

Отчет о прохождении 3 этапа внешних курсов

Продвинутые темы

Глущенко Евгений Игоревич, НКАбд-02-23

Содержание

1	Цель работы	1
2	Задание.....	1
3	Теоретическое введение	1
4	Выполнение лабораторной работы.....	2
5	Сертификат.....	19
6	Выводы.....	19
	Список литературы.....	19

1 Цель работы

Ознакомиться с функционалом операционной системы Linux.

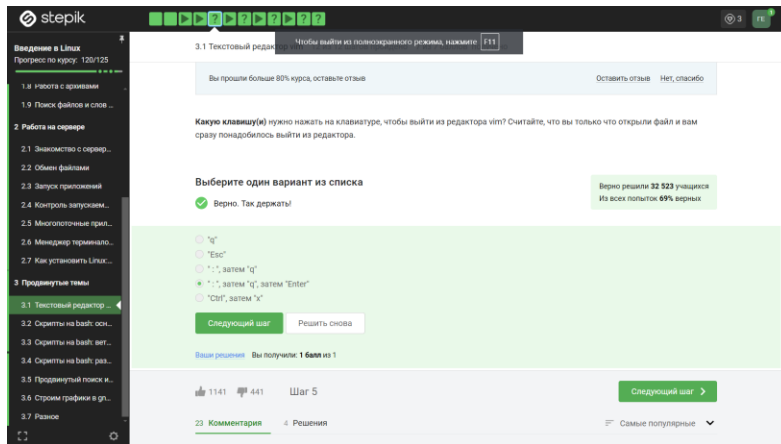
2 Задание

Просмотреть видео и на основе полученной информации пройти тестовые задания.

3 Теоретическое введение

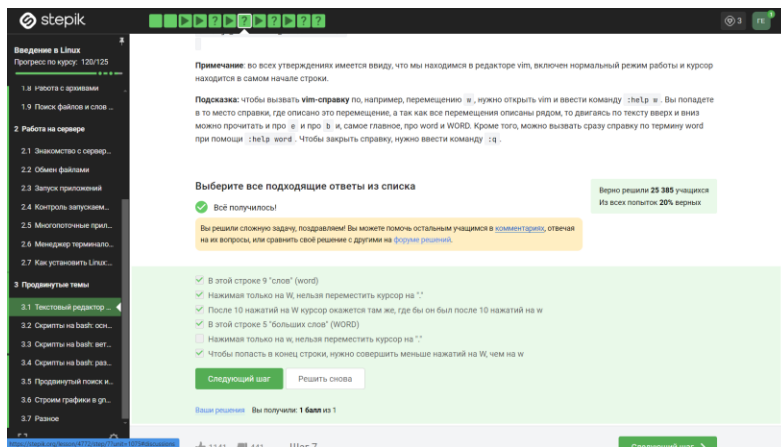
Линукс - в части случаев GNU/Linux — семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью 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

[illegible]

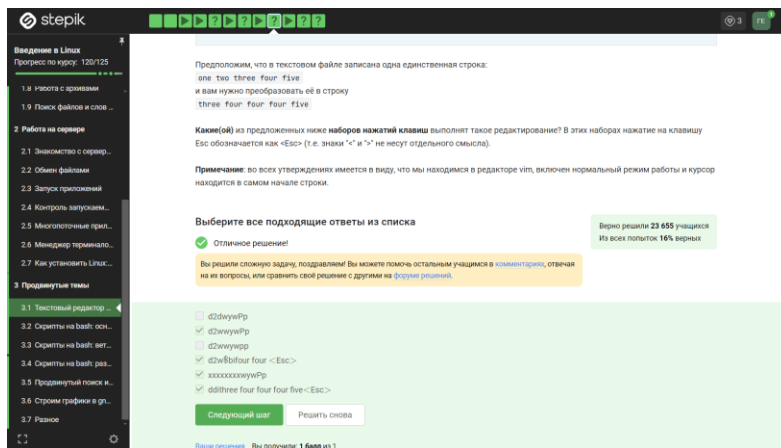
Задание 1

У редактора vim есть tutorial, который позволяет разобраться с командами, необходимыми для стандартной работы. - 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



Задание 2

Нельзя нажимая только на `w` переместить курсор на “.”



Задание 3

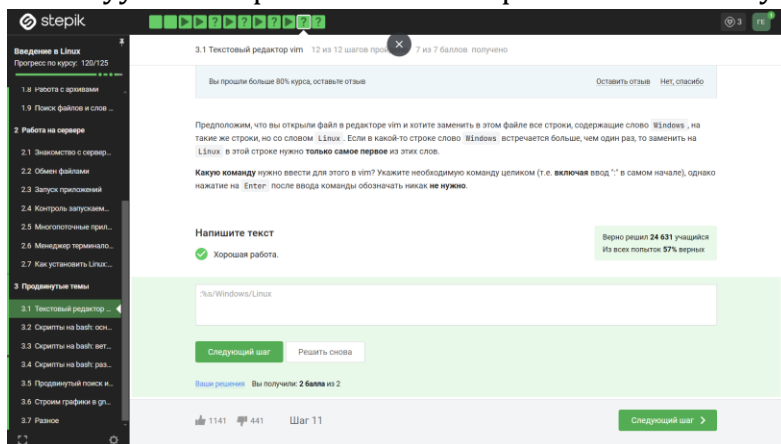
d2wwywPp

d2w&bifour four <>

xxxxxxxxxwywPp

ddithree four four four five <>

- \$ — в конец текущей строки;
- w — на слово вправо;
- b — на слово влево;
- i — начать ввод перед 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;

