

# **Отчет о прохождении 3 этапа внешних курсов**

**Продвинутые темы**

Калашникова Ольга Сергеевна, НПИбд-01-23

# **Содержание**

<b>1 Цель работы</b>	<b>6</b>
<b>2 Задание</b>	<b>7</b>
<b>3 Теоретическое введение</b>	<b>8</b>
<b>4 Выполнение лабораторной работы</b>	<b>9</b>
<b>5 Сертификат</b>	<b>36</b>
<b>6 Выводы</b>	<b>37</b>
<b>Список литературы</b>	<b>38</b>

# Список иллюстраций

4.1	Задание 1 . . . . .	9
4.2	Задание 2 . . . . .	10
4.3	Задание 3 . . . . .	10
4.4	Задание 4 . . . . .	11
4.5	Задание 5 . . . . .	12
4.6	Задание 6 . . . . .	12
4.7	Задание 7 . . . . .	13
4.8	Задание 8 . . . . .	13
4.9	Задание 9 . . . . .	14
4.10	Задание 10 . . . . .	14
4.11	Задание 11 . . . . .	15
4.12	Задание 12 . . . . .	15
4.13	Задание 13 . . . . .	16
4.14	Задание 14 . . . . .	17
4.15	Задание 14 . . . . .	18
4.16	Задание 15 . . . . .	19
4.17	Задание 16 . . . . .	20
4.18	Задание 16_2 . . . . .	20
4.19	Задание 17 . . . . .	20
4.20	Задание 18 . . . . .	21
4.21	Задание 18 . . . . .	22
4.22	Задание 19 . . . . .	23
4.23	Задание 19 . . . . .	24
4.24	Задание 20 . . . . .	24
4.25	Задание 21 . . . . .	25
4.26	Задание 22 . . . . .	26
4.27	Задание 23 . . . . .	26
4.28	Задание 24 . . . . .	27
4.29	Задание 24 . . . . .	27
4.30	Задание 25 . . . . .	28
4.31	Задание 26 . . . . .	28
4.32	Задание 27 . . . . .	29
4.33	Задание 28 . . . . .	29
4.34	Задание 29 . . . . .	30
4.35	Задание 30 . . . . .	31
4.36	Задание 31 . . . . .	32
4.37	Задание 32 . . . . .	33

4.38 Задание 33 . . . . .	34
4.39 Задание 34 . . . . .	35
4.40 Задание 35 . . . . .	35
5.1 Сертификат . . . . .	36

# **Список таблиц**

# **1 Цель работы**

Ознакомиться с функционалом операционной системы Linux.

## **2 Задание**

Просмотреть видео и на основе полученной информации пройти тестовые задания.

### **3 Теоретическое введение**

Линукс - в части случаев GNU/Linux — семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения. Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов — в форме, готовой для установки и удобной для сопровождения и обновлений, — и имеющих свой набор системных и прикладных компонентов, как свободных, так и проприетарных.

## 4 Выполнение лабораторной работы

3 Этап: (рис. fig. 4.1, fig. 4.2, fig. 4.3, fig. 4.4, fig. 4.5, fig. 4.6, fig. 4.7, fig. 4.8, fig. 4.9, fig. 4.10, fig. 4.11, fig. 4.12, fig. 4.13, fig. 4.14, fig. 4.15, fig. 4.16, fig. 4.17, fig. 4.18, fig. 4.19, fig. 4.20, fig. 4.21, fig. 4.22, fig. 4.23, fig. 4.24, fig. 4.25, fig. 4.26, fig. 4.27, fig. 4.28, fig. 4.29, fig. 4.30, fig. 4.31, fig. 4.32, fig. 4.33, fig. 4.34, fig. 4.35, fig. 4.36, fig. 4.37, fig. 4.38, fig. 4.39, fig. 4.40, fig. 5.1).

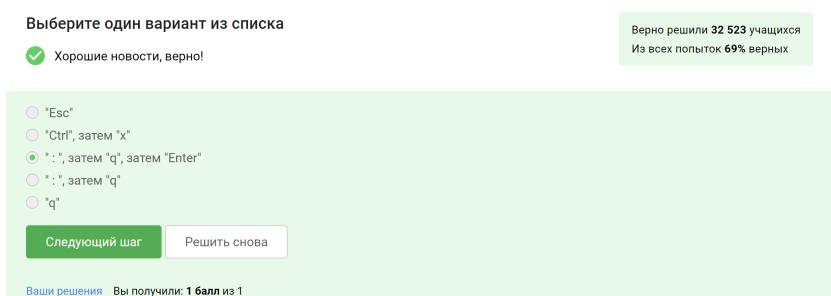


Рис. 4.1: Задание 1

Стоит упомянуть, что у редактора vim есть тьюториал, который позволяет разобраться с командами, необходимыми для стандартной работы. За выход из редактора отвечают следующие команды:

- ZQ - выйти без сохранения
- :q! - выйти без сохранения
- ZZ - записать файл и выйти (если файл не изменяли, то записываться он не будет)
- :wq - записать файл и выйти
- :x - записать файл и выйти

- :w - записать файл
- :sav filename - “сохранить как”
- :w filename - “сохранить как”
- :w! - записать файл

Как мы видим, вариантов много, при этом каждый сможет найти тот, который подойдёт под конкретную ситуацию.

Выберите все подходящие ответы из списка

Прекрасный ответ.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

В этой строке 5 "больших слов" (WORD)  
 В этой строке 9 "больших слов" (WORD)  
 В этой строке 12 "слов" (word)  
 В этой строке 5 "слов" (word)  
 Чтобы попасть в конец строки, нужно одинаковое число нажатий, что на W, что на w  
 Нажимая только на w, нельзя переместить курсор на "."

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 1 балл из 1

Верно решили 25 385 учащихся  
Из всех попыток 20% верных

Рис. 4.2: Задание 2

Strange\_ TEXT is\_here. 2=2 YES!

Точка считается “маленьким словом”, так что всего их 9: Strange\_, is\_here, ., 2, =, 2, ! и два лишних пробела.

И если посчитать нажатия на W и на W, то действительно после 10 штук попадем в одно место. 10 нажатий на W, это то же самое, что и 10 нажатий на w,

Выберите все подходящие ответы из списка

Прекрасный ответ.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

d2dwywPp  
 ddthree four four four five<Esc>  
 x2wwywPp  
 d2wwywpp  
 xxxxxxxxwywPp  
 d2wwifour four <Esc>

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 1 балл из 1

Верно решили 23 655 учащихся  
Из всех попыток 16% верных

Рис. 4.3: Задание 3

```
d2wwifour four <<Esc>>  
d2wwywPp  
d2w$$bifour four <<Esc>>
```

- \$ — в конец текущей строки;
- w — на слово вправо;
- b — на слово влево;
- i — начать ввод перед курсором;
- p — вставка содержимого неименованного буфера под курсором;
- P — вставка содержимого неименованного буфера перед курсором;
- yy (также Y) — копирование текущей строки в неименованный буфер;
- uu — копирование числа строк начиная с текущей в неименованный буфер;

Напишите текст

✓ Прекрасный ответ.

