

Отчет о прохождении 3 этапа внешнего курса

Продвинутые темы

Иванов Сергей Владимирович, НПИбд-01-23

Содержание

1 Цель работы	5
2 Задание	6
3 Теоретическое введение	7
4 Выполнение лабораторной работы	8
5 Сертификат	28
6 Вывод	29
Список литературы	30

Список иллюстраций

4.1	1 Задание	8
4.2	2 Задание	9
4.3	3 Задание	9
4.4	4 Задание	10
4.5	5 Задание	10
4.6	6 Задание	11
4.7	7 Задание	11
4.8	8 Задание	12
4.9	9 Задание	12
4.10	10 Задание	13
4.11	11 Задание	13
4.12	12 Задание	14
4.13	13 Задание	14
4.14	14 Задание	15
4.15	15 Задание	16
4.16	16 Задание	17
4.17	17 Задание	17
4.18	18 Задание	18
4.19	19 Задание	18
4.20	20 Задание	19
4.21	21 Задание	19
4.22	22 Задание	20
4.23	23 Задание	20
4.24	24 Задание	21
4.25	25 Задание	21
4.26	26 Задание	21
4.27	27 Задание	22
4.28	28 Задание	22
4.29	29 Задание	23
4.30	30 Задание	23
4.31	31 Задание	24
4.32	32 Задание	25
4.33	33 Задание	26
4.34	34 Задание	26
4.35	35 Задание	27
4.36	36 Задание	27

5.1 Сертификат	28
--------------------------	----

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

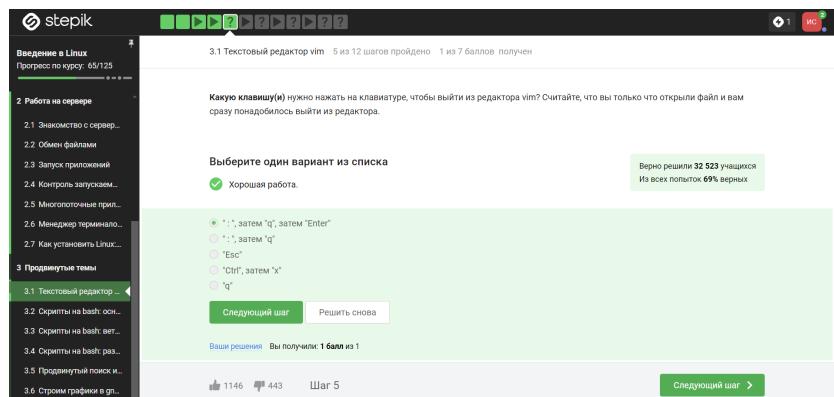


Рис. 4.1: 1 Задание

Стоит упомянуть, что у редактора vim есть тьюториал, 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.

The screenshot shows a Stepik course interface. On the left, there's a sidebar with a navigation tree for a 'Введение в Linux' course, including sections like 'Работа на сервере' and 'Текстовый редактор'. The main area displays a question about vim keyboard shortcuts. The question text is: '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 w 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 .' Below this is a section titled 'Выберите все подходящие ответы из списка' (Select all correct answers from the list). It contains several options with checkboxes, some of which are checked (e.g., 'В этой строке 5 "больших слов" (WORD)', 'Нажимая только на W, нельзя переместить курсор на " "'). At the bottom, there are buttons for 'Следующий шаг' (Next step) and 'Решить снова' (Solve again).

Рис. 4.2: 2 Задание

`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,

This screenshot shows another question from the same Stepik course. The sidebar and question text are identical to the previous one. The 'Выберите все подходящие ответы из списка' section contains several options, with 'd2wwifour four <<Esc>>' checked. At the bottom, there are buttons for 'Следующий шаг' and 'Решить снова'.

Рис. 4.3: 3 Задание

`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 неименованного буфера под курсором;
- P — вставка содержимого неименованного буфера перед курсором;
- yy (также Y) — копирование текущей строки в неименованный буфер;
- uy — копирование числа строк начиная с текущей в неименованный буфер;

3.1 Текстовый редактор vim 11 из 12 шагов пройдено 5 из 7 баллов получено

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 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

Прекрасный ответ.

:%s/Windows/Linux

Следующий шаг **Решить снова**

Ваши решения Вы получили: 2 балла из 2

Рис. 4.4: 4 Задание

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 %.

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

Здорово, всё 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](#).