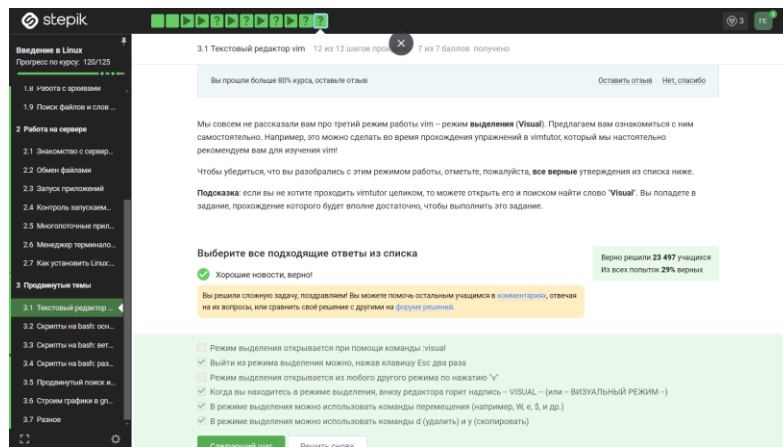


Задание 4

Поиск и замена в редакторе работают так:

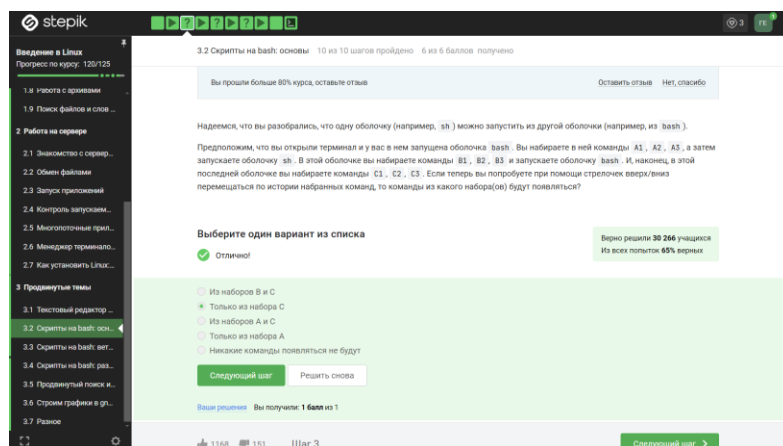
:{пределы}s/{что заменяем}/{на что заменяем}/{опции}

Для замены во всем файле мы можем использовать %.



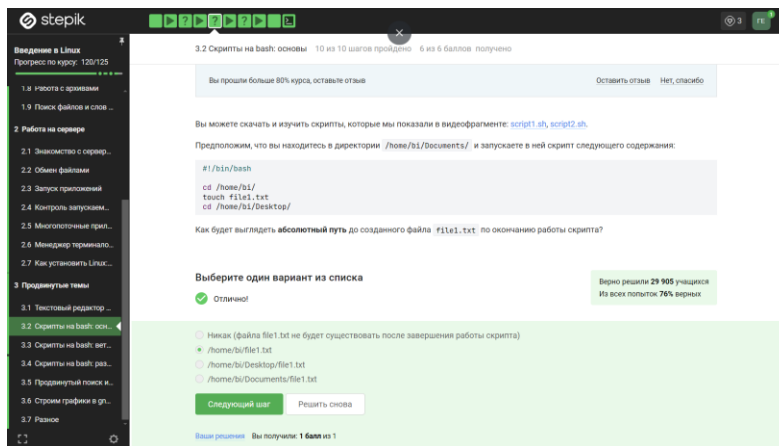
Задание 5

Режим выдеелния не открывается с помощью :visual Режим выделения не открывается из любого другого режима по нажатию “v”



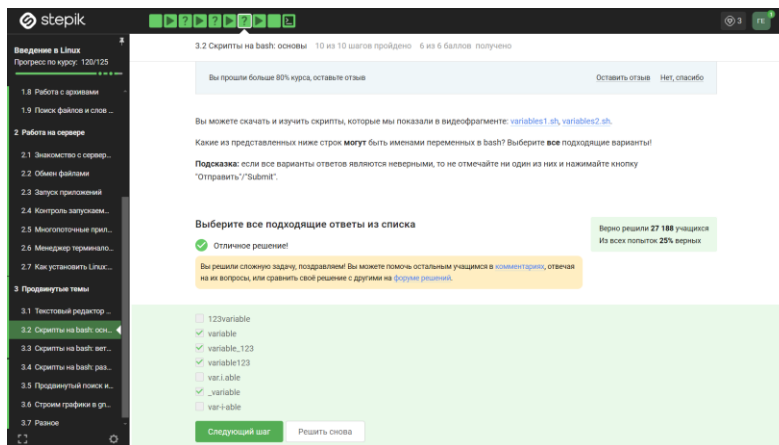
Задание 6

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



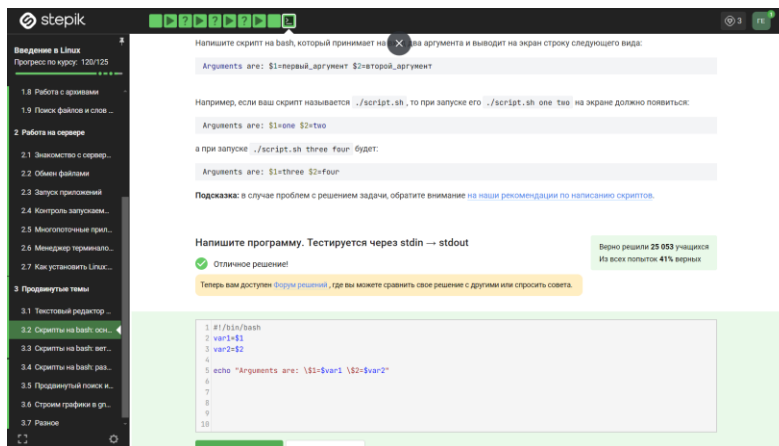
Задание 7

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



Задание 8

Имя не должно начинаться с цифры, не может иметь в своем составе специальные символы кроме “_”

