

Выполнение внешнего курса

Блок №3. Продвинутые темы

Сергеев Д. О.

9 мая 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Сергеев Даниил Олегович
- Студент
- Направление: Прикладная информатика
- Российский университет дружбы народов
- 1132246837@pfur.ru

Цель работы

Получить основные навыки работы с ОС Linux путем прохождения внешнего курса на образовательной платформе stepik.

Задание

- Пройти курс.
- Получить сертификат.
- Записать видео по каждому разделу.
- Записать итоговую презентацию по каждому этапу.
- Написать отчёт по прохождению контрольных мероприятий по каждому разделу.

Выполнение лабораторной работы

Приступим к выполнению третьего блока заданий внешнего курса – Продвинутым темам.

Текстовый редактор vim

1. Вопрос 1-й: Из только что открытого в vim файла можно выйти перейдя в командный режим с помощью : и написав q, затем ENTER. (если не производилось изменение файла)

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

Верно решено 32 523 учащиеся
Из всех попыток 69% верных

☒ Так точно!

☐ "q"

☐ "Q"

☒ ":", затем "q", затем "Enter"

☐ "Ctrl", затем "q"

☐ "q", затем "Enter"

Следующий шаг Решить снова

Ваши результаты Вы получили: 1 балл

Рис. 1: Задание №1. Условие и ответ

2. Вопрос 2-й: Введем тестовую строку и проведем на ней указанные операции, выберем правильные ответы.

При перемещении в vim "по словам" есть небольшая разница в том, используем мы маленькую (и, 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 `и`, нужно открыть vim и ввести команду `:he lp и`. Вы попадете в то место справки, где описано это перемещение, а так как все перемещения описаны рядом, то двигаться по тексту вверх и вниз можно прочитав и про `е` и про `б` и, самое главное, про word и WORD. Кроме того, можно вызвать сразу справку по термину word при помощи `:he lp word`. Чтобы закрыть справку, нужно ввести команду `:q`.

Выберите все подходящие ответы из списка

☒ Хорошая работа.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [стиментарии](#), ответив на их вопросы, или сравнить свой результат с другими на [форуме решений](#).

☒ После 10 нажатий на W курсор окажется там же, где бы он был после 10 нажатий на и
☒ В этой строке 9 "слов" (word)
☐ В этой строке 5 "слов" (word)
☐ Нажимая только на и, нельзя переместить курсор на ""
☒ Нажимая только на W, нельзя переместить курсор на ""
☐ В этой строке 12 "слов" (word)

[Следующий шаг](#) [Решить снова](#)

Ваше решение: **Вы получили: 1 балл**

Рис. 2: Задание №2. Условие и ответ

3. Вопрос 3-й: Проверим каждую комбинацию в редакторе vim. Укажем правильные ответы.

