Отчет о прохождении 3 этапа внешнего курса

Продвинутые темы

Осина Виктория Александровна, НКАбд-04-23

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Ознакомиться с функционалом операционной системы Linux.

2 Задание

Просмотреть видеоматериал и 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

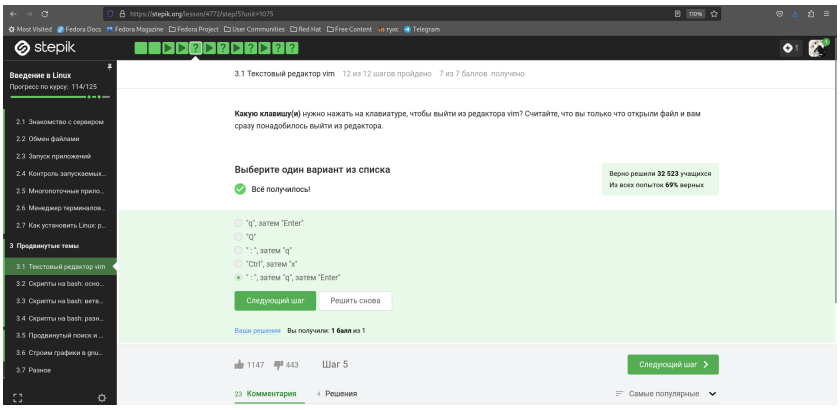


Рис. 4.1: Задание 1

Выйти из редактора можно несколькими способами:

- 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

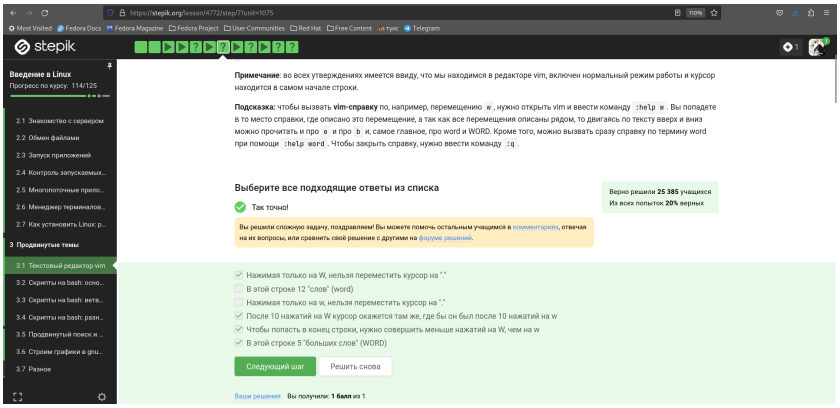


Рис. 4.2: Задание 2

`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`.