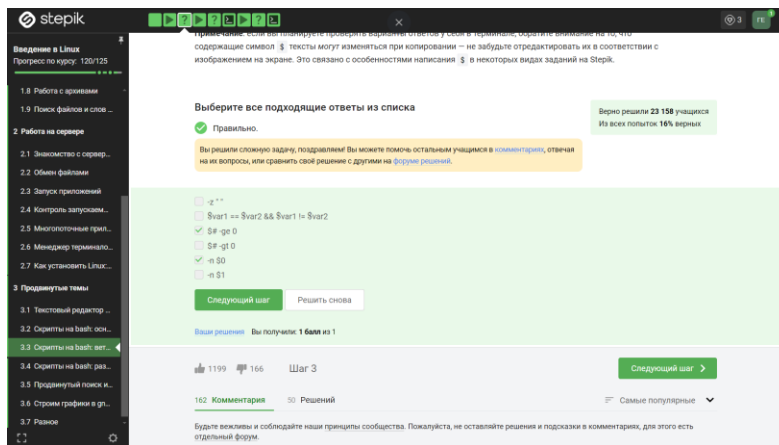


Задание 9

\$ echo опции строка Эта команда выводит переданные в качестве аргументов строки в стандартный вывод. Обычно она используется в оболочечных сценариях для отображения сообщений или результатов выполнения других команд.

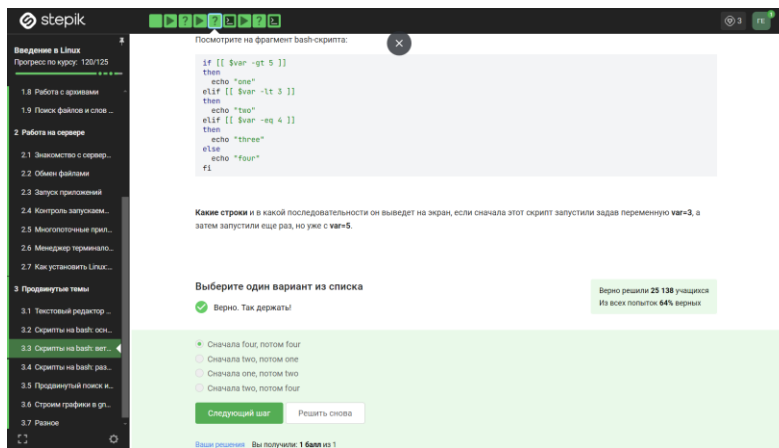
var1=\$1, var2=\$2 - обозначение переменных

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



Задание 10

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

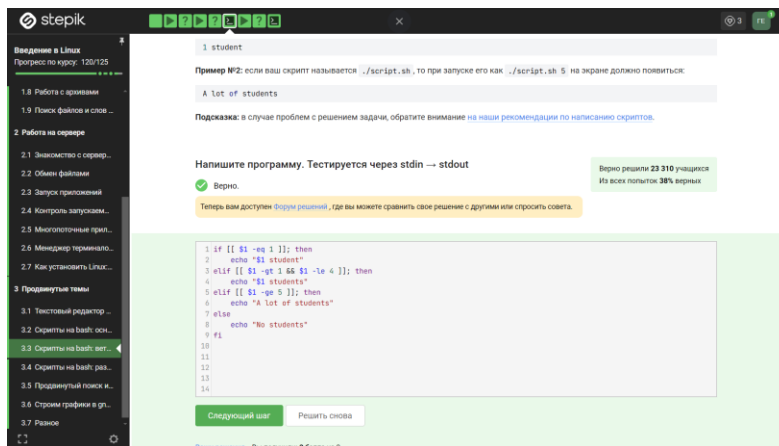


Задание 11

-lt - меньше -gt - больше -eq - равно

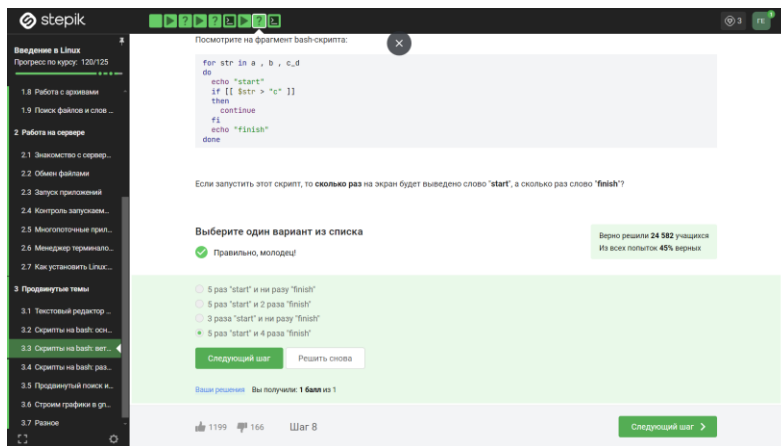
3 > 5 - не верно, 3 < 3 - не верно, 3 = 4 - не верно.

5 > 5 - не верно, 5 < 3 - не верно, 5 = 4 - не верно.



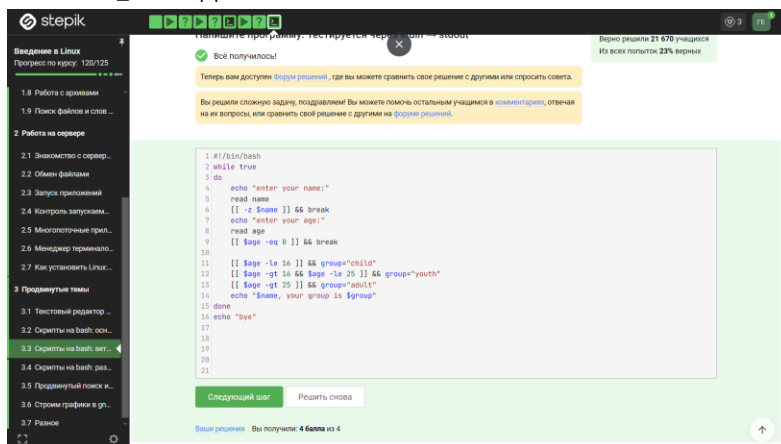
Задание 12

1. Если “число” равно одному, то вывести ” “число” студент “.
2. Если “число” больше одного, но меньше или рано 5, то вывести ” “число” студенты “.
3. Если “число” больше пяти, то вывести “Много студентов”.
4. Если ничего не подходит, то вывести “Нет судентов”.



