

Внешний курс

Выполнение внешнего курса. Основы Linux. Раздел 3

Сокирка Анна Константиновна

oday

Российский университет дружбы народов, Москва, Россия

Получить общее представление о системе Linux и её базовых возможностях.

Пройти все предложенные тесты и упражнения в третьем разделе курса.

Раздел 3 — Выход из Vim без сохранения

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

✔ Отличное решение!

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

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

[Следующий шаг](#) [Решить снова](#)

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

Рис. 1: Выход из Vim без сохранения

Раздел 3 — Замена и повторение строки в Vim

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

☒ Здорово, всё верно.

Верно решили **23 655** учащихся
Из всех попыток **16%** верных

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

- ☐ x2wwywPp
- ☐ d2dwywPp
- ☒ d2w\$b!four four <Esc>
- ☒ dd!three four four four five<Esc>
- ☒ d2wwywPp
- ☐ d2wwywpp

Следующий шаг

Решить снова

Рис. 2: Замена и повторение строки в Vim

Вы прошли больше 80% курса, оставьте отзыв

[Оставить отзыв](#) [Нет, спасибо](#)

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

✓ Отлично!

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

Следующий шаг

Решить снова

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

Рис. 3: Замена в Vim: Windows на Linux

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

☒ Хорошие новости, верно!

Верно решили **23 497** учащихся
Из всех попыток **29%** верных

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

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

Следующий шаг

Решить снова

Рис. 4: Режим Visual в Vim

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

☒ Отлично!

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

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

[Следующий шаг](#) [Решить снова](#)

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

Рис. 5: История команд оболочки

Раздел 3 — Абсолютный путь к файлу

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

☒ Так точно!

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

Следующий шаг Решить снова

Ваши решения Вы получили: **1 балл**

Рис. 6: Абсолютный путь к файлу

Раздел 3 — Примеры допустимых имен переменных

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

☒ 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](#).

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

- ☒ variable
- ☐ var-i-able
- ☐ variab\$\$le
- ☒ variable123
- ☐ vari/able
- ☒ __variable
- ☐ var.i.able

[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: ...

Рис. 7: Примеры допустимых имен переменных

Раздел 3 — Вывод аргументов в скрипте

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

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

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

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

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

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

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

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

Напишите программу. Тестируется через `stdin → stdout`

✓ Так точно!

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

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

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

Рис. 8: Вывод аргументов в скрипте

Раздел 3 — Условия, всегда выводющие True

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

☒ Всё получилось!

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

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

- ☒ -n \$0
- ☐ -z * *
- ☒ -s \$0
- ☒ -z =
- ☒ -e \$0
- ☒ \$# -ge 0

Рис. 9: Условия, всегда выводющие True

Раздел 3 — Ошибки в скрипте с переменной

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

☒ Всё правильно.

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

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

Рис. 10: Ошибки в скрипте с переменной

Раздел 3 — Условие для вывода student(s)

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



Всё 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. 11: Условие для вывода student(s)

Раздел 3 — Цикл для вывода start и finish

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

Выберите один вариант из списка

☒ Здорово, всё верно.

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

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

[Следующий шаг](#) [Решить снова](#)

Рис. 12: Цикл для вывода start и finish

Раздел 3 — Ввод имени и возраста, определение возрастной группы

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

Рис. 13: Ввод имени и возраста, определение возрастной группы

Раздел 3 — Увеличение переменной

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

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

Выберите **все подходящие** варианты!

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

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

Выберите все подходящие ответы из списка

☒ Прекрасный ответ.

Верно решили **22 116** учащихся
Из всех попыток **20%** верных

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

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

Следующий шаг

Решить снова

Рис. 14: Увеличение переменной

Раздел 3 — Команда echo для вывода звёздочки

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

☒ Отлично!

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

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

[Следующий шаг](#) [Решить снова](#)

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

Рис. 15: Команда echo для вывода звёздочки

Раздел 3 — Результат выполнения 10 вызовов счетчика

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

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

Напишите текст

✅ Правильно.

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 **20 009** учащихся
Из всех попыток **28%** верных

counters are: and 110

Рис. 16: Результат выполнения 10 вызовов счетчика

Раздел 3 — Нахождение НОД по алгоритму Евклида

Напишите программу. Тестируется через stdin → stdout

✓ Хорошие новости, верно!

Верно решили **18 148** учащихся
Из всех попыток **35%** верных

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

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

Рис. 17: Нахождение НОД по алгоритму Евклида

Раздел 3 — Ошибки в калькуляторе

Напишите программу. Тестируется через stdin → stdout

✓ Правильно.

