

# **Отчет по выполнению курса на онлайн платформе Stepik**

**Введение в Linux. Продвинутые темы**

Зарифбеков Амир Пайшанбиевич НБИбд-01-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение заданий Stepik. Продвинутые темы.</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>39</b>
<b>6</b>	<b>Используемые ресурсы</b>	<b>40</b>

## Список иллюстраций

4.1	Какую клавишу(и) нужно нажать на клавиатуре? . . . . .	8
4.2	Строка . . . . .	9
4.3	Редактирование . . . . .	10
4.4	Какую команду нужно ввести для этого в vim? . . . . .	11
4.5	Режим выделения (Visual) . . . . .	11
4.6	Работа с документом . . . . .	12
4.7	Стрелочки вверх/вниз . . . . .	13
4.8	Program . . . . .	14
4.9	Имена переменных в bash . . . . .	14
4.10	Работа с терминалом . . . . .	15
4.11	Скрипт . . . . .	16
4.12	Какие строки и в какой последовательности он выведет на экран .	17
4.13	Скрипт на bash . . . . .	18
4.14	Если запустить этот скрипт, то сколько раз на экран будет выведено слово “start”, а сколько раз слово “finish”? . . . . .	19
4.15	Работа с документом . . . . .	20
4.16	Увеличат значение переменной a на значение переменной b . . .	21
4.17	путь . . . . .	22
4.18	Работа с файлами . . . . .	22
4.19	Работа с файлами . . . . .	23
4.20	Работа скрипта . . . . .	24
4.21	Работа с калькулятором . . . . .	25
4.22	Файлы . . . . .	26
4.23	-path и -name команды find . . . . .	27
4.24	Find . . . . .	28
4.25	Файл results.txt наибольшего размера . . . . .	29
4.26	Команда grep -E . . . . .	29
4.27	sed -n . . . . .	30
4.28	Форма sed . . . . .	31
4.29	Запуске gnuplot . . . . .	32
4.30	Название . . . . .	33
4.31	Скрипт . . . . .	34
4.32	Работа с файлом . . . . .	35
4.33	Права доступа rwxrw-r- . . . . .	36
4.34	Команда . . . . .	37
4.35	Характеристики . . . . .	37
4.36	Команда . . . . .	38

4.37 Команда . . . . .	38
------------------------	----

# 1 Цель работы

Целью третьего блока “Продвинутые темы” является:

1. Научиться работать с редактором Vim.
2. Работа в интерпретаторе bash и других приложениях.

## 2 Задание

1. Для изучения и выполнения практических заданий нам дается 7 миниблоков в нашем большом блоке “Продвинутые темы”. Нам нужно просмотреть теоретические видео и выполнить тестовые задания.

## 3 Теоретическое введение

Мы будем работать в гораздо более мощном редакторе `vim`. Он позволяет выполнять многие действия на уровне продвинутых редакторов с графическим пользовательским интерфейсом или даже лучше. Кроме того, `vim` (или, как минимум, его “старший брат” `vi`), в отличие от `nano` и многих других программ, по умолчанию установлен в подавляющем большинстве дистрибутивов Linux (и не только Linux!), поэтому научившись им пользоваться, мы сможем отредактировать текстовые файлы практически в любом терминале на любой системе. Это особенно актуально, когда у нас нет возможности запустить редактор с графическим пользовательским интерфейсом, например, если вы находитесь на сервере.

## 4 Выполнение заданий Stepik.

### Продвинутые темы.

3.1) Для начала мы должны изучить, что такое vim и зачем на нем работать. Первый вопрос в блоке **“Текстовый редактор vim”**:

- 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.

3.1 Текстовый редактор vim 5 из 13 шагов пройдено 1 из 10 баллов получен

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

✓ Прекрасный ответ.

Верно решили 32 523 учащихся  
Из всех попыток 69% верных

☐ "Esc"

☒ ":", затем "q", затем "Enter"

☐ "Q"

☐ ":", затем "q"

☐ "q"

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 4.1: Какую клавишу(и) нужно нажать на клавиатуре?