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



Задание 14

#!/bin/bash while true do echo "enter your name:" read name [[-z \$name]] && break echo "enter your age:" read age [[\$age -eq 0]] && break

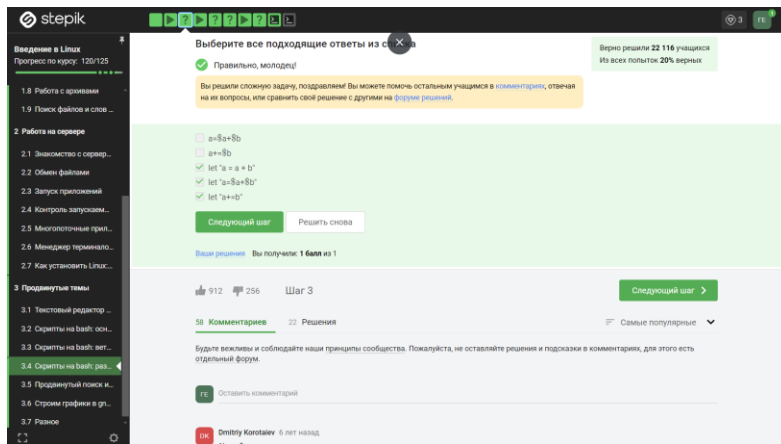
```
[[ $age -le 16 ]] && group="child"
[[ $age -gt 16 && $age -le 25 ]] && group="youth"
[[ $age -gt 25 ]] && group="adult"
echo "$name, your group is $group"
```



```
done echo "bye"
```

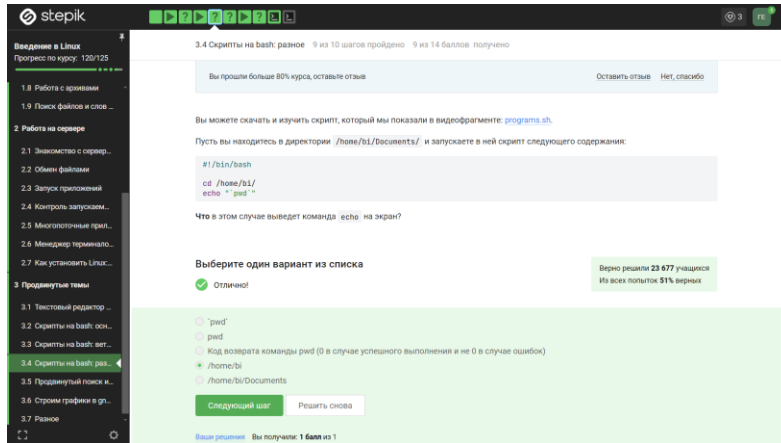
Это оболочный скрипт на языке Bash, который запрашивает у пользователя его имя и возраст, а затем классифицирует его в одну из трех групп: "ребенок", "молодежь" или "взрослый" в зависимости от возраста. Вот пошаговое объяснение скрипта:

1. `#!/bin/bash`: Это так называемый шибанг, который указывает, что скрипт должен выполняться с помощью интерпретатора Bash.
2. Цикл `while`: Цикл `while` будет выполняться до тех пор, пока условие в нем верно. В данном случае условие всегда истинно (`true`), поэтому цикл будет выполняться бесконечно, пока пользователь не выполнит команду `break`.
3. `echo "enter your name:"`: Выводит на экран запрос ввести имя.
4. `read name`: Считывает введенное пользователем имя и сохраняет его в переменной `name`.
5. `[[-z $name]] && break`: Проверяет, пуста ли переменная `name`. Если она пуста (т.е. пользователь нажал Enter без ввода имени), цикл `while` прерывается с помощью команды `break`.
6. `echo "enter your age:"`: Выводит на экран запрос ввести возраст.
7. `read age`: Считывает введенный пользователем возраст и сохраняет его в переменной `age`.
8. `[[$age -eq 0]] && break`: Проверяет, равен ли возраст 0. Если возраст равен 0 (т.е. пользователь ввел 0), цикл `while` прерывается с помощью команды `break`.
9. Далее скрипт определяет группу, к которой относится пользователь, на основе его возраста:
 - `[[$age -le 16]] && group="child"`: Если возраст меньше или равен 16, группа устанавливается в `"child"`.
 - `[[$age -gt 16 && $age -le 25]] && group="youth"`: Если возраст больше 16 и меньше или равен 25, группа устанавливается в `"youth"`.
 - `[[$age -gt 25]] && group="adult"`: Если возраст больше 25, группа устанавливается в `"adult"`.
10. `echo "$name, your group is $group"`: Выводит на экран сообщение, содержащее имя пользователя и его группу.
11. После выхода из цикла `while` скрипт выводит на экран `"bye"`.



