

# **Отчет о прохождении первого модуля внешнего курса**

**Введение в Linux. Модуль 3**

Филиппева Ксения Дмитриевна

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	34

## Список иллюстраций

2.1	Задание 1 . . . . .	6
2.2	Задание 2 . . . . .	7
2.3	Задание 3 . . . . .	8
2.4	Задание 4 . . . . .	8
2.5	Задание 5 . . . . .	9
2.6	Задание 6 . . . . .	9
2.7	Задание 7 . . . . .	10
2.8	Задание 8 . . . . .	11
2.9	Задание 9 . . . . .	12
2.10	Задание 10 . . . . .	13
2.11	Задание 11 . . . . .	14
2.12	Задание 12 . . . . .	15
2.13	Задание 13 . . . . .	16
2.14	Задание 14 . . . . .	17
2.15	Задание 15 . . . . .	18
2.16	Задание 16 . . . . .	19
2.17	Задание 17 . . . . .	19
2.18	Задание 18 . . . . .	20
2.19	Задание 19 . . . . .	21
2.20	Задание 20 . . . . .	22
2.21	Задание 21 . . . . .	23
2.22	Задание 22 . . . . .	23
2.23	Задание 23 . . . . .	24
2.24	Задание 24 . . . . .	24
2.25	Задание 25 . . . . .	25
2.26	Задание 26 . . . . .	25
2.27	Задание 27 . . . . .	26
2.28	Задание 28 . . . . .	26
2.29	Задание 29 . . . . .	27
2.30	Задание 30 . . . . .	28
2.31	Задание 31 . . . . .	29
2.32	Задание 32 . . . . .	30
2.33	Задание 33 . . . . .	31
2.34	Задание 34 . . . . .	32
2.35	Задание 35 . . . . .	32
2.36	Задание 36 . . . . .	33

## **Список таблиц**

# 1 Цель работы

Пройти курс, который дополнит уже имеющиеся и возможно даст новые знания.

## 2 Выполнение лабораторной работы

Это было и в наших лабораторных работах, поэтому без проблем нахожу верный ответ. (рис. 2.1).

Какую клавишу(и) нужно нажать на клавиатуре, чтобы выйти из редактора vim? Считайте, что вы только что открыли файл и вам сразу понадобилось выйти из редактора.

Выберите один вариант из списка

☒ Всё правильно.

Верно решили **32 523** учащихся  
Из всех попыток **69%** верных

- ☐ "q", затем "Enter"
- ☐ "q"
- ☒ " : ", затем "q", затем "Enter"
- ☐ "Esc"
- ☐ "Ctrl", затем "x"

Следующий шаг

Решить снова

Рис. 2.1: Задание 1

Тут особо нечего писать, просто использование усвоенной информации из видео. (рис. 2.2).



Предположим, что в текстовом файле записана одна единственная строка:

```
one two three four five
```

и вам нужно преобразовать её в строку

```
three four four four five
```

**Какие(ой)** из предложенных ниже **наборов нажатий клавиш** выполнят такое редактирование? В этих наборах нажатие на клавишу Esc обозначается как <Esc> (т.е. знаки "<" и ">" не несут отдельного смысла).

**Примечание:** во всех утверждениях имеется в виду, что мы находимся в редакторе vim, включен нормальный режим работы и курсор находится в самом начале строки.

Выберите все подходящие ответы из списка

☒ Отлично!

Верно решили **23 655** учащихся  
Из всех попыток **16%** 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](#).

- ☒ d2wwywPp
- ☐ x2wwywPp
- ☒ ddithree four four four five<Esc>
- ☐ d2dwywPp
- ☒ d2w\$bifour four <Esc>
- ☒ d2wwifour four <Esc>

Рис. 2.3: Задание 3

Ответ получен методом экспериментов :) (рис. 2.4).

Предположим, что вы открыли файл в редакторе vim и хотите заменить в этом файле все строки, содержащие слово `Windows`, на такие же строки, но со словом `Linux`. Если в какой-то строке слово `Windows` встречается больше, чем один раз, то заменить на `Linux` в этой строке нужно **только самое первое** из этих слов.