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 fLve
```

Какие(ой) из предложенных ниже **наборов нажатий клавиш** выполнит 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. Так держать!

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

☒ ddthree four four five<Esc>
☐ d2wyyPp
☒ xxxxxxxxxxxxwyyPp
☒ d2wytfour four<Esc>
☒ d2w\$tfour four<Esc>
☒ d2wyywPp

[Следующий шаг](#) [Решить снова](#)

Ваше решение: Вы получили: 1 балл

Верно решили 23 655 учащихся
Из всех попыток 16% верных

Рис. 3: Задание №3. Условие и ответ

4. Вопрос 4-й: В командном режиме можно совершить замену первого вхождения слова в строке. Для этого укажем строку

:%s/Windows/Linux, где:

- % – диапазон (всего файла)
- s – команда замены
- Windows – то что заменяем
- Linux – то на что заменяем

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

Верно. Верно решил 24 631 человек
Из всех попыток 97% верных

`:%s/Windows/Linux`

Следующий шаг Решить снова

Ваше решение Вы получили 2 балла

Рис. 4: Задание №4. Условие и ответ

5. Вопрос 5-й: Проверим каждую опцию из предложенного списка, выберем правильные ответы.

Мы совсем не рассказывали 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 решили **22 497** учащихся
Из всех попыток **25%** верных

✓ 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 при помощи команды :visual
- ☐ Режим выделения открывается из любого другого режима по нажатию "v"
- ☒ 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

Следующий шаг Решить снова

Ваши решения Вы получили **2 балла**

Рис. 5: Задание №5. Условие и ответ

Скрипты на bash: основы

1. Вопрос 6-й: Каждая оболочка хранит свойфайл независимо, поэтому в последней оболочке будут лишь команды C1, C2, C3.

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 попробуете при помощи стрелочек вверх/вниз перемещаться по истории набранных команд, то команды из какого набора(ов) будут появляться?

Выберите один вариант из списка

✓ Прекрасный ответ.

Верно решили 30 266 учащихся
Из всех попыток 95% верных

- ☐ Только из набора B
- ☐ Из наборов A и C
- ☐ Из наборов B и C
- ☒ Только из набора C
- ☐ Только из набора A

Следующий шаг Решить снова

Ваша реакция Вы получили: 1 балл

Рис. 6: Задание №6. Условие и ответ

2. Вопрос 7-й: Исходя из скрипта, мы перейдем в домашнюю директорию, после в ней создастся файл file1.txt и команда отправит нас в каталог /home/bi/Desktop. В итоге выбираем второй вариант.

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

Верно решили 29 905 учащихся
Из всех попыток 76% верных

☒ Хорошие новости, верно!

☐ /home/bi/Desktop/file1.txt

☒ /home/bi/file1.txt

☐ /home/bi/Documents/file1.txt

☐ Ничах (файла file1.txt не будет существовать после завершения работы скрипта)

[Следующий шаг](#) [Решить снова](#)

Ваша реакция: [Вы получили 1 балл](#)

Рис. 7: Задание №7. Условие и ответ

3. Вопрос 8-й: В языке программирования bash в названия переменных допускаются нижние подчёркивания, числа и буквы, однако имена не должны начинаться с цифры. Выберем подходящие варианты ответа.

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

✔ Все 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](#).

- ☐ 123variable
- ☒ VARiable
- ☐ var.Lable
- ☐ vari/able
- ☐ var@lable
- ☐ variab\$\$\$
- ☒ _variable

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл

Рис. 8: Задание №8. Условие и ответ

4. Вопрос 9-й: Используем команду echo для вывода сообщения. Чтобы символ доллара корректно отображался, укажем обратный слеш перед ним: ...\\$1...

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: $1первый_аргумент $2второй_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

✓ Прекрасный ответ.

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

Верно решили 29 653 учащихся
Из всех попыток 41% верных

```
1 echo Arguments are: \$1=\$1 \$2=\$2
2
3
4
5
6
```

Следующий шаг Решить снова

[Ваше решение](#) Вы получили: 3 балла

echo Arguments are: \\$1=\$1 \\$2=\$2

Рис. 9: Задание №9. Условие и ответ

Скрипты на bash: ветвления и циклы

1. Вопрос 10-й: Двойные квадратные скобки – усовершенствованная версия тестовой конструкции test, позволяющая совершать сравнения и тесты. Изучим возможные опции для этой конструкции и выберем подходящие варианты.

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 `B -eq B` **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 содержимое символа `$` тексты могут измениться при копировании – не забудьте отредактировать их в соответствии с изображением на экране. Это связано с особенностями написания `$` в некоторых видах заданий на Stepik.

Выберите все подходящие ответы из списка

Верно решили 23 158 учащихся из всех попыток 10% 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](#).

☒ `x"`
☒ `n $0`
☐ `x"`
☐ `$var1 == $var2 && $var1 != $var2`
☒ `$ -ge 5`
☐ `n $1`

Следующий шаг Решить снова

Ваши решения: Вы получили 1 балл

Рис. 10: Задание №10. Условие и ответ

2. Вопрос 11-й: Задание проверяет знание опций для `[[]]`. Пройдемся по условиям с заданными параметрами и выберем правильные варианты из списка.

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

☒ Правильно, молодец!

Верно решили 25 126 человек
Из всех попыток 64% верных

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

[Следующий шаг](#) [Решить снова](#)

[Взглянуть на решение](#) Вы получили: 1 балл

Рис. 11: Задание №11. Условие и ответ

3. Вопрос 12-й: Будем считывать первый аргумент с помощью \$1. Для проходки по возможным вариантам используем оператор case.

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!

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.

Рис. 12: Задание №12. Условие

Напишите программу. Тестируется через stdin → stdout

Верно решено 23 310 учениками
Из всех попыток 38% верных

Правильно.

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