Верно решил 24 631 учащийся  
Из всех попыток 57% верных

.%s/Windows/Linux

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: 2 балла из 2

Рис. 4.4: Задание 4

Поиск и замена в редакторе работают по следующей схеме:

:{пределы}s/{что заменяем}/{на что заменяем}/{опции}

Для замены во всем файле можно использовать символ %.

Выберите все подходящие ответы из списка

Так точно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

Чтобы выйти из режима выделения, нужно ввести :q  
 Режим выделения открывается из нормального режима по нажатию "v"  
 Когда вы находитесь в режиме выделения, внизу редактора горит надпись – VISUAL – (или – ВИЗУАЛЬНЫЙ РЕЖИМ –)  
 Выйти из режима выделения можно, нажав клавишу Esc два раза  
 В режиме выделения можно использовать команды d (удалить) и u (скопировать)  
 В режиме выделения можно использовать команды перемещения (например, W, e, \$, и др.)

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 2 балла из 2

Рис. 4.5: Задание 5

Команда \$ – в конец текущей строки, W - до пробела вправо - то есть, перемещение.

Нажать Esc достаточно один раз, но да ладно.

Надпись visual - горит.

d – используется совместно с командами перемещения. Удаляет символы с текущего положения курсора до положения после ввода команды перемещения.

уу (также Y) – копирование текущей строки в буфер;

Выберите один вариант из списка

Отличное решение!

Верно решили 30 266 учащихся  
Из всех попыток 65% верных

Из наборов В и С  
 Только из набора С  
 Из наборов А и С  
 Никакие команды появляться не будут  
 Только из набора А

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 1 балл из 1

Рис. 4.6: Задание 6

Только из набора С потому что у каждой оболочки свой буфер, который при выходе из нее будет записываться в файл истории.

Выберите один вариант из списка

Хорошие новости, верно!

- /home/bi/Desktop/file1.txt
- /home/bi/file1.txt
- Никак (файла file1.txt не будет существовать после завершения работы скрипта)
- /home/bi/Documents/file1.txt

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: **1 балл** из 1

Рис. 4.7: Задание 7

/home/bi/file1.txt - потому что именно в этой директории мы создаем новый файл, а уже после его создания мы переходим в другую папку.

Выберите все подходящие ответы из списка

Так точно!

Верно решили 27 188 учащихся

Из всех попыток 25% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- \_variable
- var-i-able
- variab\$le
- VARiable
- \_\_variable
- vari/able
- variable123

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: **1 балл** из 1

Рис. 4.8: Задание 8

Имя не может начинаться с цифры, содержать специальные символы или пробелы.

Напишите программу. Тестируется через stdin → stdout

Верно.

Теперь вам доступен [Форум решений](#), где вы можете сравнить свое решение с другими или спросить совета.

```

1 #!/bin/bash
2 var1=$1
3 var2=$2
4 echo "Arguments are: \$1=$1 \$2=$2"
5
6
7
8
9

```

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 3 балла из 3

Верно решили 25 053 учащихся  
Из всех попыток 41% верных

Рис. 4.9: Задание 9

\$ echo опции строка Эта команда печатает строки, которые передаются в качестве аргументов в стандартный вывод и обычно используется в сценариях оболочки для отображения сообщения или вывода результатов других команд.

**var1=\$1** - обозначение переменных

**var2=\$2**

**echo "Arguments are: \\$1=\$var1 \\$2=\$var2"** - строка печати.

Выберите все подходящие ответы из списка

Верно. Так держать!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

-e \$0  
 -z \*  
 -s \$0  
 \$# -gt 0  
 \$var1 == \$var2 && \$var1 != \$var2  
 -n \$0

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 1 балл из 1

Верно решили 23 158 учащихся  
Из всех попыток 16% верных

Рис. 4.10: Задание 10

- \$0 - имя скрипта
- \$# - вернет количество аргументов
- -ge - больше или равно
- -n - не пустая строка.

Имя скрипта - это не пустая строка.

`#$` Это число аргументов без учета имени скрипта, который всегда `$0`. И число аргументов всегда будет или равно нулю, или больше него, тк просто не может скатиться в отрицательную сторону.

Выберите один вариант из списка

Верно.

Сначала two, потом four  
 Сначала four, потом one  
 Сначала four, потом four  
 Сначала two, потом one

[Следующий шаг](#) [Решить снова](#)

Верно решили 25 138 учащихся  
Из всех попыток 64% верных

Рис. 4.11: Задание 11

- `-lt, (<)` - меньше
- `-gt` - больше
- `-eq` - равно

3 не больше 5, 3 не меньше 3, 3 не равно 4.

5 не больше 5, 5 не меньше 3, 5 не равно 4.

Оба раза выведет `four`.

Напишите программу. Тестируется через `stdin` → `stdout`

Отличное решение!

Теперь вам доступен [Форум решений](#), где вы можете сравнить свое решение с другими или спросить совета.

```
1#!/bin/bash
2if [[ $1 -eq 1 ]]; then
3    echo "$1 student"
4elif [[ $1 -gt 1 && $1 -le 4 ]]; then
5    echo "$1 students"
6elif [[ $1 -ge 5 ]]; then
7    echo "A lot of students"
8else
9    echo "No students"
10fi
```

[Следующий шаг](#) [Решить снова](#)

Верно решили 23 310 учащихся  
Из всех попыток 38% верных

Рис. 4.12: Задание 12

1. Задаю общую часть в каждом выводе - слово “student”: `v=student`
2. Выполняем команды для разных аргументов.

3. res - это результат для вывода
4. echo "\$res" - вывести результат

Если запустить этот скрипт, то сколько раз на экран будет выведено слово "start", а сколько раз слово "finish"?

Выберите один вариант из списка

Абсолютно точно.