**Какую команду** нужно ввести для этого в vim? Укажите необходимую команду целиком (т.е. **включая** ввод ":" в самом начале), однако нажатие на `Enter` после ввода команды обозначать никак **не нужно**.

Напишите текст

☒ Правильно.

Верно решил **24 631** учащихся  
Из всех попыток **57%** верных

```
:%s/Windows/Linux
```

Рис. 2.4: Задание 4

Ответ был получен после изучения справки по команде. (рис. 2.5).



Мы совсем не рассказали вам про третий режим работы vim – режим **выделения (Visual)**. Предлагаем вам ознакомиться с ним самостоятельно. Например, это можно сделать во время прохождения упражнений в vimtutor, который мы настоятельно рекомендуем вам для изучения vim!

Чтобы убедиться, что вы разобрались с этим режимом работы, отметьте, пожалуйста, **все верные** утверждения из списка ниже.

**Подсказка:** если вы не хотите проходить vimtutor целиком, то можете открыть его и поиском найти слово **"Visual"**. Вы попадете в задание, прохождение которого будет вполне достаточно, чтобы выполнить это задание.

#### Выберите все подходящие ответы из списка

☒ Отлично!

Верно решили **23 497** учащихся  
Из всех попыток **29%** 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](#).

- ☒ В режиме выделения можно использовать команды перемещения (например, W, e, \$, и др.)
- ☒ Когда вы находитесь в режиме выделения, внизу редактора горит надпись – VISUAL – (или – ВИЗУАЛЬНЫЙ РЕЖИМ –)
- ☒ Режим выделения открывается из нормального режима по нажатию "v"
- ☐ Режим выделения открывается из любого другого режима по нажатию "v"
- ☒ Выйти из режима выделения можно, нажав клавишу Esc два раза
- ☒ В режиме выделения можно использовать команды d (удалить) и у (скопировать)

### Рис. 2.5: Задание 5

Сделан логический вывод из данной нам информации. (рис. 2.6).

Надеемся, что вы разобрались, что одну оболочку (например, `sh`) можно запустить из другой оболочки (например, из `bash`).

Предположим, что вы открыли терминал и у вас в нем запущена оболочка `bash`. Вы набираете в ней команды `A1`, `A2`, `A3`, а затем запускаете оболочку `sh`. В этой оболочке вы набираете команды `B1`, `B2`, `B3` и запускаете оболочку `bash`. И, наконец, в этой последней оболочке вы набираете команды `C1`, `C2`, `C3`. Если теперь вы попытаете при помощи стрелочек вверх/вниз перемещаться по истории набранных команд, то команды из какого набора(ов) будут появляться?

#### Выберите один вариант из списка

☒ Хорошие новости, верно!

Верно решили **30 266** учащихся  
Из всех попыток **65%** верных

- ☐ Из наборов A и C
- ☐ Из наборов B и C
- ☒ Только из набора C
- ☐ Только из набора A
- ☐ Никакие команды появляться не будут

### Рис. 2.6: Задание 6

То, что находится после команды “тач” никак не влияет на расположение файла, поэтому он будет создан в той директории, в которой мы находились до написания программы. (рис. 2.7).

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [script1.sh](#), [script2.sh](#).

Предположим, что вы находитесь в директории `/home/bi/Documents/` и запускаете в ней скрипт следующего содержания:

```
#!/bin/bash
cd /home/bi/
touch file1.txt
cd /home/bi/Desktop/
```

Как будет выглядеть **абсолютный путь** до созданного файла `file1.txt` по окончании работы скрипта?

Выберите один вариант из списка

☒ Всё получилось!

Верно решили **29 905** учащихся  
Из всех попыток **76%** верных

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

Следующий шаг

Решить снова

Рис. 2.7: Задание 7

Проверено практическим путём. (рис. 2.8).

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [variables1.sh](#), [variables2.sh](#).

Какие из представленных ниже строк **могут** быть именами переменных в bash? Выберите **все** подходящие варианты!

**Подсказка:** если все варианты ответов являются неверными, то не отмечайте ни один из них и нажимайте кнопку "Отправить"/"Submit".