```
1 arg=$1
2 case $arg in
3 )
4     echo No students
5 ;;
6 )
7     echo 1 student
8 ;;
9 )
10    echo 2 students
11 ;;
12 )
13    echo 3 students
14 ;;
15 )
16    echo 4 students
17 ;;
18 *)
19    echo A lot of students
20 esac
21
22
23
24
25
```

Следующий шаг Решить снова

Ваше решение Выполнили: 2 балла

Рис. 13: Задание №12. Ответ


```
arg=$1
case $arg in
    0)
        echo No students
        ;;
    1)
        echo 1 student
        ;;
    2)
        echo 2 students
        ;;
    3)
        echo 3 students
        ;;
    4)
        echo 4 students
        ;;
    *)
        echo A lot of students
esac
```

4. Вопрос 13-й: Команда запуститься 5 раз, так как знак запятой тоже считается отдельным значением переменной str. Условие `[[$str > "c"]]` проверяет по длине строк, поэтому команда закончится на последнем аргументе, не выводя сообщение finish.

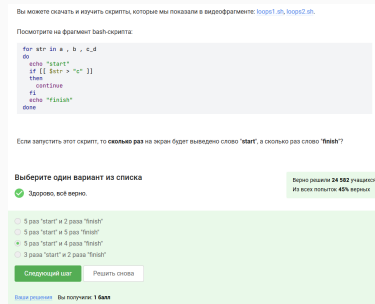


Рис. 14: Задание №13. Условие и ответ

5. Вопрос 14-й: Программу запишем в цикл while true. Сделаем отдельные переменные для имени и возраста, считаем значения через команду read, проверяя на пустую строку либо нулевой возраст.

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 "if you, 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 Petrova
enter your age:
25
Elena Petrova, your group is youth
enter your name:
bye
```

Рис. 15: Задание №14. Условие

Напишите программу. Тестируется через stdin → stdout

Верно решено 21 470 человек
Из всех попыток 22% верных

✓ Абсолютно точно.

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

Вы решаете сложную задачу, поздравляем! Вы можете помочь опытным участникам в [комментариях](#), отвечая на их вопросы, или сравнить свой решение с другими на [форум](#) решений.

```
1 name=""
2 age=""
3 while true
4 do
5     echo enter your name:
6     read name
7     if [[ -z $name ]]
8     then
9         break
10    fi
11    echo enter your age:
12    read age
13    if [[ $age -eq 0 ]]
14    then
15        break
16    fi
17
18    if [[ $age -le 16 ]]
19    then
20        echo $name, your group is child
21        elif [[ $age -gt 16 ]] && [[ $age -le 25 ]]
22    then
23        echo $name, your group is youth
24        elif [[ $age -gt 25 ]]
25    then
26        echo $name, your group is adult
27    fi
28    else
29        echo bye
30    fi
31
```

