Отчет о прохождении 3 этапа внешних курсов

Продвинутые темы

Арфонос Дмитрий, НММбд-03-23

Содержание

# 1 Цель работы

Ознакомиться с функционалом операционной системы Linux.

# 2 Задание

Просмотреть видео и на основе полученной информации пройти тестовые задания.

# 3 Теоретическое введение

Линукс - в части случаев GNU/Linux — семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения. Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов — в форме, готовой для установки и удобной для сопровождения и обновлений, — и имеющих свой набор системных и прикладных компонентов, как свободных, так и проприетарных.

# 4 Выполнение лабораторной работы

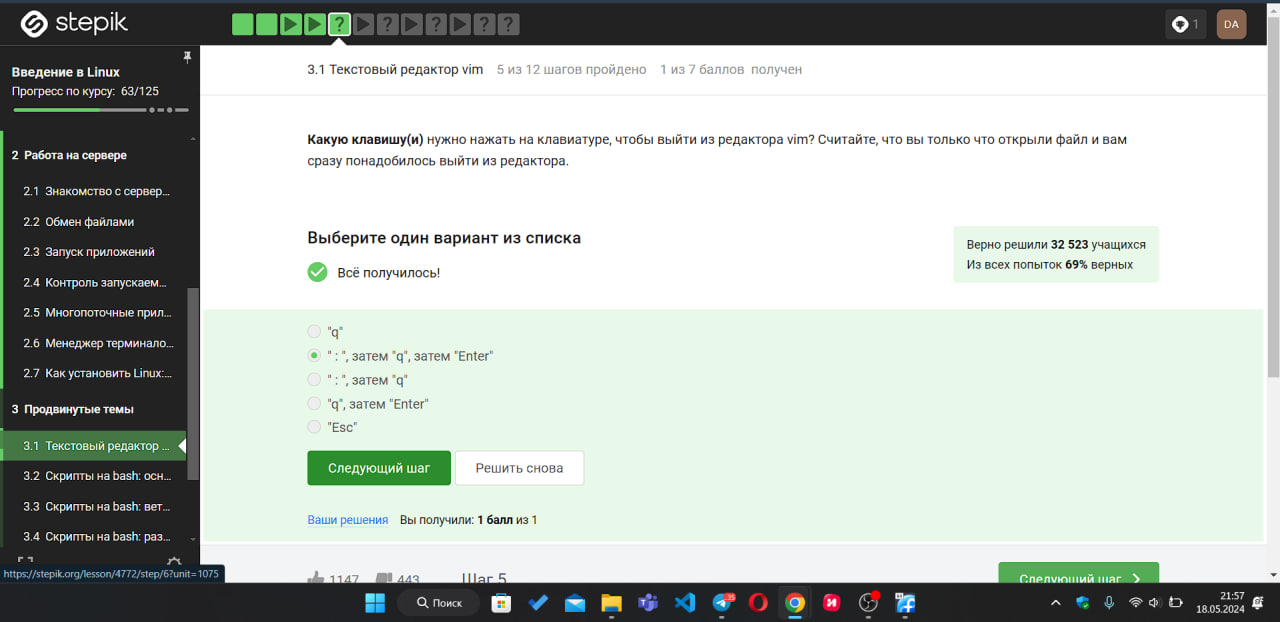


Figure 1: Задание 1

Стоит упомянуть, что у редактора vim есть туториал, который позволяет разобраться с командами, необходимыми для стандартной работы. За выход из редактора отвечают следующие команды:

* ZQ - выйти без сохранения
* :q! - выйти без сохранения
* ZZ - записать файл и выйти (если файл не изменяли, то записываться он не будет)
* :wq - записать файл и выйти
* :x - записать файл и выйти
* :w - записать файл
* :sav filename - “сохранить как”
* :w filename - “сохранить как”
* :w! - записать файл

Как мы видим, вариантов много, при этом каждый сможет найти тот, который подойдёт под конкретную ситуацию.

