

Продвинутые темы

Часть 3

Зюков Александр Валерьевич

Содержание

1	Цель работы	6
2	Выполнение лабораторной работы	7
3	Выводы	44

Список иллюстраций

2.1	sc1	7
2.2	sc2	8
2.3	sc3	9
2.4	sc4	10
2.5	sc5	11
2.6	sc6	12
2.7	sc7	13
2.8	sc8	14
2.9	sc9	15
2.10	sc10	16
2.11	sc11	17
2.12	sc12	18
2.13	sc13	19
2.14	sc14	20
2.15	sc15	21
2.16	sc16	22
2.17	sc17	23
2.18	sc18	24
2.19	sc19	25
2.20	sc20	26
2.21	sc21	27
2.22	sc22	28
2.23	sc23	29
2.24	sc24	30
2.25	sc25	31
2.26	sc26	32
2.27	sc27	33
2.28	sc28	34
2.29	sc29	34
2.30	sc30	35
2.31	sc31	35
2.32	sc30	36
2.33	sc33	37
2.34	sc34	38
2.35	sc35	39
2.36	sc36	40
2.37	sc37	41

2.38 sc38	41
2.39 sc39	42
2.40 sc40	43
2.41 sc41	43

Список таблиц

1 Цель работы

Прохождение третьей части внешнего курса stepik

2 Выполнение лабораторной работы

Задание 1

Чтобы выйти из редактора vim нужно нажать ":" , затем "q", затем "Enter"

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

☒ Хорошие новости, верно!

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

[Ваши решения](#) Вы получили: **1 балл**

Рис. 2.1: sc1

Задание 2

Главное различие между word и WORD чтобы попасть в конец строки нужно совершить меньше нажатий на W чем на w

При перемещении в vim "по словам" есть небольшая разница в том, используем мы маленькую (w, e, b) или большую (W, E, B) букву. Первые перемещают нас по "словам" (word), а вторые по "большим словам" (WORD). Посмотрите справку по этим перемещениям и разберитесь в чем заключается разница между word и WORD.

А для того, чтобы убедиться, что вы разобрались, отметьте ниже **все верные** утверждения про следующую строку:

```
Strange_ TEXT is_here. 2=2 YES!
```

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

Подсказка: чтобы вызвать **vim-справку** по, например, перемещению `w`, нужно открыть vim и ввести команду `:help w`. Вы попадете в то место справки, где описано это перемещение, а так как все перемещения описаны рядом, то двигаясь по тексту вверх и вниз можно прочитать и про `e` и про `b` и, самое главное, про word и WORD. Кроме того, можно вызвать сразу справку по термину word при помощи `:help word`. Чтобы закрыть справку, нужно ввести команду `:q`.

Выберите все подходящие ответы из списка

Верно решили **25 385** учащихся
Из всех попыток **20%** верных

☒ 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](#).

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

Следующий шаг

Решить снова

Рис. 2.2: sc2

Задание 3

Чтобы отредактировать данный текст one two three four five в three four four four five нам понадобится

1)d2w\$\$bifour four <> 2)ddithree four four four five <> 3)d2wwifour four <>

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

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

- ☒ d2w\$bifour four <Esc>
- ☐ x2wwywPp
- ☒ ddithree four four four five<Esc>
- ☐ d2dwywPp
- ☒ d2wwifour four <Esc>
- ☐ d2wwywpp

Следующий шаг

Решить снова

Рис. 2.3: sc3

Задание 4

Чтобы заменить слово Windows на Linux нам понадобится команда
сама команда :%s/Windows/Linux

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

✓ Так точно!

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

`:%s/Windows/Linux`

Следующий шаг

Решить снова

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

Рис. 2.4: sc4

Задание 5

Существует третий режим vim - режим выделения

- 1)Режим выделения открывается из нормального режима по нажатию “v” 2)В режиме выделения можно использовать команды d (удалить) и у (скопировать)
- 3)Выйти из режима выделения можно, нажав клавишу Esc два раза 4)В режиме выделения можно использовать команды перемещения (например, W, e, \$, и др.)

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

☒ Верно. Так держать!

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](#).

- ☐ Режим выделения открывается из любого другого режима по нажатию "v"
- ☒ Режим выделения открывается из нормального режима по нажатию "v"
- ☒ В режиме выделения можно использовать команды d (удалить) и у (скопировать)
- ☐ Чтобы выйти из режима выделения, нужно ввести :q
- ☒ Выйти из режима выделения можно, нажав клавишу Esc два раза
- ☒ В режиме выделения можно использовать команды перемещения (например, W, e, \$, и др.)

Следующий шаг

Решить снова

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

Рис. 2.5: sc5

Задание 6

Только С

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

✓ Хорошая работа.