Чтобы выйти из режима выделения, нужно ввести `:d`

Когда вы находитесь в режиме выделения, внизу редактора горит надпись – `VISUAL` – (или – ВИЗУАЛЬНЫЙ РЕЖИМ –)

В режиме выделения можно использовать команды `d` (удалить) и `u` (скопировать)

Выйти из режима выделения можно, нажав клавишу `Esc` два раза

В режиме выделения можно использовать команды перемещения (например, `W`, `e`, `S`, и др.)

Режим выделения открывается из нормального режима по нажатию `"`

Следующий шаг **Решить снова**

Ваши решения Вы получили: 2 балла из 2

Рис. 4.5: 5 Задание

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.

уу (также Y) — копирование текущей строки в буфер;

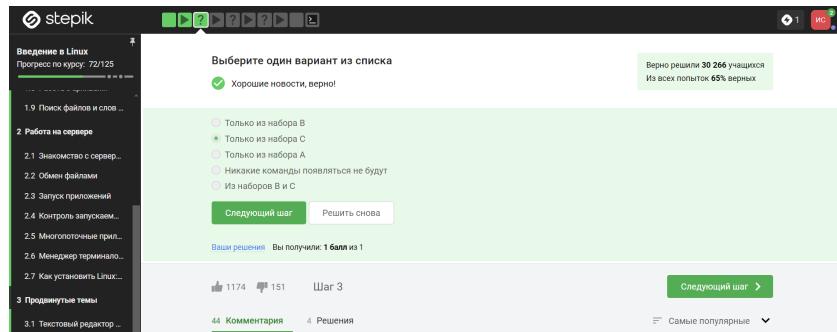


Рис. 4.6: 6 Задание

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

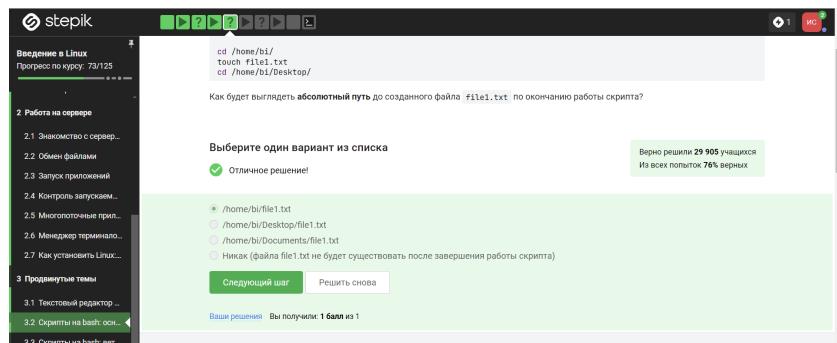


Рис. 4.7: 7 Задание

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

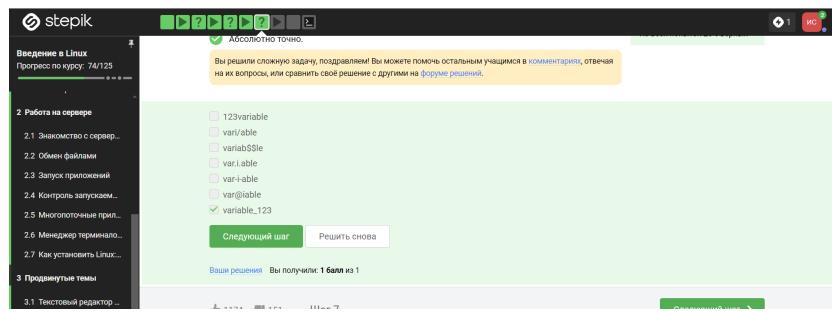


Рис. 4.8: 8 Задание

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

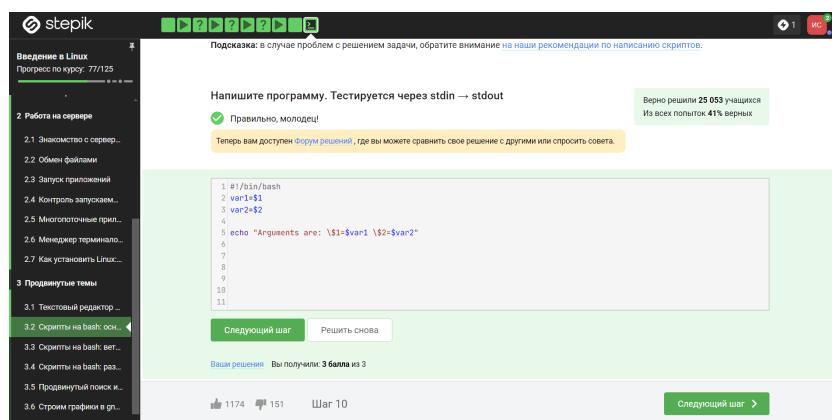


Рис. 4.9: 9 Задание

`$ 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.