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

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

```
1 #!/bin/bash
2 while [[ True ]]
3 do
4     read birinchi amal ikkinchi
5     if [[ $birinchi == "exit" ]]
6     then
7         echo "bye"
8         break
9     elif [[ "$birinchi" =~ "^[-0-9]+$" && "$ikkinchi" =~ "^[-0-9]+$" ]]
10    then
11        echo "error"
12        break
13    else
14        case $amal in
15            "+") let "result = birinchi + ikkinchi";;
16            "-") let "result = birinchi - ikkinchi";;
17            "/" ) let "result = birinchi / ikkinchi";;
18            "*" ) let "result = birinchi * ikkinchi";;
19            "%" ) let "result = birinchi % ikkinchi";;
20            "**") let "result = birinchi ** ikkinchi";;
21            *) echo "error" ; break ;;
22        esac
23        echo "$result"
24    fi
25 done
26
27
28
29
30
```

Рис. 18: Ошибки в калькуляторе

Раздел 3 — Разница между find -iname и -name

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

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

Выберите все подходящие ответы из списка

 Всё правильно.

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

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

[Следующий шаг](#) [Решить снова](#)

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

Рис. 19: Разница между find -iname и -name

Раздел 3 — Разница между -path и -name в find

Выберите все подходящие ответы из списка

✓ Здорово, всё верно.

Верно решили **18 450** учащихся
Из всех попыток **22%** 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](#).

- ☐ 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
- ☐ 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 иногда может остаться таким же

Следующий шаг Решить снова

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

Рис. 20: Разница между -path и -name в find

Раздел 3 — Поиск с -mindepth 2 и -maxdepth 3

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

☒ Хорошая работа.

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

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

[Следующий шаг](#) [Решить снова](#)

Рис. 21: Поиск с -mindepth 2 и -maxdepth 3

Раздел 3 — Команды grep -A и -B

```
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

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

[Следующий шаг](#) [Решить снова](#)

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

Рис. 22: Команды grep -A и -B

Выберите один вариант из списка

✓ Верно. Так держать!

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

☐ Будут выведены все строки файла text.txt, в которых есть только большие буквы латинского алфавита

☒ Каждая строчка будет выведена два раза

☐ Появится сообщение об ошибке

☐ На экран ничего не напечатается

Следующий шаг

Решить снова

Ваши решения

Вы получили: **1 балл**

👍 786

👎 246

Шаг 11

Следующий шаг >

Рис. 23: grep с шаблоном для строк с заглавными буквами

Раздел 3 — Замена двух заглавных букв на 'abbreviation'

не поможет).

Напишите текст

✓ Отлично!

Верно решили **16 632** учащихся
Из всех попыток **34%** верных

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

```
sed 's/[A-Z]\(2,\) /abbreviation /g' input.txt > edited.txt
```

[Следующий шаг](#) [Решить снова](#)

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

Рис. 24: Замена двух заглавных букв на 'abbreviation'

Выберите один вариант из списка

✓ Правильно, молодец!

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

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

[Следующий шаг](#) [Решить снова](#)

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

Рис. 25: Опции gnuplot для сохранения графиков

Выберите один вариант из списка

☒ Здорово, всё 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 своё решение с другими на [форуме решений](#).

- ☐ 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)
- ☐ Название – первое значение из первого столбца, нарисовано 10 точек
- ☐ Название "popame", нарисовано 10 точек
- ☐ Название – первое значение из второго столбца, нарисовано 10 точек

[Следующий шаг](#) [Решить снова](#)

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

Рис. 26: Визуализация данных в gnuplot

Напишите текст

✓ Так точно!

Верно решили **13 935** учас
Из всех попыток **44%** вер

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

Следующий шаг

Решить снова

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

Рис. 27: Подтверждение и статистика в gnuplot

Раздел 3 — Инструкция по зеркальному отражению графика в gnuplot

файле `move.rot` (т.е. **добавлять и удалять инструкции нельзя!**) таким образом, чтобы:

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

Измененный файл загрузите в форму ниже.

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

Напишите текст

✓ Верно.

Верно решили **12 854** учащихся
Из всех попыток **47%** верных


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

Следующий шаг

Решить снова

Рис. 28: Инструкция по зеркальному отражению графика в gnuplot

выберите все подходящие ответы из списка

 Так точно!

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

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

- ☒ Количество слов
- ☒ Количество строк
- ☒ Длину самой длинной строки
- ☒ Размер файла в байтах
- ☒ Количество символов

[Следующий шаг](#) [Решить снова](#)

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

Рис. 29: Множественный выбор в тесте

Раздел 3 — Команда для размера директории

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 2.0К) 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: 2.4М (также на экране может быть выведен еще и символ "", обозначающий, что это размер именно текущей директории).

Напишите текст

✓ Хорошие новости, верно!

Верно решил 16 381 учащийся
Из всех попыток 53% верных

`du -h -s`

Следующий шаг Решить снова

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

Рис. 30: Команда для размера директории

Раздел 3 — Создание нескольких директорий одной командой

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

✓ Отличное решение!

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

Следующий шаг Решить снова

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

Рис. 31: Создание нескольких директорий одной командой

Я прошла и изучила третий раздел курса по Linux на платформе Stepik.

Скриншоты иллюстрируют ключевые моменты работы с терминалом, процессами, утилитами и файлами в Linux-среде.

Дополнительно изучены команды **find**, **grep**, работа с графиками в **gnuplot**, а также использование переменных и циклов в скриптах.