Отчёт по прохождению 3 этапа внешних курсов на stepik

Продвинутые темы

Верниковская Екатерина Андреевна

Содержание

# 1 Цель работы

Ознакомиться с функционалом операционной системы Linux.

# 2 Задание

Посмотреть много видео и на основе полученной информации пройти тестовые задания.

# 3 Выполнение 3 этапа внешних курсов на stepik

Задание №1: так как я работала с редактором vim, я помню что надо сделать для того чтобы выйти из него (рис. 1)

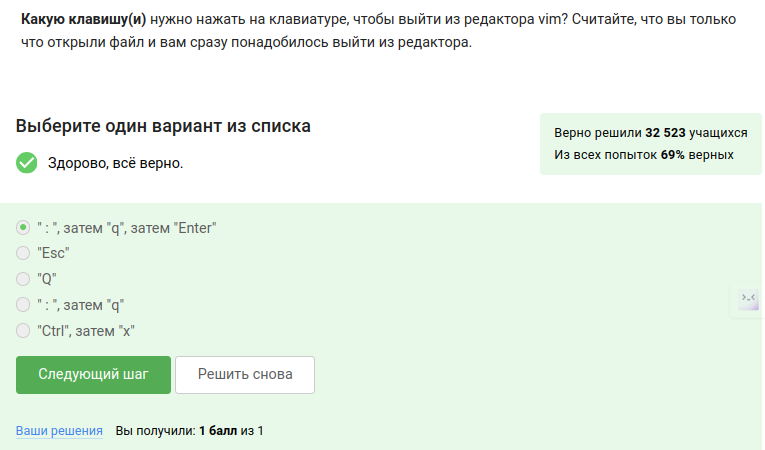


Рис. 1: Задание №1

Задание №2:

* Точка считается “маленьким словом”, так что всего их 9: Strange\_, is\_here, ., 2, =, 2, ! и два лишних пробела
* Клавиша W перемещает курсор на один символ вправо, а точка (“.”) находится на конце строки. Чтобы переместить курсор на точку, необходимо нажать W 25 раз, так как строка содержит 25 символов (включая пробелы). Однако если вы нажмете W 25 раз, курсор переместится за пределы строки. Клавиша w, с другой стороны, перемещает курсор на одно слово влево или вправо. Поскольку точка отделена от слова “YES!” пробелом, можно переместить курсор на нее, нажав w один раз.

(рис. 2)

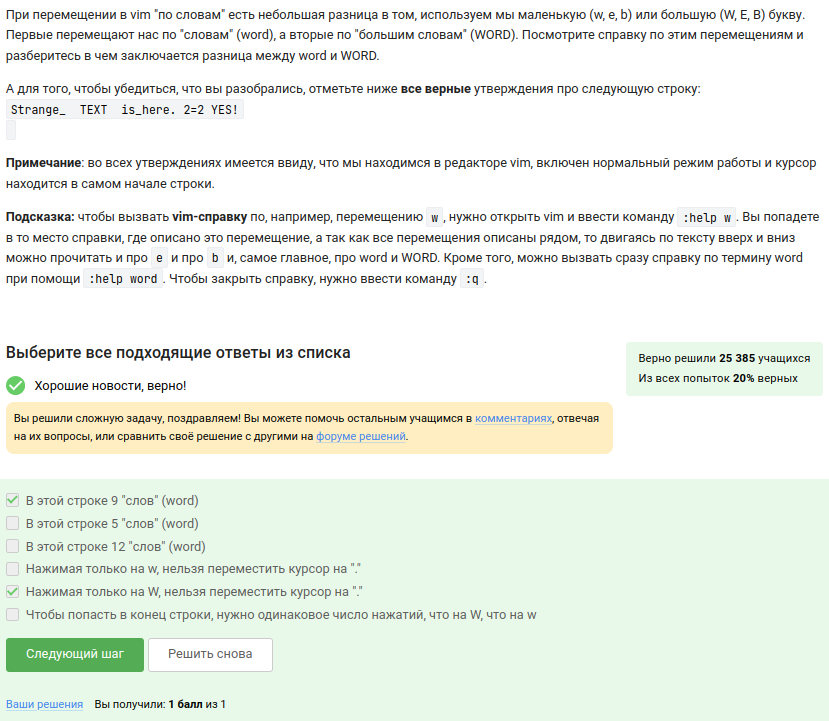


Рис. 2: Задание №2

Задание №3:

* $ — в конец текущей строки;
* w — на слово вправо;
* b — на слово влево;
* i — начать ввод перед курсором;
* p — вставка содержимого неименнованного буфера под курсором;
* P — вставка содержимого неименованного буфера перед курсором;
* yy (также Y) — копирование текущей строки в неименованный буфер;
* yy — копирование числа строк начиная с текущей в неименованный буфер;

Ответ: d2wwywPp; d2wwifour four ; ddithree four four four five

(рис. 3)

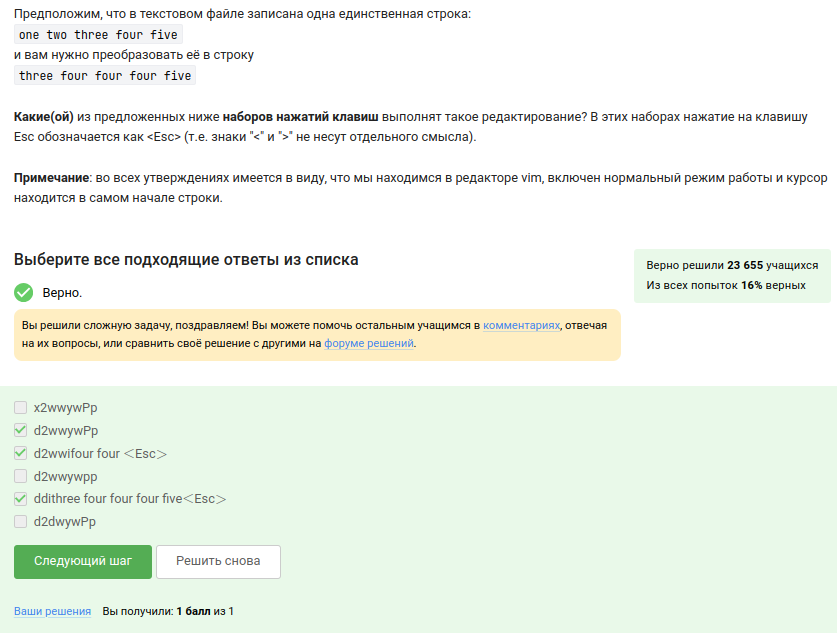


Рис. 3: Задание №3

Задание №4: поиск и замена в редакторе работают по следующей схеме: :{пределы}s/{что заменяем}/{на что заменяем}/{опции} Для замены во всем файле можно использовать символ % (рис. 4)

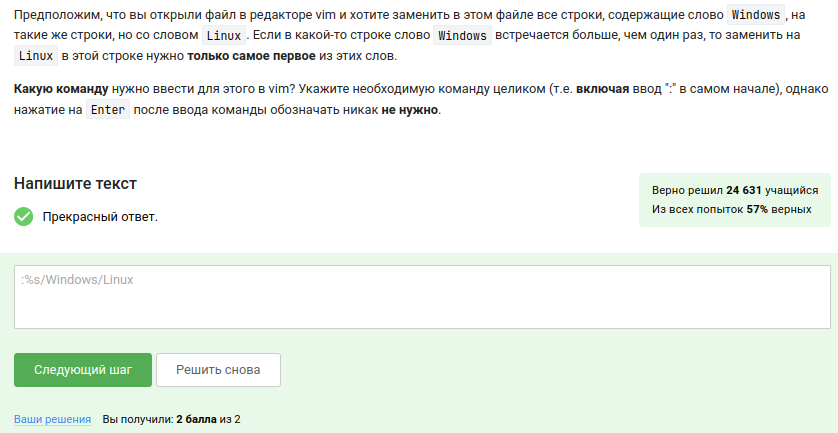


Рис. 4: Задание №4

Задание №5:

Команда $ — в конец текущей строки, W - до пробела вправо - то есть, перемещение. Нажать Esc достаточно один раз, но да ладно. Надпись visual - горит. d — используется совместно с командами перемещения. Удаляет символы с текущего положения курсора до положения после ввода команды перемещения. yy (также Y) — копирование текущей строки в буфер;

(рис. 5)