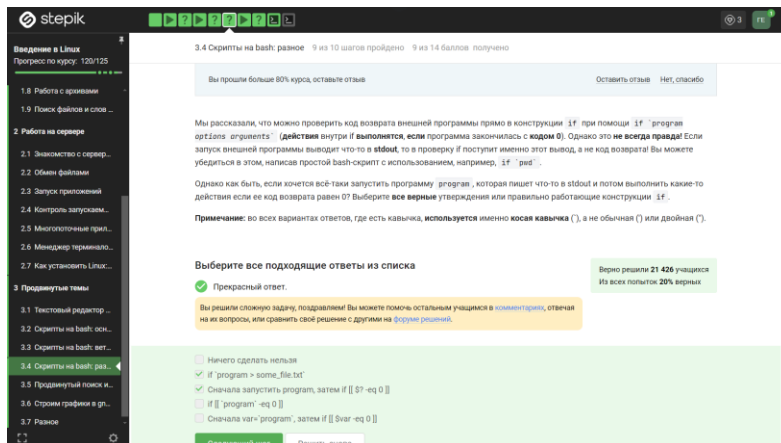
Задание 15

1. $a = \$a$
2. $a+=b$ тоже самое что и $a=a+b$



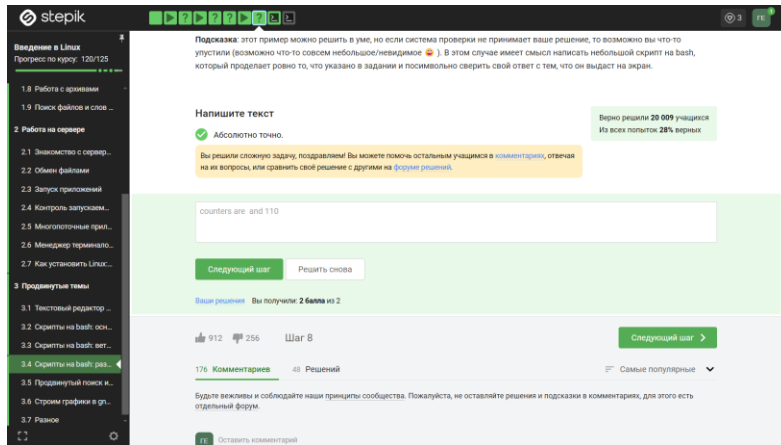
Задание 16

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



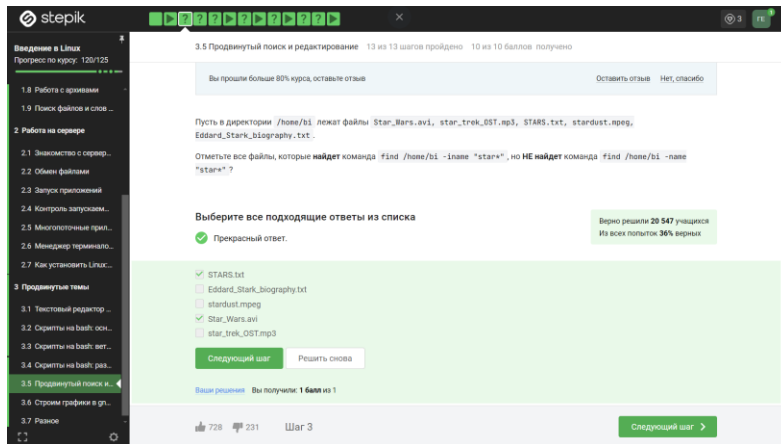
Задание 16_2

программ выполняет стандартный вывод в терминал. И нам нужно настроить вывод в файл.



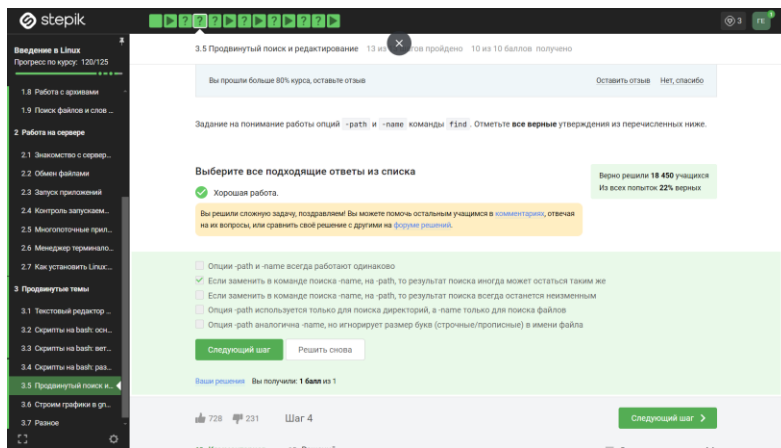
Задание 17

Локальная переменная является пустой строкой, а вторая переменная представляет собой сумму арифметической прогрессии от 1 до 10, которая равна 55. При умножении этой суммы на 2 получаем результат 110.



Задание 20

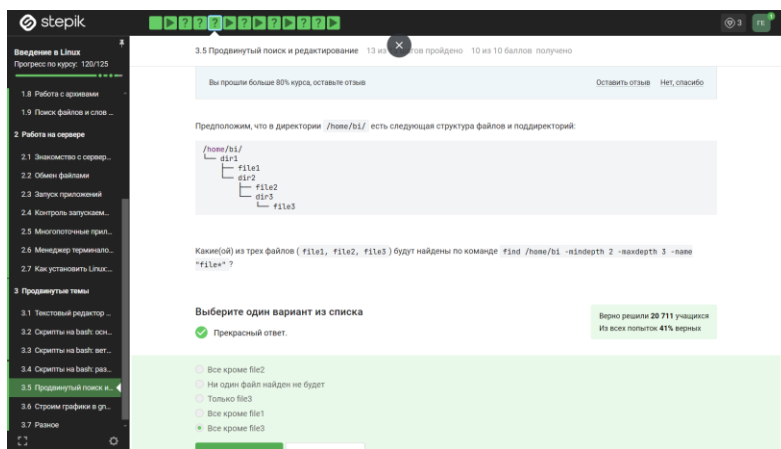
-iname ищет без учета регистра, а -name в точности как в запросе. Звездочка стоит после слова - это значит после слова может быть сколько угодно символов.