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

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: ***

Рис. 2.6: sc6

Задание 7

После изучения скрипта абсолютный путь будет выглядеть `/home/bi/file1.txt`

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

☒ Здорово, всё верно.

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

- ☐ /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**

Рис. 2.7: sc7

Задание 8

Именами переменных в bash могут быть:

1)variable123 2)variable_123 3)VARiable 4)__variable

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

☒ 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](#).

- ☒ variable123
- ☐ vari/able
- ☒ VARiable
- ☐ var-i-able
- ☒ variable_123
- ☐ variab\$\$le
- ☒ __variable

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 2.8: sc8

Задание 9

Напишем скрипт на bash который принимает на вход два аргумента и выводит на экран строку следующего вида - Arguments are: \$1=первый_аргумент \$2=второй_аргумент

Сам код

```
var1=$1 var2=$2
```

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

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

✓ Всё получилось!

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 var1=$1
2 var2=$2
3
4 echo "Arguments are: \${1=$var1} \${2=$var2}"
5
6
7
8
9
```

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: ...

Рис. 2.9: sc9

Задание 10

1)-z "" 2)\$var1 ==

var2||

var1 != \$var23)# -ge 0 4) -n \$0 5)!(4 -le 3)

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [branching1.sh](#).

Предположим, вы пишете скрипт на `bash` и хотите использовать в нем конструкцию `if` в следующем фрагменте:

```
if [[ ... ]]
then
  echo "True"
fi
```

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

☒ Правильно.

Верно решили **23 158** 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](#).

- ☒ `-z ""`
- ☒ `$var1 == $var2 || $var1 != $var2`
- ☒ `$# -ge 0`
- ☒ `-n $0`
- ☐ `$var1 == $var2 && $var1 != $var2`
- ☒ `! (4 -le 3)`

Следующий шаг

Решить снова

Ваши решения Вы получили: **1 балл**

Рис. 2.10: sc10

Задание 11

Сначала `var=3` затем `var=5` у нас он выведет Сначала four, потом four

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [branching2.sh](#), [branching3.sh](#).

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

☒ Абсолютно точно.

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

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

Следующий шаг

Решить снова

Ваши решения Вы получили: ...

Рис. 2.11: sc11

Задание 12

Напишем скрипт на bash который принимает на вход один аргумент который будет обозначать число студентов в аудитории

Сам скрипт `if [[$1 -eq 1]]; then echo "$1 student" elif [[$1 -gt 1 && $1 -le 4]]; then echo "$1 students" elif [[$1 -ge 5]]; then echo "A lot of students" else echo "No students" fi`

Напишите скрипт на bash, который принимает на вход один аргумент (целое число от 0 до бесконечности), который будет обозначать число студентов в аудитории. В зависимости от значения числа нужно вывести разные сообщения.

Соответствие входа и выхода должно быть таким:

```
0 --> No students
1 --> 1 student
2 --> 2 students
3 --> 3 students
4 --> 4 students
5 и больше --> A lot of students
```

Примечание а): выводить нужно только строку справа, т.е. "-->" выводить не нужно.

Примечание б): в последней строке слово "lot" с маленькой буквы!

Примечание 2: в этой и всех последующих задачах на написание скриптов, если не указано явно, что нужно **проверять вход** (например, что он будет именно числом и именно от 0 до бесконечности), то этого делать **не нужно!**

Пример №1: если ваш скрипт называется `./script.sh`, то при запуске его как `./script.sh 1` на экране должно появиться:

```
1 student
```

Пример №2: если ваш скрипт называется `./script.sh`, то при запуске его как `./script.sh 5` на экране должно появиться:

```
A lot of students
```

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

Напишите программу. Тестируется через stdin → stdout

✓ Отлично!

Верно решили **23 310** учащихся
Из всех попыток **38%** верных

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

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: ...

Рис. 2.12: sc12

Задание 13

После запуска скрипта у на выведится 5 start раз и 4 раза finish

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

☒ Так точно!

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

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: ***

Рис. 2.13: sc13

Задание 14

Напишем скрипт на bash который будет определять в какую возрастную группу попадают пользователи скрипт будет представлен на скрине

Напишите скрипт на bash, который будет определять в какую возрастную группу попадают пользователи. При запуске скрипт 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](#).

Рис. 2.14: sc14