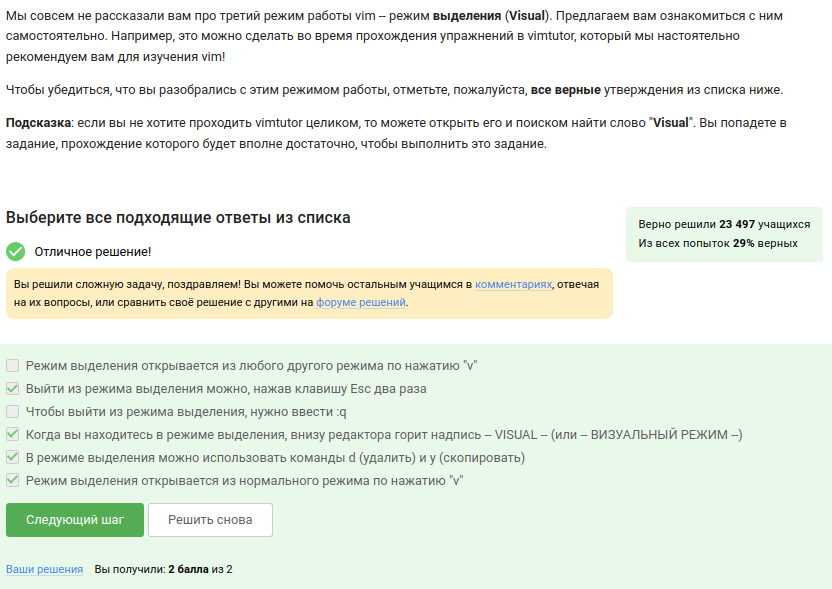


Рис. 5: Задание №5

Задание №6: только из набора С потому что у каждой оболочки свой буфер, который при выходе из нее будет записываться в файл истории (рис. 6)

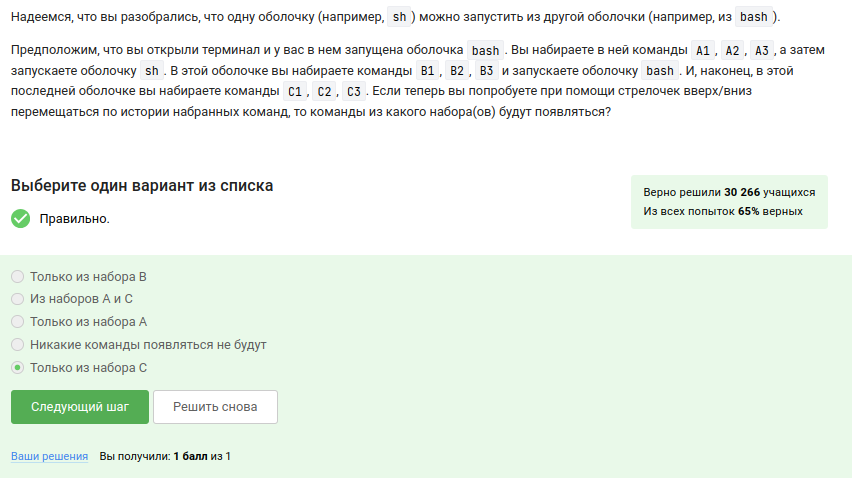


Рис. 6: Задание №6

Задание №7: /home/bi/file1.txt - потому что именно в этой директории мы создаем но- вый файл, а уже после его создания мы переходим в другую папку (рис. 7)

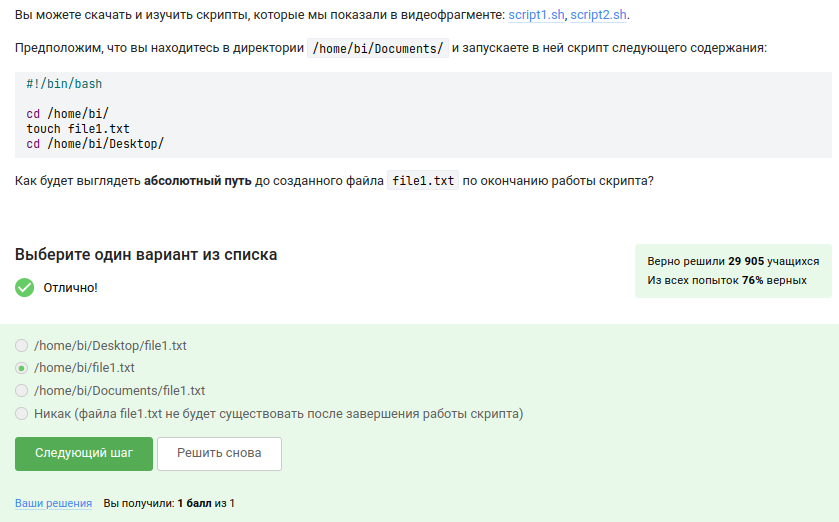


Рис. 7: Задание №7

Задание №8: в имени только буквы, цифры и подчеркивание. Больше никаких символов! (рис. 8)

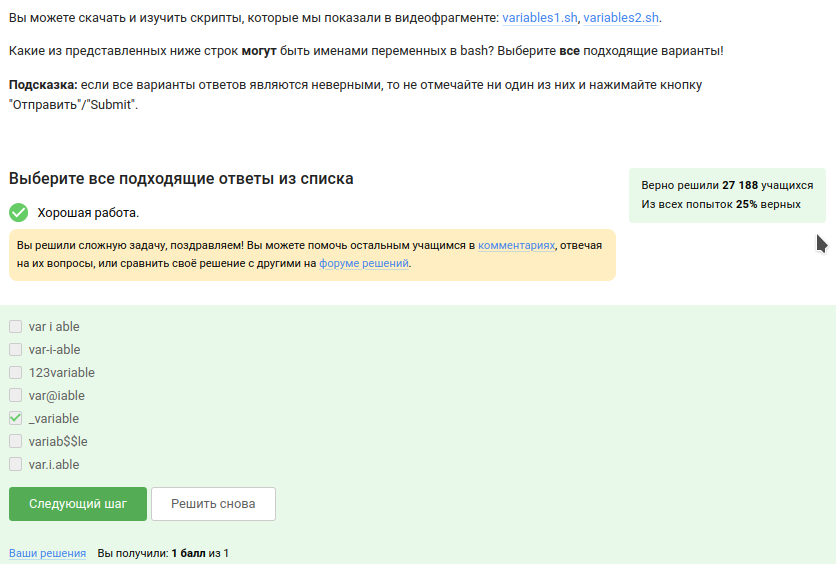


Рис. 8: Задание №8

Задание №9: создаю файл sh, пишу программу и проверяю работу командного файла (рис. 9), (рис. 10), (рис. 11), (рис. 12)

Создание файла для задания №9

Рис. 9: Создание файла для задания №9

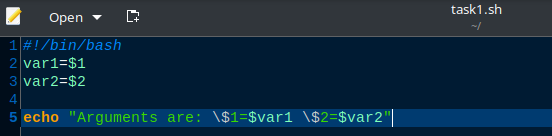


Рис. 10: Программа для задания №9

Работа программы для задания №9

Рис. 11: Работа программы для задания №9

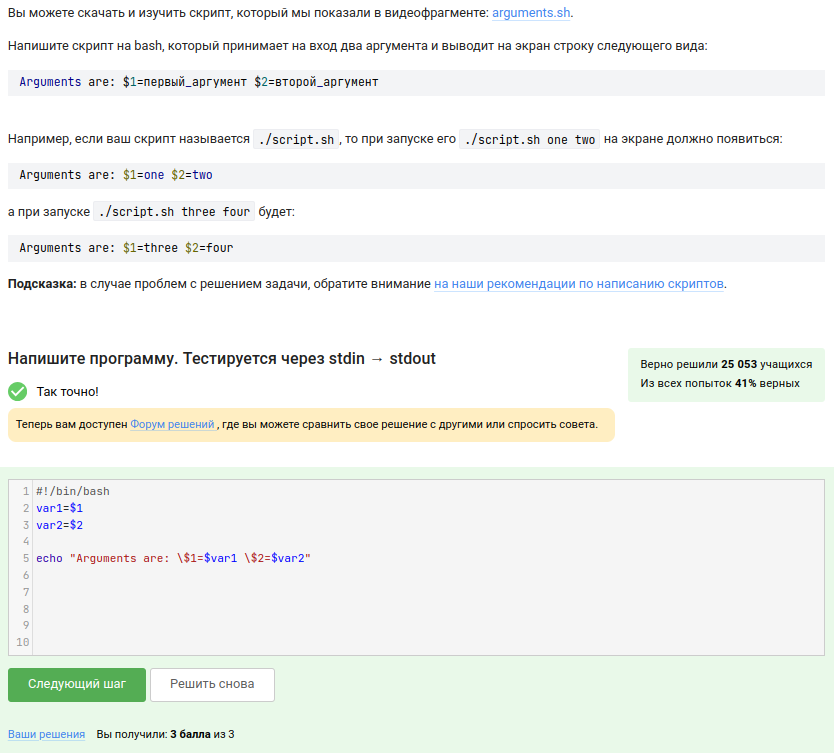


Рис. 12: Задание №9

Программа для задания №9:

#!/bin/bash  
var1=$1  
var2=$2  
  
echo "Arguments are: \$1=$var1 \$2=$var2"

Задание №10: на скрине всё видно (рис. 13)