Выберите все подходящие ответы из списка

Верно решили 27 188 учащихся  
Из всех попыток 25% верных

✓ Всё получилось!

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](#).

- ☐ variab\$\$le
- ☒ variable
- ☐ var@iable
- ☒ VARiable
- ☒ variable123
- ☒ \_variable
- ☒ \_\_variable

Следующий шаг

Решить снова

Рис. 2.8: Задание 8

Просто написан скрипт, я не знаю как еще это комментировать... (рис. 2.9).

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [arguments.sh](#).

Напишите скрипт на bash, который принимает на вход два аргумента и выводит на экран строку следующего вида:

```
Arguments are: $1=первый_аргумент $2=второй_аргумент
```

Например, если ваш скрипт называется `./script.sh`, то при запуске его `./script.sh one two` на экране должно появиться:

```
Arguments are: $1=one $2=two
```

а при запуске `./script.sh three four` будет:

```
Arguments are: $1=three $2=four
```

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

Напишите программу. Тестируется через stdin → stdout

✓ Так точно!

Верно решили **25 053** учащихся  
Из всех попыток **41%** верных

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
5 echo "Arguments are: \ $1=$var1 \ $2=$var2"
```

Рис. 2.9: Задание 9

Ответ получен практическим путём. (рис. 2.10).

Вы можете вписать вместо "..." (внутри `[[ ]]` и **не забудьте про пробелы** после `[[` и перед `]]`!) любое из перечисленных ниже условий. Однако мы просим вас выбрать только те из них, при которых `echo` напечатает на экран `True` вне зависимости от того, с какими параметрами был запущен ваш скрипт и какие в нем есть переменные.

Например, условие `0 -eq 0` **подходит**, т.к. ноль всегда равен нулю вне зависимости от аргументов и переменных внутри скрипта и на экран будет напечатано `True`. В то же время условие `$var1 -eq 0` **не подходит**, так как в переменной `var1` как может быть записан ноль (тогда будет напечатано `True`), так его может и не быть (тогда ничего напечатано не будет).

**Примечание:** если вы планируете проверять варианты ответов у себя в терминале, обратите внимание на то, что содержащие символ `$` тексты *могут* изменяться при копировании — не забудьте отредактировать их в соответствии с изображением на экране. Это связано с особенностями написания `$` в некоторых видах заданий на Stepik.

Выберите все подходящие ответы из списка

Верно решили **23 158** учащихся  
Из всех попыток **16%** верных

☒ 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](#).

- ☒ -n \$0
- ☒ 5 -ge 5
- ☒ -s \$0
- ☐ -n \$1
- ☐ \$# -gt 0
- ☐ -z " "

Рис. 2.10: Задание 10

Выполнено, пораскинув мозгами :) (рис. 2.11).

Посмотрите на фрагмент bash-скрипта:

```
if [[ $var -gt 5 ]]
then
  echo "one"
elif [[ $var -lt 3 ]]
then
  echo "two"
elif [[ $var -eq 4 ]]
then
  echo "three"
else
  echo "four"
fi
```

Какие строки и в какой последовательности он выведет на экран, если сначала этот скрипт запустили задав переменную **var=3**, а затем запустили еще раз, но уже с **var=5**.

Выберите один вариант из списка

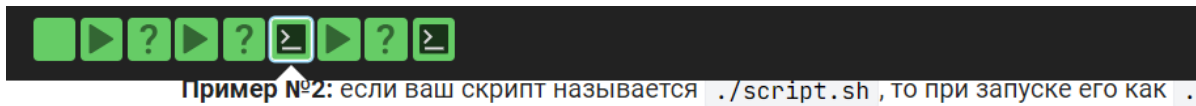
☒ Здорово, всё верно.

Верно решили **25 138** учащихся  
Из всех попыток **64%** верных

- ☐ Сначала four, потом one
- ☐ Сначала two, потом four
- ☐ Сначала one, потом two
- ☒ Сначала four, потом four

Рис. 2.11: Задание 11

Также, как и в задании 10... Нет комментариев по простому написанию скрипта. (рис. 2.12).



A lot of students

**Подсказка:** в случае проблем с решением задачи, обратите внимание [на наши рек](#)

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

✓ Отлично!

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