```

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

```

Следующий шаг

Решить снова

Ваши решения Вы получили: **4 балла**

Рис. 2.15: sc15

Задание 15

Из ниже предложенных вариантов ответа увеличивает число a на значение b

1)let $a=$

$a+$

b 2)let $a=a+b$ 3)let “ $a=$

$a+$

b” 4)let “ $a = a + b$ ”

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

Верно решили **22 116** 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](#).

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 2.16: sc16

Задание 16

Команда выведет `/home/bi`

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

☒ Хорошие новости, верно!

Верно решили **23 677** учащихся
Из всех попыток **51%** верных

- ☐ ``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

[Ваши решения](#) Вы получили: **1 балл**

Рис. 2.17: sc17

Задание 17

Ответ

1)if program > some_file.txt 2)Сначала запустить program, затем if [[\$? -eq 0]]

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

☒ Здорово, всё 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](#).

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

980 278

Шаг 6

Следующий шаг >

Рис. 2.18: sc18

Задание 18

Команда `echo "counters are $c1 and $c2"` выведет на экран `counters are and 110`

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



Абсолютно точно.

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

Рис. 2.19: sc19

Задание 19

Напишем скрипт на bash который будет искать наибольший общий делитель

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 отличие 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. 2.20: sc20

Напишите программу. Тестируется через stdin → stdout

✓ Прекрасный ответ.

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 while [ true ]
2 do
3     read n1 n2
4     if [ -z $n1 ]; then
5         echo "bye"
6         break
7     else
8         gcd () {
9             remainder=1
10            if [ $n2 -eq 0 ]
11            then
12                echo "bye"
13            fi
14            while [ $remainder -ne 0 ]
15            do
16                remainder=$((n1%n2))
17                n1=$n2
18                n2=$remainder
19            done
20        }
21        gcd $1 $2
22        echo "GCD is $n1"
23    fi
24 done
25
26
```

Следующий шаг

Решить снова

Рис. 2.21: sc21

Задание 20

Напишем калькулятор на bash

Сам скрипт

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 операнд (целое число), операция (одна из `"+"`, `"-"`, `"*"`, `"/"`, `"%"`, `"**"`) и второй операнд (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](#).

Рис. 2.22: `sc22`

Напишите программу. Тестируется через stdin → stdout

✓ Отличное решение!

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

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

```
1 while true
2 do
3     read num1 op num2
4     if [[ $num1 == "exit" ]]
5     then
6         echo "bye"
7         exit
8     elif [[ ! $num1 =~ ^[0-9]+$ || ! $num2 =~ ^[0-9]+$ ]]
9     then
10        echo "error"
11        exit
12    else
13        case $op in
14            "+") let "res = num1 + num2" ;;
15            "-") let "res = num1 - num2" ;;
16            "/" let "res = num1 / num2" ;;
17            "*" let "res = num1 * num2" ;;
18            "%" let "res = num1 % num2" ;;
19            "**") let "res = num1 ** num2" ;;
20            *) echo "error" ; exit ;;
21        esac
22        echo "$res"
23    fi
24 done
```

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **5 баллов**

Рис. 2.23: sc23

Задание 21

Команда `find /home/bi -iname "star"` *но не найдет* `find /home/bi -name "star"`

1) STARS.txt 2) Star_Wars.avi

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

☒ Всё правильно.

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

☐ Eddard_Stark_biography.txt

☐ stardust.mpeg

☒ STARS.txt

☐ star_trek OST.mp3

☒ Star_Wars.avi

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 2.24: sc24

Задание 22

Команды `-path` и `-name` если заменить в команде `-name` на `-path` то результат поиска иногда может остаться таким же

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, 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](#).

- ☐ Опции `-path` и `-name` всегда работают одинаково
- ☐ 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
- ☒ Если заменить в команде поиска `-name`, на `-path`, то результат поиска иногда может остаться таким же
- ☐ Если заменить в команде поиска `-name`, на `-path`, то результат поиска всегда останется неизменным

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: ...

👍 786 👎 246

Шаг 4

Следующий шаг >

Рис. 2.25: sc25

Задание 23

Из данных трех файлов (`file1`, `file2`, `file3`) будут найдены все кроме `file3`

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

☒ Отлично!

Верно решили **20 711** учащихся
Из всех попыток **41%** верных

- ☐ Все три файла
- ☒ Все кроме file3
- ☐ Только file2
- ☐ Только file1
- ☐ Ни один файл найден не будет

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: ...

Рис. 2.26: sc26

Задание 24

После выполнения всех команда у нас будет `results.txt` будет одинакового размера во всех случаях

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

Верно решили **20 237** учащихся
Из всех попыток **41%** верных

☒ Так точно!

- ☐ `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`
- ☐ `grep -C 1 "word" file.txt > results.txt`
- ☐ Все, кроме `grep "word" file.txt > results.txt`

Следующий шаг

Решить снова

Ваши решения Вы получили: ...