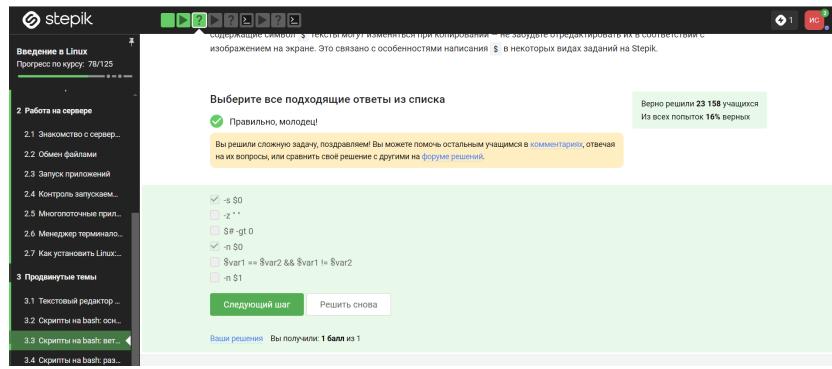


Рис. 4.10: 10 Задание

- **\$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.

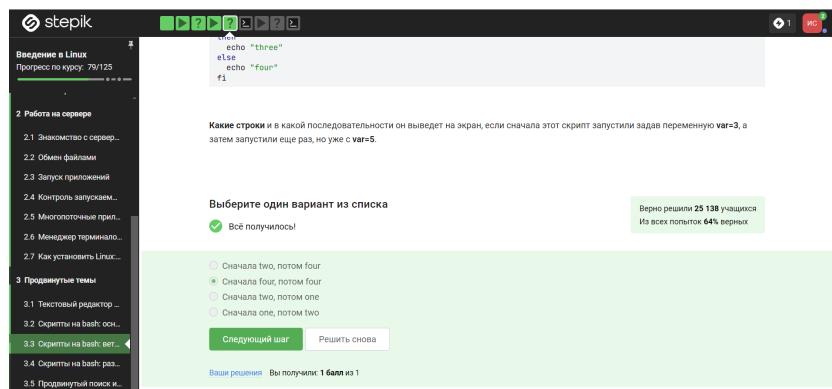


Рис. 4.11: 11 Задание

- **-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.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 2 and 3. Chapter 3.3 'Скрипты на bash: нет...' is selected. A task window titled 'Напишите программу. Тестируется через stdin → stdout' contains the following code:

```
1 if [[ $1 -eq 1 ]]; then
2   echo "$1 student"
3 elif [[ $1 -gt 1 && $1 -le 4 ]]; then
4   echo "$1 students"
5 elif [[ $1 -ge 5 ]]; then
6   echo "A lot of students"
7 else
8   echo "No students"
9 fi
10
11
12
13
14
```

Below the code are two buttons: 'Следующий шаг' and 'Решить снова'. A message at the bottom says 'Ваше решение: Вы получили 3 балла из 3'.

Рис. 4.12: 12 Задание

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

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 2 and 3. Chapter 3.3 'Скрипты на bash: нет...' is selected. A task window contains the following code:

```
for str in a b c d
do
  echo "start"
  if [[ $str > "c" ]]
  then
    continue
  fi
  echo "finish"
done
```

A note below the code says: 'Если запустить этот скрипт, то сколько раз на экран будет выведено слово "start", а сколько раз слово "finish"?' Below the code are two buttons: 'Следующий шаг' and 'Решить снова'. A message at the bottom says 'Ваше решение: Вы получили 1 балл из 1'.