```
1 #!/bin/bash
2
3 # Получаем количество студентов из первого аргумента
4 students=$1
5
6 # Проверяем количество студентов и выводим соответствующее сообщение
7 if [[ "$students" -eq 0 ]]; then
8     echo "No students"
9 elif [[ "$students" -eq 1 ]]; then
10    echo "1 student"
11 elif [[ "$students" -eq 2 ]]; then
12    echo "2 students"
13 elif [[ "$students" -eq 3 ]]; then
14    echo "3 students"
15 elif [[ "$students" -eq 4 ]]; then
16    echo "4 students"
17 else
18    echo "A lot of students"
19 fi
20
```

Рис. 2.12: Задание 12

Скрипт был запущен на собственной виртуальной машине. (рис. 2.13).

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [loops1.sh](#), [loops2.sh](#).

Посмотрите на фрагмент bash-скрипта:

```
for str in a , b , c_d
do
    echo "start"
    if [[ $str > "c" ]]
    then
        continue
    fi
    echo "finish"
done
```

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

Выберите один вариант из списка

☒ Верно. Так держать!

Верно решили **24 582** учащихся  
Из всех попыток **45%** верных

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

Рис. 2.13: Задание 13

Также, как и в заданиях 10 и 12... (рис. 2.14).



✓ Так точно!

Из вс

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 своё решение с другими на [форуме решений](#).

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

Рис. 2.14: Задание 14

Ответ получен практическим путём. (рис. 2.15).

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [math1.sh](#), [math2.sh](#).

Какие(ая) из предложенных ниже инструкций увеличат значение переменной `a` на значение переменной `b`? Например, если в `a` было записано 10, в `b` было 5, то в `a` должно записаться 15.

Выберите **все подходящие** варианты!

**Примечание:** если вы планируете проверять варианты ответов у себя в терминале, обратите внимание на то, что содержащие символ `$` тексты *могут* изменяться при копировании — не забудьте отредактировать их в соответствии с изображением на экране. Это связано с особенностями написания `$` в некоторых видах заданий на Stepik.

**Подсказка:** обратите особое внимание на кавычки и **пробелы**, они могут как принципиально изменить команду, так и ни на что не повлиять (в зависимости от команды и контекста)!

Выберите все подходящие ответы из списка

☒ Здорово, всё верно.

Верно решили **22 116** учащихся  
Из всех попыток **20%** 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](#).

- ☐ `let a = a + b`
- ☐ `let "a+=b"`
- ☒ `let "a+=b"`
- ☐ `a+=$b`
- ☐ `a=$a+$b`

Рис. 2.15: Задание 15

Скрипт был запущен на собственной виртуальной машине. (рис. 2.16).

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [programs.sh](#).

Пусть вы находитесь в директории `/home/bi/Documents/` и запускаете в ней скрипт следующего содержания:

```
#!/bin/bash
cd /home/bi/
echo "`pwd`"
```

Что в этом случае выведет команда `echo` на экран?

Выберите один вариант из списка

Верно решили 23 677 учащихся  
Из всех попыток 51% верных

✓ Хорошая работа.

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

Следующий шаг

Решить снова

Рис. 2.16: Задание 16

Ответ получен благодаря просмотру видео перед заданием. (рис. 2.17).

Мы рассказали, что можно проверить код возврата внешней программы прямо в конструкции `if` при помощи `if `program options arguments`` (действия внутри `if` выполняются, если программа закончилась с кодом 0). Однако это не всегда правда! Если запуск внешней программы выводит что-то в `stdout`, то в проверку `if` поступит именно этот вывод, а не код возврата! Вы можете убедиться в этом, написав простой `bash`-скрипт с использованием, например, `if `pwd``.

Однако как быть, если хочется всё-таки запустить программу `program`, которая пишет что-то в `stdout` и потом выполнить какие-то действия если ее код возврата равен 0? Выберите все верные утверждения или правильно работающие конструкции `if`.

**Примечание:** во всех вариантах ответов, где есть кавычка, используется именно косая кавычка (```), а не обычная (`'`) или двойная (`"`).

Выберите все подходящие ответы из списка

Верно решили 21 426 учащихся  
Из всех попыток 20% верных

✓ Правильно, молодец!

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](#).

- ☐ `if [[ `program` -eq 0 ]]`
- ☐ Ничего сделать нельзя
- ☒ Сначала запустить `program`, затем `if [[ $? -eq 0 ]]`
- ☒ `if `program > some_file.txt``
- ☐ Сначала `var=`program``, затем `if [[ $var -eq 0 ]]`

Рис. 2.17: Задание 17

Скрипт был запущен на собственной виртуальной машине. (рис. 2.18).

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [functions1.sh](#), [functions2.sh](#).

Посмотрите на функцию из bash-скрипта:

```
counter () # takes one argument
{
  local let "c1+=1"
  let "c2+=${1}*2"
}
```

Впишите в форму ниже **строку**, которую выведет на экран команда `echo "counters are $c1 and $c2"` если она находится в скрипте **после десяти вызовов** функции `counter` с параметрами сначала 1, затем 2, затем 3 и т.д., последний вызов с параметром 10.

**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



Правильно.

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

Рис. 2.18: Задание 18

Просто написанный скрипт... (рис. 2.19).



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

Рис. 2.19: Задание 19

21

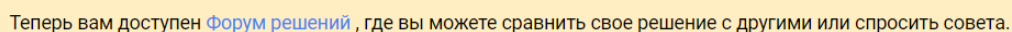
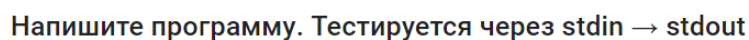


Рис. 2.20: Задание 20

22

Пусть в директории `/home/bi` лежат файлы `Star_Wars.avi`, `star_trek OST.mp3`, `STARS.txt`, `stardust.mpeg`, `Eddard_Stark_biography.txt`.

Отметьте все файлы, которые **найдет** команда `find /home/bi -iname "star*"`, но **НЕ найдет** команда `find /home/bi -name "star*" ?`

Выберите все подходящие ответы из списка

Верно решили **20 547** учащихся  
Из всех попыток **36%** верных

☒ Хорошие новости, верно!

- ☐ `star_trek OST.mp3`
- ☐ `stardust.mpeg`
- ☐ `Eddard_Stark_biography.txt`
- ☒ `STARS.txt`
- ☒ `Star_Wars.avi`

Следующий шаг

Решить снова

Рис. 2.21: Задание 21

Ответ получен практическим путем. (рис. 2.22).

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

Выберите все подходящие ответы из списка

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

☒ Правильно.

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](#).

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

Следующий шаг

Решить снова

Рис. 2.22: Задание 22

Файлы были в точности воспроизведены в собственной виртуальной машине и использованы команды, поэтому таков ответ был получен (рис. 2.23).

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