- 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!

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. Так держать!

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

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

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

Следующий шаг    Решить снова

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

Рис. 4.2: Строка

- 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 (т.е. знаки "<" и ">" не несут отдельного смысла).

о. 1 текстовый редактор vim → эта задача принадлежит → на то чтобы получить

---

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

☒ 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](#).

☐ d2wwywp  
☒ d2wwywPp  
☒ ddithree four four four five<Esc>  
☒ d2w\$bifour four <Esc>  
☒ d2wwifour four <Esc>  
☐ d2dwywPp

Следующий шаг    Решить снова

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

Рис. 4.3: Редактирование

- 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.

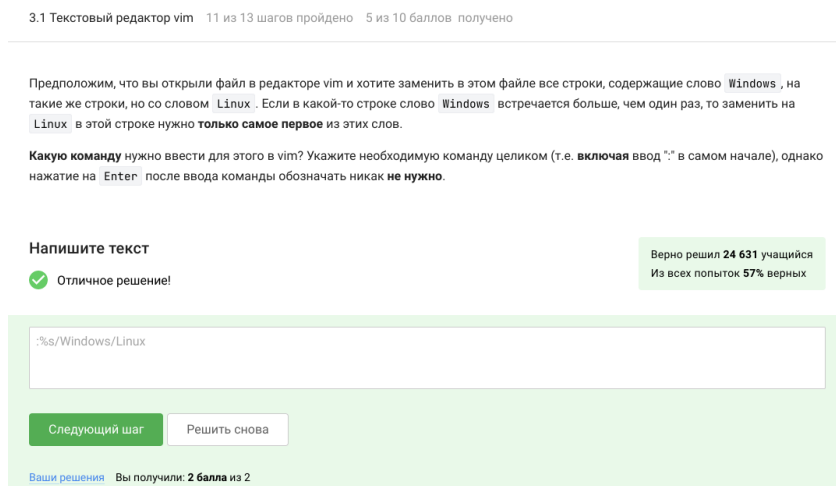


Рис. 4.4: Какую команду нужно ввести для этого в vim?

- Третий режим работы vim - режим выделения (Visual).

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

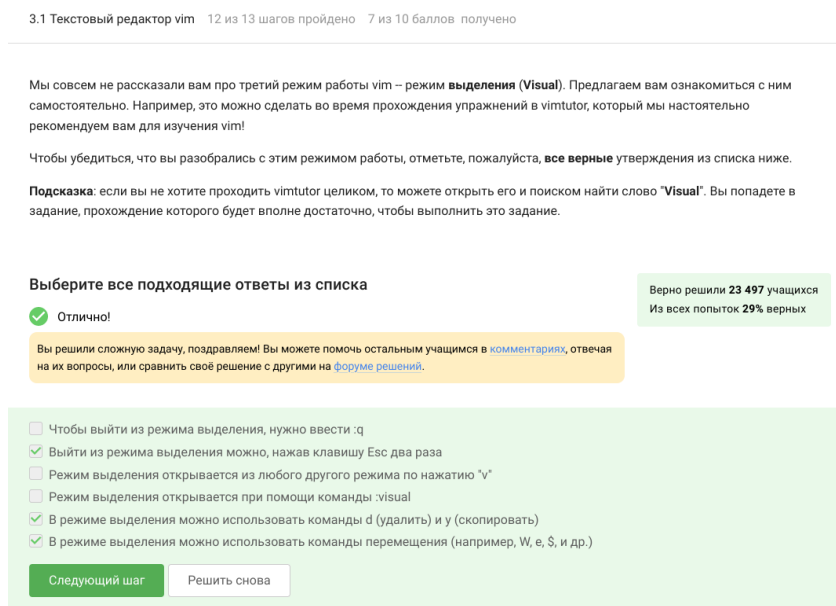


Рис. 4.5: Режим выделения (Visual)

- Практическое задание в терминале: Откройте в vim файл `/home/box/byron.txt`.

