

# **Отчёт по прохождению внешнего курса “Введение в Linux”. Этап 3.**

**НКАбд-06-23**

Улитина Мария Максимовна

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	27

## Список иллюстраций

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

## Список таблиц

# 1 Цель работы

Изучить основы Linux.

## 2 Выполнение лабораторной работы

### ##Задание 1

(рис. 2.1).

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

Выберите один вариант из списка

✓ Всё правильно.

Верно решили 32 523 учащихся  
Из всех попыток 69% верных

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

Следующий шаг    Решить снова

Ваши решения    Вы получили: 1 балл из 1

Рис. 2.1: Задание 1

### ##Задание 2

Пояснение: ответ на это задание можно получить эмпирическим путем, выполнив все эти команды в vim (рис. 2.2).

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

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

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

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

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

Выберите все подходящие ответы из списка

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

✓ 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](#).

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

Следующий шаг

Решить снова

Рис. 2.2: Задание 2

## ##Задание 3

Пояснение: ответ на это задание можно получить эмпирическим путем, выполнив все эти команды в vim (рис. 2.3).

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

✓ 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](#).

- ☐ d2dwywPp
- ☒ d2wwywPp
- ☐ x2wwywPp
- ☒ ddithree four four four five<Esc>
- ☒ d2w\$bifour four <Esc>
- ☐ d2wwywpp

Следующий шаг

Решить снова

Рис. 2.3: Задание 3

## ##Задание 4

Пояснение: ответ на это задание можно получить эмпирическим путем, выполнив все эти команды в vim (рис. 2.4).

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

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

Напишите текст

✓ Абсолютно точно.

Верно решил 24 631 учащийся  
Из всех попыток 57% верных

Следующий шаг    Решить снова

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

Рис. 2.4: Задание 4

## ##Задание 5

Пояснение: ответ на это задание можно получить эмпирическим путем, выполнив все эти команды в vim (рис. 2.5).

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

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

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

Выберите все подходящие ответы из списка

✓ Прекрасный ответ.

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

Верно решили 23 497 учащихся  
Из всех попыток 29% верных

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

Следующий шаг    Решить снова

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

Рис. 2.5: Задание 5

## ##Задание 6



## Пояснение: только из набора С (рис. 2.6).

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

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

Выберите один вариант из списка

✓ Хорошие новости, верно!

Верно решили 30 266 учащихся  
Из всех попыток 65% верных

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 2.6: Задание 6

## ##Задание 7

Пояснение: файл создастся в той директории, где мы сейчас находимся (рис. 2.7).

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

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

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

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

Выберите один вариант из списка

✓ Так точно!

Верно решили 29 905 учащихся  
Из всех попыток 76% верных

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 2.7: Задание 7

## ##Задание 8

Пояснение: правила схожи с именованием переменных в других языках программирования (рис. 2.8).

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

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

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

Выберите все подходящие ответы из списка

Верно решили **27 188** учащихся  
Из всех попыток **25%** верных

✓ Так точно!

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

- ☐ var.l.able
- ☒ variable
- ☒ VARiable
- ☒ variable123
- ☐ var@lable
- ☒ \_\_variable
- ☐ vari/able

Следующий шаг

Решить снова

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

Рис. 2.8: Задание 8

## ##Задание 9

Пояснение: (рис. 2.9).

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

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

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

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

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

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

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

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

Напишите программу. Тестируется через stdin → stdout

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

✓ Прекрасный ответ.

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

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

Следующий шаг

Решить снова

Рис. 2.9: Задание 9

## ##Задание 10

Пояснение: (рис. 2.10).

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

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

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

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

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

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

Выберите все подходящие ответы из списка

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

☒ Верно.

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

- ☒ `-n $0`
- ☒ `! (4 -le 3)`
- ☒ `5 -ge 5`
- ☒ `-s $0`
- ☐ `$var1 == $var2 && $var1 != $var2`
- ☒ `$var1 == $var2 || $var1 != $var2`

Следующий шаг

Решить снова

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

Рис. 2.10: Задание 10

## ##Задание 11

Пояснение: оба раза будет `else` (рис. 2.11).

