb 20 13:27 .emacs.d
drwx-----T 3 corgi corgi 4096 Feb 20 16:19 .texlive
drwxr-xr-x  3 corgi corgi 4096 Feb 22 16:54 git-repositories
-rw-----  1 corgi corgi   124 Feb 22 22:50 .git-credentials
drwxr-xr-x  4 root  root  4096 Feb 23 16:58 ..
-rw-----  1 corgi corgi 11691 Feb 24 12:24 .bash_history
drwxr-xr-x  3 corgi corgi 4096 Feb 24 12:25 Pictures
-rw-r--r--  1 corgi corgi   186 Feb 24 12:31 .bashrc
-rw-----  1 corgi corgi  2778 Feb 24 12:34 .xsession-errors.old
drwxr-xr-x  3 corgi corgi 4096 Feb 24 12:35 git-repos
-rw-----  1 corgi corgi    50 Feb 24 12:38 .Xauthority
-rw-r----- 1 corgi corgi     4 Feb 24 12:38 .vboxclient-clipboard-tty7-control.pid
-rw-r----- 1 corgi corgi     4 Feb 24 12:38 .vboxclient-clipboard-tty7-service.pid
-rw-r----- 1 corgi corgi     4 Feb 24 12:38 .vboxclient-draganddrop-tty7-service.pid
-rw-r----- 1 corgi corgi     4 Feb 24 12:38 .vboxclient-draganddrop-tty7-control.pid
-rw-r----- 1 corgi corgi     4 Feb 24 12:38 .vboxclient-hostversion-tty7-control.pid
-rw-r----- 1 corgi corgi     4 Feb 24 12:38 .vboxclient-seamless-tty7-control.pid
-rw-r----- 1 corgi corgi     4 Feb 24 12:38 .vboxclient-seamless-tty7-service.pid
-rw-----  1 corgi corgi  3683 Feb 24 14:53 .xsession-errors
-rw-----  1 corgi corgi   118 Feb 24 15:08 .lessht
-rw-----  1 corgi corgi 24335 Feb 24 15:09 .viminfo
drwx----- 23 corgi corgi 4096 Feb 24 15:09 .
[corgi@corgi lab3]$ █

```

Задание 19

Создайте в домашней директории папку *bin*, в ней файл *bubble* с правами 744. В файл добавьте:

```
#!/bin/bash
```

```
ls
```

Запустите её из папки как *./bubble* (добейтесь корректного запуска).

Теперь измените переменную *PATH* системы - внесите запись в */.bashrc* - так, чтобы программу *bubble* можно было запустить просто словом *bubble* (не затрите старое значение *\$PATH*, а добавь к ней нужную папку). Молодцы, теперь Вы знаете, зачем нужен *\$PATH*.

```
1 #!/bin/bash
2
3 declare FOLDER_NAME=$HOME/bin
4 declare FILE_NAME=bubble
5 declare FILE=$FOLDER_NAME/$FILE_NAME
6 declare BASH_SCRIPT="#!/bin/bash\nls"
7 declare PATH_STRING="PATH=$PATH:$FILE"
8
9 echo "mkdir $FOLDER_NAME"
10 mkdir $FOLDER_NAME
11
12 echo "touch $FILE"
13 touch $FILE
14
15 echo "sudo chmod 744 $FILE"
16 sudo chmod 744 $FILE
17
18 echo "echo -e $BASH_SCRIPT > $FILE"
19 echo -e $BASH_SCRIPT > $FILE
20
21 echo "cd $FOLDER_NAME"
22 cd $FOLDER_NAME
23 echo "./$FILE_NAME"
24 ./$FILE_NAME
25
26 echo "bubble"
27 bubble
28
29 echo "export PATH=$PATH:$FOLDER_NAME"
30 export PATH=$PATH:$FOLDER_NAME
31
32 echo "cd $FOLDER_NAME"
33 cd $FOLDER_NAME
34 echo "bubble"
35 bubble
```