Удалите все строки с 1001 по 2000 (включительно). Скопируйте строки с 5 по 11 (включительно) и вставьте их в самый конец файла, добавив одну пустую строку перед этой вставкой (т.е. строчка 5 должна следовать за ровно одной пустой строкой, а та за строкой с номером 6277). Замените в тексте все “Harold” на “Ivan”.

Сохраните отредактированный файл с именем /home/box/byron\_edited.txt (это можно сделать прямо из vim!).

Выполнение:

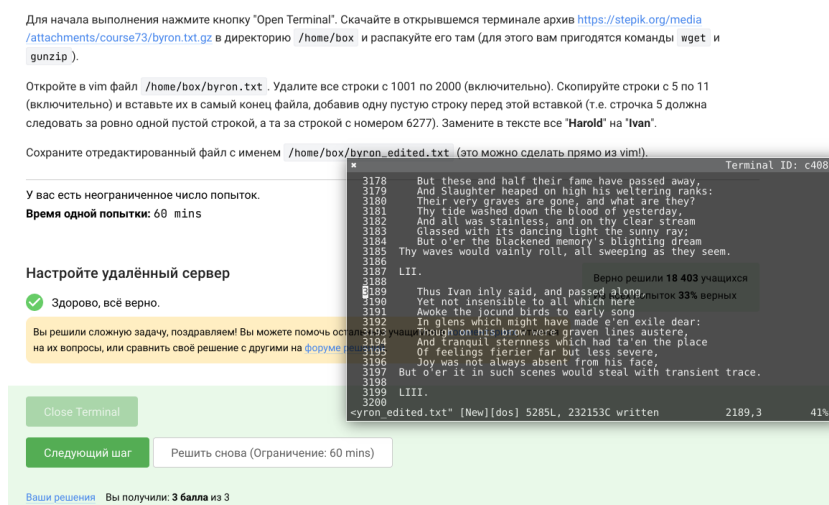


Рис. 4.6: Работа с документом

3.2) Переходим к следующему мини-блоку “Скрипты на bash”. После изучения теории решаем три тестовых задания и одно практическое

Что такое программирование вообще и что такое bash. Затем перейдем и к практической части - посмотрим, что должна включать в себя программа (скрипт) на bash, напишем первый простой скрипт и запустим его в терминале. Рассмотрим и попробуем на практике такие понятия как переменные и аргументы скрипта.

- 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?

3.2 Скрипты на `bash`: основы 3 из 10 шагов пройдено 1 из 6 баллов получен

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

Верно решили 30 266 учащихся  
Из всех попыток 65% верных

☒ Здорово, всё верно.

☐ Только из набора C

☐ Только из набора A

☐ Только из набора B

☐ Никакие команды появляться не будут

☐ Из наборов B и C

Следующий шаг Решить снова

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

Рис. 4.7: Стрелочки вверх/вниз

- 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?

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

☒ Хорошая работа.

Верно решили 29 905 учащихся  
Из всех попыток 76% верных

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 4.8: Program

- Какие из представленных ниже строк могут быть именами переменных в bash?

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

☒ Здорово, всё верно.

Верно решили 27 188 учащихся  
Из всех попыток 25% верных

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

- ☐ `var@iabile`  
☒ `._variable`  
☒ `__variable`  
☐ `123variable`  
☒ `variable123`  
☒ `variable_123`  
☒ `variable`

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 4.9: Имена переменных в bash