Следующий шаг

Решить ошибку

Моя попытка Из получено: 4 балла

Рис. 16: Задание №14. Ответ

Скрипты на bash: ветвления и циклы

```
name=" "  
age=" "  
while true  
do  
    echo enter your name:  
    read name  
    if [[ -z $name ]]  
    then  
        break  
    fi  
    echo enter your age:  
    read age  
    if [[ $age -eq 0 ]]  
    then
```

```
        break  
    fi  
    if [[ $age -le 16 ]]  
    then  
        echo $name, your group is child  
    elif [[ $age -gt 16 ]] && [[ $age -lt 25 ]]  
    then  
        echo $name, your group is youth  
    elif [[ $age -gt 25 ]]  
    then  
        echo $name, your group is adult  
    fi  
done  
echo bye
```

Скрипты на bash: разное

1. Вопрос 15-й: Выражения с командой `let` можно писать как в кавычках, так и без них. Во втором варианте происходит сложение (объединение) строк, а первый вариант не работает из-за неправильного операнда.

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

Какие(их) из предложенных ниже инструкций увеличат значение переменной `a` на значение переменной `b`?
Например, если в `a` было записано 10, в `b` было 3, то в `a` должно записаться 13.
Выберите **все подходящие** варианты!

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

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

Выберите все подходящие ответы из списка

Верно решили 22 116 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](#).

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

Следующий шаг Решить снова

[Ваше решение](#) Вы получили 1 балл

Рис. 17: Задание №15. Условие и ответ

2. Вопрос 16-й: Обратные одинарные ковычки позволяют выполнять команды внутри сценария оболочки, поэтому вывод будет равен пути `rwd`, указанном в файле.

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

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

```
#!/bin/bash
cd /home/bl/
echo "$rwd"
```

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

Выберите один вариант из списка

Верно решили 23 677 учащихся
Из всех попыток 51% верных

☒ Здорово, всё верно.

☐ "rwd"

☐ /home/bl/Documents

☐ Код возврата команды `rwd` (0 в случае успешного выполнения и не 0 в случае ошибок)

☒ /home/bl

☐ rwd

[Следующий шаг](#) [Решить снова](#)

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

Рис. 18: Задание №16. Условие и ответ

3. Вопрос 17-й: Достать вывод потока ошибок можно через встроенную переменную `$?` либо через запись результата в какой-нибудь файл с последующим считыванием кода выхода уже этого условия.

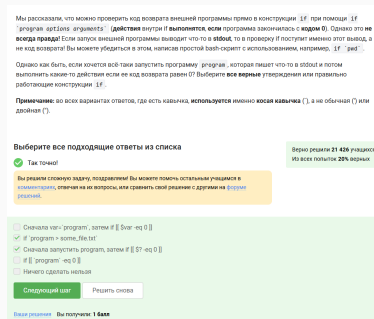


Рис. 19: Задание №17. Условие и ответ

4. Вопрос 18-й: Подвох этой задачи заключается в том, что мы пытаемся вывести значение переменной `$c1`, хотя она является локальной в теле функции `counter`. На выходе её значение будет равно пустой строке.

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 по возможности сверить свой ответ с тем, что он выведет на экран.

Напишите текст

✔️ Правильно, молодец!

Вы решили задание правильно! Вы можете помочь остальным ученикам в комментариях, отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

Верно решили 20 000 учеников из всех попыток 28% верных

counters are and 110

Следующий шаг Решить снова

Ваше решение Вы получили: 2 балла

Рис. 20: Задание №18. Условие и ответ

5. Вопрос 19-й: Напишем программу для вычисления НОД, используя алгоритм Евклида и рекурсивный вызов функции в bash.

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 ждать ввода двух натуральных чисел через пробел (для этого можно использовать `read` и указать ему два переменные – см. пример в видеорекомендации). После ввода чисел скрипт считает их НОД и выводит на экран сообщение "GCD is *определённое 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 ввёл 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 вывести 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
15 25
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. 21: Задание №19. Условие

Напишите программу. Тестируется через stdin → stdout

Верно решено 18 148 учащихся
Из всех попыток 35% верных

✓ Правильно.