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

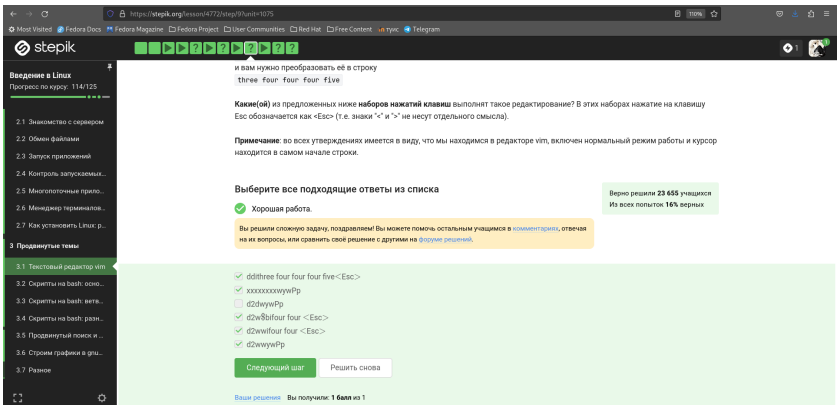


Рис. 4.3: Задание 3

```
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;

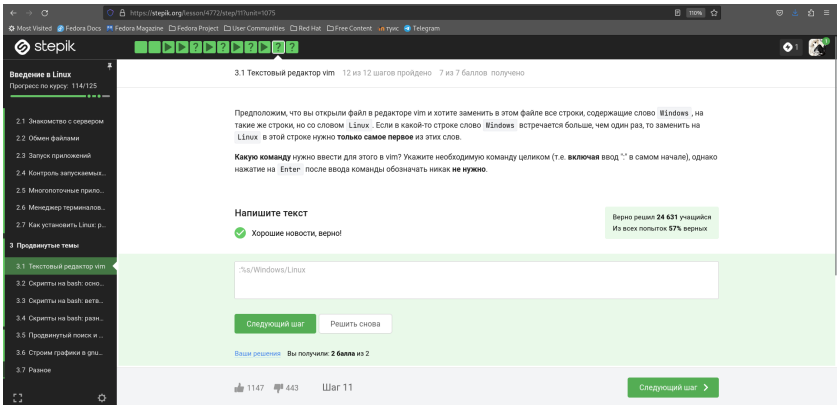


Рис. 4.4: Задание 4

Поиск и замена в редакторе работают по следующей схеме:

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

Для замены во всем файле используется символ %.

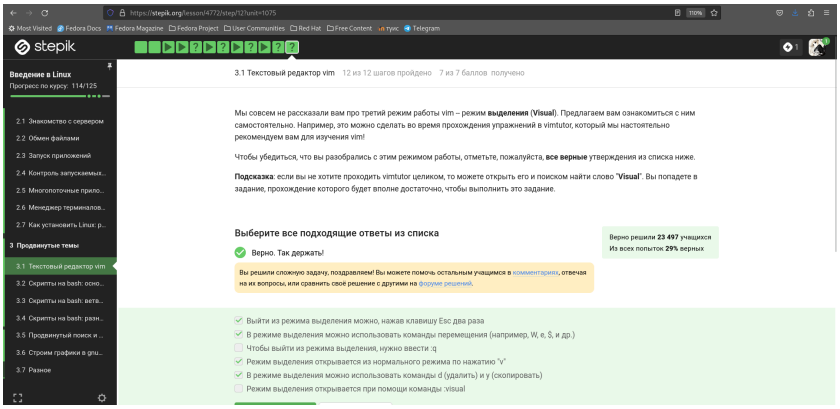


Рис. 4.5: Задание 5

Команда $ — в конец текущей строки, W - до пробела вправо.

Нажать Esc два раза.