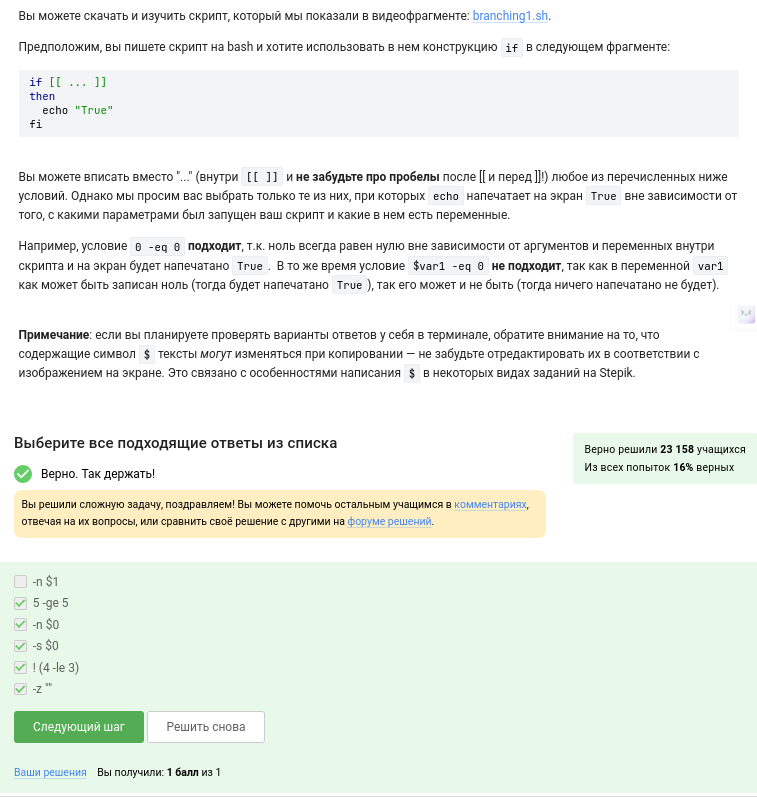


Рис. 13: Задание №10

Задание №11:

* -lt, (<) - меньше
* -gt - больше
* -eq - равно

3 не больше 5, 3 не меньше 3, 3 не равно 4. 5 не больше 5, 5 не меньше 3, 5 не равно 4. Оба раза выведет four

(рис. 14)

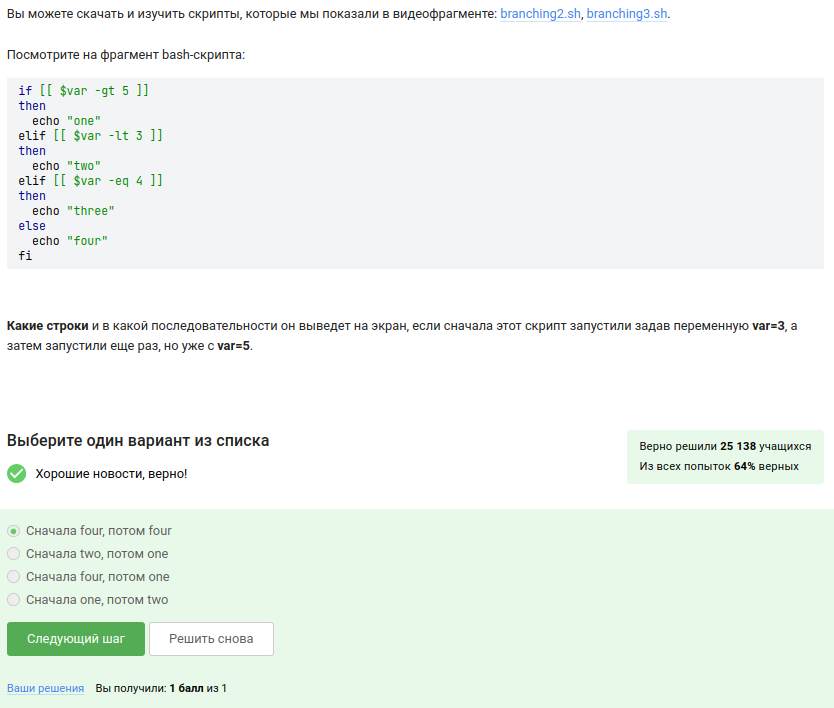


Рис. 14: Задание №11

Задание №12: создаю файл sh, пишу программу и проверяю работу командного файла (рис. 15), (рис. 16), (рис. 17), (рис. 18)

Создание файла для задания №12

Рис. 15: Создание файла для задания №12

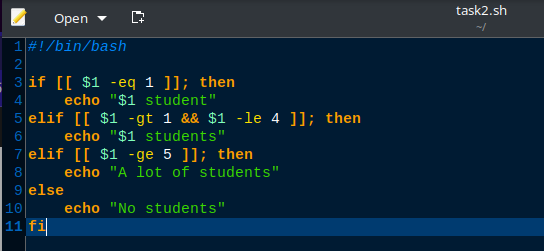


Рис. 16: Программа для задания №12

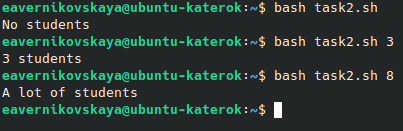


Рис. 17: Работа программы для задания №12

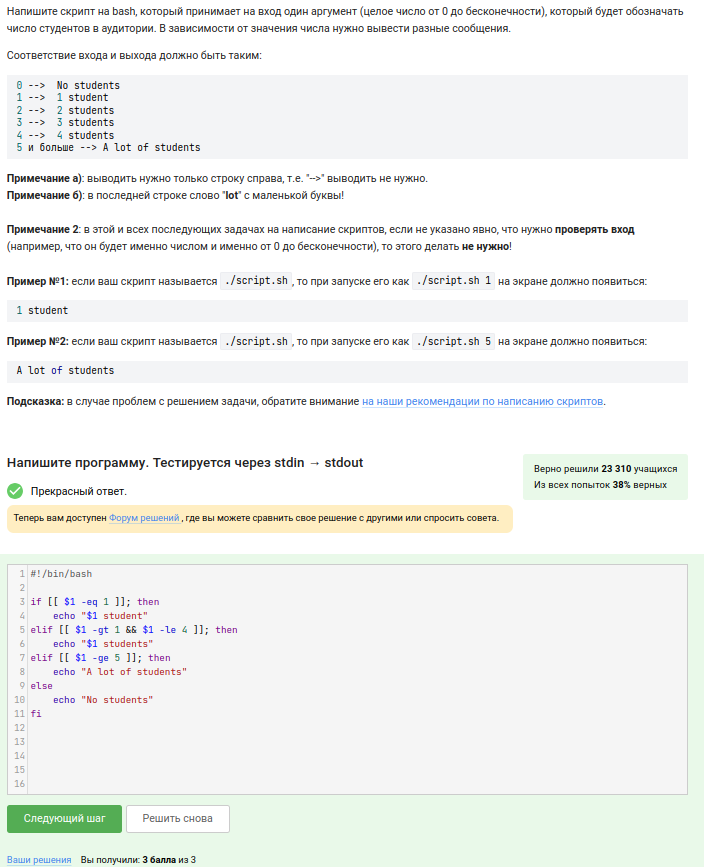


Рис. 18: Задание №12

Программа для задания №12:

#!/bin/bash  
  
if [[ $1 -eq 1 ]]; then  
 echo "$1 student"  
elif [[ $1 -gt 1 && $1 -le 4 ]]; then  
 echo "$1 students"  
elif [[ $1 -ge 5 ]]; then  
 echo "A lot of students"  
else  
 echo "No students"  
fi

Задание №13:

* (Start)
* a > c нет (Finish)
* (Start)
* , > c нет (Finish)
* (Start)
* b > c нет (Finish)
* (Start)
* , > c нет (Finish)
* (Start)
* c\_d > c да

(рис. 19)

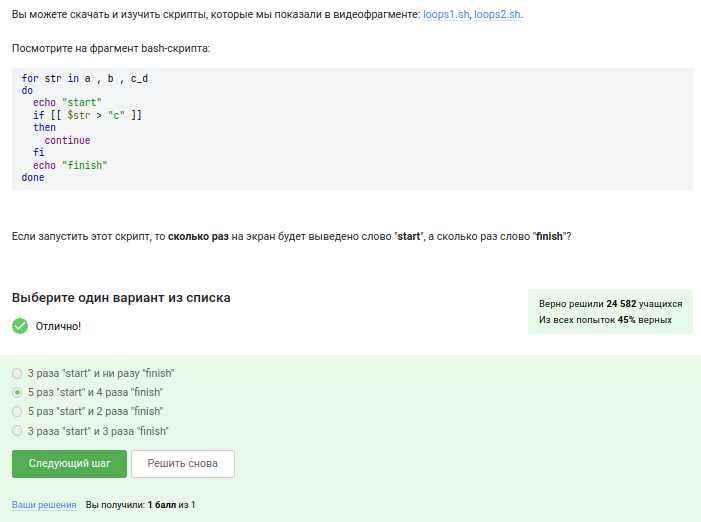


Рис. 19: Задание №13

Задание №14: создаю файл sh, пишу программу и проверяю работу командного файла (рис. 20), (рис. 21), (рис. 22), (рис. 23), (рис. 24)