- 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

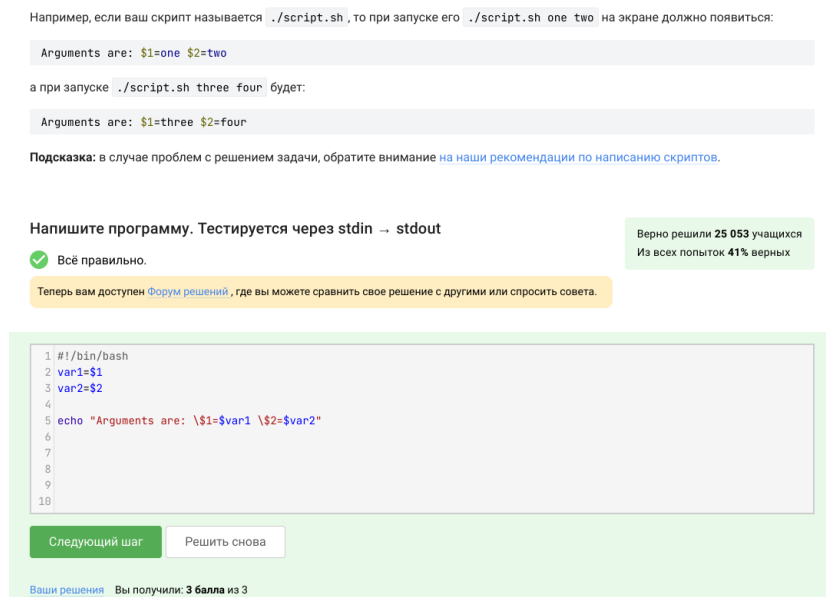


Рис. 4.10: Работа с терминалом

3.3) Однако такое поведение подходит далеко не всегда. Мы изучим управляющие конструкции языка bash, которые позволят нам писать скрипты, где часть инструкций выполняется только при определенных условиях (ветвления), а часть инструкций выполняется по много раз подряд (циклы).. В мини-блок “Скрипты на bash: ветвления и циклы” рассмотрим управляющие конструкции.

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).

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, 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](#).

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

- ☐ `-n $1`
- ☒ `-z "`
- ☒ `-s $0`
- ☒ `5 -ge 5`
- ☒ `$var1 == $var2 || $var1 != $var2`
- ☐ `-z " "`

Следующий шаг    Решить снова

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

Рис. 4.11: Скрипт

- 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`.

Посмотрите на фрагмент кода 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

Верно решили 25 138 учащихся  
Из всех попыток 64% верных

☒ Здорово, всё верно.

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

Следующий шаг    Решить снова

Ваши решения    Вы получили: 1 балл из 1

Рис. 4.12: Какие строки и в какой последовательности он выведет на экран

- 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

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`

✓ Верно. Так держать!

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 #!/bin/bash
2 v=student #задаем общую часть в каждом выводе - слово "student"
3 case $1 in #выполнить следующее для разных аргументов
4 0) res="No ${v}s";; #res - это результат для вывода
5 1) res="${1}${v}s";;
6 [2-4]) res="${1} ${v}s";;
7 *) res="A lot of ${v}s";;
8 esac
9 echo "$res" #вывести результат
10
11
12
13
14

```

Следующий шаг    Решить снова

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

Рис. 4.13: Скрипт на bash

- 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
```

3.3 Скрипты на bash: ветвления и циклы 8 из 9 шагов пройдено 6 из 10 баллов получено

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

Отличное решение!

Верно решили 24 582 учащихся  
Из всех попыток 45% верных

☒ 5 раз "start" и 4 раза "finish"

☐ 5 раз "start" и ни разу "finish"

☐ 3 раза "start" и ни разу "finish"

☐ 3 раза "start" и 3 раза "finish"

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 4.14: Если запустить этот скрипт, то сколько раз на экран будет выведено слово “start”, а сколько раз слово “finish”?

- 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 “, your group is ”, где 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!).

```
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                 echo "$name, your group is adult" ; else
24                 if [[ ($age -ge 17) && ($age -le 25) ]]; then
25                     echo "$name, your group is youth" ; fi
26                 fi ; break
27             done ; fi
28         done
29     fi
30 done
```

Следующий шаг    Решить снова

Ваши решения    Вы получили: 4 балла из 4

Рис. 4.15: Работа с документом

3.4) Следующий блок **Скрипты на bash**: разноемы заканчиваем изучение основ программирования на bash. С использованием уже пройденного материала можно написать довольно сложные и полезные скрипты на bash, но на этом занятии будет рассмотрено еще несколько полезных тем. Среди них будут: арифметические операции; запуск внешних программ и обработка результатов их работы; понятие функций в языке bash и их использование.