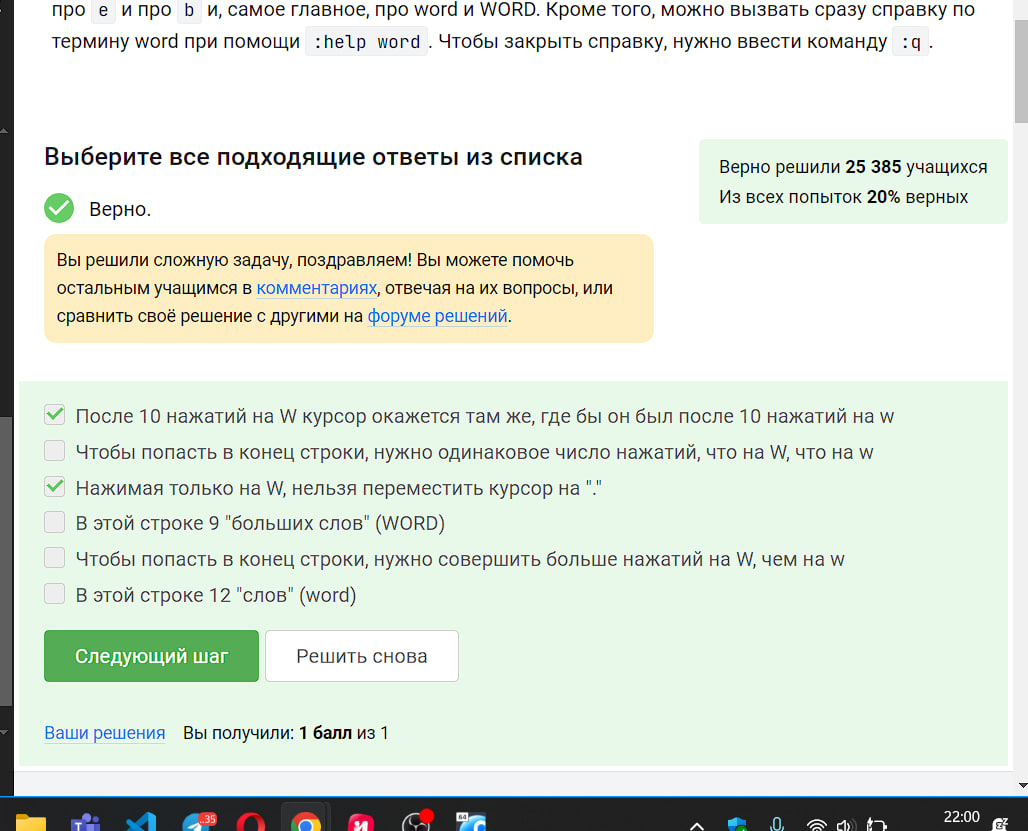


Figure 2: Задание 2

Strange\_ TEXT is\_here. 2=2 YES!

Точка считается “маленьким словом”, так что всего их 9: Strange\_, is\_here, ., 2, =, 2, ! и два лишних пробела.

И если посчитать нажатия на w и на W, то действительно после 10 штук попадем в одно место. 10 нажатий на W, это то же самое, что и 10 нажатий на w,

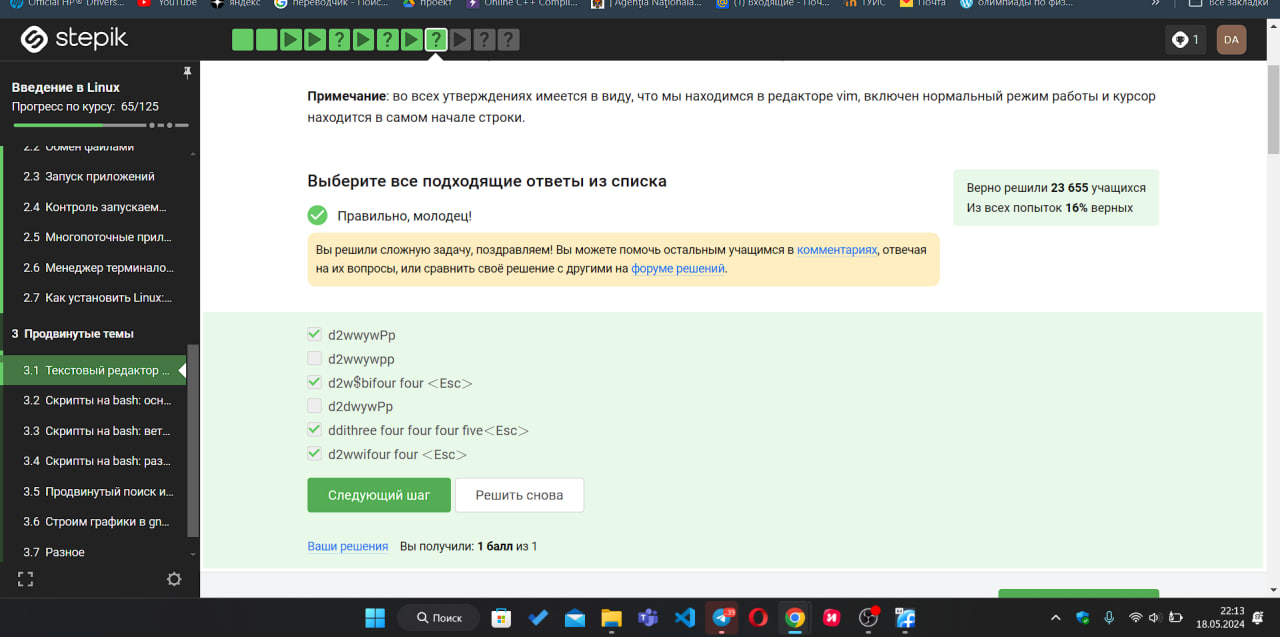


Figure 3: Задание 3

d2wwifour four <<Esc>>

d2wwywPp

d2w$$bifour four <<Esc>>

* $ — 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;

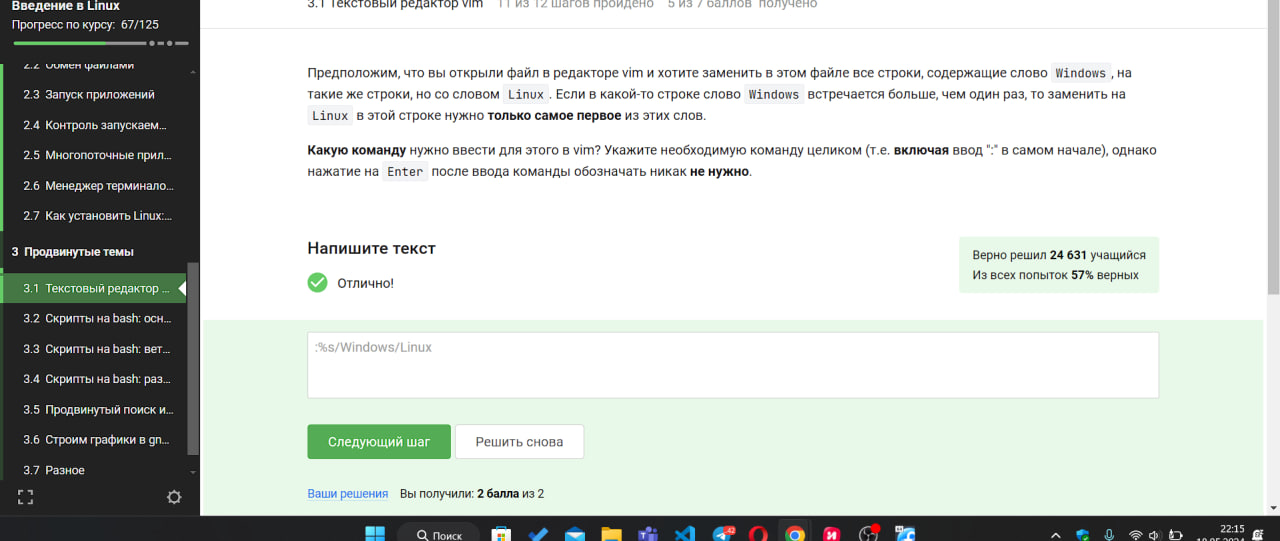


Figure 4: Задание 4

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 %.