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;

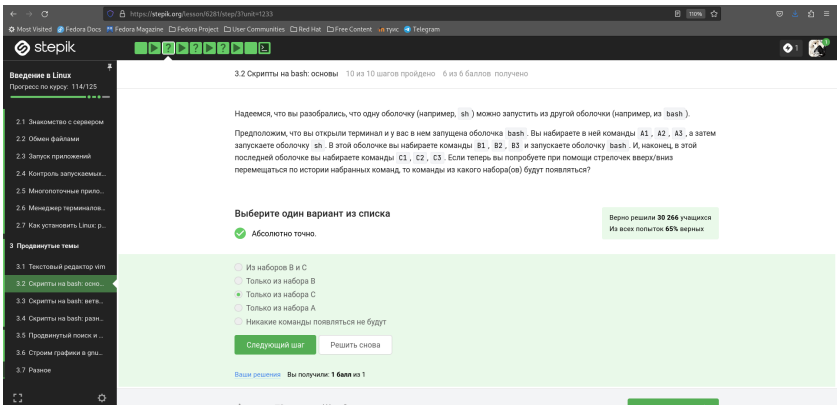


Рис. 4.6: Задание 6

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

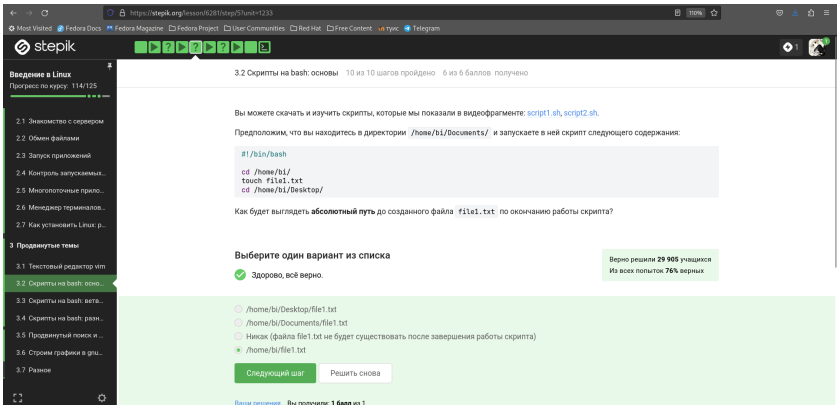


Рис. 4.7: Задание 7

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

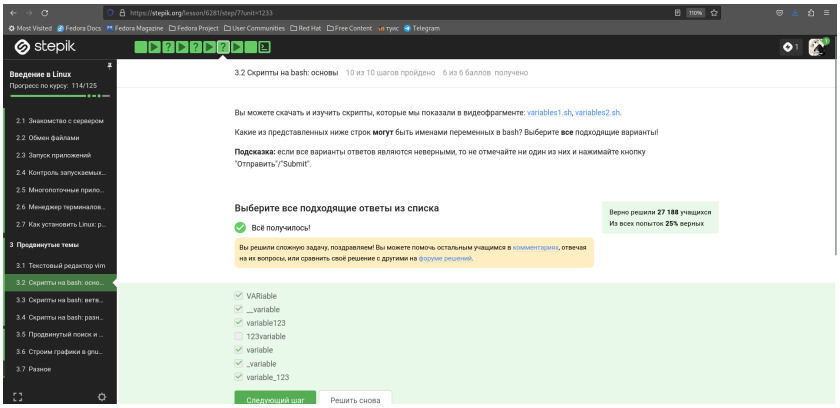


Рис. 4.8: Задание 8

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

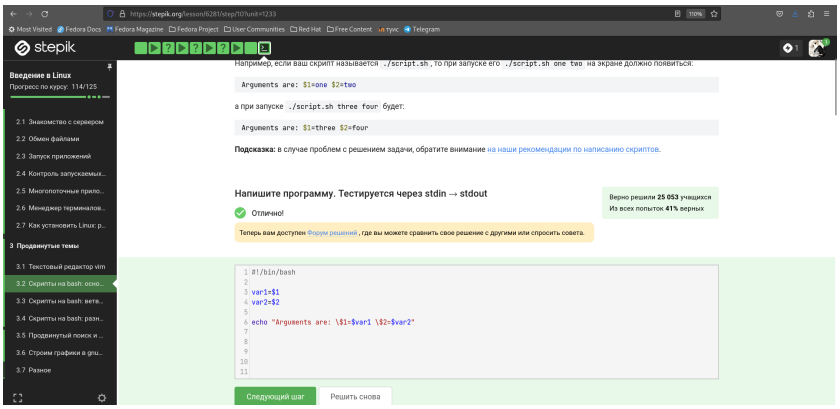


Рис. 4.9: Задание 9

`$ 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.

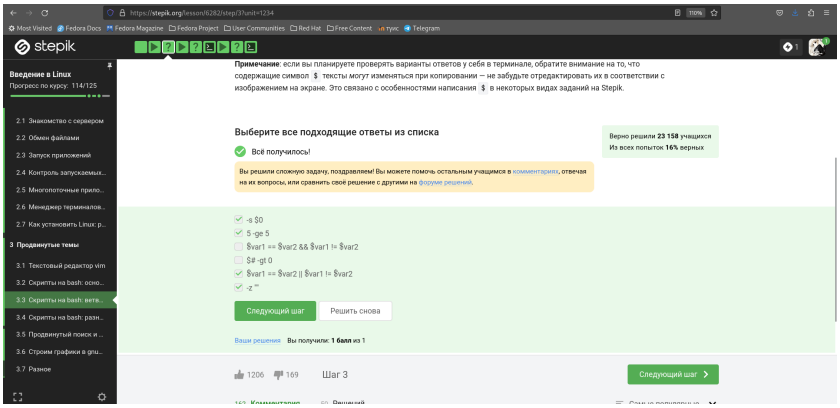