Создание файла для задания №14

Рис. 20: Создание файла для задания №14

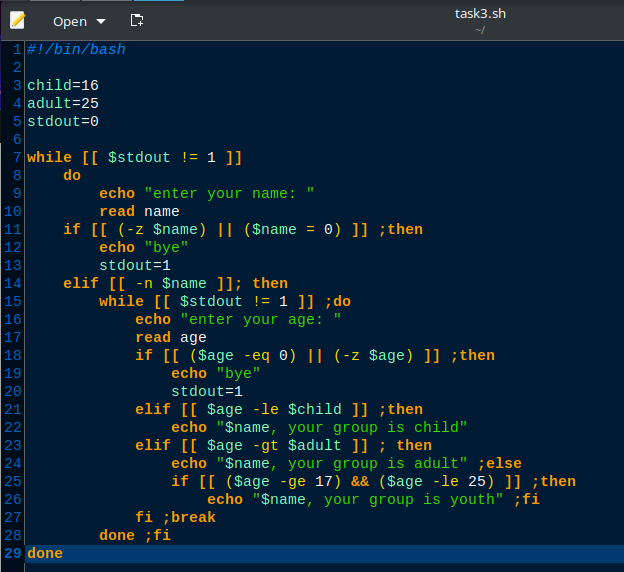


Рис. 21: Программа для задания №14

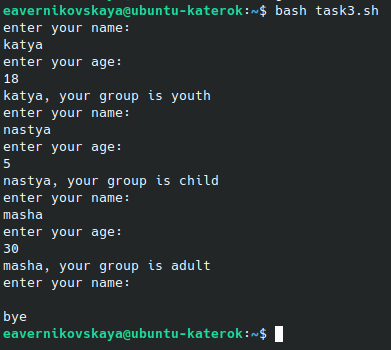


Рис. 22: Работа программы для задания №14



Рис. 23: Задание №14 (1)

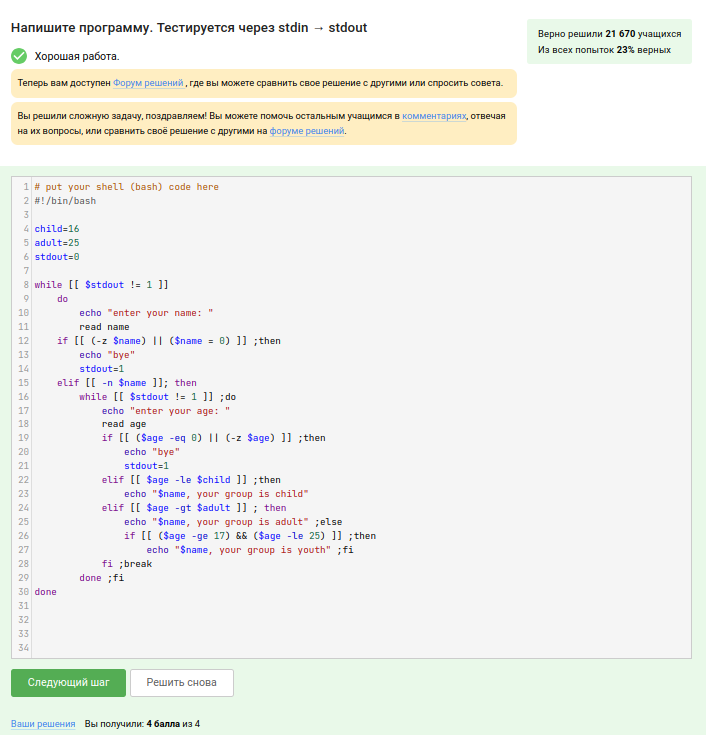


Рис. 24: Задание №14 (2)

Программа для задания №14:

#!/bin/bash  
  
child=16  
adult=25  
stdout=0  
  
while [[ $stdout != 1 ]]  
 do  
 echo "enter your name: "  
 read name  
 if [[ (-z $name) || ($name = 0) ]] ;then  
 echo "bye"  
 stdout=1  
 elif [[ -n $name ]]; then  
 while [[ $stdout != 1 ]] ;do  
 echo "enter your age: "  
 read age  
 if [[ ($age -eq 0) || (-z $age) ]] ;then  
 echo "bye"  
 stdout=1  
 elif [[ $age -le $child ]] ;then  
 echo "$name, your group is child"  
 elif [[ $age -gt $adult ]] ; then  
 echo "$name, your group is adult" ;else  
 if [[ ($age -ge 17) && ($age -le 25) ]] ;then  
 echo "$name, your group is youth" ;fi  
 fi ;break  
 done ;fi  
done

Задание №15: на скрине всё видно (рис. 25)

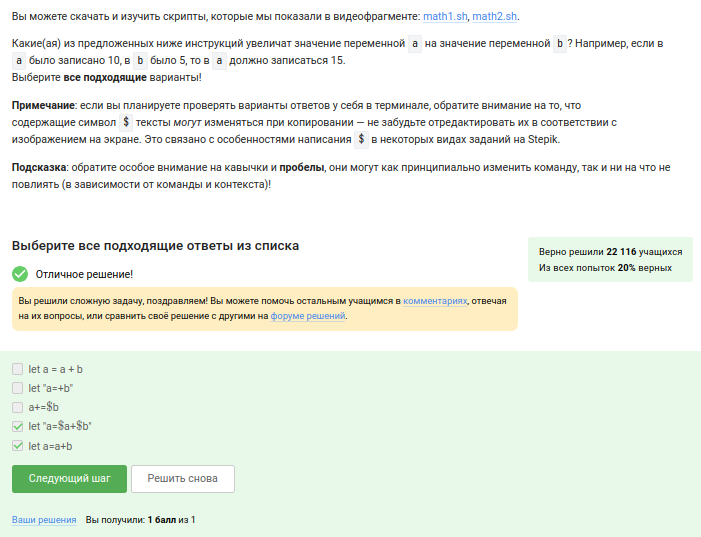


Рис. 25: Задание №15

Задание №16: выведет путь до директории, в которую мы перешли,так как “pwd”- это команда (рис. 26)

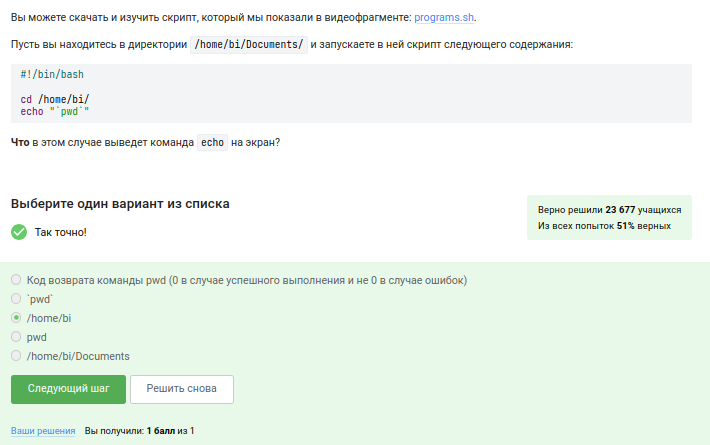


Рис. 26: Задание №16

Задание №17: на скрине всё видно (рис. 27)

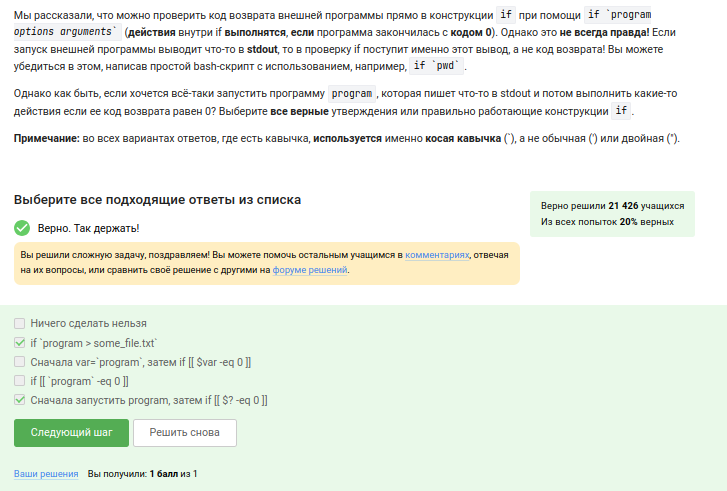


Рис. 27: Задание №17

Задание №18: создаю файл sh, пишу программу и проверяю работу командного файла (рис. 28), (рис. 29), (рис. 30), (рис. 31)