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

✓ Прекрасный ответ.

Верно решили 25 138 учащихся  
Из всех попыток 64% верных

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 2.11: Задание 11

## ##Задание 12

Пояснение: 5 start, 4 finish (рис. 2.12).

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

Соответствие входа и выхода должно быть таким:

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

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

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

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

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

```
1 student
```

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

```
A lot of students
```

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

Напишите программу. Тестируется через `stdin → stdout`

✓ Правильно.

Верно решили 23 310 учащихся  
Из всех попыток 38% верных

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

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

Рис. 2.12: Задание 12

## ##Задание 13

Пояснение: 5 start, 4 finish (рис. 2.13).

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

Посмотрите на фрагмент bash-скрипта:

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

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

Выберите один вариант из списка

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

☒ Верно. Так держать!

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

Следующий шаг

Решить снова

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

Рис. 2.13: Задание 13

## ##Задание 14

Пояснение: напишем цикл с чтением переменных и условным ветвлением (рис. 2.14).

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

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

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

**Примеры** корректной работы скрипта:

№1

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

№2:

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

bye
```

Рис. 2.14: Задание 14

(рис. 2.15).

```

1 # put your shell (bash) code here
2 while :
3
4 do
5
6
7
8 echo "enter your name:"
9
10 read NAME
11
12 if [[ -z "$NAME" ]]; then
13     echo "bye"
14     break
15 fi
16
17 echo "enter your age:"
18 read age
19
20
21 if [[ "$age" = 0 ]]; then
22     echo "bye"
23     break
24
25 elif [[ "$age" -ge 1 && "$age" -le 16 ]]; then
26     echo "$NAME, your group is child"
27
28 elif [[ "$age" -ge 17 && "$age" -le 25 ]]; then
29     echo "$NAME, your group is youth"
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

Рис. 2.15: Задание 14

(рис. 2.16).

```

46
47
48     elif [[ "$age" -ge 25 ]]; then
49         echo "$NAME, your group is adult"
50
51
52
53
54 fi
55
56 done
57
58
59
60

```

Следующий шаг    Решить снова

Ваши решения    Вы получили: 4 балла из 4

Рис. 2.16: Задание 14

## ##Задание 15

Пояснение: мы перешли в /home/bi/ и вывели путь к ней (рис. 2.17).



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

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

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

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

Выберите один вариант из списка

Верно решили 23 677 учащихся  
Из всех попыток 51% верных

☒ Отличное решение!

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

☒ `/home/bi`

☐ ``pwd``

☐ `pwd`

☐ `/home/bi/Documents`

Следующий шаг    Решить снова

Ваши решения    Вы получили: 1 балл из 1

Рис. 2.17: Задание 15

## ##Задание 16

Пояснение: первая переменная локальная (рис. 2.18).

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

Посмотрите на функцию из bash-скрипта:

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

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

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

Напишите текст

Верно решили 20 009 учащихся  
Из всех попыток 28% верных

☒ Хорошие новости, верно!

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

counters are and 110

Следующий шаг    Решить снова

Ваши решения    Вы получили: 2 балла из 2

Рис. 2.18: Задание 16

## ##Задание 17

Пояснение: напишем скрипт для отыскания НОД (рис. 2.19).

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 две переменные – см. пример в видеофрагменте). После ввода чисел скрипт считает их НОД и выводит на экран сообщение "GCD is <посчитанное значение>", например, для чисел 15 и 25 это будет "GCD is 5". После этого скрипт опять входит в режим ожидания двух натуральных чисел. Если в какой-то момент работы пользователь ввел вместо этого пустую строку, то нужно написать на экран "bye" и закончить свою работу.

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

Пример корректной работы скрипта:

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

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

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

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

## Рис. 2.19: Задание 17

(рис. 2.20).

Напишите программу. Тестируется через stdin → stdout

Верно решили 18 148 учащихся  
Из всех попыток 35% верных

✓ Всё правильно.