- 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.

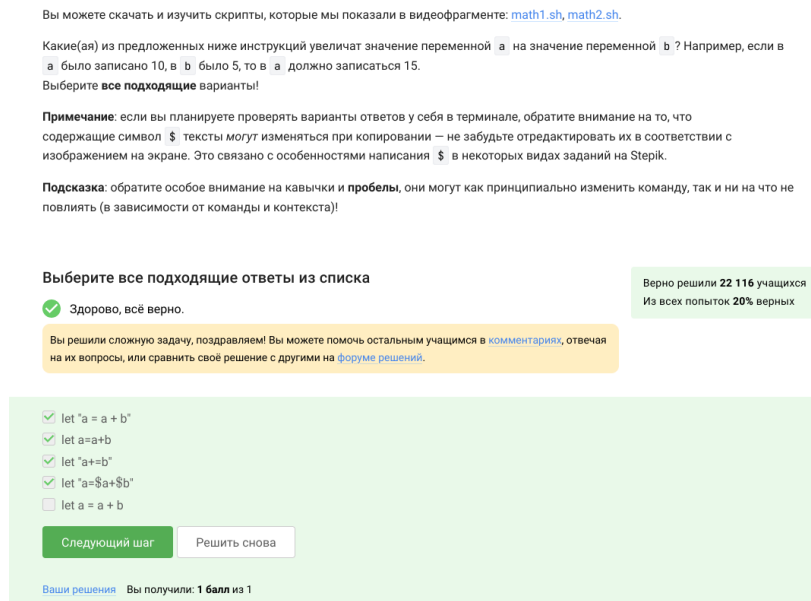


Рис. 4.16: Увеличат значение переменной `a` на значение переменной `b`

- 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?

3.4 Скрипты на `bash`: разное 5 из 10 шагов пройдено 2 из 14 баллов получено

Вы прошли больше 80% курса, оставьте отзыв

Оставить отзыв Нет, спасибо

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

✓ Отличное решение!

Верно решили 23 677 учащихся  
Из всех попыток 51% верных

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

Следующий шаг

Решить снова

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

Рис. 4.17: путь

- Выберите все верные утверждения или правильно работающие конструкции `if`.

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

✓ Всё получилось!

Верно решили 21 426 учащихся  
Из всех попыток 20% верных

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

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

Следующий шаг

Решить снова

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

Рис. 4.18: Работа с файлами

- Посмотрите на функцию из bash-скрипта:

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

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

✓ Верно.