Создание файла для задания №18

Рис. 28: Создание файла для задания №18

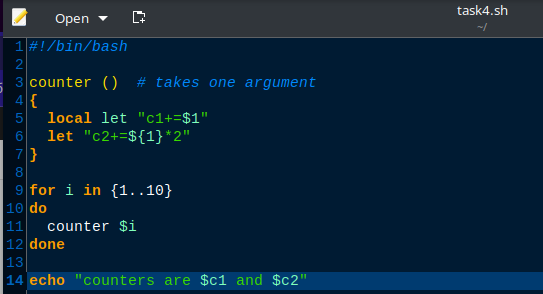


Рис. 29: Программа для задания №18

Работа программы для задания №18

Рис. 30: Работа программы для задания №18

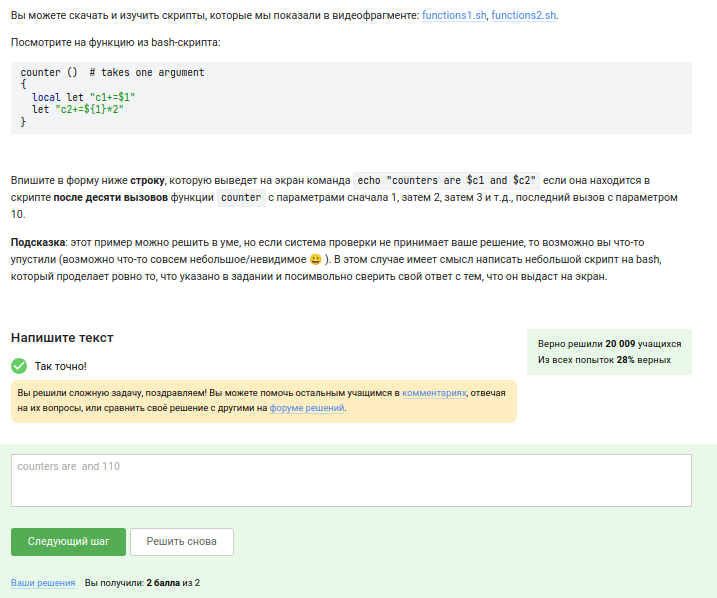


Рис. 31: Задание №18

Программа для задания №18:

#!/bin/bash  
  
counter () # takes one argument  
{  
 local let "c1+=$1"  
 let "c2+=${1}\*2"  
}  
  
for i in {1..10}  
do  
 counter $i  
done  
  
echo "counters are $c1 and $c2"

Задание №19: создаю файл sh, пишу программу и проверяю работу командного файла (рис. 32), (рис. 33), (рис. 34), (рис. 35), (рис. 36)

Создание файла для задания №19

Рис. 32: Создание файла для задания №19

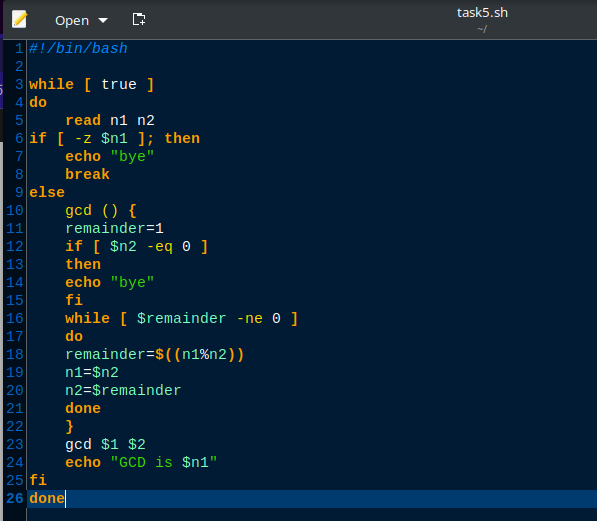


Рис. 33: Программа для задания №19

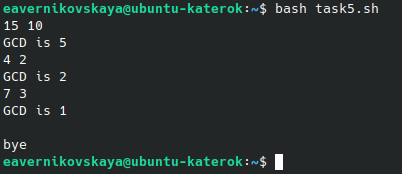


Рис. 34: Работа программы для задания №19

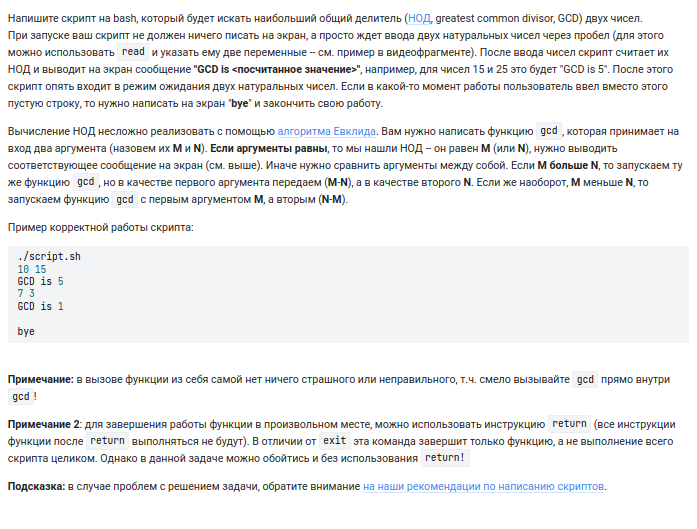


Рис. 35: Задание №19 (1)

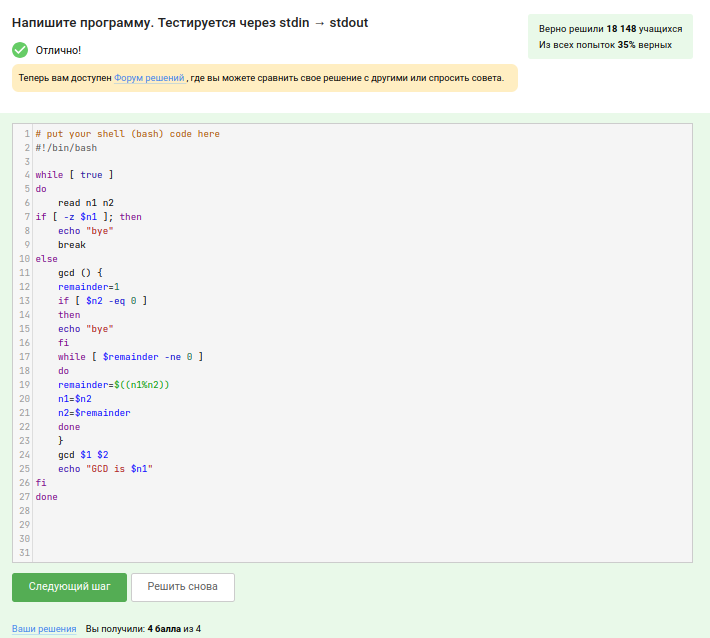


Рис. 36: Задание №19 (2)

Программа для задания №19:

#!/bin/bash  
  
while [ true ]  
do  
 read n1 n2  
if [ -z $n1 ]; then  
 echo "bye"  
 break  
else  
 gcd () {  
 remainder=1  
 if [ $n2 -eq 0 ]  
 then  
 echo "bye"  
 fi  
 while [ $remainder -ne 0 ]  
 do  
 remainder=$((n1%n2))  
 n1=$n2  
 n2=$remainder  
 done  
 }  
 gcd $1 $2  
 echo "GCD is $n1"  
fi  
done

Задание №20: создаю файл sh, пишу программу и проверяю работу командного файла (рис. 37), (рис. 38), (рис. 39), (рис. 40), (рис. 41)

Создание файла для задания №20

Рис. 37: Создание файла для задания №20

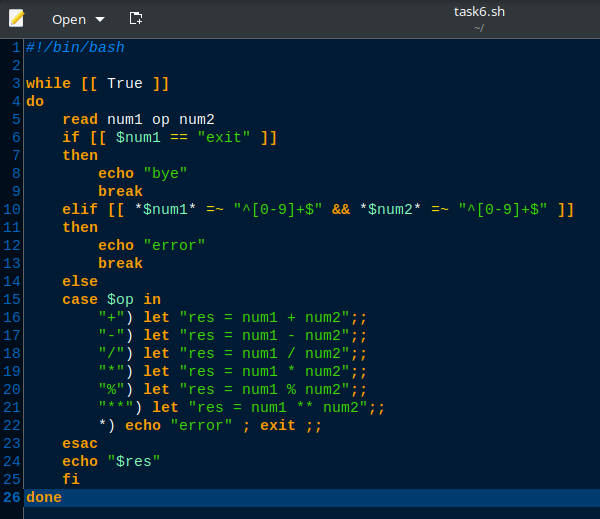


Рис. 38: Программа для задания №20

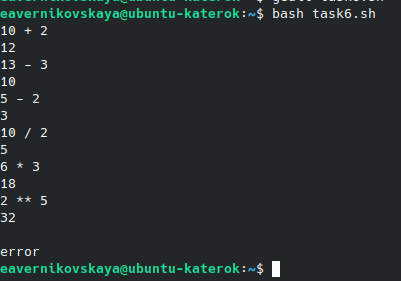


Рис. 39: Работа программы для задания №20

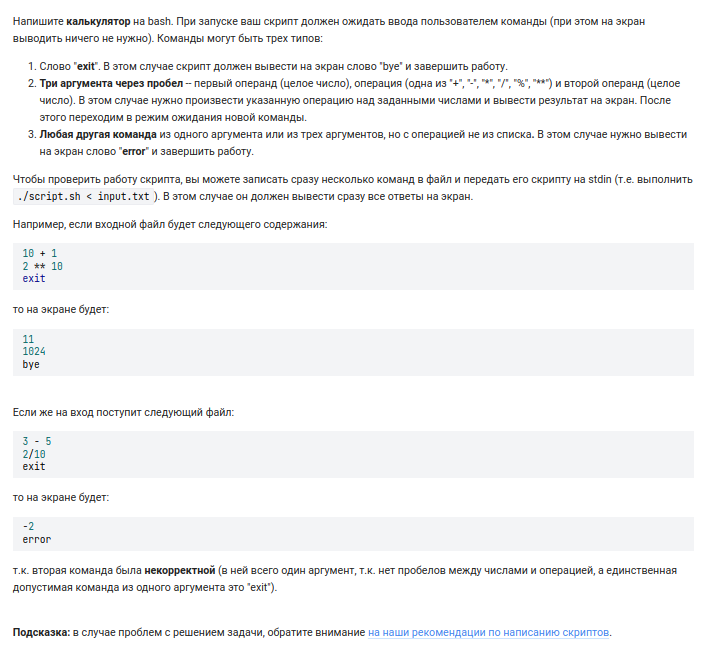


Рис. 40: Задание №20 (1)