Задание 21

`find [path] [expression]`

где: `path` - это путь к директории, в которой нужно выполнить поиск файлов (по умолчанию, поиск производится в текущей директории и всех ее поддиректориях).



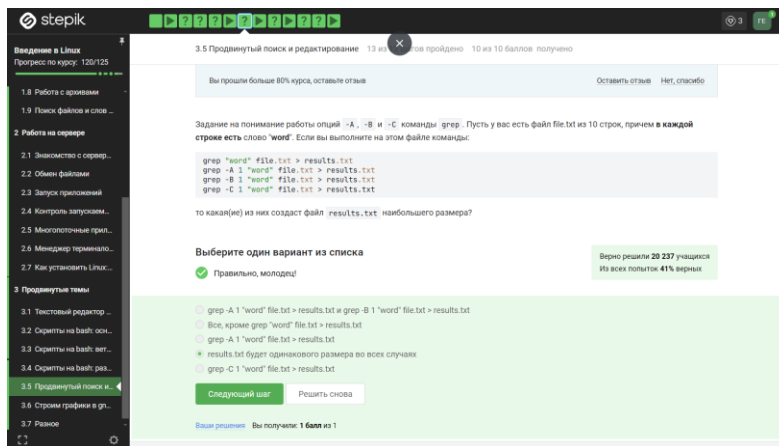
Задание 22

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

`/home/bi -> depth=1`

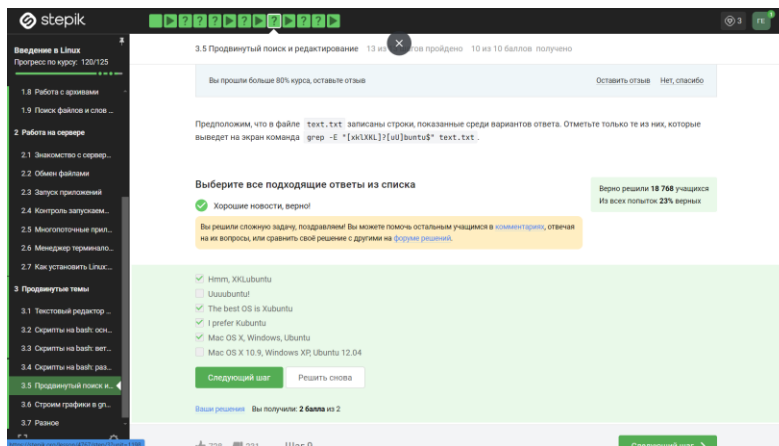
`/home/bi/dir1 -> depth=2`

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



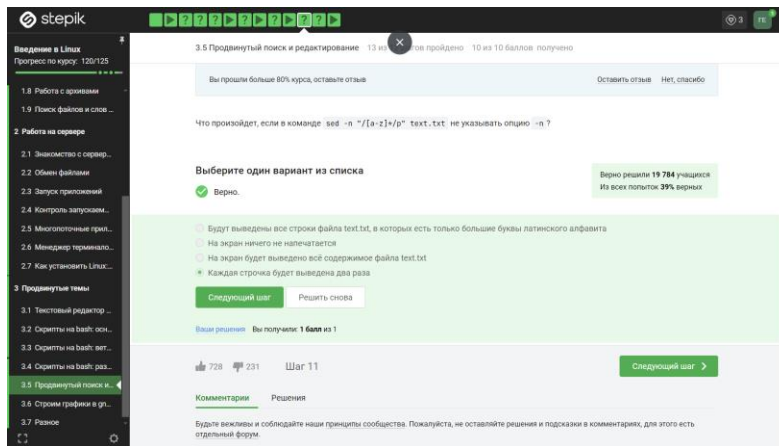
Задание 23

Print NUM lines of trailing context after/before matching lines “matching lines”. Если в последовательных строках текста (например, со 2-й по 100-ю) обнаружено совпадение, то вывод контекста будет производиться до и после всей группы этих строк, а не до и после каждой отдельной строки в данной группе.



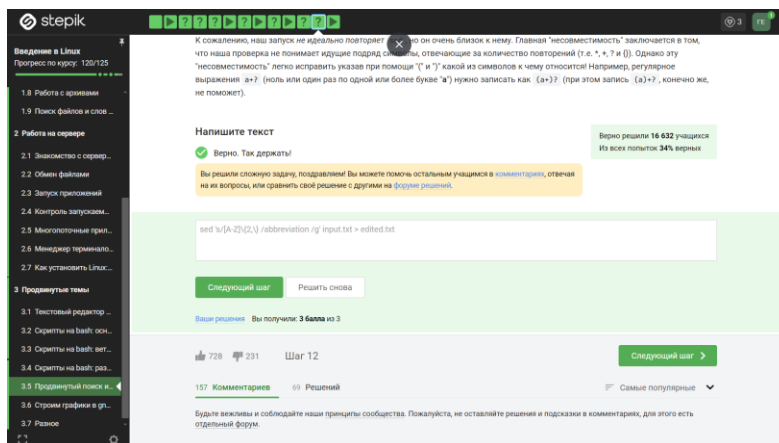
Задание 24

Написал программу и пргнал ее, видно на скринкасте.



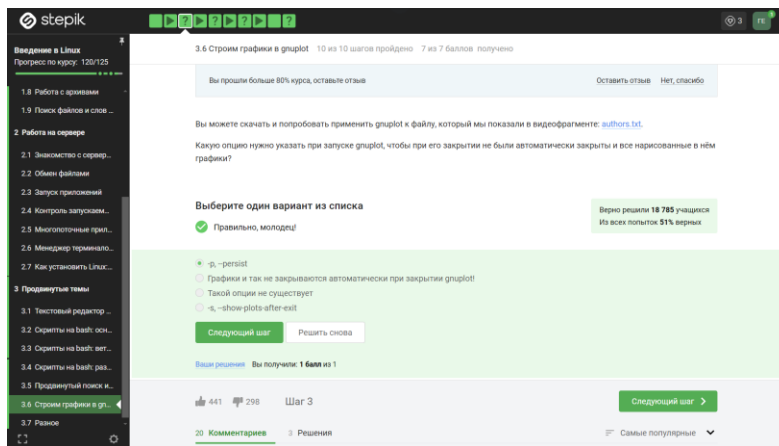
Задание 25

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.