Рис. 2.27: sc27

Задание 25

После команды `grep -E "[xklXKL]?[uU]buntu$" text.txt` у нас выведется

1)Linux is not always Ubuntu 2)Hmm, XKLUbuntu 3) Mac OS X, Windows, Ubuntu 4)I prefer Kubuntu 5)Lubuntu is better than Ubuntu 6)The best OS is Xubuntu

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

Верно решили **18 768** учащихся
Из всех попыток **23%** верных

✓ 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](#).

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: ...

Рис. 2.28: sc28

Задание 26

Если в команде `sed -n "/[a-z]*/p" text.txt` не указывать опцию `-n` у нас будет каждая строка выведена два раза

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

Выберите один вариант из списка

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

✓ Верно.

- ☐ 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
- ☐ Появится сообщение об ошибке

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: ...

Рис. 2.29: sc29

Задание 27

Ответ `sed "s/ [A-Z]+[A-Z]+ / abbreviation /g" input.txt > edited.txt`

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 повторов (т.е. *, +, ? и {}). Однако эту "несовместимость" легко исправить указав при помощи "(" и ")" какой из символов к чему относится! Например, регулярное выражения `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

Верно решили **16 632** учащихся
...

Рис. 2.30: sc30

Напишите текст

✓ Всё правильно.

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

```
sed "s/ [A-Z]+[A-Z]+ / abbreviation /g" input.txt > edited.txt
```

Следующий шаг

Решить снова

Ваши решения Вы получили: **3 балла**

Рис. 2.31: sc31

Задание 28

При запуске `gnuplot` можно указать опцию `-p, -persist`

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

☒ Абсолютно точно.

Верно решили **18 785** учащихся
Из всех попыток **51%** верных

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 2.32: sc30

Задание 29

После выполнения кода у нас будет построена и нарисована Название – первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)

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

☒ Всё получилось!

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 2.33: sc33

Задание 30

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

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 OX (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

✓ Отличное решение!

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)
```

Следующий шаг

Решить снова

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

Рис. 2.34: sc34

Задание 31

После изменения файла мы получим

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

✓ Верно.

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

[Ваши решения](#) Вы получили: **3 балла**

Рис. 2.35: sc35

Задание 32

Команды установят файлу `file.txt` права доступа `rw-rw-r--`

1) `chmod u+wx file.txt; chmod g+w file.txt` 2) `chmod a+wx file.txt; chmod o-wx file.txt; chmod g-x file.txt` 3) `chmod 764 file.txt` 4) `chmod ug+w file.txt; chmod u+x file.txt`

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

✔ Отлично!

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 u+wx file.txt; chmod g+w file.txt`
- ☒ `chmod a+wx file.txt; chmod o-wx file.txt; chmod g-x file.txt`
- ☐ `chmod 777 file.txt`
- ☒ `chmod 764 file.txt`
- ☒ `chmod ug+w file.txt; chmod u+x file.txt`
- ☐ `chmod 467 file.txt`

Следующий шаг

Решить снова

Ваши решения Вы получили: **1 балл**

Рис. 2.36: sc36

Задание 33

Ответ

1)sudo chown user:group dir 2)sudo chmod o+w dir 3)sudo chown user dir 4)sudo
chmod a+w dir

Предположим вы использовали команду `sudo` для создания директории `dir`. По умолчанию для `dir` были выставлены права доступа `rw-r-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, `v` 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!**

Рис. 2.37: sc37

Выберите все подходящие ответы из списка

☒ Здорово, всё 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:group dir`
- ☒ `sudo chmod o+w dir`
- ☐ `sudo chown :group dir`
- ☒ `sudo chown user dir`
- ☐ `chmod o+w dir`
- ☒ `sudo chmod a+w dir`

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 2.38: sc38

Задание 34

Характеристики файла можно посчитать с использованием команды `wc`. 1)Количество символов 2)Количество слов 3)Количество строк

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

Выберите все подходящие ответы из списка

✔ Отлично!

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 определенных букв (например, количество букв "А")
- ☒ Количество символов
- ☒ Количество слов
- ☐ Количество предложений
- ☒ Количество строк

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: 1 балл

Рис. 2.39: sc39

Задание 35

Напишем форму которая выведет сколько места на диске занимает текущая директория `du -h -s`

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

✓ Отлично!

Верно решил 16 381 учащийся
Из всех попыток 53% верных

```
du -h -s
```

Следующий шаг

Решить снова

Ваши решения Вы получили: 2 балла

Рис. 2.40: sc40

Задание 36

Команда которая может в текущей директории создать 3 поддиректории с именами `mkdir dir{1,2,3}`

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

✓ Прекрасный ответ.

Верно решили 16 720 учащихся
Из всех попыток 40% верных

```
mkdir dir{1,2,3}
```

Следующий шаг

Решить снова

Ваши решения Вы получили: 2 балла

Рис. 2.41: sc41

3 Выводы

Я завершил курс stepik