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

counters are and 110

Следующий шаг    Решить снова

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

Рис. 4.19: Работа с файлами

- 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 ”, 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
```

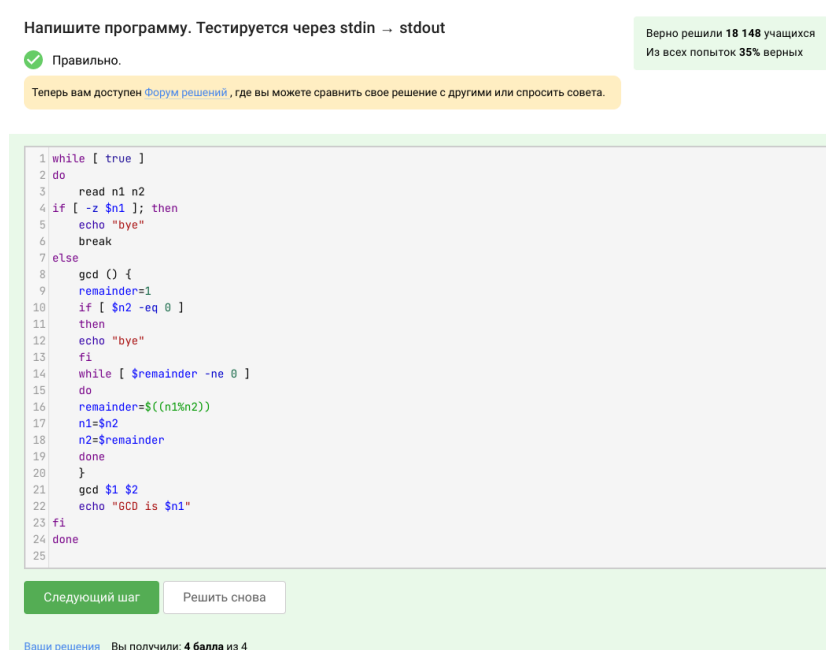


Рис. 4.20: Работа скрипта

- 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:

Слово “exit”. В этом случае скрипт должен вывести на экран слово “bye” и завершить работу. 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. 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

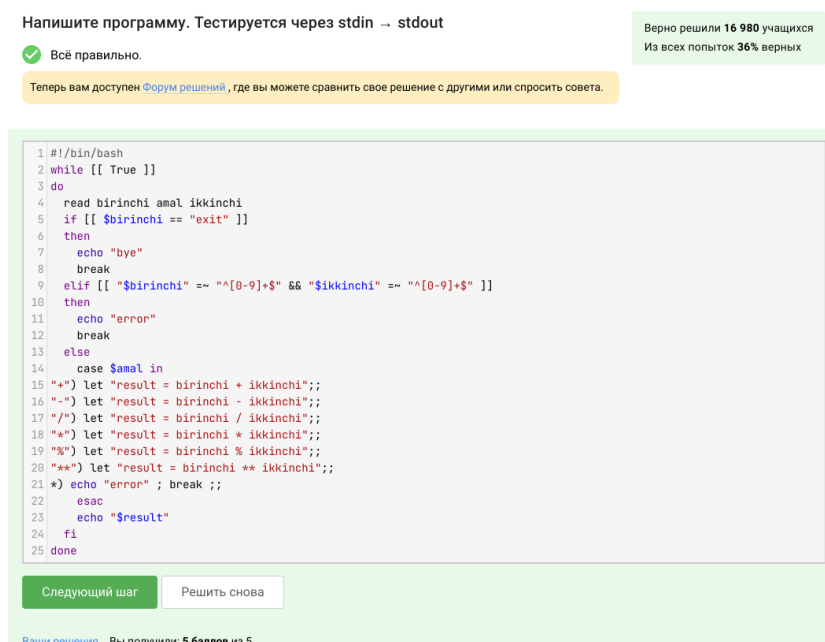


Рис. 4.21: Работа с калькулятором

3.5) Мини-блок **“Продвинутый поиск и редактирование”**. Данный блок предназначен для знакомимства с командой (точнее потоковым текстовым редактором) `sed`, который позволяет не только искать слова в файлах, но и сразу же

эти файлы редактировать.

- 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"`?

3.5 Продвинутый поиск и редактирование 3 из 13 шагов пройдено 1 из 10 баллов получен

Вы прошли больше 80% курса, оставьте отзыв [Оставить отзыв](#) [Нет, спасибо](#)

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

☒ Прекрасный ответ.

Верно решили 20 547 учащихся  
Из всех попыток 36% верных

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

[Следующий шаг](#) [Решить снова](#)

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

Рис. 4.22: Файлы

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

3.5 Продвинутый поиск и редактирование 4 из 13 шагов пройдено 2 из 10 баллов получено

Вы прошли больше 80% курса, оставьте отзыв	Оставить отзыв	Нет, спасибо
Поздравляем! Вы набрали достаточно баллов для получения сертификата. Сертификат появится в вашем профиле в течение суток.		ОК

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

Выберите все подходящие ответы из списка

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

✔ Правильно.

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

☒ В некоторых случаях `find` с `-name` найдет меньше файлов, чем `find` с таким же запросом, но с `-path`  
☐ 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  
☒ В некоторых случаях `find` с `-name` найдет больше файлов, чем `find` с таким же запросом, но с `-path`

Следующий шаг    Решить снова

Рис. 4.23: `-path` и `-name` команды `find`

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

`/home/bi/` `xxx` `dir1` `xxx` `file1` `xxx` `dir2` `xxx` `file2` `xxx` `dir3` `xxx` `file3`

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

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

☒ Верно. Так держать!

Верно решили 20 711 учащихся  
Из всех попыток 41% верных

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

Следующий шаг

Решить снова

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

Рис. 4.24: Find

- 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?

3.5 Продвинутый поиск и редактирование 7 из 13 шагов пройдено 4 из 10 баллов получено

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

☒ Правильно, молодец!

☐ `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`

☐ `grep -A 1 "word" file.txt > results.txt`

☒ `results.txt` будет одинакового размера во всех случаях

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

Следующий шаг Решить снова

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

Рис. 4.25: Файл `results.txt` наибольшего размера

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

3.5 Продвинутый поиск и редактирование 9 из 13 шагов пройдено 6 из 10 баллов получено

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

Выберите все подходящие ответы из списка

Верно решили 18 768 учащихся  
Из всех попыток 23% верных

☒ Правильно.

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

☒ Mac OS X, Windows, Ubuntu

☐ Ubuntu is better than Windows