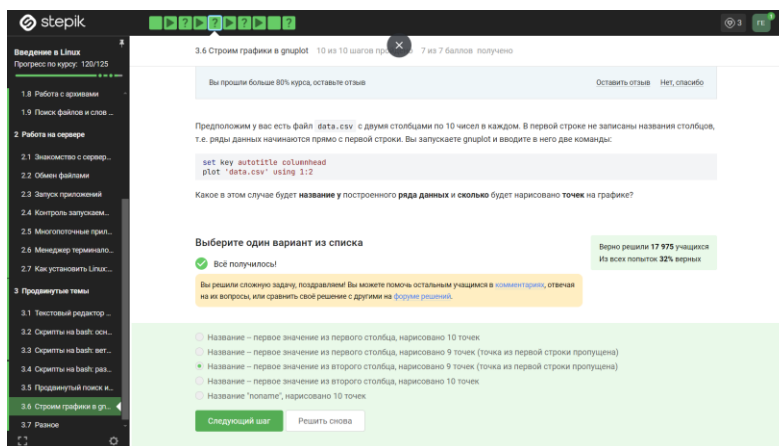
Задание 26

Из-за наличия запятой без пробела после аббревиатуры ABBA ("ABBA,") она отличается от других аббревиатур, которые соответствуют следующим шаблонам: [XX] или [XXX] (и более). Чтобы привести ABBA к указанному формату, необходимо добавить пробел внутри квадратных скобок с обеих сторон: [ABBA].



Задание 27

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

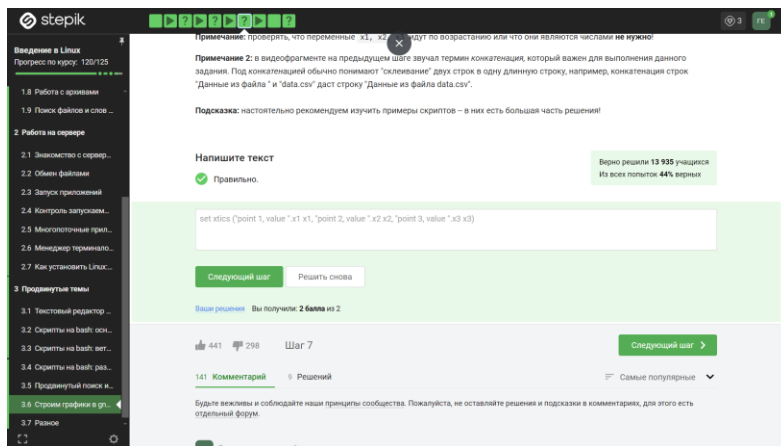


Задание 28

plot 'data.csv' using 1:2 даст ошибку:

warning: Skipping data file with no valid points ^ x range is invalid

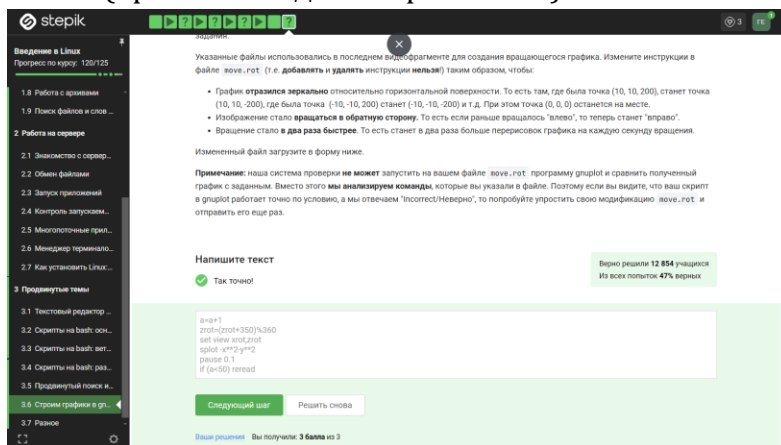
Вероятной причиной необычного поведения парсера является спецификация формата CSV, которая допускает наличие строк, где значения в столбцах разделяются запятыми.



Задание 29

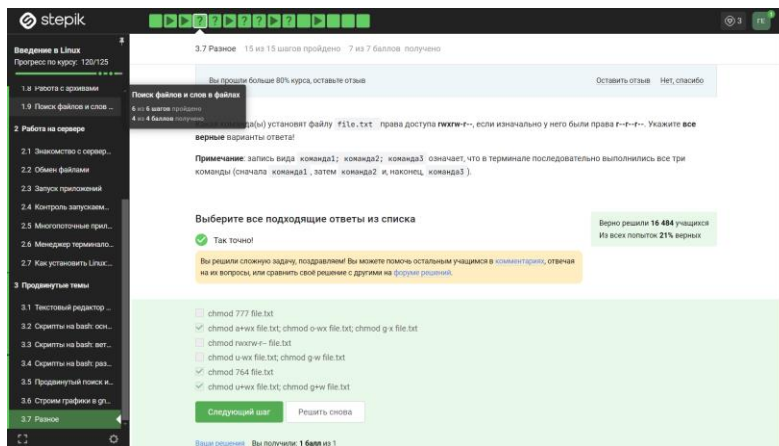
Процесс установки подписей в заданной последовательности происходит следующим образом:

1. Формируется команда установки подписей.
2. В скобках указывается подпись, состоящая из текста задания и переменной с координатой, разделенных пробелом и запятой.
3. Этот процесс повторяется для каждой переменной, разделяя их запятыми (кроме последней переменной).