Теперь вам доступен [Формат решения](#), где вы можете сравнить свое решение с другим или попросить совета.

```
1 gcd()
2 {
3     local let "M=${1}"
4     local let "N=${2}"
5     if [[ M -eq N ]]
6     then
7         return $M
8     elif [[ M -gt N ]]
9     then
10        let "M=$((M-N))"
11    else
12        let "N=$((N-M))"
13    fi
14    gcd $M $N
15}
16
17
18
19 while true
20 do
21     read num1 num2
22     if [[ -z $num1 ]] || [[ -z $num2 ]]
23     then
24         echo bye
25         break
26     fi
27     gcd $num1 $num2
28     echo GCD is $?
29 done
30
```

Следующий шаг

Решить снова

Ваше решение: Вы получили 4 балла

Рис. 22: Задание №19. Ответ

Скрипты на bash: разное

```
gcd()  
{  
    local let "M+=${1}"  
    local let "N=${2}"  
    if [[ M -eq N ]]  
    then  
        return $M  
    elif [[ M -gt N ]]  
    then  
        let "r=$M-$N"  
        gcd $r $N  
    elif [[ M -lt N ]]  
    then  
        let "r=$N-$M"
```

```
    gcd $M $r  
    fi  
}  
while true  
do  
    read num1 num2  
    if [[ -z $num1 ]] || [[ -z $num2 ]]  
    then  
        echo bye  
        break  
    fi  
    gcd $num1 $num2  
    echo GCD is $?  
done
```

6. Вопрос 20-й: Реализуем калькулятор. Будем считывать уже 3 аргумента командой read. Чтобы оператор case не путал умножение "*") и оставшиеся варианты *), поместим знак умножения в кавычки.

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 `./calc1pt.sh < input.txt`). В этом случае он должен вывести сразу все ответы на экран.

Например, если входной файл содержит 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. 23: Задание №20. Условие

Напишите программу. Тестируется через stdin → stdout

Верно. Так держать!

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

Ваше решение 16 988 учащихся
Из всех попыток 30% верных

```
1 while true
2 do
3     read num1 op num2
4     if [[ $num1 == "exit" ]]
5     then
6         echo bye
7         break
8     elif [[ "$num1" =~ ^[0-9]+$ && "$num2" =~ ^[0-9]+$ ]]
9     then
10        echo error
11        break
12    fi
13    case $op in
14        "+")
15        let "res=$num1+$num2"
16        ;;
17        "-")
18        let "res=$num1-$num2"
19        ;;
20        "*")
21        let "res=$num1*$num2"
22        ;;
23        "/")
24        let "res=$num1/$num2"
25        ;;
26        "%")
27        let "res=$num1%$num2"
28        ;;
29        "**")
30        let "res=$num1**$num2"
31        ;;
32        *)
33        echo error
34        break
35    esac
36    echo $res
37 done
```

Следующий шаг

Решить снова

Ваше решение

Вы получили: 5 баллов

Рис. 24: Задание №20. Ответ

Продвинутый поиск и редактирование

1. Вопрос 21-й: Команда `-iname` не учитывает регистр, поэтому найдутся те варианты, которые начинаются с `star` и заканчиваются любым набором в любом регистре.

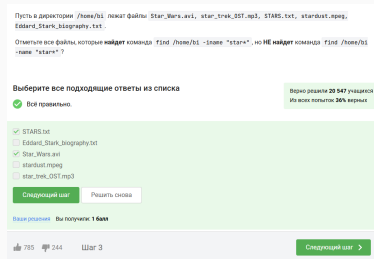


Рис. 25: Задание №21. Условие и ответ

2. Вопрос 22-й: Изучим каждый вариант ответа и выберем верные.

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

Выберите все подходящие ответы из списка

Всё получилось!

