

Отчёт по выполнению 3 этапа внешнего курса

Операционные системы

Бережной Иван Александрович

Содержание

1	Цель работы	6
2	Задание	7
3	Выполнение работы	8
4	Выводы	36
	Список литературы	37

Список иллюстраций

3.1	Задание 1	8
3.2	Задание 2	9
3.3	Задание 3	10
3.4	Задание 4	10
3.5	Задание 5	11
3.6	Задание 6	11
3.7	Задание 7	12
3.8	Задание 8	13
3.9	Задание 9	14
3.10	Задание 10	15
3.11	Задание 11	16
3.12	Задание 12	17
3.13	Условие задания 13	17
3.14	Задание 13	18
3.15	Задание 14	19
3.16	Задание 15	20
3.17	Задание 16	21
3.18	Условие задания 17	21
3.19	Задание 17	22
3.20	Условие задания 18	23
3.21	Задание 18	23
3.22	Задание 19	24
3.23	Задание 20	24
3.24	Задание 21	25
3.25	Задание 22	26
3.26	Задание 23	26
3.27	Задание 24	27
3.28	Задание 25	28
3.29	Задание 26	28
3.30	Задание 27	29
3.31	Задание 28	30
3.32	Задание 29	31

3.33 Задание 30	32
3.34 Задание 31	33
3.35 Задание 32	34
3.36 Задание 33	34
3.37 Задание 34	35

Список таблиц

1 Цель работы

Научиться работать с операционной системой Linux

2 Задание

Выполнить тестовые задания на основе теории

3 Выполнение работы

Выполняем третий этап: (рис. 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.20, 3.21, 3.22, 3.23, 3.24, 3.25, 3.26, 3.27, 3.28, 3.29, 3.30, 3.31, 3.32, 3.33, 3.34, 3.35, 3.36, 3.37).

Проверим, какой из вариантов сработает в редакторе и выберем его.

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 получилось!

Верно решили 32 523 учащихся
Из всех попыток 69% верных

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

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 3.1: Задание 1

Создаём текстовый файл `vim tmp.txt`, копируем туда строку “Strange_TEXT is_here. 2=2 YES!” и проверяем каждый вариант.

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

☒ Прекрасный ответ.

Верно решили **25 385** 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](#).

- ☒ В этой строке 9 "слов" (word)
- ☒ Нажимая только на W, нельзя переместить курсор на "."
- ☒ Чтобы попасть в конец строки, нужно совершить меньше нажатий на W, чем на w
- ☐ Чтобы попасть в конец строки, нужно совершить больше нажатий на W, чем на w
- ☒ После 10 нажатий на W курсор окажется там же, где бы он был после 10 нажатий на w
- ☒ В этой строке 5 "больших слов" (WORD)

Следующий шаг

Решить снова

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

Рис. 3.2: Задание 2

Проверим каждый набор клавиш и выберем верные варианты ответа.

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

☒ Правильно.

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

☐ d2dwywPp

☒ ddithree four four four five<Esc>

☒ d2w\$bifour four <Esc>

☒ xxxxxxxxwywPp

☒ d2wwyywPp

☒ d2wwifour four <Esc>

Следующий шаг

Решить снова

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

Рис. 3.3: Задание 3

Поиск и замена в Vim осуществляется командой :substitute, однако куда удобнее использовать для нее аббревиатуру :s. Общий синтаксис этой команды примерно такой:

:{пределы}s/{что заменяем}/{на что заменяем}/{опции}

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

☒ Отличное решение!

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

:%s/Windows/Linux

Следующий шаг

Решить снова

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

Рис. 3.4: Задание 4

Опять же, на практике проверим каждое утверждение.

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

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

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

Выберите все подходящие ответы из списка

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

☒ Правильно, молодец!

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](#).

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

Следующий шаг

Решить снова

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

Рис. 3.5: Задание 5

Только из набора C потому что у каждой оболочки свой буфер, который при выходе из нее буде записываться в файл истории.

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

☒ Отличное решение!

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

Следующий шаг

Решить снова

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

Рис. 3.6: Задание 6

/home/bi/file1.txt - потому что именно в этой директории мы создаем новый файл, а уже после его создания мы переходим в другую папку.

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

✓ Отличное решение!

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

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

Следующий шаг Решить снова

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

Рис. 3.7: Задание 7

В имени могут быть только буквы, цифры и подчеркивание.

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

☒ Хорошие новости, верно!