Рис. 4.10: Задание 10

- $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.

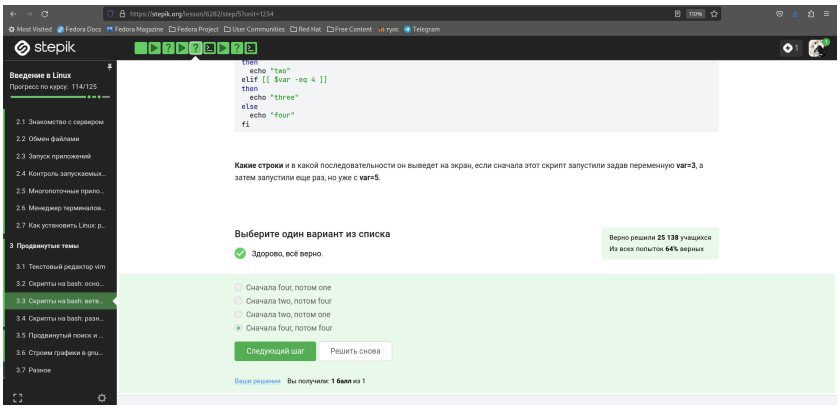


Рис. 4.11: Задание 11

- -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.

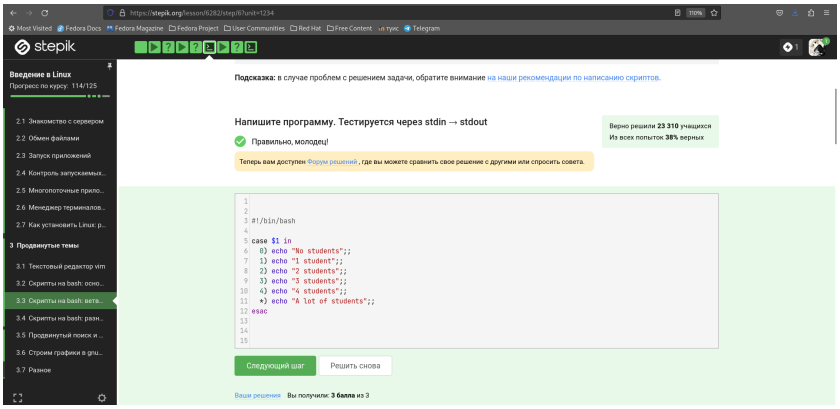


Рис. 4.12: Задание 12

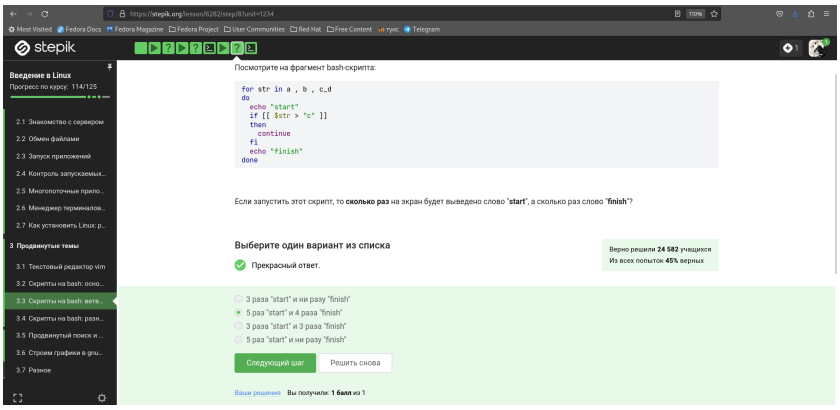


Рис. 4.13: Задание 13

- (Start)
- a > c нет (Finish)
- (Start)
- , > c нет (Finish)

- (Start)
- b > c нет (Finish)
- (Start)
- , > c нет (Finish)
- (Start)
- c_d > c да

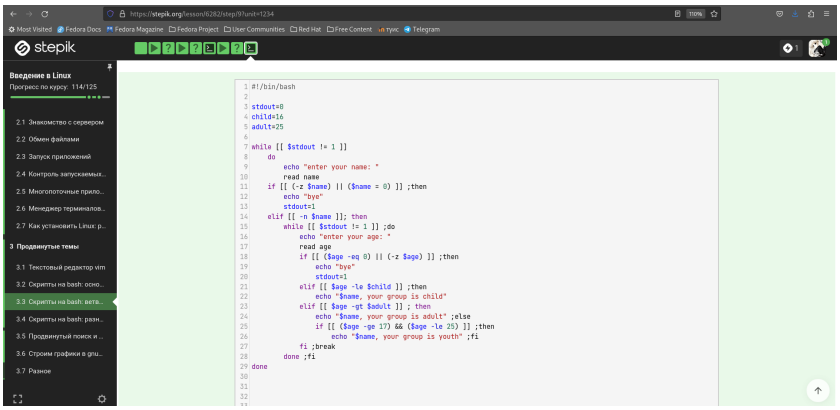


Рис. 4.14: Задание 14