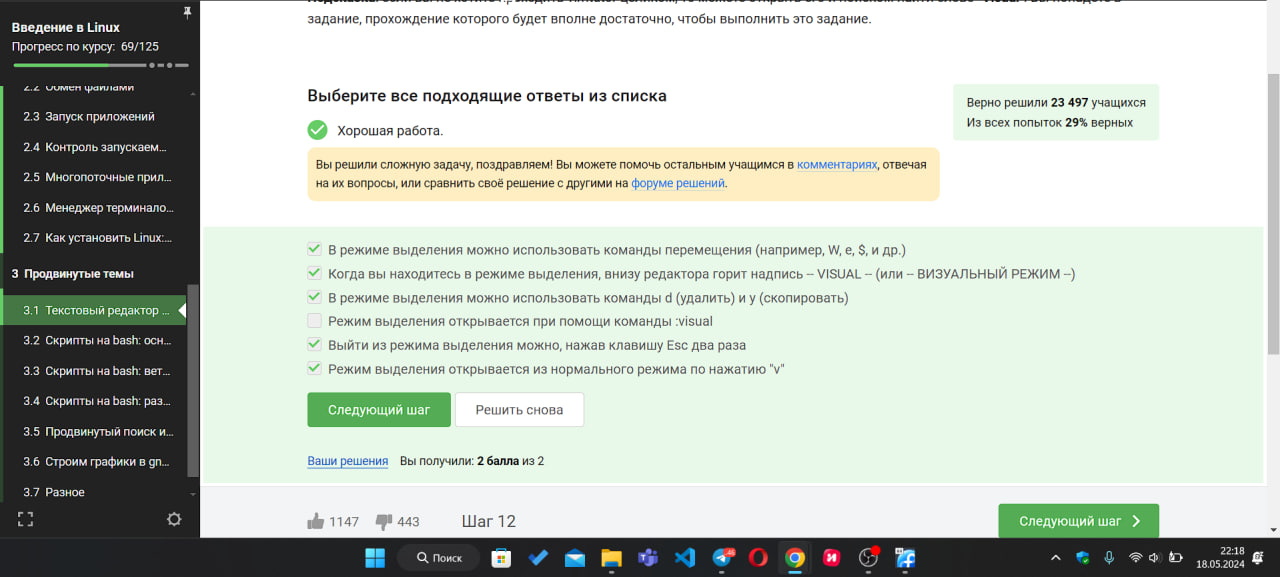


Figure 5: Задание 5

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;