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

Верно решили 18 450 учащихся
Из всех попыток 22% верных

- ☐ Опции `-path` и `-name` всегда работают одинаково
- ☐ Опция `-path` аналогична `-name`, но игнорирует размер букв (строчные/прописные) в имени файла
- ☐ Если заменить в команде поиска `-name` на `-path`, то результат поиска всегда останется неизменным
- ☐ 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

Следующий шаг Решить снова

[Ваши решения](#) Вы получили 1 балл

Рис. 26: Задание №22. Условие и ответ

3. Вопрос 23-й: Глубина поиска начинается с текущей директории, поэтому он будет проходить в области **dir1** \leftrightarrow **dir2**, не заходя в каталог **dir3**.

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

```
graph TD
    bi["/home/bi/"] --> dir1["dir1"]
    bi --> file1["file1"]
    bi --> file2["file2"]
    bi --> file3["file3"]
    dir1 --> dir2["dir2"]
    dir1 --> dir3["dir3"]
    dir2 --> file2
```

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

Выберите один вариант из списка

Верно решили 20 711 учащихся
Из всех попыток 41% верных

☒ Здорово, всё верно.

☐ Только file1

☐ Все кроме file1

☐ Все кроме file2

☒ Все кроме file3

☐ Только file2

Следующий шаг Решить снова

Ваше решение Вы получили 1 балл

Рис. 27: Задание №23. Условие и ответ

4. Вопрос 24-й: Рассмотрим каждую из опций:

- -A n – Вывод n строк контекста после группы найденных вхождений;
- -B n – Вывод n строк контекста перед группой найденных вхождений;
- -C n – Вывод n строк контекста до и после группы найденных вхождений;

Так как до и после 10 строк слов word нет текста, то контекста не будет, а размер всех файлов будет одинаковый.

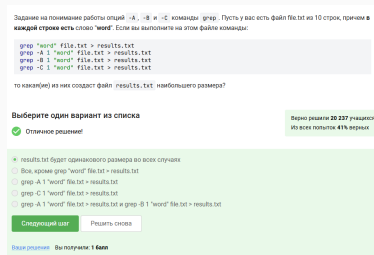


Рис. 28: Задание №24. Условие и ответ

5. Вопрос 25-й: Команда выведет только строки, которые: заканчиваются на слове Ubuntu или ubuntu, перед которым слитно находится не более одного вхождения символов из набора xklXKL.

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]*[u]buntu$" text.txt`.

Выберите все подходящие ответы из списка

Верно решено 18 705 учащимся
Из всех попыток 23% верные

☒ Так точно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в комментариях, ответив на их вопросы, или сравнить своё решение с другими на форуме решений.

- ☒ The best OS is Xubuntu
- ☒ Mac OS X, Windows, Ubuntu
- ☒ Hmmm, XXLubuntu
- ☒ Lubuntu is better than Ubuntu
- ☐ Lubuntu is better than Windows
- ☐ Uuubuntu!

Ваши решения: Вы получили 2 балла

Рис. 29: Задание №25. Условие и ответ

6. Вопрос 26-й: Без опции `-n` происходит автоматический вывод всех строк, при этом `r` в конце выражения указывает на то, что дополнительно должны быть выведены строки, удовлетворяющие шаблону. Произойдет задвоение строк.

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

Выберите один вариант из списка

✔ Правильно.

Верно решили 19 784 учащихся
из всех попыток 39% верных

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

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл

Рис. 30: Задание №26. Условие и ответ

7. Вопрос 27-й: Используем команду sed

`s/ [A-Z]\{2,} / abbreviation`

`/g` input.txt > edited.txt, в

которой:

- `s` – указание на копирование;
- `[A-Z]` – любые символы из набора от A до Z;
- `\{2,}` – хотя бы две буквы из набора;
- `abbreviation` – то на что заменяем;
- `g` – каждое вхождение в строке;

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 `"I YEU YEU and YEU!"` является **некорректным** (в нем есть две "аббревиатуры", но сам код подради и на таких примерах мы проверять вашу инструкцию не будем).