5 раз "start" и 4 раза "finish"  
 5 раз "start" и ни разу "finish"  
 3 раза "start" и 2 раза "finish"  
 3 раза "start" и 3 раза "finish"

[Следующий шаг](#) [Решить снова](#)

Верно решили 24 582 учащихся  
Из всех попыток 45% верных

Ваши решения Вы получили: 1 балл из 1

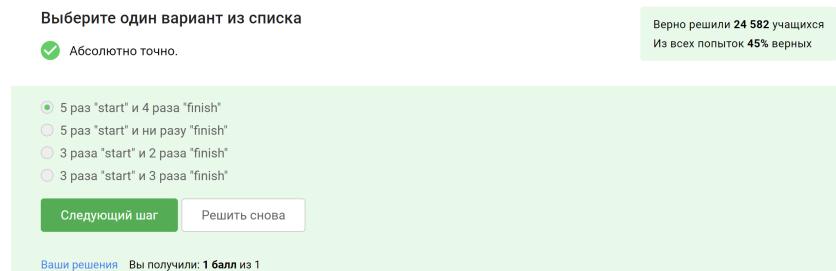


Рис. 4.13: Задание 13

- (Start)
- a > c нет (Finish)
- (Start)
- , > c нет (Finish)
- (Start)
- b > c нет (Finish)
- (Start)
- , > c нет (Finish)
- (Start)
- c\_d > c да

Напишите скрипт на bash, который будет определять в какую возрастную группу попадают пользователи. При запуске скрипта должен вывести сообщение "enter your name:" и ждать от пользователя ввода имени (используйте `read`, чтобы прочитать его). Когда имя введено, то скрипт должен написать "enter your age:" и ждать ввода возраста (опять нужен `read`). Когда возраст введен, скрипт пишет на экран <Имя>, your group is <группа>, где <группа> определяется на основе возраста по следующим правилам:

- младше либо равно 16: "child",
- от 17 до 25 (включительно): "youth",
- старше 25: "adult".

После этого скрипт опять выводит сообщение "enter your name:" и всё начинается по новой (бесконечный цикл!). Если в какой-то момент работы скрипта будет введено **пустое имя** или **возраст 0**, то скрипт должен написать на экран "bye" и закончить свою работу (выход из цикла).

**Примеры** корректной работы скрипта:

№1

```
./script.sh
enter your name:
Egor
enter your age:
16
Egor, your group is child
enter your name:
Elena
enter your age:
0
bye
```

№2:

```
./script.sh
enter your name:
Elena Petrovna
enter your age:
25
Elena Petrovna, your group is youth
enter your name:
bye
```

**Подсказка:** в случае проблем с решением задачи, обратите внимание [на наши рекомендации по написанию скриптов](#).

Напишите программу. Тестируется через `stdin` → `stdout`



Верно.

Верно решили 21 670 учащихся

Из всех попыток 23% верных

Теперь вам доступен [Форум решений](#), где вы можете сравнить свое решение с другими или спросить совета.

Рис. 4.14: Задание 14

Напишите программу. Тестируется через stdin → stdout

✓ Верно.

Верно решили 21 670 учащихся

Из всех попыток 23% верных

Теперь вам доступен [Форум решений](#), где вы можете сравнить свое решение с другими или спросить совета.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свое решение с другими на [форуме решений](#).

```
1#!/bin/bash
2child=16
3adult=25
4who=0
5
6while [[ $who != 1 ]]
7do
8    echo "enter your name: "
9    read name
10   if [[ (-z $name) ]]; then
11       echo "bye"
12       who=1
13   elif [[ -n $name ]]; then
14       while [[ $who != 1 ]]; do
15           echo "enter your age: "
16           read age
17           if [[ ($age -eq 0) ]]; then
18               echo "bye"
19               who=1
20           elif [[ $age -le $child ]]; then
21               echo "$name, your group is child"
22           elif [[ $age -gt $adult ]]; then
23               echo "$name, your group is adult" ;else
24                   if [[ ($age -ge 17) && ($age -le 25) ]]; then
25                       echo "$name, your group is youth" ;fi
26                   fi ;break
27   done ;fi
28done ;fi
```

Следующий шаг

Решить снова

Ваши решения Вы получили: 4 балла из 4

Рис. 4.15: Задание 14

child=16

adult=25

stdout=0

```
while [[ $stdout != 1 ]] #конструкция типа while-True
do
    echo "enter your name: " #Пользователь вводит имя
    read name
    if [[ (-z $name) || ($name = 0) ]]; then #Если имя не по параметрам, простишься
        echo "bye"
        stdout=1
    elif [[ -n $name ]]; then #А вот если имя нормальное
        while [[ $stdout != 1 ]] ;do
            echo "enter your age: " #То пусть вводит возраст
```

```

read age #Считываем возраст

if [[ ($age -eq 0) || (-z $age) ]] ;then #Если возраст 0 или строка пуста -
прощаемся

    echo "bye"
    stdout=1

    elif [[ $age -le $child ]] ;then #Если меньше или равен ребенку, то ребенок
        echo "$name, your group is child"

    elif [[ $age -gt $adult ]] ; then #Больше взрослого -
то взрослый

        echo "$name, your group is adult" ;else
        if [[ ($age -ge 17) && ($age -le 25) ]] ;then #Если от 17 до 25, то подросток
            echo "$name, your group is youth" ;fi
        fi ;break

done ;fi

done

```

Выберите все подходящие ответы из списка

Хорошие новости, верно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

let "a = a + b"  
 let "a+=b"  
 let a=a+b  
 let a = a + b  
 a+=\$b

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 1 балл из 1

Верно решили 22 116 учащихся  
Из всех попыток 20% верных

Рис. 4.16: Задание 15

1.  $a = \$a$
2.  $a += b$  это то же самое, что и  $a = a + b$ , но с символами “ $+=$ ” != “ $=+$ ”
3. если выражение не в скобках, но с пробелами - работать не будет. ( $let a=a+b$  - сработает;  $let a = a + b$  - нет)

Выберите один вариант из списка

Всё получилось!

Верно решили 23 677 учащихся  
Из всех попыток 51% верных

Код возврата команды pwd (0 в случае успешного выполнения и не 0 в случае ошибок)  
 /home/bi  
 pwd  
 /home/bi/Documents  
 `pwd`

**Следующий шаг**    **Решить снова**

Ваши решения   Вы получили: 1 балл из 1

Рис. 4.17: Задание 16

Выведет путь до директории, в которую мы перешли, так как “`pwd`” - это команда

Выберите все подходящие ответы из списка

Верно.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

Сначала запустить program, затем if [[ \$? -eq 0 ]]  
 if [[ 'program' -eq 0 ]]  
 Сначала var='program', затем if [[ \$var -eq 0 ]]  
 Ничего сделать нельзя  
 if 'program > some\_file.txt'

**Следующий шаг**    **Решить снова**

Ваши решения   Вы получили: 1 балл из 1

Рис. 4.18: Задание 16\_2

programm выполняет стандартный вывод в терминал (если это принцип работы программы). И нам нужно настроить вывод в файл.

Подсказка: этот пример можно решить в уме, но если система проверки не принимает ваше решение, то возможно вы что-то упустили (возможно что-то совсем небольшое/невидимое 😊). В этом случае имеет смысл написать небольшой скрипт на bash, который проделает ровно то, что указано в задании и посимвольно сверить свой ответ с тем, что он выдаст на экран.

Напишите текст

Прекрасный ответ.

Верно решили 20 009 учащихся  
Из всех попыток 28% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

counters are and 110

**Следующий шаг**    **Решить снова**

Ваши решения   Вы получили: 2 балла из 2

Рис. 4.19: Задание 17

Первая переменная локальная, и это просто пустая строка, вторая переменная - это сумма арифметической прогрессии от 1 до 10, равна 55, но при умножении на 2 даст 110.