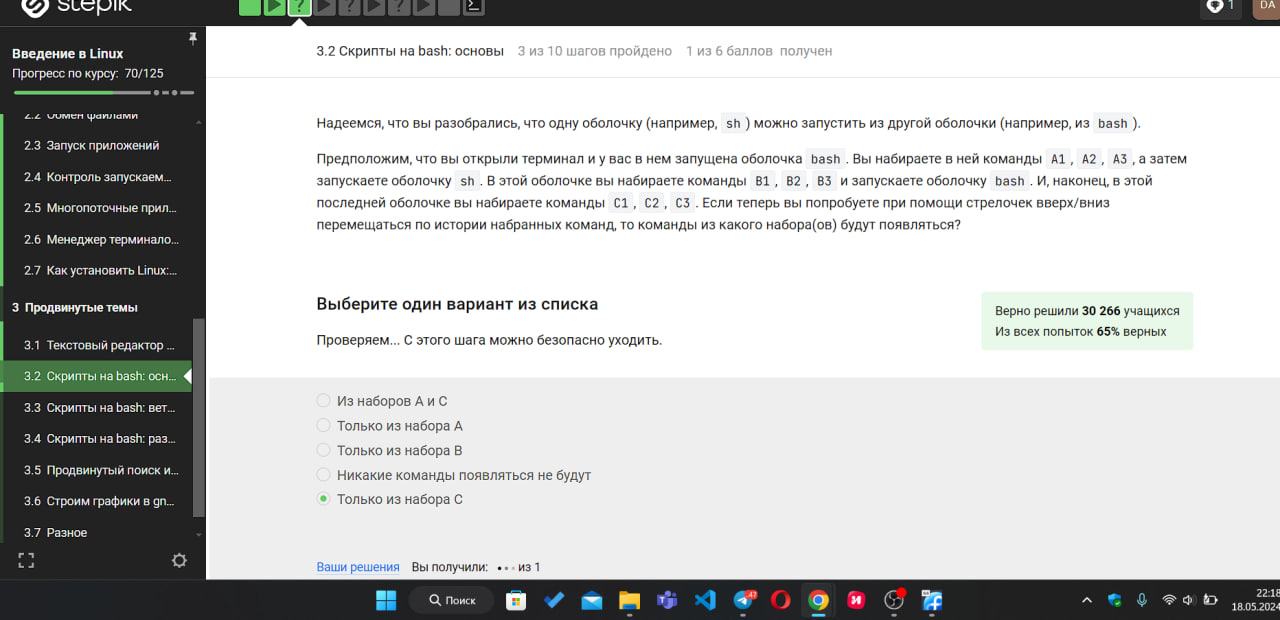


Figure 6: Задание 6

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

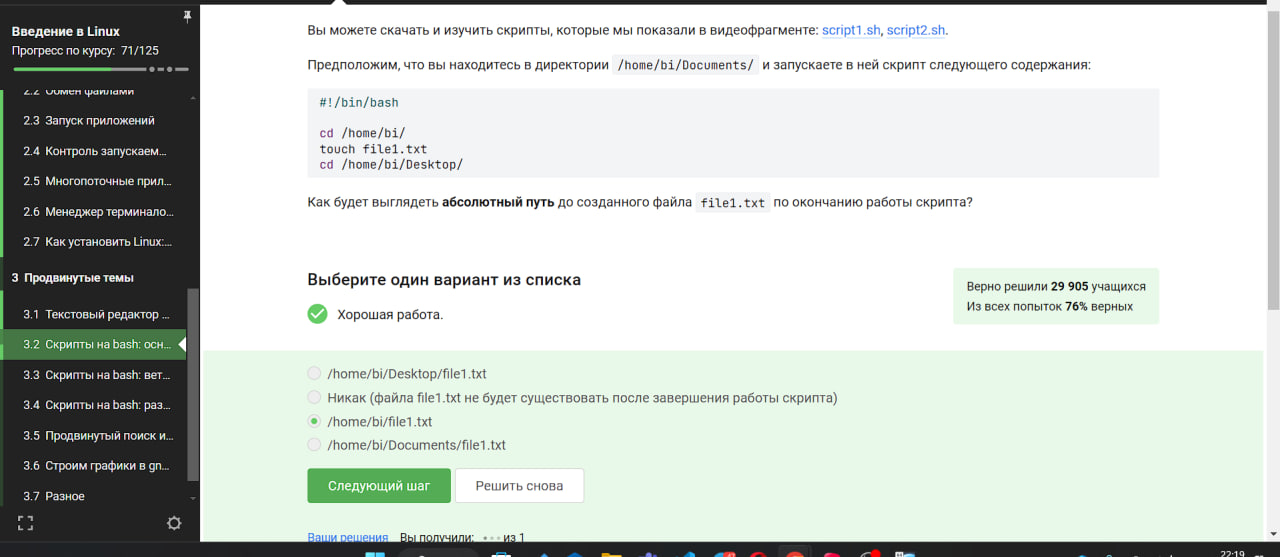


Figure 7: Задание 7

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

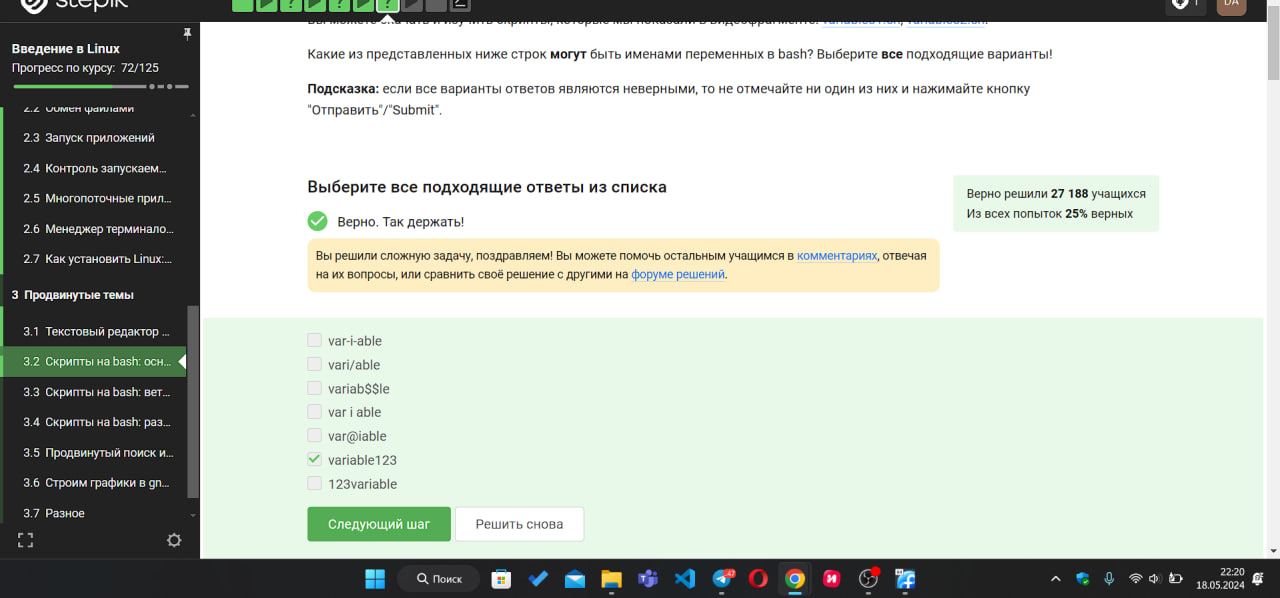


Figure 8: Задание 8

Имя не может начинаться с цифры, содержать специальные символы или пробелы.