```
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, простимс
        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 п
                echo "bye"
                stdout=1
            elif [[ $age -le $child ]] ;then #Если меньше или равен ребенку, то р
                echo "$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,
                    echo "$name, your group is youth" ;fi
            fi ;break
        done ;fi
done
```

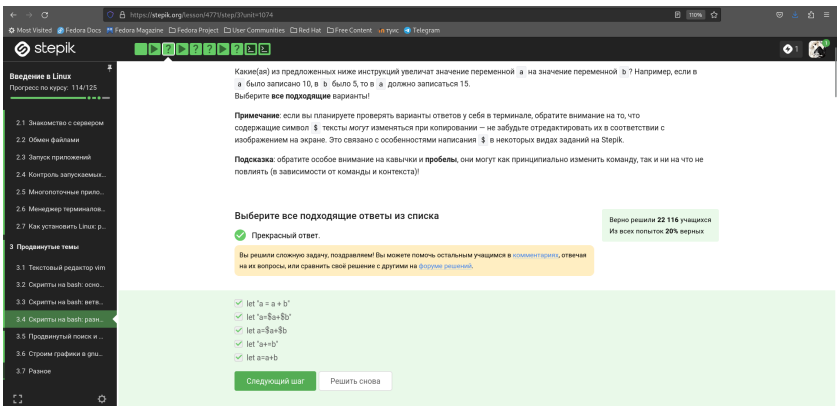


Рис. 4.15: Задание 15

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)

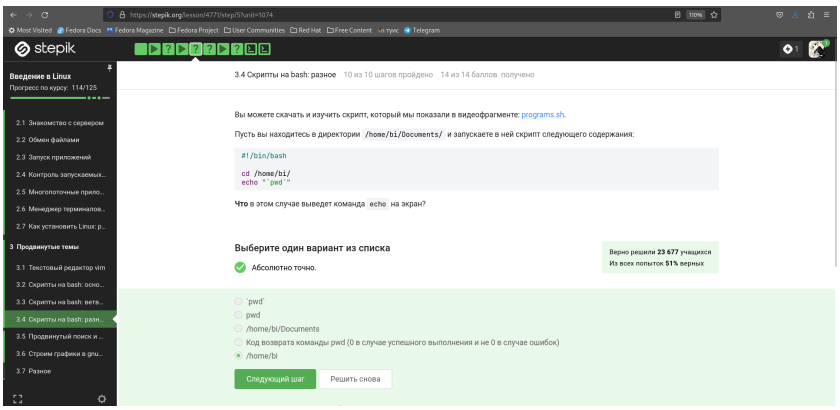


Рис. 4.16: Задание 16