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

```
1 while [ true ]
2 do
3   read n1 n2
4   if [ -z $n1 ]; then
5     echo "bye"
6     break
7   fi
8   gcd () {
9     remainder=1
10    if [ $n2 -eq 0 ]
11    then
12      echo "bye"
13    fi
14    while [ $remainder -ne 0 ]
15    do
16      remainder=$((n1%n2))
17      n1=$n2
18      n2=$remainder
19    done
20    gcd $1 $2
21    echo "GCD is $n1"
22  fi
23 fi
24 done
25
26
27
28
29
```

Следующий шаг

Решить снова

Ваши решения

Вы получили: 4 балла из 4

## Рис. 2.20: Задание 17

## ##Задание 18

Пояснение: напишем калькулятор на `bash`, считывая переменные и знак (рис. 2.21).

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

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

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

Например, если входной файл будет следующего содержания:

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

то на экране будет:

```
11
1024
bye
```

Если же на вход поступит следующий файл:

```
3 ~ 5
2/10
exit
```

то на экране будет:

```
~2
error
```

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

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

Рис. 2.21: Задание 18

(рис. 2.22).

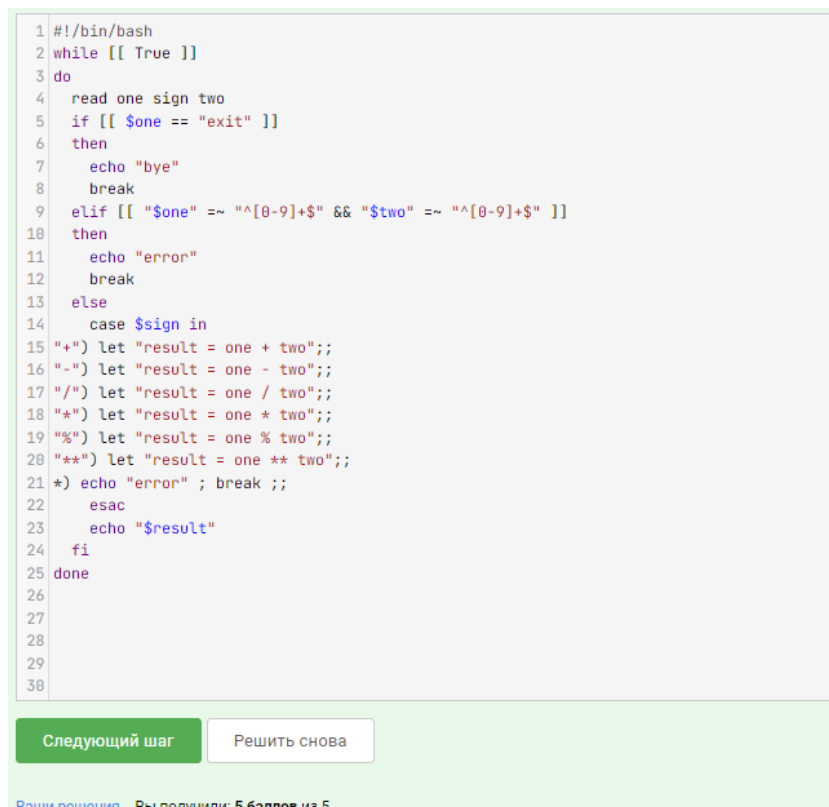


Рис. 2.22: Задание 18

## ##Задание 19

Пояснение: find не чувствителен к регистру, но необходимо чтобы имя файла начиналось с заданной маски (рис. 2.23).