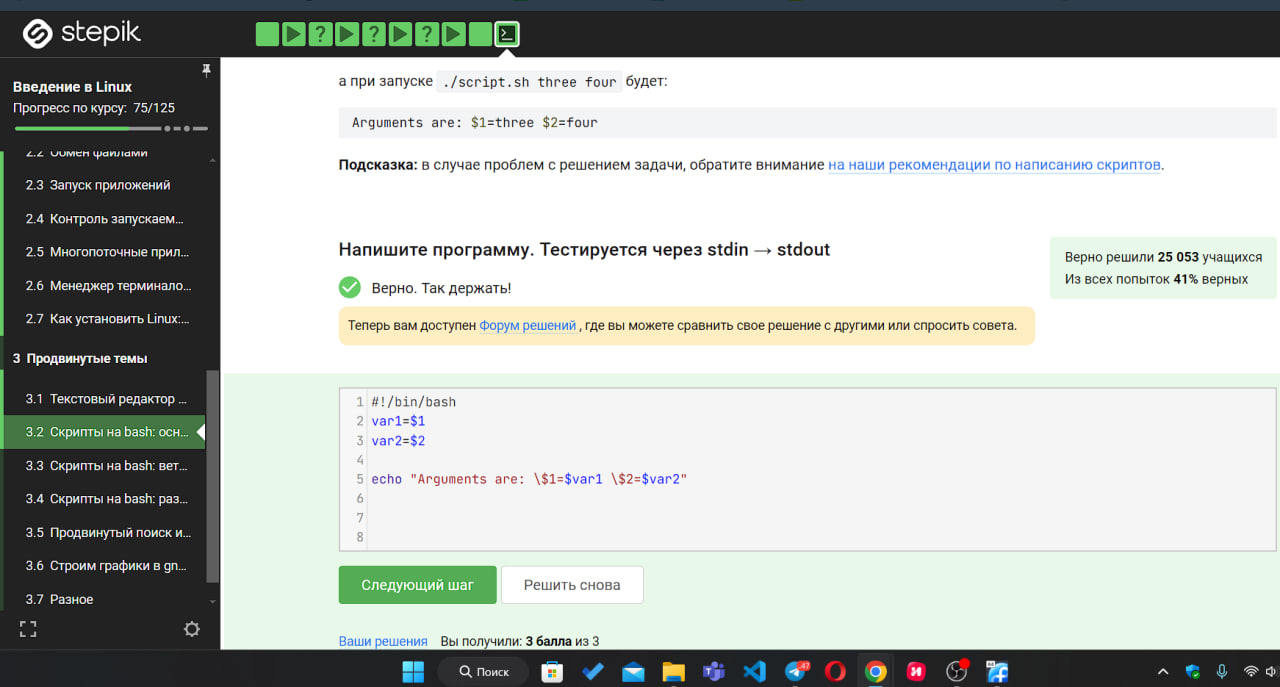


Figure 9: Задание 9

$ 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

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

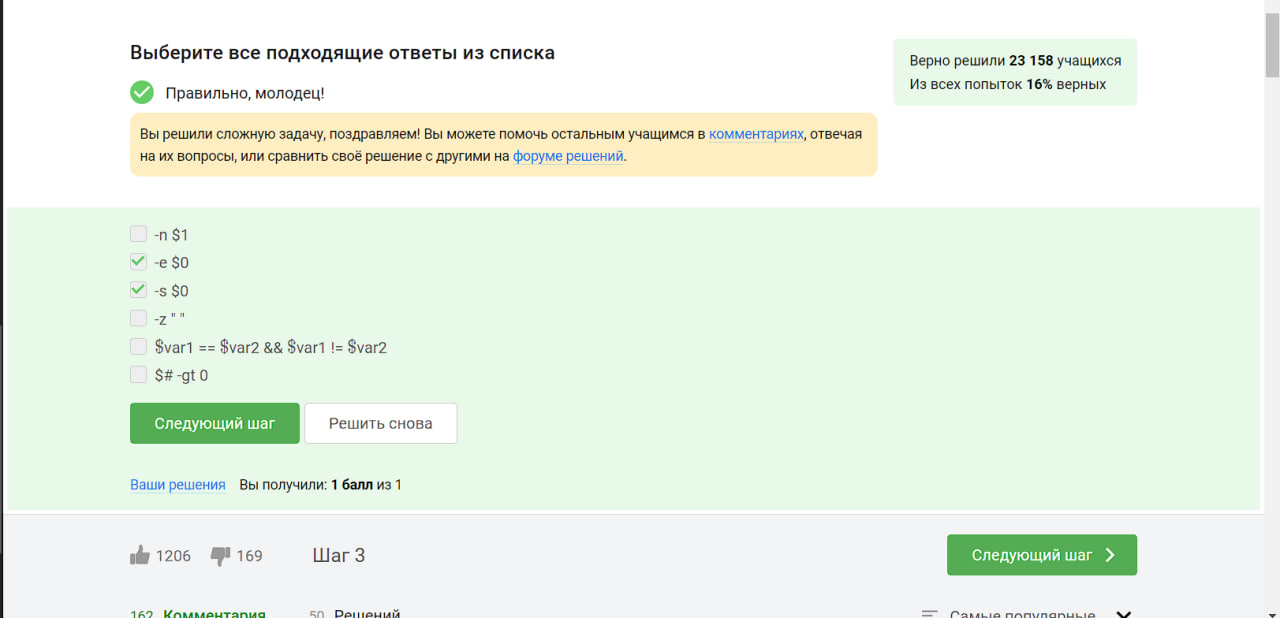


Figure 10: Задание 10

* $0 - имя скрипта
* $# - вернет количество аргументов
* -ge - больше или равно
* -n - не пустая строка.

Имя скрипта - это не пустая строка.

$# Это число аргументов без учета имени скрипта, который всегда $0. И число аргументов всегда будет или равно нулю, или больше него, тк просто не может скатиться в отрицательную сторону.