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

Задание 16_2

programm выполняет стандартный вывод в терминал (если это принцип работы программы). И нам нужно настроить вывод в файл.

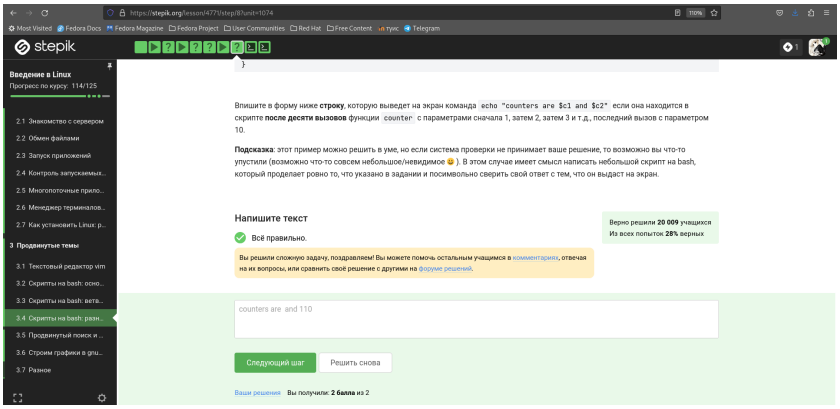


Рис. 4.17: Задание 17

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.

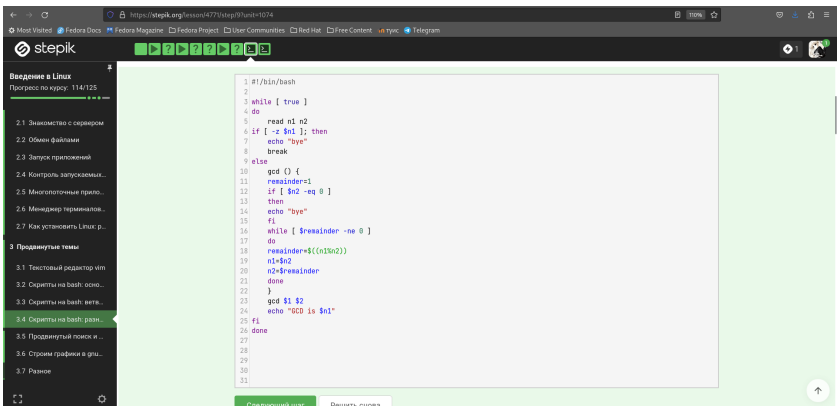


Рис. 4.18: Задание 18

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.

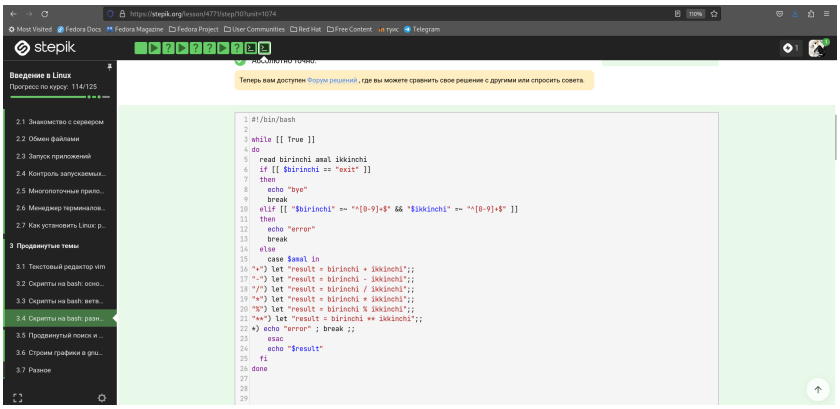


Рис. 4.19: Задание 19

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

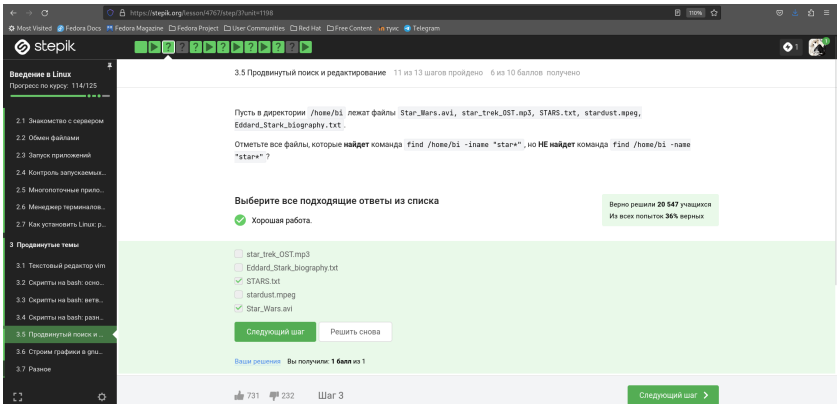


Рис. 4.20: Задание 20