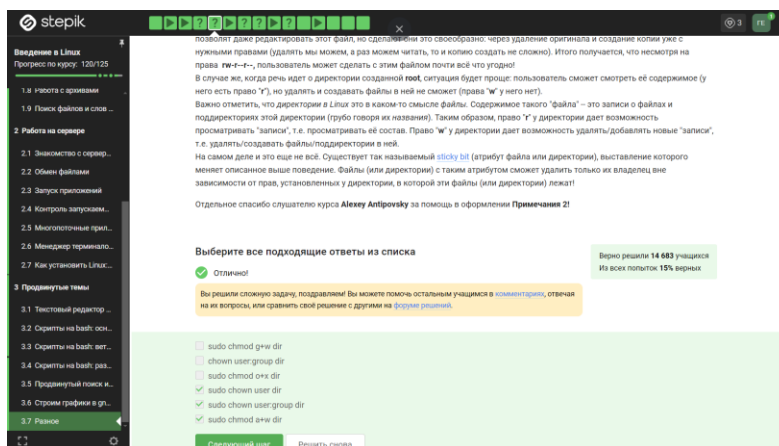
Задание 30

1. График строится командой "splot x2+y2".
2. Чтобы вращать объект в противоположную сторону, необходимо изменить команду "zrot=(zrot+10)%360" на "zrot=(zrot+350)%360".
3. Для более частой перерисовки нужно уменьшить паузу в команде "pause", например: "pause 0.1".

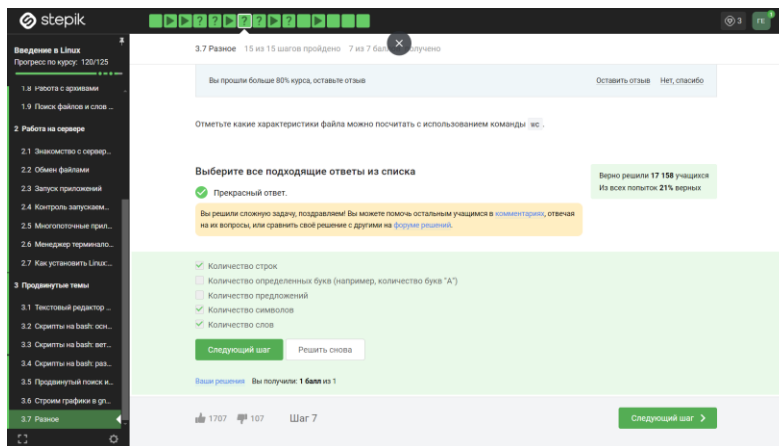


Задание 31

r - чтение; w - запись; x - выполнение; s - выполнение от имени суперпользователя (дополнительный); u - владелец файла; g - группа файла; o - все остальные пользователи; 0 - никаких прав; 1 - только выполнение; 2 - только запись; 3 - выполнение и запись; 4 - только чтение; 5 - чтение и выполнение; 6 - чтение и запись; 7 - чтение запись и выполнение.

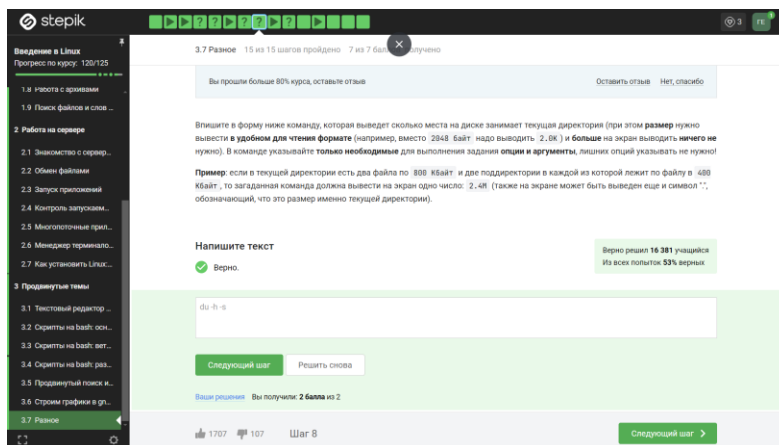


Задание 32



Задание 33

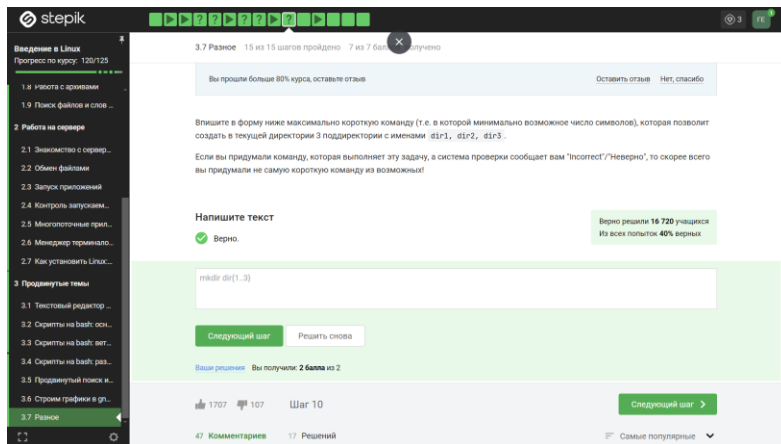
`wc -l` вывести количество строк `wc -c` вывести количество байт `wc -m` вывести количество символов `wc -L` вывести длину самой длинной строки `wc -w` вывести количество слов



Задание 34

`-h`, `--human-readable` print sizes in human readable format (e.g., 1K 234M 2G)

`-s`, `--summarize` display only a total for each argument



Задание 35

Самая короткая команда, которая создаст три директории от dir1 до dir3.

5 Сертификат

Сертификат

Сертификат

6 Выводы

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

Список литературы

1. Введение в Linux