3.4 Скрипты на bash: разное 10 из 10 шагов пройдено 14 из 14 баллов получено

Напишите скрипт на bash, который будет искать наибольший общий делитель (НОД, greatest common divisor, GCD) двух чисел. При запуске ваш скрипт не должен ничего писать на экран, а просто ждет ввода двух натуральных чисел через пробел (для этого можно использовать `read` и указать ему две переменные – см. пример в видеофрагменте). После ввода чисел скрипт считает их НОД и выводит на экран сообщение "GCD is <посчитанное значение>", например, для чисел 15 и 25 это будет "GCD is 5". После этого скрипт опять входит в режим ожидания двух натуральных чисел. Если в какой-то момент работы пользователь ввел вместо этого пустую строку, то нужно написать на экран "bye" и закончить свою работу.

Вычисление НОД несложно реализовать с помощью алгоритма Евклида. Вам нужно написать функцию `gcd`, которая принимает на вход два аргумента (назовем их **M** и **N**). Если аргументы равны, то мы нашли НОД – он равен **M** (или **N**), нужно выводить соответствующее сообщение на экран (см. выше). Иначе нужно сравнить аргументы между собой. Если **M** больше **N**, то запускаем ту же функцию `gcd`, но в качестве первого аргумента передаем (**M-N**), а в качестве второго **N**. Если же наоборот, **M** меньше **N**, то запускаем функцию `gcd` с первым аргументом **M**, а вторым (**N-M**).

Пример корректной работы скрипта:

```
./script.sh  
10 15  
GCD is 5  
7 3  
GCD is 1  
  
bye
```

**Примечание:** в вызове функции из себя самой нет ничего страшного или неправильного, т.ч. смело вызывайте `gcd` прямо внутри `gcd`!

**Примечание 2:** для завершения работы функции в произвольном месте, можно использовать инструкцию `return` (все инструкции функции после `return` выполняться не будут). В отличии от `exit` эта команда завершит только функцию, а не выполнение всего скрипта целиком. Однако в данной задаче можно обойтись и без использования `return`!

**Подсказка:** в случае проблем с решением задачи, обратите внимание [на наши рекомендации по написанию скриптов](#).

Рис. 4.20: Задание 18

Напишите программу. Тестируется через stdin → stdout

✓ Так точно!

Верно решили 18 148 учащихся  
Из всех попыток 35% верных

Теперь вам доступен [Форум решений](#), где вы можете сравнить свое решение с другими или спросить совета.

```
1#!/bin/bash
2while [ true ]
3do
4    read n1 n2
5    if [ -z $n1 ]; then
6        echo "bye"
7        break
8    else
9        gcd () {
10            remainder=1
11            if [ $n2 -eq 0 ]
12            then
13                echo "bye"
14            fi
15            while [ $remainder -ne 0 ]
16            do
17                remainder=$((n1%n2))
18                n1=$n2
19                n2=$remainder
20            done
21        }
22        gcd $1 $2
23        echo "GCD is $n1"
24    fi
25done
```

Следующий шаг

Решить снова

Ваши решения Вы получили: 4 балла из 4

Рис. 4.21: Задание 18

## Алгоритм нахождения НОД делением

1. Большее число делим на меньшее.
2. Если делится без остатка, то меньшее число и есть НОД (следует выйти из цикла).
3. Если есть остаток, то большее число заменяем на остаток от деления.
4. Переходим к пункту 1.

Напишите **калькулятор** на bash. При запуске ваш скрипт должен ожидать ввода пользователем команды (при этом на экран выводить ничего не нужно). Команды могут быть трех типов:

1. Слово **"exit"**. В этом случае скрипт должен вывести на экран слово **"bye"** и завершить работу.
2. **Три аргумента через пробел** – первый operand (целое число), операция (одна из **"+", "-", "\*", "/", "%", "\*\*"**) и второй operand (целое число). В этом случае нужно произвести указанную операцию над заданными числами и вывести результат на экран. После этого переходим в режим ожидания новой команды.
3. **Любая другая команда** из одного аргумента или из трех аргументов, но с операцией не из списка. В этом случае нужно вывести на экран слово **"error"** и завершить работу.

Чтобы проверить работу скрипта, вы можете записать сразу несколько команд в файл и передать его скрипту на stdin (т.е. выполнить `./script.sh < input.txt`). В этом случае он должен вывести сразу все ответы на экран.

Например, если входной файл будет следующего содержания:

```
10 + 1  
2 ** 10  
exit
```

то на экране будет:

```
11  
1024  
bye
```

Если же на вход поступит следующий файл:

```
3 - 5  
2/10  
exit
```

то на экране будет:

```
-2  
error
```

т.к. вторая команда была **некорректной** (в ней всего один аргумент, т.к. нет пробелов между числами и операцией, а единственная допустимая команда из одного аргумента это **"exit"**).

**Подсказка:** в случае проблем с решением задачи, обратите внимание [на наши рекомендации по написанию скриптов](#).

Напишите программу. Тестируется через stdin → stdout

Верно.

Теперь вам доступен [Форум решений](#), где вы можете сравнить свое решение с другими или спросить совета.

Верно решили **16 980** учащихся  
Из всех попыток **36%** верных

Рис. 4.22: Задание 19

Напишите программу. Тестируется через stdin → stdout

Верно.

Теперь вам доступен [Форум решений](#), где вы можете сравнить свое решение с другими или спросить совета.

```

1#!/bin/bash
2while [[ $True ]]
3do
4    read birinchi amal ikkinchi
5    if [[ $birinchi == "exit" ]]
6    then
7        echo "bye"
8        break
9    elif [[ "$birinchi" =~ ^[0-9]+\$ && "$ikkinchi" =~ ^[0-9]+\$ ]]
10   then
11       echo "error"
12       break
13    else
14        case $amal in
15        "+") let "result = birinchi + ikkinchi";;
16        "-") let "result = birinchi - ikkinchi";;
17        "/") let "result = birinchi / ikkinchi";;
18        "*") let "result = birinchi * ikkinchi";;
19        "%") let "result = birinchi % ikkinchi";;
20        "**") let "result = birinchi ** ikkinchi";;
21    *) echo "error"; break ;;
22    esac
23    echo "$result"
24  fi
25 done

```

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 5 баллов из 5

Рис. 4.23: Задание 19

Калькулятор выглядит обычно - мы вводим два числа, пишем, что с ними надо сделать, и потом, учитывая случаи ошибок, выводим результат.

### Выберите все подходящие ответы из списка

Правильно, молодец!

Eddard\_Stark\_biography.txt

star\_trek\_OST.mp3

Star\_Wars.avi

stardust.mpeg

STARS.txt

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 1 балл из 1

Рис. 4.24: Задание 20

-iname ищет без учета регистра, а -name в точности как в запросе. Звездочка

СТОИТ ПОСЛЕ СЛОВА - ЭТО ЗНАЧИТ ПОСЛЕ СЛОВА МОЖЕТ БЫТЬ СКОЛЬКО УГОДНО СИМВОЛОВ.