-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.

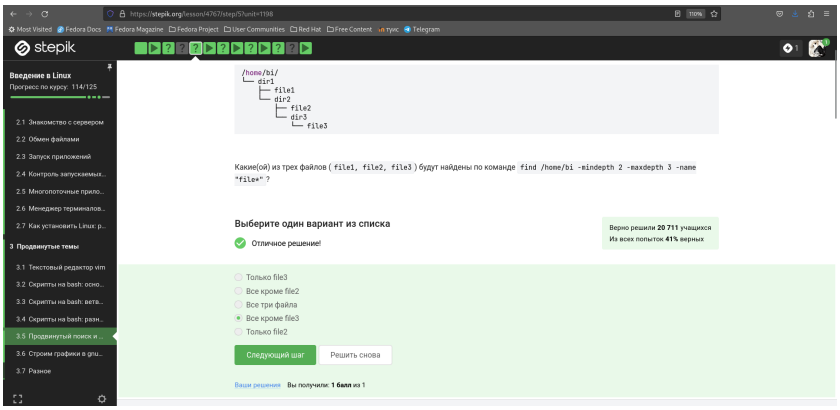


Рис. 4.21: Задание 22

Текущий каталог - это depth=1, а остальное считается просто:

/home/bi -> depth=1

/home/bi/dir1 -> depth=2

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

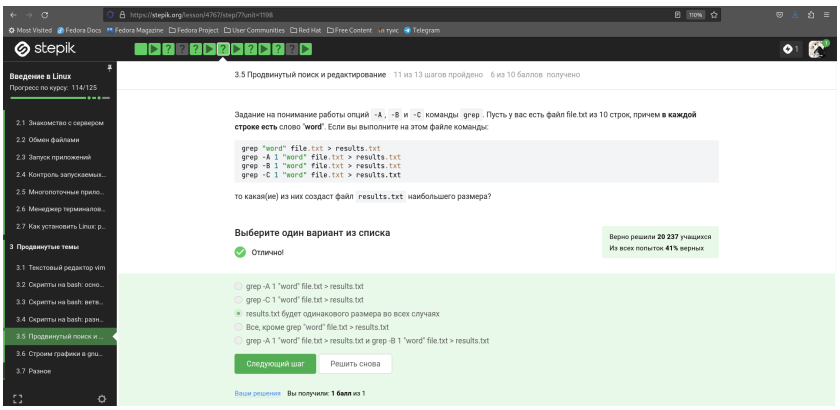


Рис. 4.22: Задание 23

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

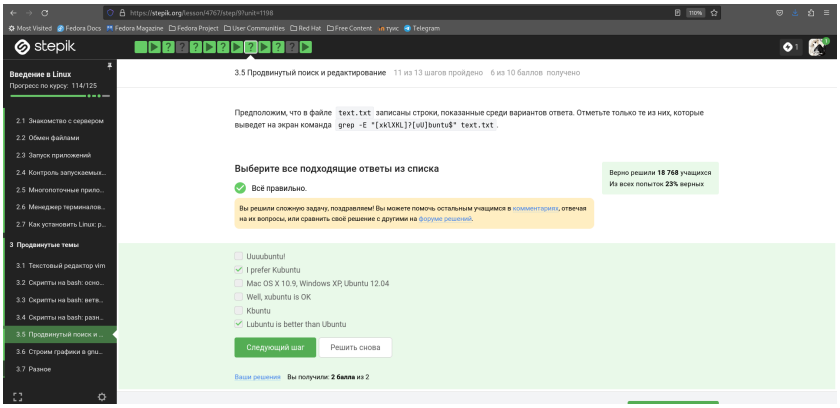


Рис. 4.23: Задание 24

Объяснение на втором скриншоте.

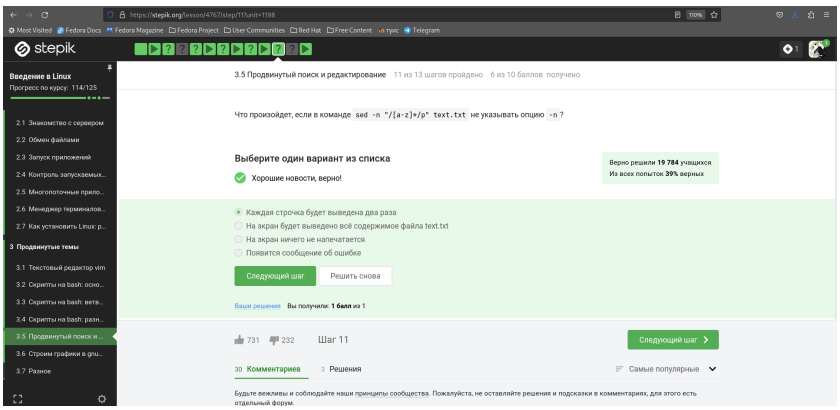


Рис. 4.24: Задание 25