Рис. 4.13: 13 Задание

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

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



Рис. 4.14: 14 Задание

```

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: " #To пустъ 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

```

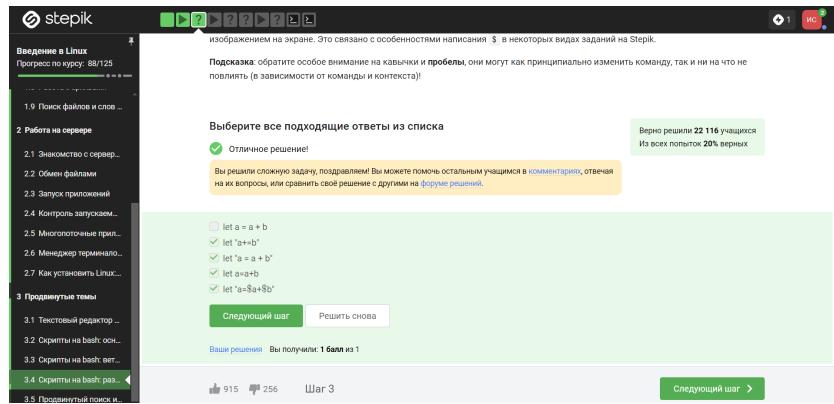


Рис. 4.15: 15 Задание

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)

The screenshot shows a Stepik course titled "Введение в Linux". The current step is "3. Продвинутые темы" and the specific question is "3.4 Скрипты на bash: раз...". The question text is: "Пусть вы находитесь в директории /home/b1/Documents/ и запускаете в ней скрипт следующего содержания: #!/bin/bash cd /home/b1/ echo \$PWD". The question asks: "Что в этом случае выведет команда echo на экран?". Below the question, a list of options is shown: "/home/b1/Documents", "pwd", "/home/b1", "'pwd'", and "Код возврата команды pwd (0 в случае успешного выполнения и не 0 в случае ошибок)". The user has selected the option "pwd". At the bottom of the screen, there are buttons for "Следующий шаг" and "Решить снова". The overall interface is clean with a white background and standard UI elements.

Рис. 4.16: 16 Задание

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

The screenshot shows a Stepik course titled "Введение в Linux". The current step is "3. Продвинутые темы" and the specific question is "3.4 Скрипты на bash: раз...". The question text is: "Выберите все подходящие ответы из списка". It includes a note: "Вы решали сложную задачу, поздравляем! Вы можете помочь остальным учащимся в комментариях, отвечая на их вопросы, или сравнить свое решение с другими на форуме решений." Below the note, a list of options is shown: "Ничего сделать нельзя", "Сначала запустить program, затем if [[\$? -eq 0]]", "if [[program -eq 0]]", "Сначала var=program, затем if [[\$var -eq 0]]", and "if [program > some_file.txt]". The first three options are checked. At the bottom of the screen, there are buttons for "Следующий шаг" and "Решить снова". The overall interface is clean with a white background and standard UI elements.

Рис. 4.17: 17 Задание

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

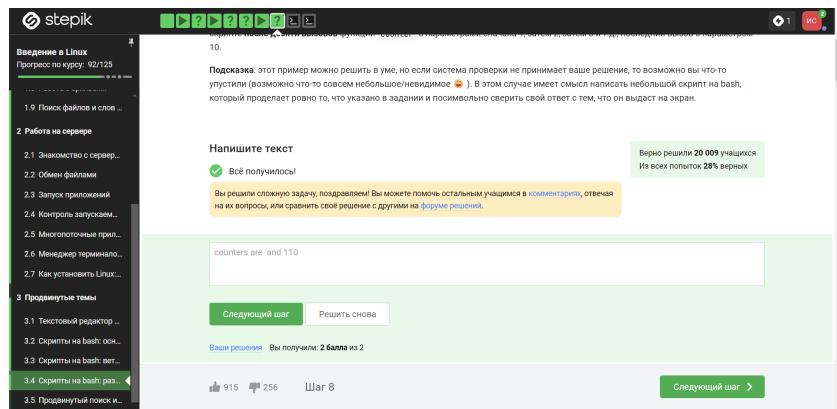


Рис. 4.18: 18 Задание

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.

```

5. if [ -z $1 ]; then
6.     echo "bye"
7.     break
8. else
9.     gcd () {
10.         remainder=$1
11.         if [ $2 -eq 0 ]
12.             then
13.                 echo "bye"
14.             fi
15.             while [ $remainder -ne 0 ]
16.             do
17.                 remainder=$((($1*$2)%$1))
18.                 n1=$n2
19.                 n2=$remainder
20.             done
21.         }
22.         gcd $1 $2
23.         echo "GCD is $1"
24.     fi
25. done
26.
27.
28.
29.