Задание на понимание работы опций `-path` и `-name` команды `find`. Отметьте **все верные** утверждения из перечисленных ниже.

Выберите все подходящие ответы из списка

Правильно, молодец!

Верно решили **18 450** учащихся  
Из всех попыток **22%** верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- Опция `-path` аналогична `-name`, но игнорирует размер букв (строчные/прописные) в имени файла
- В некоторых случаях `find -name` найдет меньше файлов, чем `find` с таким же запросом, но с `-path`
- Если заменить в команде поиска `-name` на `-path`, то результат поиска иногда может оставаться таким же
- В некоторых случаях `find -path` найдет больше файлов, чем `find` с таким же запросом, но с `-path`
- Если заменить в команде поиска `-name` на `-path`, то результат поиска всегда останется неизменным

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: **1 балл из 1**

Рис. 4.25: Задание 21

`find [path] [expression]`

где: `path` - это путь к директории, в которой нужно выполнить поиск файлов (по умолчанию, поиск производится в текущей директории и всех ее поддиректориях);

`expression` - это выражение, которое определяет критерии поиска файлов.

`-name`: поиск файлов по имени. Например: `find /home/user -name myfile.txt`

Предположим, что в директории `/home/b1/` есть следующая структура файлов и поддиректорий:



Какие(ой) из трех файлов (`file1`, `file2`, `file3`) будут найдены по команде `find /home/b1 -mindepth 2 -maxdepth 3 -name "file*"`?

Выберите один вариант из списка

Всё получилось!

Верно решили 20 711 учащихся  
Из всех попыток 41% верных

- Только `file1`
- Все три файла
- Все кроме `file3`
- Только `file2`
- Все кроме `file1`

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 4.26: Задание 22

Текущий каталог - это depth=1, а остальное считается просто:

`/home/b1` -> depth=1

`/home/b1/dir1` -> depth=2

`/home/b1/dir1/dir2` -> depth=3

Задание на понимание работы опций `-A`, `-B` и `-C` команды `grep`. Пусть у вас есть файл `file.txt` из 10 строк, причем **в каждой строке есть** слово "word". Если вы выполните на этом файле команды:

```

grep "word" file.txt > results.txt
grep -A 1 "word" file.txt > results.txt
grep -B 1 "word" file.txt > results.txt
grep -C 1 "word" file.txt > results.txt

```

то какая(ие) из них создаст файл `results.txt` наибольшего размера?

Выберите один вариант из списка

Отличное решение!

Верно решили 20 237 учащихся  
Из всех попыток 41% верных

- `grep -A 1 "word" file.txt > results.txt` и `grep -B 1 "word" file.txt > results.txt`
- `grep -C 1 "word" file.txt > results.txt`
- Все, кроме `grep "word" file.txt > results.txt`
- `grep -A 1 "word" file.txt > results.txt`
- `results.txt` будет одинакового размера во всех случаях

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 4.27: Задание 23

Из описания man: Print NUM lines of trailing context after/before matching lines  
“matching lines” - множественное число, строки в которых нашлось совпадение

Т.е. если идут 2...10...100 строк подряд, в которых обнаружилось совпадение, контекст будет выведен до и после этой ГРУППЫ строк, а не до и после каждой строки в этой группе

3.5 Продвинутый поиск и редактирование 13 из 13 шагов пройдено 10 из 10 баллов получено

Предположим, что в файле `text.txt` записаны строки, показанные среди вариантов ответа. Отметьте только те из них, которые выведет на экран команда `grep -E "[xkLXKL]?[uU]buntu$" text.txt`.

Выберите все подходящие ответы из списка

Так точно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свое решение с другими на [форуме решений](#).

Верно решили 18 768 учащихся  
Из всех попыток 23% верных

- Well, xubuntu is OK
- Mac OS X 10.9, Windows XP, Ubuntu 12.04
- Kubuntu
- Uuuubuntu!
- Linux is not always Ubuntu
- The best OS is Xubuntu

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: 2 балла из 2

Рис. 4.28: Задание 24

Предположим, что в файле `text.txt` записаны строки, показанные среди вариантов ответа. Отметьте только те из них, которые выведет на экран команда `grep -E "[xkLXKL]?[uU]buntu$" text.txt`.

Выберите все подходящие ответы из списка

Так точно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

Верно решили 18 768 учащихся  
Из всех попыток 23% верных

- Well, xubuntu is OK
- Mac OS X 10.9, Windows XP, Ubuntu 12.04
- Kubuntu
- Uuuubuntu!
- Linux is not always Ubuntu
- The best OS is Xubuntu

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: 2 балла из 2

Рис. 4.29: Задание 24

Объяснение на втором скриншоте.

Что произойдет, если в команде `sed -n "/[a-z]*p" text.txt` не указывать опцию `-n`?

Выберите один вариант из списка

✓ Верно. Так держать!

Верно решили 19 784 учащихся  
Из всех попыток 39% верных

- На экран ничего не напечатается
- Каждая строка будет выведена два раза
- Будут выведены все строки файла `text.txt`, в которых есть только большие буквы латинского алфавита
- На экран будет выведено всё содержимое файла `text.txt`

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 4.30: Задание 25

The `-n` option disables the automatic printing, which means the lines you don't specifically tell it to print do not get printed, and lines you do explicitly tell it to print (e.g. with `p`) get printed only once.

3.5 Продвинутый поиск и редактирование 13 из 13 шагов пройдено 10 из 10 баллов получено

Запишите в форму ниже инструкцию `sed`, которая заменит все "аббревиатуры" в файле `input.txt` на слово "abbreviation" и запишет результат в файл `edited.txt` (на экран при этом ничего выводить не нужно). Обратите внимание, что в инструкции должны быть указаны и сам `sed`, и оба файла!

Под "аббревиатурой" будем понимать слово, которое удовлетворяет следующим условиям:

- состоит только из больших букв латинского алфавита,
- состоит из хотя бы двух букв,
- окружено одним пробелом с каждой стороны.

При этом будем считать, что в тексте **не может быть две "аббревиатуры" подряд**. Например, текст "YOU YOU and YOU!" является **некорректным** (в нем есть две "аббревиатуры", но они идут подряд) и на таких примерах мы проверять вашу инструкцию **не будем**.

**Пример:** если у вас был текст "Hi, I heard these songs by ABBA, TLA and DM!", то он должен быть преобразован в "Hi, I heard these songs by ABBA, abbreviation and abbreviation!".

**Примечание:** после вашей замены "аббревиатуры" на слово "abbreviation" **количество пробелов** в тексте **не должно меняться**!