☒ Ubuntu is better than Ubuntu

☐ Well, xubuntu is OK

☒ Linux is not always Ubuntu

☐ Uuuubuntu!

Следующий шаг Решить снова

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

Рис. 4.26: Команда `grep -E`

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

3.5 Продвинутый поиск и редактирование 11 из 13 шагов пройдено 7 из 10 баллов получено

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

☒ Отличное решение!

☒ Каждая строка будет выведена два раза  
☐ Появится сообщение об ошибке  
☐ Будут выведены все строки файла `text.txt`, в которых есть только большие буквы латинского алфавита  
☐ На экран будет выведено всё содержимое файла `text.txt`

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 4.27: `sed -n`

- 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.

**Внимание!** Во время проверки мы **не запускаем** команду, которую вы ввели на реальном файле с «аббревиатурами» (это небезопасно, мы всего же вводим `cat -rf / ./a.out`) Вместо этого мы сперва анализируем структуру вашей инструкции (например, что в ней использовано 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** подобно `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 «( )» 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).

✓ Верно. Так держать!

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](#).

kjhkhjkhjkknk

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

Следующий шаг

Решить снова

Ваши решения

Вы получили: 3 балла из 3

3.6) Мини-блок “**Строим графики в gnuplot**”. Данный блок предназначен для рассмотрения программы gnuplot. Эта программа используется для построения двух- и трехмерных графиков. Мы изучим базовые команды gnuplot для работы в интерактивном режиме, а затем познакомимся с потоковым режимом работы и напишем несколько gnuplot-скриптов. В завершение занятия будут продемонстрированы некоторые из продвинутых возможностей gnuplot.

- 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?

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.

Верно решили 18 785 учащихся  
Из всех попыток 51% верных

- ☐ -s, --show-plots-after-exit
- ☐ -raise
- ☐ Такой опции не существует
- ☒ -p, --persist

Следующий шаг

Решить снова

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

Рис. 4.29: Запуске gnuplot

- 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?



3.6 Строим графики в gnuplot 5 из 10 шагов пройдено 2 из 7 баллов получено

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

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

Какое в этом случае будет **название** у построенного **ряда данных** и **сколько** будет нарисовано **точек** на графике?

Выберите один вариант из списка

Отличное решение!

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 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)
- ☒ Название – первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)

Следующий шаг Решить снова

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

Рис. 4.30: Название

- Предположим, что вы пишете `gnuplot`-скрипт и у вас в нем есть три переменные `x1`, `x2`, `x3`, в которых записаны координаты важных точек по оси ОХ (по возрастанию). Вы хотите, чтобы на этой оси было только три деления (т.е. три черточки) в этих самых координатах, а подписи этих делений были оформлены в виде “point , value”.

Например, для  $x_1=0$ ,  $x_2=10$ ,  $x_3=20$ , 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,  $x_1=100$ ,  $x_2=150$ ,  $x_3=250$ , 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.

координатах, а подписи этих деления 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

✓ Правильно.

Верно решили 13 935 учащихся

Из всех попыток 44% верных

Следующий шаг

Решить снова

Ваши решения    Вы получили: 2 балла из 2

Рис. 4.31: Скрипт

- 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.

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

✓ Правильно, молодец!