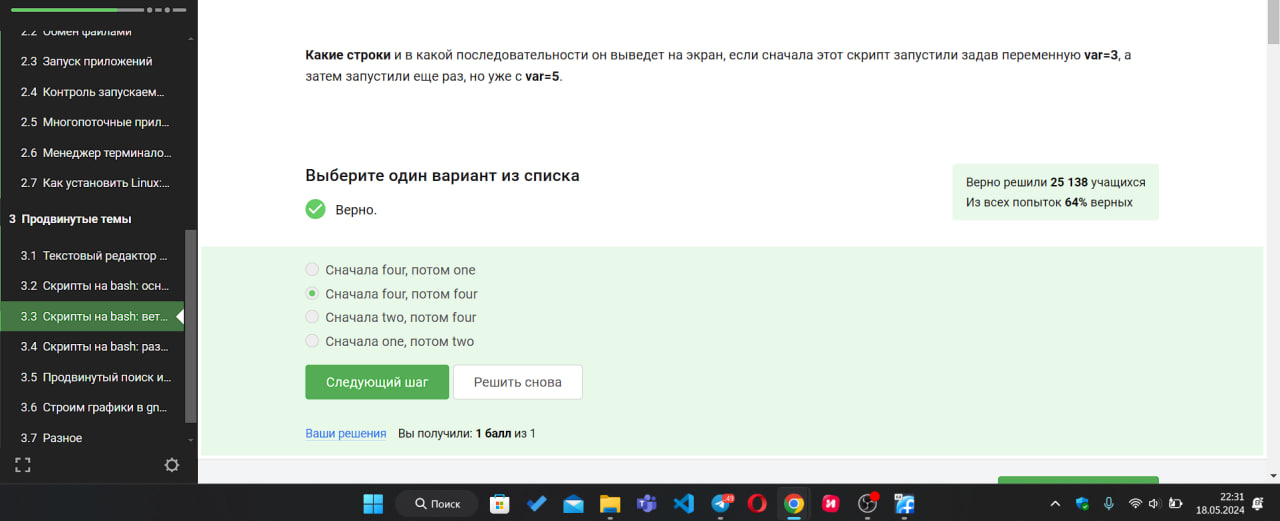


Figure 11: Задание 11

* -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.