**Внимание!** Во время проверки мы **не запускаем команду**, которую вы ввели на реальном файле с "аббревиатурами" (это небезопасно, можно же ввести `rm -rf /*`!). Вместо этого мы сперва анализируем структуру вашей инструкции (например, что в ней использован именно `sed` и сделано это ровно один раз, что на вход подается `input.txt`, а результат будет записан в `edited.txt` и т.д.), а затем **запускаем её смысловую часть** (т.е. поиск по регулярному выражению и замена на "abbreviation") на тестовых примерах. К сожалению, наш запуск **не идеально повторяет** `sed`, но он очень близок к нему. Главная "несовместимость" заключается в том, что наша проверка не понимает идущие подряд символы, отвечающие за количество повторений (т.е. `*, +, ?` и `{}`). Однако эту "несовместимость" легко исправить указав при помощи `(")"` какой из символов к чему относится! Например, регулярное выражение `a{+}` (ноль или один раз по одной или более букве "a") нужно записать как `(a){+}`? (при этом запись `(a){+?}`, конечно же, не поможет).

Напишите текст

✓ Отлично!

Верно решили 16 632 учащихся  
Из всех попыток 34% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свое решение с другими на [форуме решений](#).

```
sed 's/[A-Z]{2,}/abbreviation /g' input.txt > edited.txt
```

Следующий шаг

Решить снова

Ваши решения Вы получили: 3 балла из 3

Рис. 4.31: Задание 26

аббревиатура АВВА отличается от двух других аббревиатур тем, что справа он неё стоит запятая без пробела: “ABBA,”.

При этом по условию аббревиатура должна выглядеть как [ XX ] или [ XXX ] (и ещё больше X). Следовательно, для этой проверки надо добавить пробел квадратными скобками [ ] слева и, соответственно, с права.

Вы можете скачать и попробовать применить gnuplot к файлу, который мы показали в видеофрагменте: [authors.txt](#).  
Какую опцию нужно указать при запуске gnuplot, чтобы при его закрытии не были автоматически закрыты и все нарисованные в нём графики?

Выберите один вариант из списка

Верно.

Графики и так не закрываются автоматически при закрытии gnuplot!  
-s, --show-plots-after-exit  
 -p, --persist  
Такой опции не существует

[Следующий шаг](#) [Решить снова](#)

Ваше решение Вы получили: 1 балл из 1

Верно решили 18 785 учащихся  
Из всех попыток 51% верных

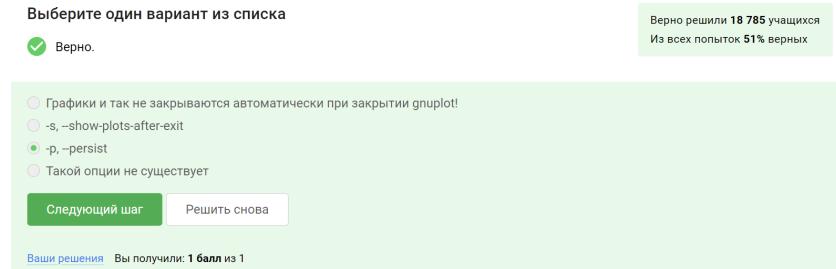


Рис. 4.32: Задание 27

-persist lets plot windows survive after main gnuplot program exits.

3.6 Строим графики в gnuplot 10 из 10 шагов пройдено 7 из 7 баллов получено

Предположим у вас есть файл `data.csv` с двумя столбцами по 10 чисел в каждом. В первой строке не записаны названия столбцов, т.е. ряды данных начинаются прямо с первой строки. Вы запускаете gnuplot и вводите в него две команды:

```
set key autotitle columnhead  
plot 'data.csv' using 1:2
```

Какое в этом случае будет **название** **ярда** **данных** и **сколько** будет нарисовано **точек** на графике?

Выберите один вариант из списка

Абсолютно точно.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свое решение с другими на [форуме решений](#).

Название -- первое значение из первого столбца, нарисовано 10 точек  
Название "нопаме", нарисовано 10 точек  
Название -- первое значение из второго столбца, нарисовано 10 точек  
Название -- первое значение из первого столбца, нарисовано 9 точек (точка из первой строки пропущена)  
 Название -- первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)

[Следующий шаг](#) [Решить снова](#)

Ваше решение Вы получили: 1 балл из 1

Верно решили 17 975 учащихся  
Из всех попыток 32% верных

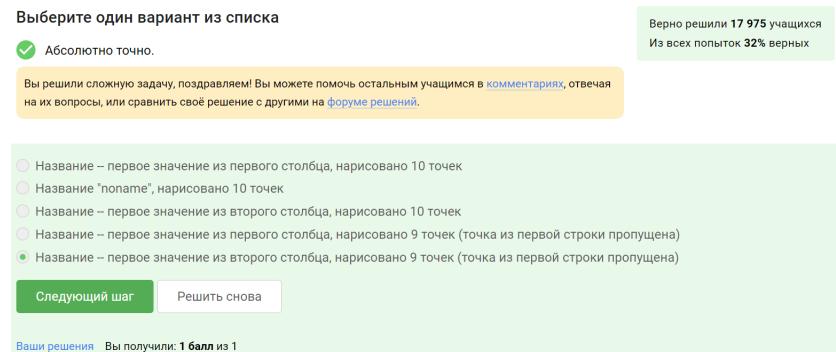


Рис. 4.33: Задание 28

`plot 'data.csv' using 1:2` даст ошибку:

`warning: Skipping data file with no valid points ^ x range is invalid`

Скорее всего причиной такого поведения является тот факт, что формат CSV содержит строки, где столбцы разделены запятой? Содержимое файла:

1,21  
2,22  
3,23  
4,24  
5,25  
6,26  
7,27  
8,28  
9,29  
10,30

3.6 Строим графики в gnuplot 10 из 10 шагов пройдено 7 из 7 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [plot.gnu](#), [plot\\_advanced.gnu](#), [plot\\_advanced2.gnu](#). Все три скрипта основаны на [этой заметке](#), данные также взяты оттуда.

Предположим, что вы пишите gnuplot-скрипт и у вас в нем есть три переменные `x1`, `x2`, `x3`, в которых записаны координаты важных точек по оси ОХ (по возрастанию). Вы хотите, чтобы на этой оси было только три деления (т.е. три черточки) в этих самых координатах, а подписи этих делений были оформлены в виде "`point <номер точки>, value <значение соответствующей переменной>`". Например, для `x1=0, x2=10, x3=20`, это были бы надписи "point 1, value 0" в точке с координатой 0 по горизонтали, "point 2, value 10" в точке с координатой 10 и "point 3, value 20" в точке с координатой 20. Или, например, `x1=100, x2=150, x3=250`, это были бы надписи "point 1, value 100" в точке с координатой 100, "point 2, value 150" в точке с координатой 150 и "point 3, value 250" в точке с координатой 250.

Впишите в форму ниже **одну команду** (т.е. одну строку), которую нужно добавить в скрипт, для выполнения этой задачи.