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@lable
- ☐ var.i.able
- ☒ VARiable
- ☒ variable_123
- ☒ variable123
- ☐ var i able
- ☒ _variable

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 3.8: Задание 8

\$ echo опции строка Эта команда печатает строки, которые передаются в качестве аргументов в стандартный вывод и обычно используется в сценариях оболочки для отображения сообщения или вывода результатов других команд.

var1=\$1 - обозначение переменных

var2=2`"Arguments are : \$1 =var1 \$2=\$var2" - строка печати.

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

✓ Прекрасный ответ.

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 3 балла из 3

Рис. 3.9: Задание 9

`$0` #имя скрипта

`-ge, #` больше или равно

`-gt, (>)` # больше

`-s <path> #` разм файла больше 0 `-le, #` меньше или равно

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

☒ Хорошие новости, верно!

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](#).

- ☐ \$# -gt 0
- ☒ \$# -ge 0
- ☒ 5 -ge 5
- ☒ -s \$0
- ☒ ! (4 -le 3)
- ☐ \$var1 == \$var2 && \$var1 != \$var2

Следующий шаг

Решить снова

Рис. 3.10: Задание 10

1. Задаю общую часть в каждом выводе - слово “student”: `v=student`
2. Выполняем команды для разных аргументов.
3. `res` - это результат для вывода
4. `echo “$res”` - вывести результат

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

✓ Так точно!

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
3 case $1 in
4 0) res="No ${v}s";;
5 1) res="${1} ${v}";;
6 [2-4]) res="${1} ${v}s";;
7 *) res="A lot of ${v}s";;
8 esac
9 echo "$res"
```

Следующий шаг

Решить снова

Рис. 3.11: Задание 11

a > c нет (Start, Finish)

, > c нет (Start, Finish)

b > c нет (Start, Finish)

c_d > c да (Start)

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

☒ Здорово, всё верно.

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

- ☐ 3 раза "start" и ни разу "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

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

Рис. 3.12: Задание 12

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. 3.13: Условие задания 13

Напишите программу. Тестируется через stdin → stdout

✓ Так точно!

Верно решили 21 670 учащихся
Из всех попыток 23% верных

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

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](#).

```
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
```

Следующий шаг

Решить снова

Рис. 3.14: Задание 13

1. $a = \$a$
2. $a += b$ это то же самое, что и $a = a + b$, но с символами “+=” != “=+”
3. если выражение не в скобках, но с пробелами - работать не будет.
(let a=a+b - сработает; let a = a + b - нет)

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

Отлично!

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

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 3.15: Задание 14

`programm` в `condition` выполняет стандартный вывод в терминал. Если программа не должна выводить в терминал `rm` `cp` etc... то она возвращает `true` || `false`. Нужно перенаправить вывод (для тех, которые выводят) и `programm` возвращает `true` || `false` как в случае с теми прогами которые не выводят.

Мы рассказали, что можно проверить код возврата внешней программы прямо в конструкции `if` при помощи `if "program options arguments"` (действия внутри `if` выполняются, если программа закончилась с кодом 0). Однако это **не всегда правда!** Если запуск внешней программы выводит что-то в `stdout`, то в проверку `if` поступит именно этот вывод, а не код возврата! Вы можете убедиться в этом, написав простой `bash`-скрипт с использованием, например, `if `pwd``.

Однако как быть, если хочется всё-таки запустить программу `program`, которая пишет что-то в `stdout` и потом выполнить какие-то действия если ее код возврата равен 0? Выберите **все верные** утверждения или правильно работающие конструкции `if`.

Примечание: во всех вариантах ответов, где есть кавычка, используется именно **косая кавычка** (`'`), а не обычная (`"`) или двойная (`"`).

Выберите все подходящие ответы из списка

Верно решили **21 426** учащихся
Из всех попыток **20%** верных

☒ Верно.

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

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

Следующий шаг

Решить снова

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

Рис. 3.16: Задание 15

Первая переменная локальная, и это просто пустая строка, вторая переменная - это сумма арифметической прогрессии от 1 до 10, равна 55, но при умножении на 2 даст 110.

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

Посмотрите на функцию из bash-скрипта:

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

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

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

Напишите текст

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

counters are and 110

Следующий шаг

Решить снова

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

Рис. 3.17: Задание 16

Алгоритм нахождения НОД делением 1. Большее число делим на меньшее. 2. Если делится без остатка, то меньшее число и есть НОД (следует выйти из цикла). 3. Если есть остаток, то большее число заменяем на остаток от деления. 4. Переходим к пункту 1

Условие задания 17

Рис. 3.18: Условие задания 17

Напишите программу. Тестируется через stdin → stdout

✓ 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
```

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: 4 балла из 4

Рис. 3.19: Задание 17

Калькулятор выглядит обычно - мы вводим два числа, пишем, что с ними надо сделать, и потом, учитывая случаи ошибок, выводим результат.