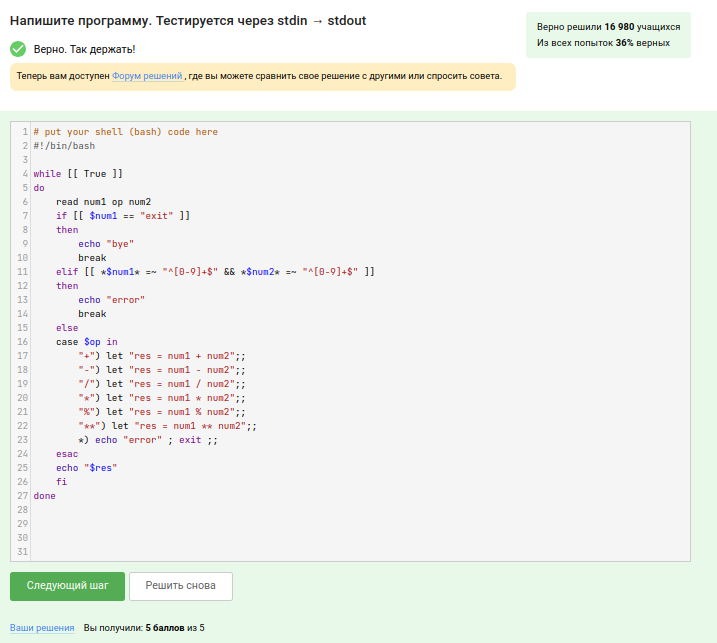


Рис. 41: Задание №20 (2)

Программа для задания №20:

#!/bin/bash  
  
while [[ True ]]  
do  
 read num1 op num2  
 if [[ $num1 == "exit" ]]  
 then  
 echo "bye"  
 break  
 elif [[ \*$num1\* =~ "^[0-9]+$" && \*$num2\* =~ "^[0-9]+$" ]]  
 then  
 echo "error"  
 break  
 else  
 case $op in  
 "+") let "res = num1 + num2";;  
 "-") let "res = num1 - num2";;  
 "/") let "res = num1 / num2";;  
 "\*") let "res = num1 \* num2";;  
 "%") let "res = num1 % num2";;  
 "\*\*") let "res = num1 \*\* num2";;  
 \*) echo "error" ; exit ;;  
 esac  
 echo "$res"  
 fi  
done

Задание №21: -iname ищет без учета регистра, а -name в точности как в запросе. Звездочка стоит после слова, значит после слова бесконечное количество символов, до слова символов не должно быть. (рис. 42)



Рис. 42: Задание №21

Задание №22: на скрине всё видно (рис. 43)

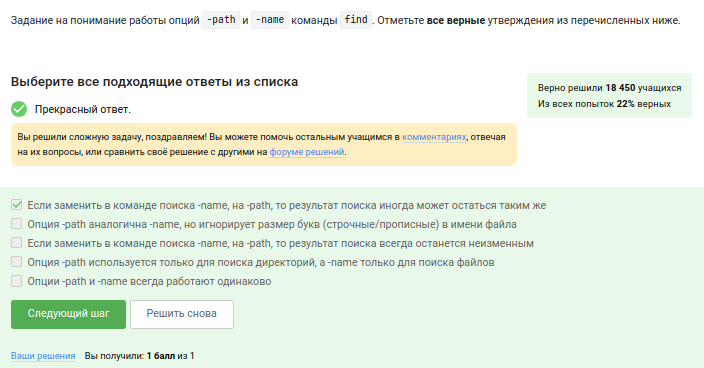


Рис. 43: Задание №22

Задание №23: текущий каталог - это depth=1, а остальное считается просто:

/home/bi -> depth=1 /home/bi/dir1 -> depth=2 /home/bi/dir1/dir2 -> depth=3 etc.

(рис. 44)

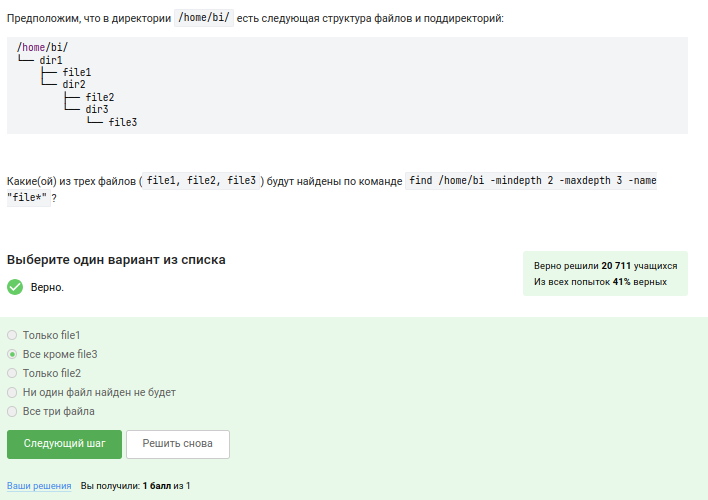


Рис. 44: Задание №23

Задание №24: на скрине всё видно (рис. 45)



Рис. 45: Задание №24

Задание №25: создаю файл txt и записываю в него строчки, показанные среди вариантов ответа. Далее использую команду *grep -E “[xklXKL]?[uU]buntu$” text.txt* и получаю ответ (рис. 46), (рис. 47), (рис. 48), (рис. 49)

Создание файла

Рис. 46: Создание файла

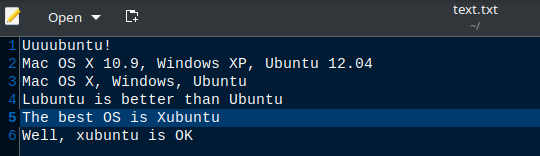


Рис. 47: Файл



Рис. 48: Вывод ответа

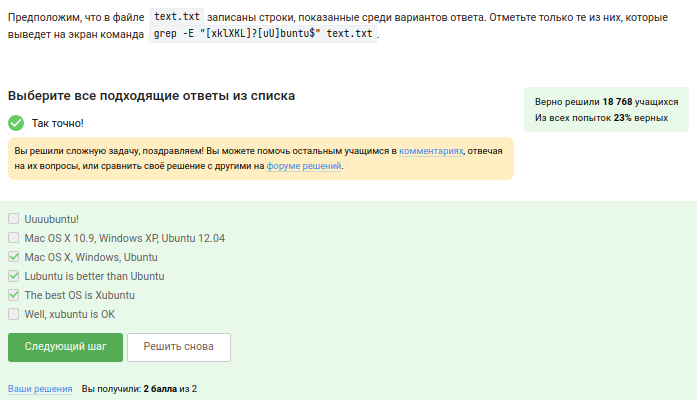


Рис. 49: Задание №25

Задание №26: опция -n отключает автоматическую печать, что означает, что строки, которые вы специально не указываете на печать, не печатаются, а строки, которые вы явно указываете на печать (например, с помощью p), печатаются только один раз (рис. 50), (рис. 51)

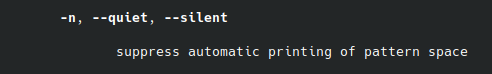


Рис. 50: Справка по команде sed

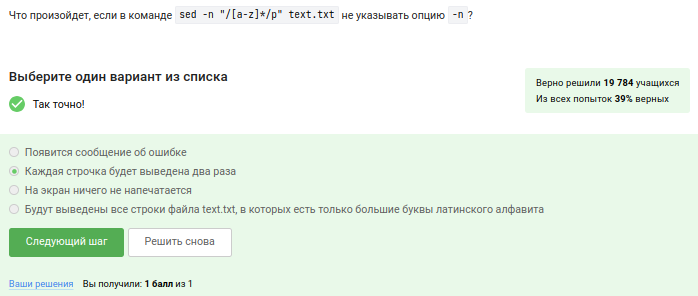


Рис. 51: Задание №26

Задание №27: создаю файл sh и пишу нужную программу. Далее ссоздаю файл txt и записываю в него строчки из примера. Далее проверяю работу командного файла. В процессе создался файл edited.txt, где абривиатуры заменены на слово abbreviation (рис. 52), (рис. 53), (рис. 54), (рис. 55), (рис. 56), (рис. 57), (рис. 58)