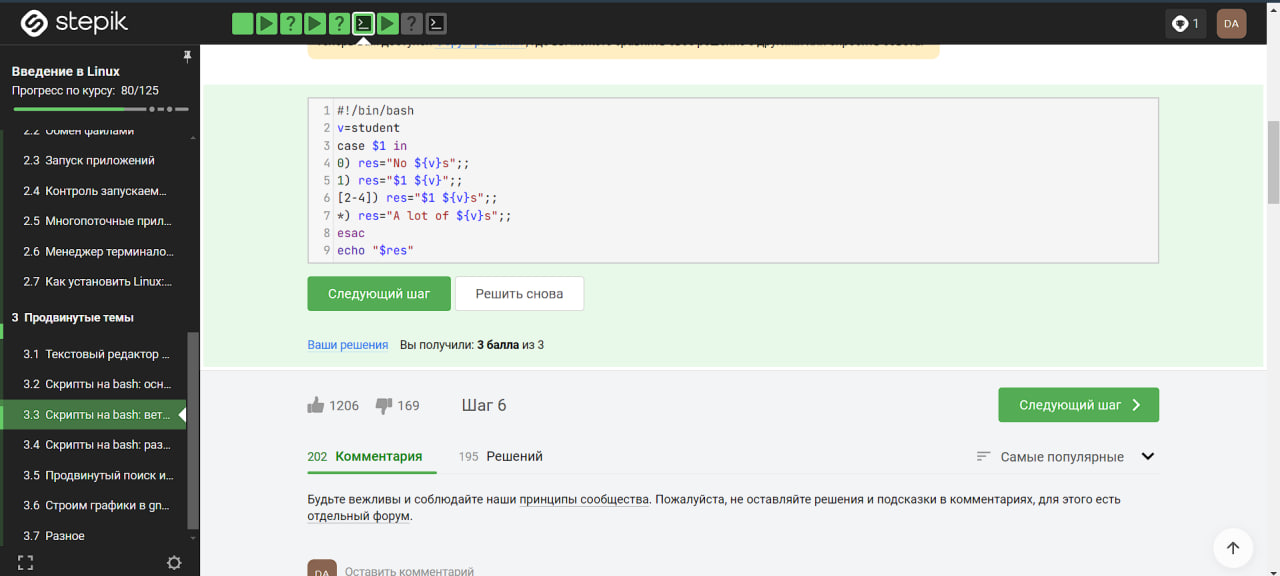


Figure 12: Задание 12

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

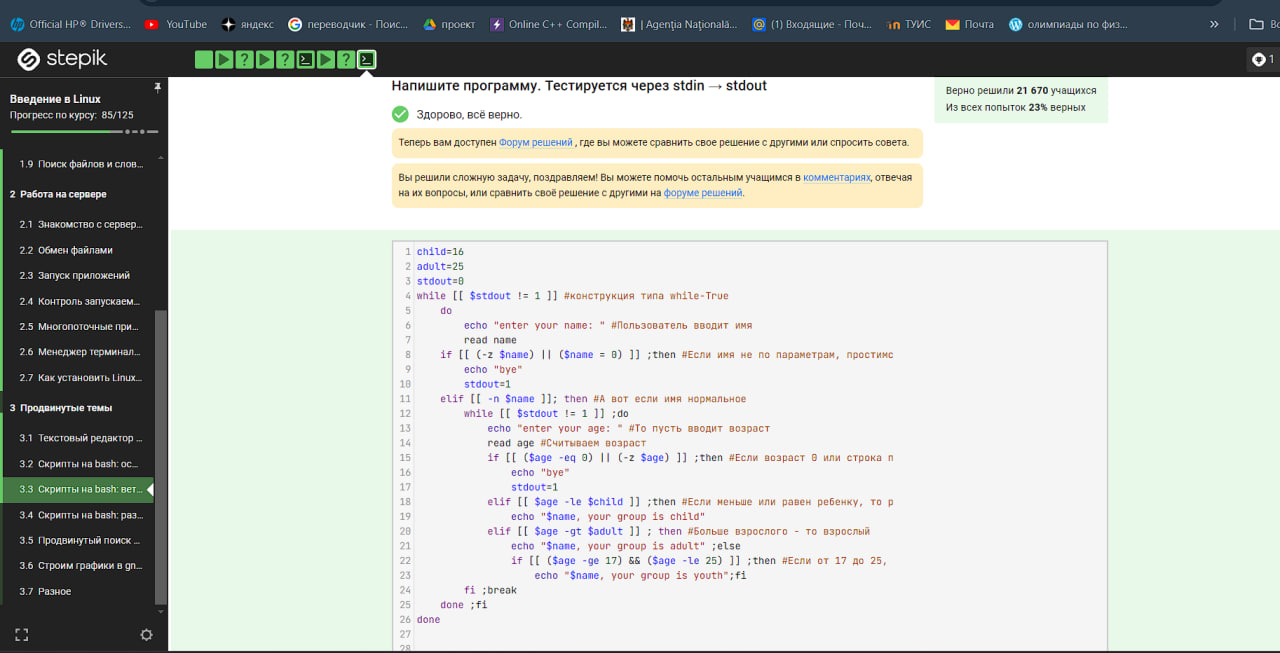


Figure 13: Задание 13

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

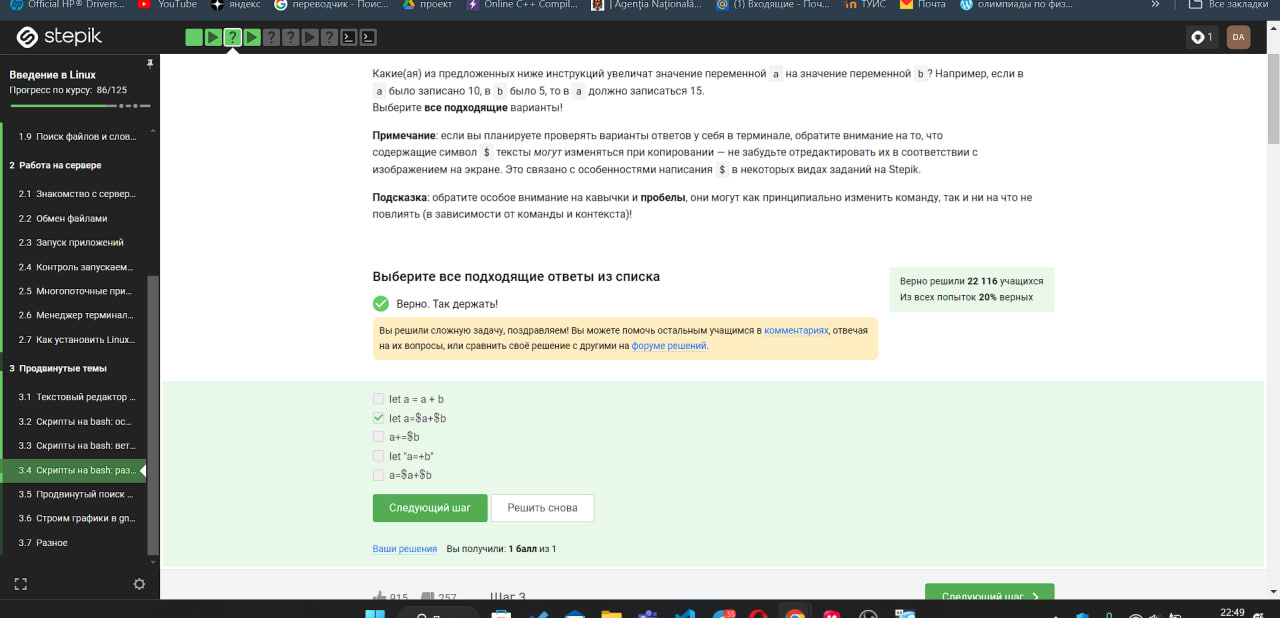


Figure 14: Задание 14

child=16  
adult=25  
stdout=0  
  
while [[ $stdout != 1 ]] #конструкция типа while-True  
 do  
 echo "enter your name: " #Пользователь вводит имя  
 read name  
 if [[ (-z $name) || ($name = 0) ]] ;then #Если имя не по параметрам, простимся  
 echo "bye"  
 stdout=1  
 elif [[ -n $name ]]; then #А вот если имя нормальное  
 while [[ $stdout != 1 ]] ;do   
 echo "enter your age: " #То пусть вводит возраст  
 read age #Считываем возраст  
 if [[ ($age -eq 0) || (-z $age) ]] ;then #Если возраст 0 или строка пуста - прощаемся  
 echo "bye"  
 stdout=1  
 elif [[ $age -le $child ]] ;then #Если меньше или равен ребенку, то ребенок  
 echo "$name, your group is child"  
 elif [[ $age -gt $adult ]] ; then #Больше взрослого - то взрослый  
 echo "$name, your group is adult" ;else  
 if [[ ($age -ge 17) && ($age -le 25) ]] ;then #Если от 17 до 25, то подросток.  
 echo "$name, your group is youth" ;fi  
 fi ;break  
 done ;fi  
done