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"**).

Рис. 3.20: Условие задания 18

Напишите программу. Тестируется через stdin → stdout

✓ Все получилось!

Верно решили 16 980 учащихся
Из всех попыток 36% верных

Теперь вам доступен форум решений, где вы можете сравнить свое решение с другими или спросить совета.

```
1 #!/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
```

Следующий шаг

Решить снова

Ваши решения Вы получили: 5 баллов из 5

Рис. 3.21: Задание 18

-iname ищет без учета регистра, а -name в точности как в запросе. Звездочка стоит после слова - это значит после слова может быть сколько угодно символов.

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

Верно решили 20 547 учащихся
Из всех попыток 36% верных

☒ Так точно!

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

Следующий шаг Решить снова

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

Рис. 3.22: Задание 19

Проверим каждое утверждение на практике и найдём нужные ответы.

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

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

Следующий шаг Решить снова

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

Рис. 3.23: Задание 20

Текущий каталог - это `depth=1`

/home/bi -> depth=1

/home/bi/dir1 -> depth=2

/home/bi/dir1/dir2 -> depth=3

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

✓ Хорошая работа.

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

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

Следующий шаг Решить снова

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

Рис. 3.24: Задание 21

Из описания `man`: Print NUM lines of trailing context after/before matching lines “matching lines” - множественное число, строки в которых нашлось совпадение. Т.е. если идут 2...10...100 строк подряд, в которых обнаружилось совпадение, контекст будет выведен до и после этой ГРУППЫ строк, а не до и после каждой строки в этой группе.

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.

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

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

Следующий шаг

Решить снова

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

Рис. 3.25: Задание 22

Снова проверяем на практике.

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. Так держаты!

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](#).

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

Следующий шаг

Решить снова

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

Рис. 3.26: Задание 23

Напишем без `-n` и посмотрим.

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

Выберите один вариант из списка

✓ Прекрасный ответ.

Верно решили 19 784 учащихся
Из всех попыток 39% верных

- ☒ Каждая строка будет выведена два раза
- ☐ На экран ничего не напечатается
- ☐ На экран будет выведено всё содержимое файла text.txt
- ☐ Появится сообщение об ошибке

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл из 1

Рис. 3.27: Задание 24

аббревиатура АВВА отличается от двух других аббревиатур тем, что справа он неё стоит запятая без пробела: “АВВА,”. При этом по условию аббревиатура должна выглядеть как [XX] или [XXX] (и ещё больше X). Следовательно, для этой проверки надо добавить пробел квадратными скобками [] слева и, соответственно, с права.

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 выражения `a+?` (ноль или один раз по одной или более букве "a") нужно записать как `(a+)?` (при этом запись `(a)+?`, конечно же, не поможет).

Напишите текст

✓ Прекрасный ответ.

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.

kjhkjhkjhkknk

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

Следующий шаг

Решить снова

Рис. 3.28: Задание 25

Перебором посмотрим и отметим нужный вариант.

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

✓ Прекрасный ответ.

Верно решили 18 785 учащихся
Из всех попыток 51% верных

☒ -p, --persist

☐ Такой опции не существует

☐ Графики и так не закрываются автоматически при закрытии gnuplot!

☐ -s, --show-plots-after-exit

Следующий шаг

Решить снова

Ваши решения

Вы получили: 1 балл из 1

Рис. 3.29: Задание 26

Формат CSV содержит строки, где столбцы разделены запятой.

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

☒ Отлично!

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 10 точек
- ☐ Название – первое значение из второго столбца, нарисовано 10 точек
- ☐ Название "data.csv" using 1:2, нарисовано 10 точек
- ☒ Название – первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)
- ☐ Название "попате", нарисовано 10 точек

Следующий шаг

Решить снова

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

Рис. 3.30: Задание 27

Сначала идет команда установки подписей, а потом в скобках: подпись - пробел - переменная с координатой - запятая. Повторяется это количество раз соответствующее числу переменных, и без запятой (в случае с последней переменной). А подпись в свою очередь получается конкатенацией текста из задания и переменной с координатой.

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [plot.gnu](#), [plot_advanced.gnu](#), [plot_advanced2.gnu](#). Все три скрипта основаны на [этой заметке](#), данные также взяты оттуда.