Создание файла для задания №27

Рис. 52: Создание файла для задания №27

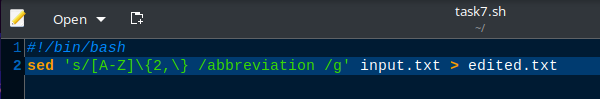


Рис. 53: Программа для задания №27

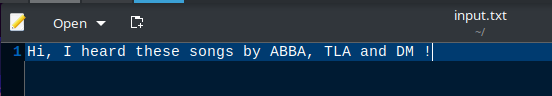


Рис. 54: Файл input.txt

ПРоверка работы программы для задания №27

Рис. 55: ПРоверка работы программы для задания №27

Открытие созданного фала

Рис. 56: Открытие созданного фала

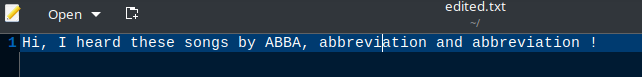


Рис. 57: Файл edited.txt

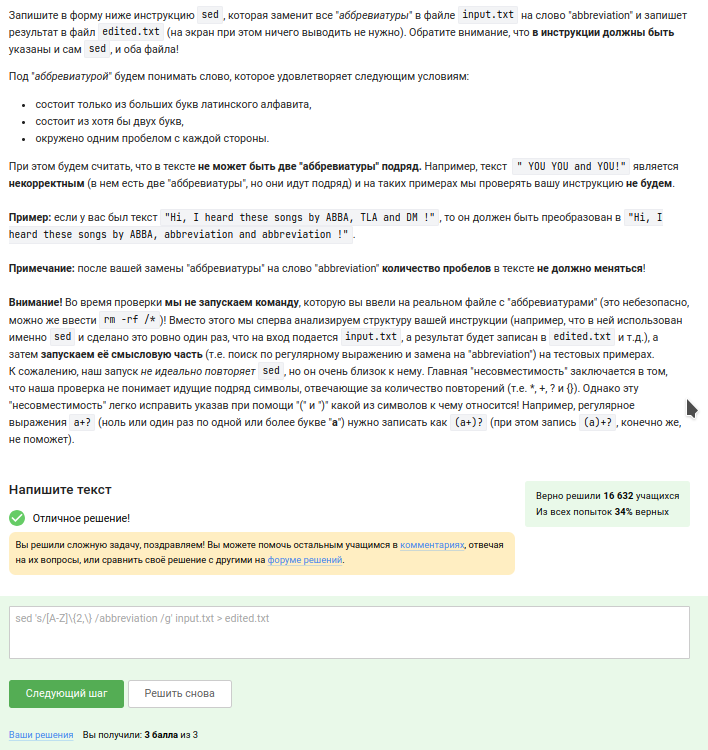


Рис. 58: Задание №27

Программа для задания №27:

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

Задание №28: -p, –persist позволяет окнам графиков сохраняться после выхода из основной программы gnuplot (рис. 59)

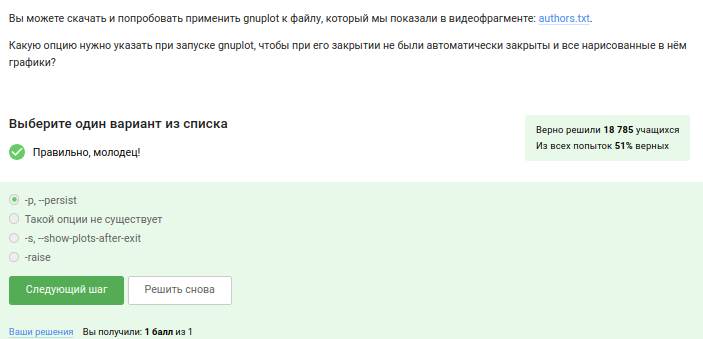


Рис. 59: Задание №28

Задание №29: на скрине всё видно (рис. 60)

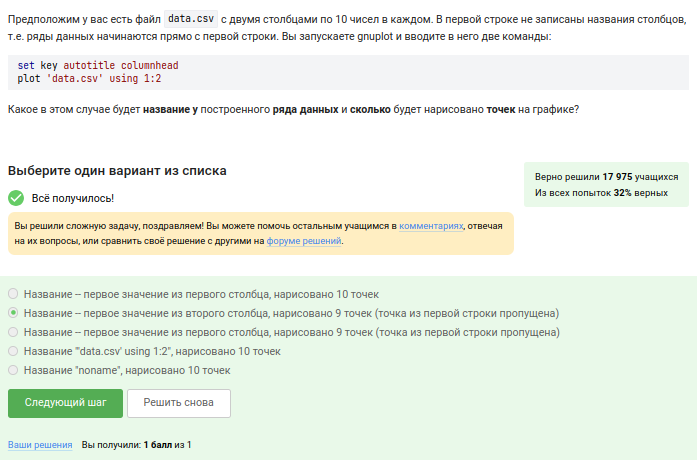


Рис. 60: Задание №29

Задание №30: на скрине всё видно (рис. 61)

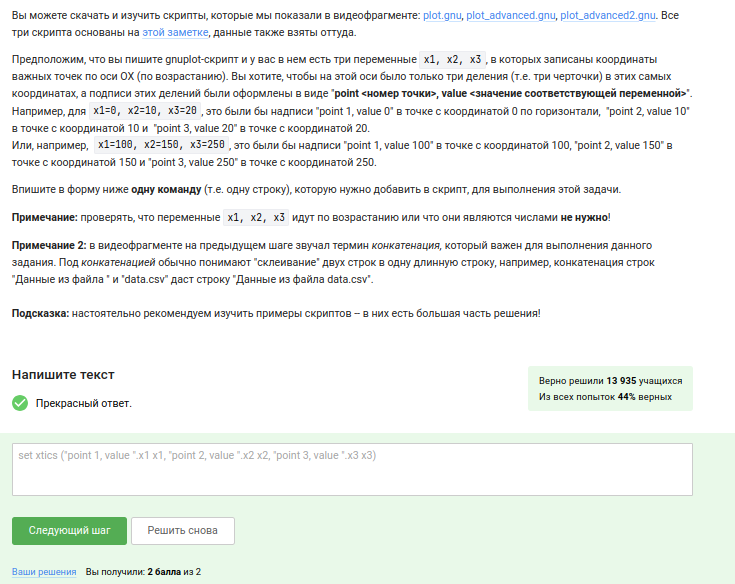


Рис. 61: Задание №30

Задание №31: создала файлы animated.gnu и move.rot. Написала нужные программы и далее проверила с помощью команды *gnuplot –persist animated.gnu* (рис. 62), (рис. 63), (рис. 64), (рис. 65), (рис. 66), (рис. 67), (рис. 68)