**Примечание:** проверять, что переменные `x1`, `x2`, `x3` идут по возрастанию или что они являются числами **не нужно!**

**Примечание 2:** в видеофрагменте на предыдущем шаге звучал термин конкатенация, который важен для выполнения данного задания. Под конкатенацией обычно понимают "склеивание" двух строк в одну длинную строку, например, конкатенация строк "Данные из файла " и "data.csv" даст строку "Данные из файла data.csv".

**Подсказка:** настоятельно рекомендуем изучить примеры скриптов – в них есть большая часть решения!

Напишите текст

Прекрасный ответ.

Верно решили **13 935** участников  
Из всех попыток **44%** верных

```
set xtics ("point 1, value ".x1 x1, "point 2, value ".x2 x2, "point 3, value ".x3 x3)
```

Следующий шаг

Решить снова

Ваши решения Ваши получили: **2 балла из 2**

Рис. 4.34: Задание 29

Сначала идет команда установки подписей, а потом в скобках:

подпись - пробел - переменная с координатой - запятая

Повторяется это количество раз соответствующее числу переменных, и без запятой (в случае с последней переменной)

А подпись в свою очередь получается конкатенацией текста из задания и переменной с координатой.

3.6 Строим графики в gnuplot 10 из 10 шагов пройдено 7 из 7 баллов получено

Если вы не скачали на предыдущем шаге файлы [animated.gnu](#) и [move.rot](#), то скачайте их теперь, т.к. они понадобятся для выполнения задания.

Указанные файлы использовались в последнем видеофрагменте для создания вращающегося графика. Измените инструкции в файле `move.rot` (т.е. **добавлять и удалять** инструкции **нельзя**) таким образом, чтобы:

- График **отразился зеркально** относительно горизонтальной поверхности. То есть там, где была точка (10, 10, 200), станет точка (10, 10, -200), где была точка (-10, -10, 200) станет (-10, -10, -200) и т.д. При этом точка (0, 0, 0) останется на месте.
- Изображение **стало вращаться в обратную сторону**. То есть если раньше вращалось "влево", то теперь станет "вправо".
- Вращение **стало в два раза быстрее**. То есть станет в два раза больше перерисовок графика на каждую секунду вращения.

Измененный файл загрузите в форму ниже.

**Примечание:** наша система проверки **не может** запустить на вашем файле `move.rot` программу gnuplot и сравнить полученный график с заданным. Вместо этого мы **анализируем команды**, которые вы указали в файле. Поэтому если вы видите, что ваш скрипт в gnuplot работает точно по условию, а мы отвечаем "Incorrect/Неверно", то попробуйте упростить свою модификацию `move.rot` и отправить его еще раз.

Напишите текст

Так точно!

Верно решили 12 854 учащихся  
Из всех попыток 47% верных

```
a=a+1
zrot=(zrot+350)%360
set view xrot,zrot
splot -x**2-y**2
pause 0.1
if (a<50) reread
```

Следующий шаг

Решить снова

Ваше решение Вы получили: 3 балла из 3

Рис. 4.35: Задание 30

1. График строится строкой “`splot x2+y2`”.

2. Вращение задается строкой “`zrot=(zrot+10)%360`”. Значит, смещение вперед (которое было изначально) можно также задать строкой “`zrot=(zrot+360+10)%360`” или иначе говоря “`zrot=(zrot+370)%360`”. А теперь посмотрим на наше требование - чтоб вращалось в другую сторону, значит, по аналогии, необходимо вместо перебора на 10 сделать недобор.

“`zrot=(zrot+350)%360`”

3. Стока “pause 0.2” ставит выполнение на паузу на определенный промежуток времени. В задании сказали перерисовывать чаще, значит пауза должна быть меньше.

3.7 Разное 15 из 15 шагов пройдено 7 из 7 баллов получено

Какая команда(ы) установят файлу `file.txt` права доступа `rwxrw-r--`, если изначально у него были права `r--r--r--`. Укажите **все верные** варианты ответа!

Примечание: запись вида `команда1; команда2; команда3` означает, что в терминале последовательно выполнились все три команды (сначала `команда1`, затем `команда2` и, наконец, `команда3`).

Выберите все подходящие ответы из списка

Хорошие новости, верно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

`chmod 777 file.txt`  
 `chmod u-wx file.txt; chmod g-w file.txt`  
 `chmod u+wx file.txt; chmod g+w file.txt`  
 `chmod rwxrw-- file.txt`  
 `chmod o-wx file.txt; chmod g-x file.txt; chmod a+wx file.txt`  
 `chmod 764 file.txt`

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: **1 балл** из 1

Рис. 4.36: Задание 31

- `r` - чтение;
- `w` - запись;
- `x` - выполнение;
- `s` - выполнение от имени суперпользователя (дополнительный);
- `u` - владелец файла;
- `g` - группа файла;
- `o` - все остальные пользователи;
- `0` - никаких прав;
- `1` - только выполнение;

- 2 - только запись;
- 3 - выполнение и запись;
- 4 - только чтение;
- 5 - чтение и выполнение;
- 6 - чтение и запись;
- 7 - чтение запись и выполнение.

Предположим вы использовали команду `sudo` для создания директории `dir`. По умолчанию для `dir` были выставлены права доступа `rwxr-xr-x` (владелец `root`, группа `root`). Таким образом никто кроме пользователя `root` не может ничего записывать в эту директорию, например, не может создавать файлы в ней.

После выполнения какой команды `user` из группы `group` всё-таки сможет создать файл внутри `dir`? Укажите **все верные** варианты ответов!

**Примечание:** считаем, что все команды выполняются от имени `user`, если явно не указано, что команда выполнена с `sudo`.

**Примечание 2:** мы выбрали пример с директорией, а не с файлом не случайно.

Дело в том, что если создать при помощи `sudo` файл с правами `rw-r--r--` в директории, которая принадлежит пользователю, то возникнет любопытная ситуация. С одной стороны пользователь может удалить этот файл (т.к. ему разрешено удалять **все** файлы внутри его директории) и может прочитать его содержимое (т.к. право "r" у файла установлено для всех), с другой стороны он не может этот файл редактировать (т.к. право "w" у файла есть только для `root`). При этом некоторые "умные" редакторы, например, `vim` позволят даже редактировать этот файл, но сделают они это своеобразно: через удаление оригинала и создание копии уже с нужными правами (удалить мы можем, а раз можем читать, то и копию создать не сложно). Итого получается, что несмотря на права `rw-r--r--`, пользователь может сделать с этим файлом почти всё что угодно!

В случае же, когда речь идет о директории созданной `root`, ситуация будет проще: пользователь сможет смотреть её содержимое (у него есть право "r"), но удалять и создавать файлы в ней не сможет (права "w" у него нет).

Важно отметить, что директории в *Linux* это в каком-то смысле *файлы*. Содержимое такого "файла" – это записи о файлах и поддиректориях этой директории (грубо говоря их *названия*). Таким образом, право "r" у директории дает возможность просматривать "записи", т.е. просматривать её состав. Право "w" у директории дает возможность удалять/добавлять новые "записи", т.е. удалять/создавать файлы/поддиректории в ней.