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

Например, для `x1=0`, `x2=10`, `x3=20`, это были бы надписи "point 1, value 0" в точке с координатой 0 по горизонтали, "point 2, value 10" в точке с координатой 10 и "point 3, value 20" в точке с координатой 20.

Или, например, `x1=100`, `x2=150`, `x3=250`, это были бы надписи "point 1, value 100" в точке с координатой 100, "point 2, value 150" в точке с координатой 150 и "point 3, value 250" в точке с координатой 250.

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

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

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

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

Напишите текст

✓ Правильно, молодец!

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. 3.31: Задание 28

1. График строится строкой `splot x2+y2`.
2. Вращение задается строкой `zrot=(zrot+10)%360`. Значит, смещение вперед (которое было изначально) можно также задать строкой `zrot=(zrot+360+10)%360` или иначе говоря `zrot=(zrot+370)%360`. А теперь посмотрим на наше требование - чтоб вращалось в другую сторону, значит, по аналогии, необходимо вместо перебора на 10 сделать недобор. `zrot=(zrot+350)%360`.
3. Строка `pause 0.2` ставит выполнение на паузу на определенный промежуток времени. В задании сказали перерисовывать чаще, значит пауза должна быть меньше.

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

✓ Отлично!

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. 3.32: Задание 29

• r - чтение; • w - запись; • x - выполнение; • s - выполнение от имени суперпользователя; • u - владелец файла; • g - группа файла; • o - все остальные пользователи; • 0 - никаких прав;

Какая команда(ы) установят файлу `file.txt` права доступа `rwXrw-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

☒ Верно. Так держать!

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

Следующий шаг

Решить снова

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

Рис. 3.33: Задание 30

Предположим вы использовали команду `sudo` для создания директории `dir`. По умолчанию для `dir` были выставлены права доступа `rw-r--r--` (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 команды `user` из группы `group` всё-таки сможет создать файл внутри `dir`? Укажите **все верные** варианты ответов!

Примечание: считаем, что все команды выполняются от имени `user`, если явно не указано, что команда выполнена с `sudo`.

Примечание 2: мы выбрали пример с директорией, а не с файлом не случайно.

Дело в том, что если создать при помощи `sudo` файл с правами `rw-r--r--` в директории, которая принадлежит пользователю, то возникнет любопытная ситуация. С одной стороны пользователь может удалить этот файл (т.к. ему разрешено удалять **все** файлы внутри его директории) и может прочитать его содержимое (т.к. право `"r"` у файла установлено для всех), с другой стороны он не может этот файл редактировать (т.к. право `"w"` у файла есть только для `root`). При этом некоторые "умные" редакторы, например, `vim` позволяют даже редактировать этот файл, но сделают они это своеобразно: через удаление оригинала и создание копии уже с нужными правами (удалять мы можем, а раз можем читать, то и копию создать не сложно). Итого получается, что несмотря на права `rw-r--r--`, пользователь может сделать с этим файлом почти всё что угодно!

В случае же, когда речь идет о директории созданной `root`, ситуация будет проще: пользователь сможет 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

☒ Хорошая работа.

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

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

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

Рис. 3.34: Задание 31

• `wc -l` вывести количество строк • `wc -c` вывести количество байт • `wc -m` вывести количество символов • `wc -L` вывести длину самой длинной строки • `wc -w` вывести количество слов

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

Выберите все подходящие ответы из списка

✓ Всё получилось!

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

Следующий шаг

Решить снова

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

Рис. 3.35: Задание 32

Впишите в форму ниже команду, которая выведет сколько места на диске занимает текущая директория (при этом **размер** нужно вывести в **удобном для чтения формате** (например, вместо `2848 байт` надо вывести `2.8К`) и **больше** на экран выводить **ничего не** нужно). В команде указывайте **только необходимые** для выполнения задания **опции и аргументы**, лишних опций указывать не нужно!

Пример: если в текущей директории есть два файла по `800 Кбайт` и две поддиректории в каждой из которой лежит по файлу в `400 Кбайт`, то загаданная команда должна вывести на экран одно число: `2.4К` (также на экране может быть выведен еще и символ `"`, обозначающий, что это размер именно текущей директории).

Напишите текст

✓ Отличное решение!

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. 3.36: Задание 33

Создаётся три директории: `dir1`, `dir2` и `dir3`.

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** из 2

Рис. 3.37: Задание 34

4 Выводы

В ходе выполнения работы мы просмотрели курс и узнали/вспомнили возможности операционной системы Linux

Список литературы

::: Stepik