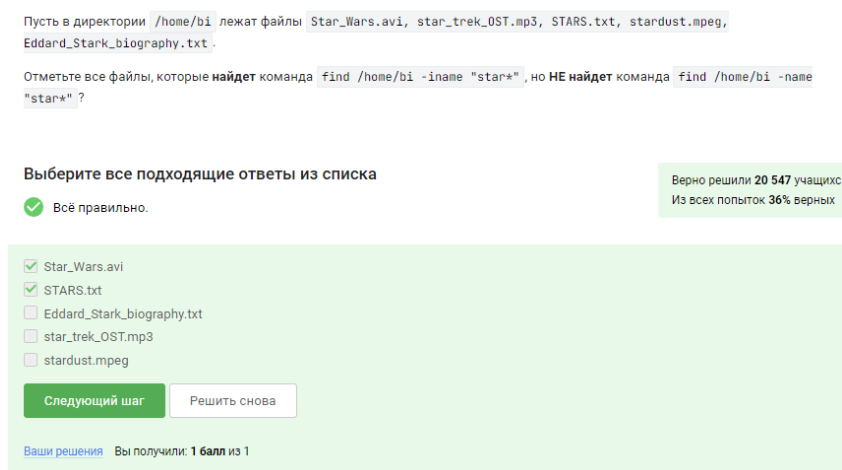


Рис. 2.23: Задание 19

## ##Задание 20

Пояснение: (рис. 2.24).

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

✓ 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](#).

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

Следующий шаг    Решить снова

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

Рис. 2.24: Задание 20

## ##Задание 21

Пояснение: `maxdepth` определяет глубину перехода как 3 (рис. 2.25).

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

✓ Хорошая работа.

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

Следующий шаг    Решить снова

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

Рис. 2.25: Задание 21

## ##Задание 22

Пояснение: будут выводиться группы строк (рис. 2.26).

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

☒ Абсолютно точно.

Верно решили 20 237 учащихся  
Из всех попыток 41% верных

- ☐ `grep -C 1 "word" file.txt > results.txt`
- ☐ `grep -A 1 "word" file.txt > results.txt`
- ☐ `grep -A 1 "word" file.txt > results.txt` и `grep -B 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. 2.26: Задание 22

## ##Задание 23

Пояснение: работа будет с поиском по маске (рис. 2.27).

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

☒ Всё получилось!

Верно решили 18 768 учащихся  
Из всех попыток 23% верных

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

- ☒ Linux is not always Ubuntu
- ☐ Lubuntu is better than Windows
- ☒ Lubuntu is better than Ubuntu
- ☒ I prefer Kubuntu
- ☐ Kbuntu
- ☒ Hmm, XKLUbuntu

Следующий шаг

Решить снова

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

Рис. 2.27: Задание 23

## ##Задание 24

Пояснение: каждая строка будет выведена 2 раза (рис. 2.28).

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

☒ Хорошие новости, верно!

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 2.28: Задание 24

## ##Задание 25

Пояснение: напишем инструкцию, указав файл ввода и вывода (рис. 2.29).

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

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

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

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

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

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

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

Рис. 2.29: Задание 25

напишем инструкцию, указав файл ввода и вывода (рис. 2.30).

Напишите текст

☒ Отличное решение!

Верно решили 16 632 учащихся  
Из всех попыток 34% верных

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

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 3 балла из 3

Рис. 2.30: Задание 25

## ##Задание 26

Пояснение: указать -p (рис. 2.31).

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

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

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 2.31: Задание 26

## ##Задание 27

Пояснение:

(рис. 2.32).

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

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

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

Выберите один вариант из списка

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

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

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 2.32: Задание 32

## ##Задание 28

Пояснение:

(рис. 2.33).



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=8`, `x2=18`, `x3=28`, 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=188`, `x2=158`, `x3=258`, это были бы надписи "point 1, value 100" в точке с координатой 100, "point 2, value 150" в точке с координатой 150 и "point 3, value 250" в точке с координатой 250.

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

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

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

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

Напишите текст

✓ Прекрасный ответ.

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

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

Следующий шаг

Решить снова

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

Рис. 2.33: Задание 33

## ##Задание 29

Пояснение:

(рис. 2.34).

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

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

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

Измененный файл загрузите в форму ниже.

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

Напишите текст

✓ Здорово, всё верно.

Верно решили 12 854 учащихся  
Из всех попыток 47% верных

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 3 балла из 3

Рис. 2.34: Задание 34

## 3 Выводы

Я выполнила первый этап внешнего курса по Linux.