The -n option disables the automatic printing, which means the lines you don’t specifically tell it to print do not get printed, and lines you do explicitly tell it to print (e.g. with p) get printed only once.

Задание 26

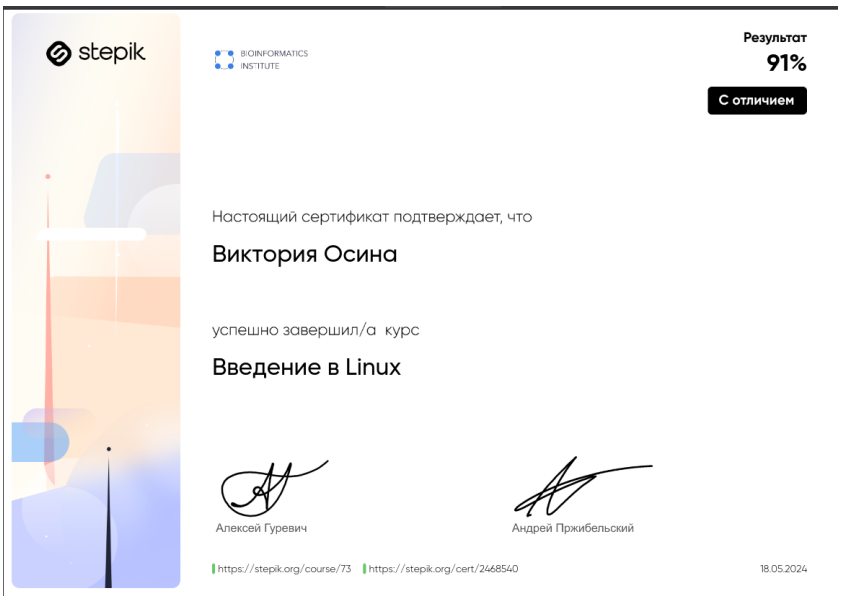
аббревиатура ABBA отличается от двух других аббревиатур тем, что справа он неё стоит запятая без пробела: “ABBA,”.

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.

Задание 27

-persist lets plot windows survive after main gnuplot program exits.

5 Сертификат



(https://stepik.org/cert/2468540)

6 Выводы

Я просмотрела курс и освежила в памяти навыки работы с более сложными командами в Линукс.

Список литературы

1. Введение в Linux