Пример: если у нас был текст `"Hi, I heard these songs by ABBA, TLA and BH!"`, 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 скорее анализируем структуру вашей инструкции (например, что в ней использованы именно `sed` и сделано это ровно один раз, что на вход подради `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 идеален: платформе `sed`, но он очень близок к нему. Главные "несовместимости" заключаются 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 `[^]` и `]` какой из символов к чему относится! Например, регулярное выражение `[a-z]` (здесь или один раз по одной или более букв "a") нужно записать как `[a^]z` (при этом запись `[a^]z`, конечно же, не поможет).

Напишите текст

✓ Хорошо, новость, верно!

Верно решили 16 632 учащихся
Из всех попыток 34% верных

Вы решили задание задним числом! Вы можете помочь остальным учащимся в [подсказках](#), отвечая на их вопросы, или сравнить свой 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

Рис. 31: Задание №27. Условие и ответ

Строим графики в gnuplot

1. Вопрос 28-й: Изучим справку man gnuplot.
Из неё получим, что существует ключ `-p`
или же `--persist`, который позволяет
оставить окно графика после закрытия
терминала.

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 решили 18 785 учащихся
из всех попыток 91% верных

☒ Отличное решение!

☐ -x, --show-plot-after-exit

☒ -p, --persist

☐ Такой опции не существует

☐ -raise

Следующий шаг Решить снова

Вопрос решен Вы получили: 1 балл

Рис. 32: Задание №28. Условие и ответ

2. Вопрос 29-й: Первая строка указывает gnuplot использовать первую строку файла как название столбцов, так как файлы названий столбцов отсутствуют, gnuplot присвоит ряду стандартное название во второй строке – 'data.csv' using 1:2.

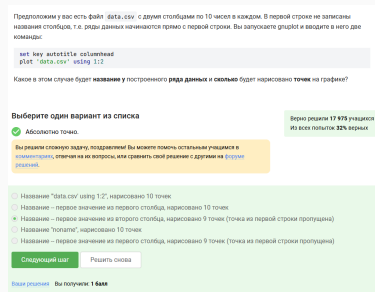


Рис. 33: Задание №29. Условие и ответ

3. Вопрос 30-й: Установим деления командой set, укажем ось Oх с помощью xtics и введем через запятую точки в формате (“название”, значение).

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=250`, 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: проверить, что переменные `x1`, `x2`, `x3` идут по возрастанию или что они являются числами **не нужно!**

Примечание 2: в видеофрагменте на предыдущем шаге явучал 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 решил 13 935 учащихся
Из всех попыток 44% верные

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

Следующий шаг Решить снова

Ваше решение Вы получили 2 балла

Рис. 34: Задание №30. Условие и ответ

4. Вопрос 31-й: В данном случае график:

- отразится зеркально, если поставить отрицание перед функцией и раскрыть скобки:
`plot -x**2-y**2`
- станет вращаться в обратную сторону, если поменять с 10 до 350 градусов:
`zrot=(zrot+350)%360`
- станет вращаться в два раза быстрее, если уменьшить задержку в два раза: `pause 0.1`

Если вы не скачали на предыдущем шаге файлы [animated.gnup](#) и [move.txt](#), 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.txt` (т.е. **добавить** и **удалить** инструкции **нельзя**) таким образом, чтобы:

- График **отразится зеркально** относительно горизонтальной оси абсцисс. То есть там, где была точка (10, 10, 250), станет точка (10, -10, 250), где была точка (-10, -10, 250) станет (-10, 10, 250) 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 переключков графика на каждую секунду вращения.

Измененный файл загрузите в форму ниже.

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

Напишите текст

☒ Здорово, всё верно.