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 из 3

Рис. 4.32: Работа с файлом

3.7) В этом блоке **Разное** мы рассмотрим несколько небольших, но довольно важных тем. Эти моменты не вошли в другие занятия курса, но мы считаем важным упомянуть о них хотя бы на завершающем занятии.

Сначала мы обсудим понятие прав доступа в Linux, затем изучим несколько новых команд терминала, а напоследок рассмотрим несколько новых возможностей применения уже известных нам команд.

- 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!

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 выполнялись все три команды (сначала `команда1`, затем `команда2` и, наконец, `команда3`).

Выберите все подходящие ответы из списка

☒ Всё правильно.

Верно решили 16 484 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](#).

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

Следующий шаг

Решить снова

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

Рис. 4.33: Права доступа `rw-rw-r--`

- 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!

В случае же, когда речь идет о директории созданной **root**, ситуация будет проще: пользователь сможет посмотреть её содержимое (у него есть право "r"), но удалять и создавать файлы в ней не сможет (права "w" у него нет). Важно отметить, что директории в *Linux* это в каком-то смысле *файлы*. Содержимое такого "файла" – это записи о файлах и поддиректориях этой директории (грубо говоря их *названия*). Таким образом, право "r" у директории дает возможность просматривать "записи", т.е. просматривать её состав. Право "w" у директории дает возможность удалять/добавлять новые "записи", т.е. удалять/создавать файлы/поддиректории в ней.

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

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

Выберите все подходящие ответы из списка

Верно решили **14 683** учащихся  
Из всех попыток **15%** верных

✓ Правильно.

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

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

Следующий шаг    Решить снова

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

Рис. 4.34: Команда

- Отметьте какие характеристики файла можно посчитать с использованием команды **wc**.

3.7 Разное    7 из 15 шагов пройдено    3 из 7 баллов    получено

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

Выберите все подходящие ответы из списка

Верно решили **17 158** учащихся  
Из всех попыток **21%** верных

✓ Отличное решение!

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 строк  
☒ Длину самой длинной строки  
☒ Размер файла в байтах  
☒ Количество символов  
☐ Количество предложений

Следующий шаг    Решить снова

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

Рис. 4.35: Характеристики

- 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).

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

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: 2.4K (также на экране может быть выведен еще и символ "", обозначающий, что это размер именно текущей директории).

Напишите текст

✓ Правильно, молодец!

Верно решил 16 381 учащийся  
Из всех попыток 53% верных

du -h -s

Следующий шаг Решить снова

Ваши решения Вы получили: 2 балла из 2

Рис. 4.36: Команда

- 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.

3.7 Разное 10 из 15 шагов пройдено 7 из 7 баллов получено

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

✓ Отлично!

Верно решили 16 720 учащихся  
Из всех попыток 40% верных

mkdir dir{1..3}

Следующий шаг Решить снова

Ваши решения Вы получили: 2 балла из 2

Рис. 4.37: Команда

## 5 Выводы

Курс знакомит слушателей с операционной системой Linux и основами её использования. В рамках третьего заключительного блока курса мы научились работать на редакторе Vim, интерпретаторе bash и других приложениях.

## 6 Используемые ресурсы

**Введение в Linux** <https://stepik.org/course/73>