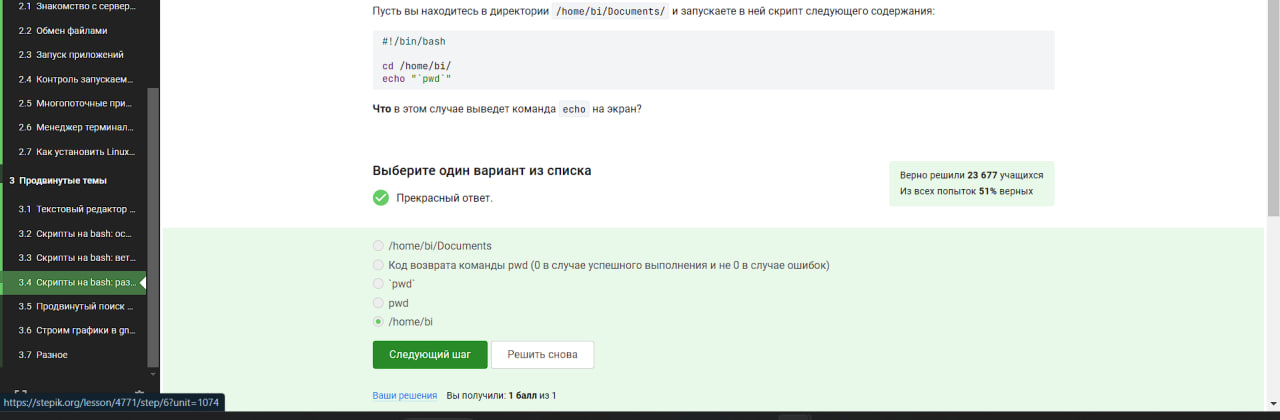


Figure 15: Задание 15

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)

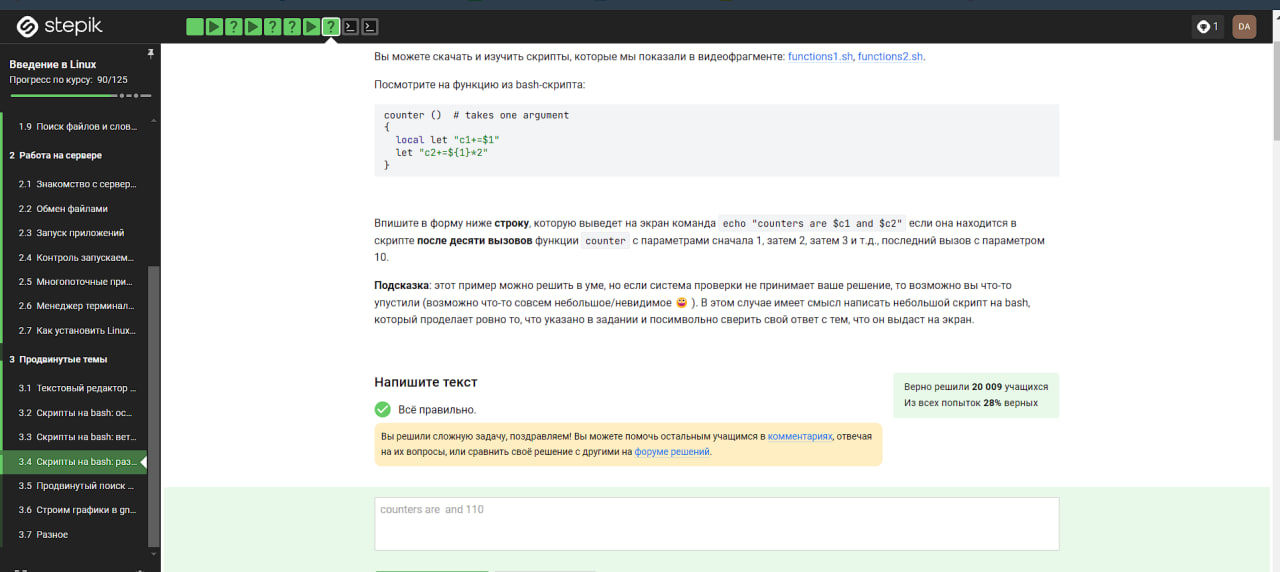


Figure 16: Задание 16

Выведет путь до директории, в которую мы перешли, так как “pwd” - это команда

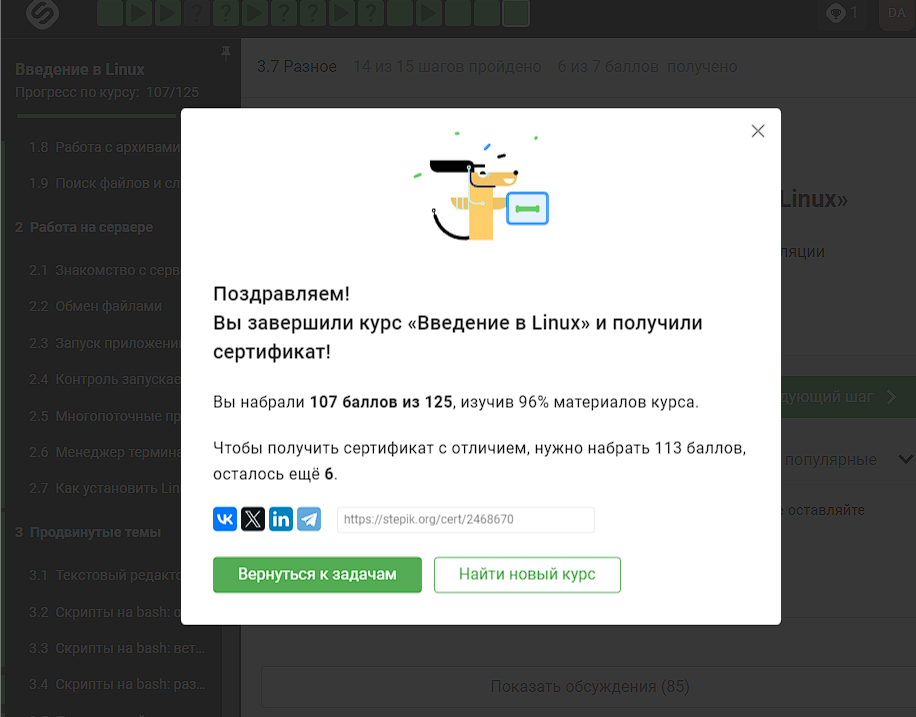


Figure 17: Сертификат

# 5 Выводы

Я просмотрел курс и освежил в памяти навыки работы с более сложными командами в Линукс.

# Список литературы

1. Введение в Linux