Ваше решение 10/104 решений
Итак, вы набрали 47% верных

[Назад к вопросу](#)

[Следующий шаг](#) [Решить снова](#)

Вопрос решен! Вы получили 3 балла

```
move (1).txt - Блокнот
Файл  Правка  Формат  Вид  Справка
a=a+1
zrot=(zrot+350)%360
set view xrot,zrot
plot -x**2-y**2
pause 0.1
if (a<50) reread
|
```

Рис. 36: Задание №31. Файл

Разное

1. Вопрос 32-й: Проверим каждую строку на правильность изменения прав файла и выберем нужные варианты ответа.

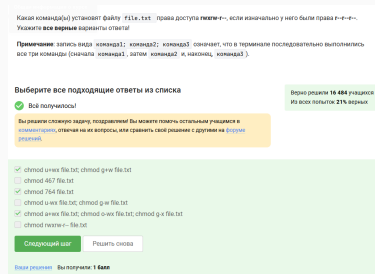


Рис. 37: Задание №32. Условие и ответ

2. Вопрос 33-й: Так как изначальные права у root, то владелец с помощью sudo должен дать права пользователю. Без sudo права не поменяются. Выберем верные варианты ответа.

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 `rwxr-xr-x` 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 получится, что несмотря на права `rwxr-xr-x`, 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 удалить и создавать файлы в ней не сможет (права `w` у него нет).

Важно отметить, что директория в `Linux` это в каком-то смысле файл. Содержимое такого "файла" – это записи о файлах и поддиректориях этой директории (глубоко говоря их название). 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

✓ Все получилось!

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

Верно решили 14 063 учеников из всех попыток 19% верных

- ☐ `sudo chmod o+r dir`
- ☒ `sudo chmod a+w dir`
- ☒ `sudo chmod o+w dir`
- ☐ `sudo chown group dir`
- ☐ `chown user:group dir`
- ☐ `chmod o+w dir`

Следующий шаг

Решить снова

Ваше решение Вы получили: 1 балл

Рис. 39: Задание №33. Ответ

Рис. 38: Задание №33. Условие

3. Вопрос 34-й: Откроем справку по команде `wc` и посмотрим ключи:
- `-l имя_файла` – вывести количество строк;
 - `-с имя_файла` – вывести количество байт;
 - `-m имя_файла` – вывести количество символов;
 - `-L имя_файла` – вывести длину самой длинной строки;
 - `-w имя_файла` – вывести количество слов;

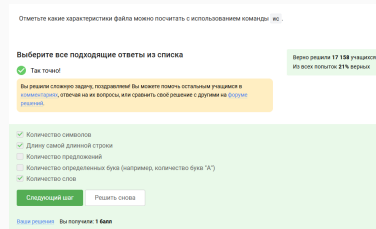


Рис. 40: Задание №34. Условие и ответ

4. Вопрос 35-й: Выведем команду `du`, анализирующую занимаемое файловое пространство, с двумя ключами `-h` или `--human-readable` и `-s` или `--summarize`.

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 2848 байт - надо вывести 2.8к) 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: 2.4к (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% верные

du -h -s

Следующий шаг Решить снова

[Вам решение](#) Вы получили 2 балла

Рис. 41: Задание №35. Условие и ответ

5. Вопрос 36-й: Создадим директорию с помощью, а номера укажем с помощью команды {1..3}, которая проходит по значениям от 1 до 3.

Введите 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

Верно. Так держать!

Верно решили 18 720 учащихся
Из всех попыток 40% верных

`mkdir dir{1..3}`

Следующий шаг Решить снова

Всего решений Вы получили 2 балла

Рис. 42: Задание №36. Условие и ответ

Вывод

В результате выполнения блока внешнего курса №3 “Продвинутые темы” я ознакомился с редактором `vim` и написанием скриптов на `bash`, изучил продвинутый поиск файлов и их продвинутое редактирование и научился строить графики в `gnuplot`.