```
/home/bi/  
├── dir1  
│   ├── file1  
│   └── dir2  
│       ├── file2  
│       └── dir3  
│           └── file3
```

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

Выберите один вариант из списка

✓ Здорово, всё верно.

Верно решили 20 711 учащихся  
Из всех попыток 41% верных

- ☐ Все кроме file1
- ☐ Только file2
- ☐ Только file3
- ☒ Все кроме file3
- ☐ Все кроме file2

Рис. 2.23: Задание 23

Ответ был получен практическим путём. (рис. 2.24).

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

✓ Хорошая работа.

Верно решили 20 237 учащихся  
Из всех попыток 41% верных

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

Следующий шаг

Решить снова

Рис. 2.24: Задание 24

Маска, данная по условию, подходит под все утверждения. (рис. 2.25).



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

Выберите все подходящие ответы из списка

☒ Прекрасный ответ.

Верно решили **18 768** учащихся  
Из всех попыток **23%** 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](#).

- ☒ I prefer Kubuntu
- ☒ The best OS is Xubuntu
- ☒ Lubuntu is better than Ubuntu
- ☒ Mac OS X, Windows, Ubuntu
- ☒ Hmm, XKUbuntu
- ☒ Linux is not always Ubuntu

Следующий шаг

Решить снова

Рис. 2.25: Задание 25

Ответ был получен практическим путём. (рис. 2.26).

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

Выберите один вариант из списка

☒ Хорошие новости, верно!

Верно решили **19 784** учащихся  
Из всех попыток **39%** верных

- ☒ 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`
- ☐ На экран ничего не напечатается

Следующий шаг

Решить снова

Рис. 2.26: Задание 26

Просто написанная команда, которая подходит под требования. (рис. 2.27).



Был установлен gnuplot и ответ получен практическим путём. (рис. 2.29).

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** у построенного **ряда данных** и **сколько** будет нарисовано **точек** на графике?

Выберите один вариант из списка

☒ Верно. Так держать!

Верно решили 17 975 учащихся  
Из всех попыток 32% 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](#).

- ☐ Название – первое значение из первого столбца, нарисовано 10 точек
- ☐ Название – первое значение из второго столбца, нарисовано 10 точек
- ☒ Название – первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)
- ☐ Название "noname", нарисовано 10 точек
- ☐ Название "data.csv' using 1:2", нарисовано 10 точек

Следующий шаг

Решить снова

Рис. 2.29: Задание 29

Просто написанная команда, которая подходит под требования. (рис. 2.30).

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 пишете 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 учащихся  
Из всех попыток 44% верных

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

Рис. 2.30: Задание 30

Отредактированный код был получен практическим путем, методом проб и ошибок :) (рис. 2.31).

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



Хорошая работа.

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
```

Рис. 2.31: Задание 31

Проанализированы команды и выбран правильный ответ. (рис. 2.32).

Какая команда(ы) установят файлу `file.txt` права доступа `rw-rw-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

Верно решили **16 484** учащихся  
Из всех попыток **21%** верных

✓ 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 a+wx file.txt; chmod o-wx file.txt; chmod g-x file.txt`
- ☐ `chmod o-wx file.txt; chmod g-x file.txt; chmod a+wx file.txt`
- ☐ `chmod 467 file.txt`
- ☒ `chmod 764 file.txt`
- ☒ `chmod u+wx file.txt; chmod g+w file.txt`
- ☒ `chmod ug+w file.txt; chmod u+x file.txt`

Следующий шаг

Решить снова

Рис. 2.32: Задание 32

Выполнено, после просмотра видео, потому что возникали трудности. (рис. 2.33).



позволят даже редактировать этот файл, но сделают они это своеобразно: через удаление оригинала и создание копии уже с нужными правами (удалять 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 описанное выше поведение. Файлы (или директории) с таким атрибутом сможет удалить 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

 Верно.

Верно решили 14 683 учащихся

Из всех попыток **15%** 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](#).

- ☒ sudo chown user dir
- ☐ sudo chmod g+w dir
- ☐ sudo chown :group dir
- ☐ chown user:group dir
- ☒ sudo chown user:group dir
- ☐ sudo chmod o+x dir

Рис. 2.33: Задание 33

Информация получена из справки о команде (рис. 2.34).

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

### Выберите все подходящие ответы из списка

✓ Всё получилось!

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 слов
- ☐ Количество определенных букв (например, количество букв "А")
- ☒ Размер файла в байтах
- ☐ Количество предложений
- ☒ Количество символов

Следующий шаг

Решить снова

Рис. 2.34: Задание 34

Выявлено практическим путем и с использованием справки по команде. (рис. 2.35).

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 которой 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

✓ Правильно.

Верно решил **16 381** учащихся  
Из всех попыток **53%** верных

`du -h -s`

Следующий шаг

Решить снова

Рис. 2.35: Задание 35



Самое простенькое задание из всего блока, тут все ясно :) (рис. 2.36).

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

✓ Правильно, молодец!

Верно решили **16 720** учащихся  
Из всех попыток **40%** верных

```
mkdir dir{1..3}
```

Следующий шаг

Решить снова

Рис. 2.36: Задание 36

## 3 Выводы

Мы прошли третий и завершающий модуль внешнего курса “Введение в Linux” и освежили уже имеющиеся знания, а также получили некоторые новые.