На самом деле и это еще не всё. Существует так называемый *sticky bit* (атрибут файла или директории), выставление которого меняет описание выше поведение. Файлы (или директории) с таким атрибутом смогут удалять только их владелец вне зависимости от прав, установленных у директории, в которой эти файлы (или директории) лежат!

Отдельное спасибо слушателю курса **Alexey Antipovsky** за помощь в оформлении **Примечания 2!**

Выберите **все подходящие ответы из списка**

Всё правильно.

Верно решили 14 683 учащихся  
Из всех попыток 15% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на ваши вопросы, или сравнить свое решение с другими на [форуме решений](#).

- `chmod o+w dir`  
 `sudo chown user dir`  
 `sudo chown :group dir`  
 `chown user:group dir`  
 `sudo chmod o+w dir`  
 `sudo chmod g+w dir`

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: **1 балл из 1**

Рис. 4.37: Задание 32

Решений два типа:

- Сменить права гостей, добавив W
- Сделать владельцем нужную группу или пользователя, в зависимости от того, у кого из них уже есть права на W
- Помнить, что root - владелец и остальные для него - others.

3.7 Разное 15 из 15 шагов пройдено 7 из 7 баллов получено

Отметьте какие характеристики файла можно посчитать с использованием команды `wc`.

Выберите все подходящие ответы из списка

Верно. Так держать!

Верно решили 17 158 учащихся  
Из всех попыток 21% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- Количество символов
- Размер файла в байтах
- Длину самой длинной строки
- Количество строк
- Количество слов

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: 1 балл из 1

Рис. 4.38: Задание 33

- `wc -l` вывести количество строк
- `wc -c` вывести количество байт
- `wc -m` вывести количество символов
- `wc -L` вывести длину самой длинной строки
- `wc -w` вывести количество слов

Впишите в форму ниже команду, которая выведет сколько места на диске занимает текущая директория (при этом **размер** нужно вывести в **удобном для чтения формате** (например, вместо 2048 байт надо выводить 2,0K) и **больше** на экран выводить **ничего не нужно**). В команде указывайте **только необходимые** для выполнения задания **опции и аргументы**, лишних опций указывать не нужно!

**Пример:** если в текущей директории есть два файла по 800 Кбайт и две поддиректории в каждой из которых лежит по файлу в 400 Кбайт, то загаданная команда должна вывести на экран одно число: 2.4M (также на экране может быть выведен еще и символ "", обозначающий, что это размер именно текущей директории).

Напишите текст

Всё получилось!

du -h -s

Следующий шаг    Решить снова

Ваши решения    Вы получили: 2 балла из 2

Верно решил 16 381 учащийся  
Из всех попыток 53% верных

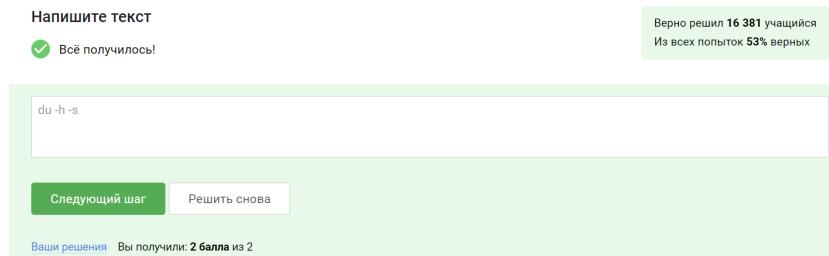


Рис. 4.39: Задание 34

-h, –human-readable print sizes in human readable format (e.g., 1K 234M 2G)  
-s, –summarize display only a total for each argument

Впишите в форму ниже максимально короткую команду (т.е. в которой минимально возможное число символов), которая позволит создать в текущей директории 3 поддиректории с именами dir1, dir2, dir3.

Если вы придумали команду, которая выполняет эту задачу, а система проверки сообщает вам "Incorrect"/"Неверно", то скорее всего вы придумали не самую короткую команду из возможных!

Напишите текст

Всё правильно.

mkdir dir{1..3}

Следующий шаг    Решить снова

Ваши решения    Вы получили: 2 балла из 2

Верно решили 16 720 учащихся  
Из всех попыток 40% верных

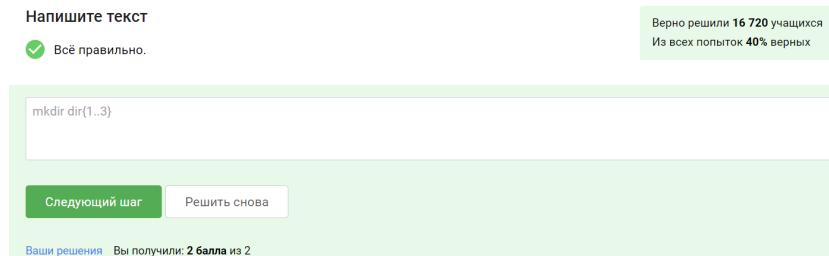


Рис. 4.40: Задание 35

Команда создаст три директории от dir1 до dir3.

## 5 Сертификат

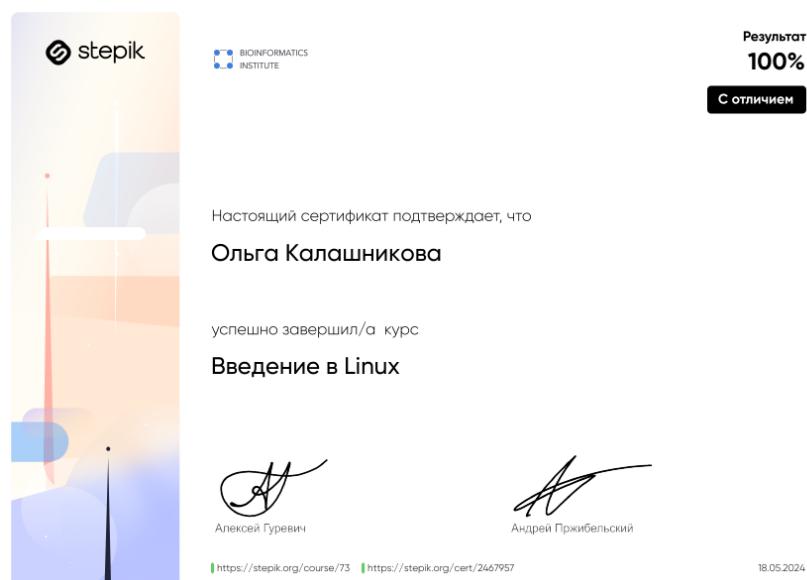


Рис. 5.1: Сертификат

## **6 Выводы**

Я просмотрела курс и освежила в памяти навыки работы с более сложными командами в Линукс.

# **Список литературы**

1. Введение в Linux