```
[corgi@corgi lab3]$ make run-task19
Running a task19..
mkdir /home/corgi/bin
mkdir: cannot create directory '/home/corgi/bin': File exists
touch /home/corgi/bin/bubble
sudo chmod 744 /home/corgi/bin/bubble
echo -e #!/bin/bash\nls > /home/corgi/bin/bubble
cd /home/corgi/bin
./bubble
bubble
bubble
./src/task19/main.sh: line 27: bubble: command not found
export PATH=/usr/local/bin:/usr/bin:/bin:/usr/sbin:/usr/bin:/usr/site_perl:/usr/bin/vendor_perl:/usr/bin/core_perl:/home/corgi/bin
cd /home/corgi/bin
bubble
bubble
bubble
[corgi@corgi lab3]$ █
```

Задание 20

Заархивируйте три файла в *tarball* со сжатием *bzip2*.

```

1 #!/bin/bash
2
3 declare CURRENT_FOLDER=src/task20/target
4 declare -i LEN=3
5 declare -a FILES
6 declare TARGET_FOLDER_NAME=tarball
7 declare TARGET_FOLDER=$CURRENT_FOLDER/$TARGET_FOLDER_NAME
8
9 for (( i=0; i<$LEN; i++ ))
10 do
11     FILES[i]=$CURRENT_FOLDER/file$i
12     echo "touch ${FILES[i]}"
13     touch ${FILES[i]}
14 done
15
16 echo "tar --totals -cvjf $TARGET_FOLDER ${FILES[0]} ${FILES[1]} ${FILES[2]}"
17 tar --totals -cvjf $TARGET_FOLDER ${FILES[0]} ${FILES[1]} ${FILES[2]}
18
19 echo "ls -l $TARGET_FOLDER"
20 ls -l $CURRENT_FOLDER

```

```

[corgi@corgi lab3]$ make run-task20
Running a task20..
touch src/task20/target/file0
touch src/task20/target/file1
touch src/task20/target/file2
tar --totals -cvjf src/task20/target/tarball src/task20/target/file0 src/task20/target/file1 src/task20/target/file2
src/task20/target/file0
src/task20/target/file1
src/task20/target/file2
Total bytes written: 10240 (10KiB, 2.2MiB/s)
ls -l src/task20/target/tarball
total 4
-rw-r--r-- 1 corgi corgi  0 Feb 25 17:16 file0
-rw-r--r-- 1 corgi corgi  0 Feb 25 17:16 file1
-rw-r--r-- 1 corgi corgi  0 Feb 25 17:16 file2
-rw-r--r-- 1 corgi corgi 168 Feb 25 17:16 tarball
[corgi@corgi lab3]$

```

Задание 21

Теперь создайте папку *PAPKA3* и разархивируйте в неё ваш архив, не перемещая (!) архив внутрь папки.

```

1 #!/bin/bash
2
3 declare ARCHIVE_FOLDER=src/task20/target
4 declare ARCHIVE_NAME=tarball
5 declare ARCHIVE=$ARCHIVE_FOLDER/$ARCHIVE_NAME
6 declare CURRENT_FOLDER=src/task21/target
7 declare TARGET_FOLDER_NAME=PAPKA3
8 declare TARGET_FOLDER=$CURRENT_FOLDER/$TARGET_FOLDER_NAME
9
10 echo "mkdir -p $TARGET_FOLDER"
11 mkdir -p $TARGET_FOLDER
12
13 echo "tar --strip-components=3 -xjvf $ARCHIVE -C $TARGET_FOLDER"
14 tar --strip-components=3 -xjvf "$ARCHIVE" -C "$TARGET_FOLDER"
15
16 echo "ls -l $TARGET_FOLDER"
17 ls -l $TARGET_FOLDER

```

```
[corgi@corgi lab3]$ make run-task21
Running a task21..
mkdir -p src/task21/target/PAPKA3
tar --strip-components=3 -xjvf src/task20/target/tarball -C src/task21/target/PAPKA3
src/task20/target/file0
src/task20/target/file1
src/task20/target/file2
ls -l src/task21/target/PAPKA3
total 0
-rw-r--r-- 1 corgi corgi 0 Feb 25 16:48 file0
-rw-r--r-- 1 corgi corgi 0 Feb 25 16:48 file1
-rw-r--r-- 1 corgi corgi 0 Feb 25 16:48 file2
[corgi@corgi lab3]$ █
```