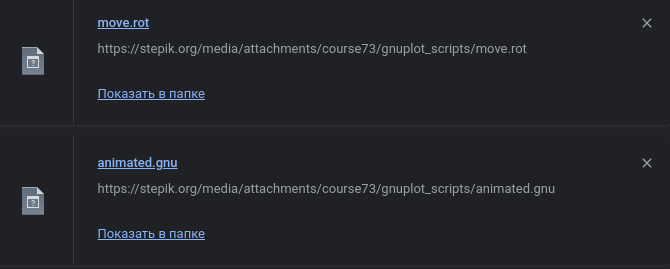


Рис. 62: Скачивание нужных файлов

Создание файла animated.gnu

Рис. 63: Создание файла animated.gnu

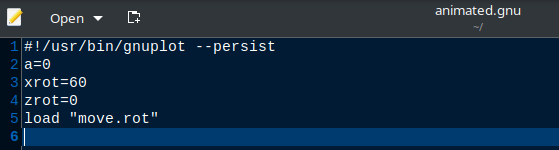


Рис. 64: Программа для animated.gnu

Создание файла move.rot

Рис. 65: Создание файла move.rot

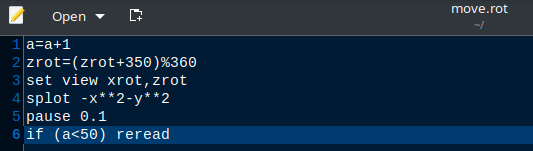


Рис. 66: Программа для move.rot

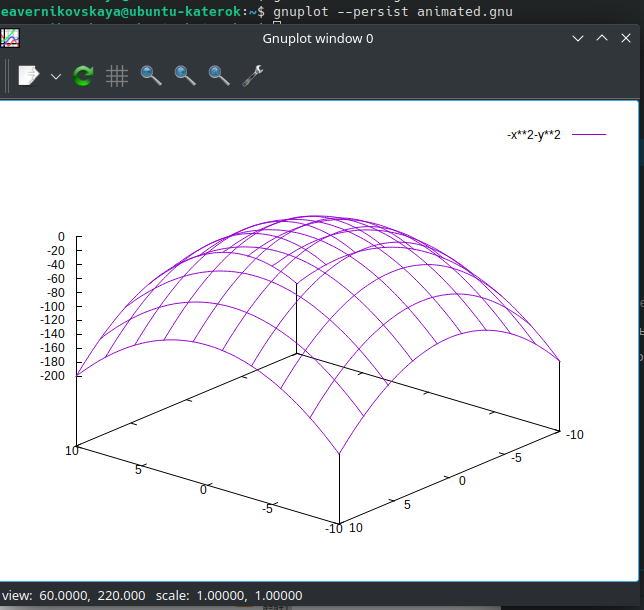


Рис. 67: Работа программы для задания №31

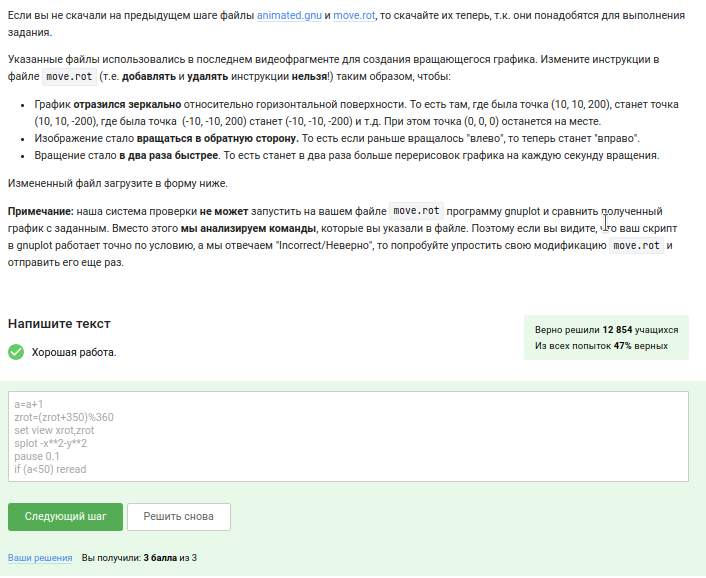


Рис. 68: Задание №31

Программы для задания №31:

1. animated.gnu

#!/usr/bin/gnuplot --persist  
a=0   
xrot=60   
zrot=0   
load "move.rot"

1. move.rot

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

Задание №32: на скрине всё видно (рис. 69)

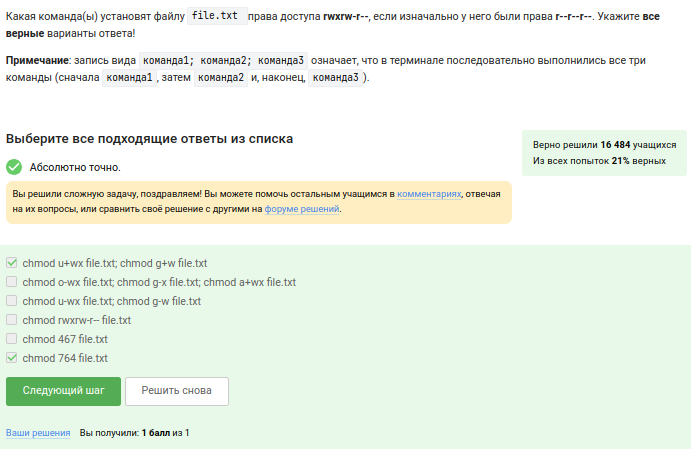


Рис. 69: Задание №32

Задание №33: на скрине всё видно (рис. 70)

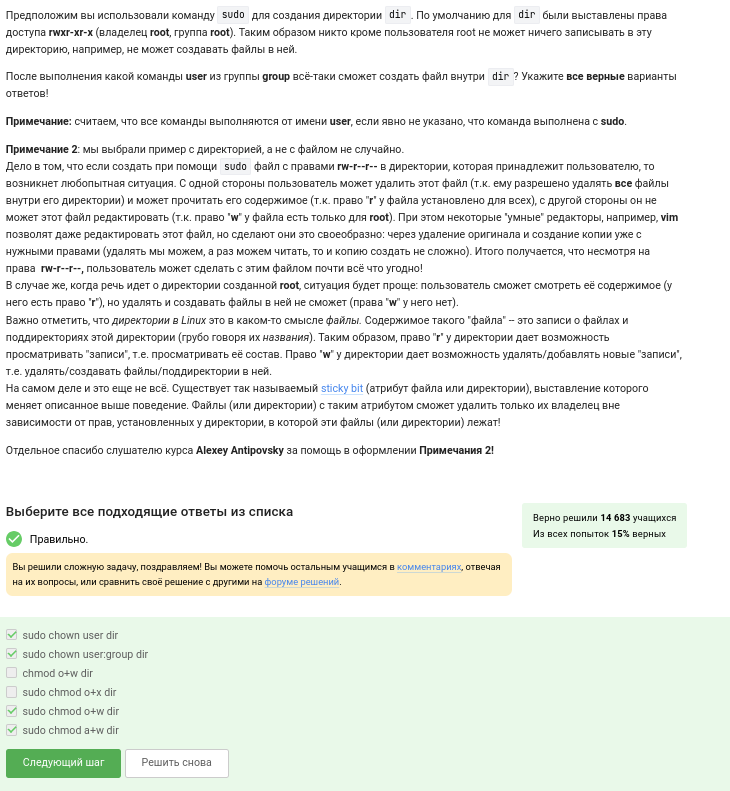


Рис. 70: Задание №33

Задание №34:

* 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

(рис. 71)

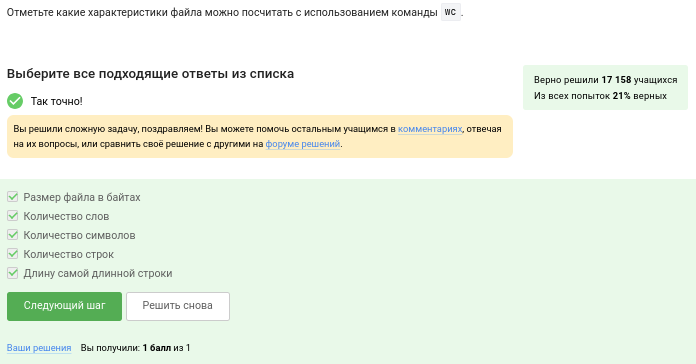


Рис. 71: Задание №34

Задание №35: на скрине всё видно (рис. 72), (рис. 73)

Команда du -h -s

Рис. 72: Команда du -h -s

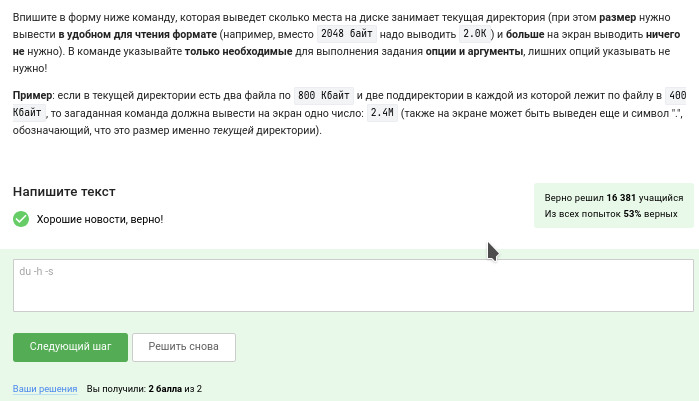


Рис. 73: Задание №35

Задание №36: на скрине всё видно (рис. 74)

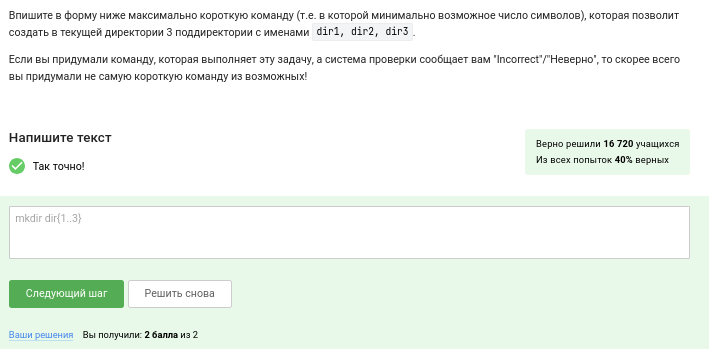


Рис. 74: Задание №36

# 4 Выводы

В ходе выполнения 3 этапа внешних курсов на stepik я освоила linux, научилась пользоваться редактором vim, научилась писать скрипты на bash, а также строить графики в gnuplot.

# 5 Список литературы

1. Курс на stepik. Продвинутые темы [Электронный ресурс] URL: https://stepik.org/course/73/syllabus?id=1098930