```

Рис. 4.19: 19 Задание

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.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The terminal window displays a script for a calculator. The script uses variables \$birinchi and \$ikkinch1 to store user input. It handles operations like addition (+), subtraction (-), multiplication (*), division (/), modulus (%), and exponentiation (**). It also checks for exit, bye, error, and assignment operations. The user has completed 101/125 steps.

```

6. if [[ $birinchi == "exit" ]]
7. then
8. echo "bye"
9. break
10. elif [[ "$birinchi" =~ ^[0-9]*$ && "$ikkinch1" =~ ^[0-9]*$ ]]
11. then
12. echo "error"
13. break
14. else
15. case $samol in
16. "+") let "result = $birinchi + $ikkinch1";;
17. "-") let "result = $birinchi - $ikkinch1";;
18. "/") let "result = $birinchi / $ikkinch1";;
19. "%") let "result = $birinchi % $ikkinch1";;
20. "**") let "result = $birinchi ** $ikkinch1";;
21. *) echo "error" ; break ;;
22. esac
23. echo "$result"
24. fi
25. done
26.
27.
28.
29.
30.

```

Следующий шаг | **Решить снова**

Ваши решения Вы получили 5 баллов из 5

Рис. 4.20: 20 Задание

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

The screenshot shows a Stepik course interface for 'Введение в Linux'. The terminal window displays a script for finding files. It uses the find command with -iname to search for files containing 'star*' in the /home/b1 directory. The user has completed 102/125 steps.

Путь в директории /home/b1 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/b1 -iname "star*"`, но **НЕ найдет** команда `find /home/b1 -name "star*"`.

Выберите все подходящие ответы из списка

Правильно, молодец!

Верно решили 20 547 учащихся
Из всех попыток 36% верных

star_trek_OST.mp3
 Star_Wars.avi
 STARS.txt
 stardust.mpeg
 Eddard_Stark_biography.txt

Следующий шаг | **Решить снова**

Ваши решения Вы получили 1 балл из 1

Рис. 4.21: 21 Задание

-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.

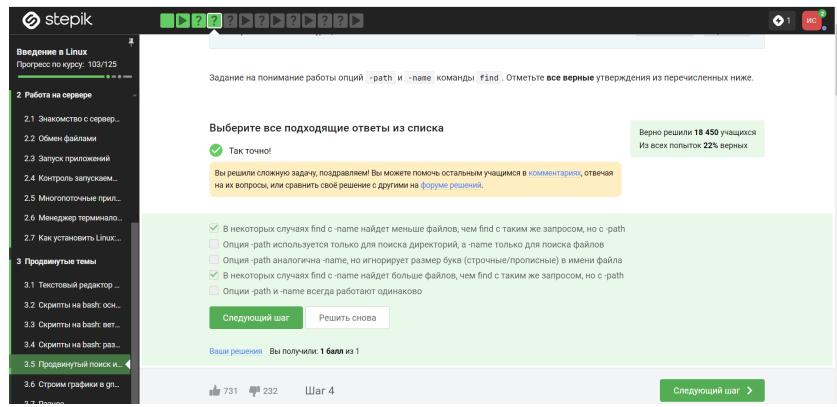


Рис. 4.22: 22 Задание

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);

expression - это выражение, которое определяет критерии поиска файлов.

-name: поиск файлов по имени. Например: `find /home/user -name myfile.txt`

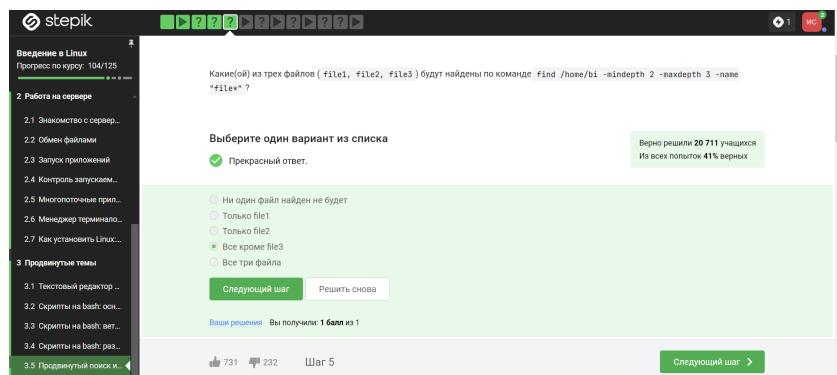


Рис. 4.23: 23 Задание

Текущий каталог - это $\text{depth}=1$, а остальное считается просто:

`/home/bi` -> $\text{depth}=1$

`/home/bi/dir1` -> $\text{depth}=2$

`/home/bi/dir1/dir2` -> $\text{depth}=3$

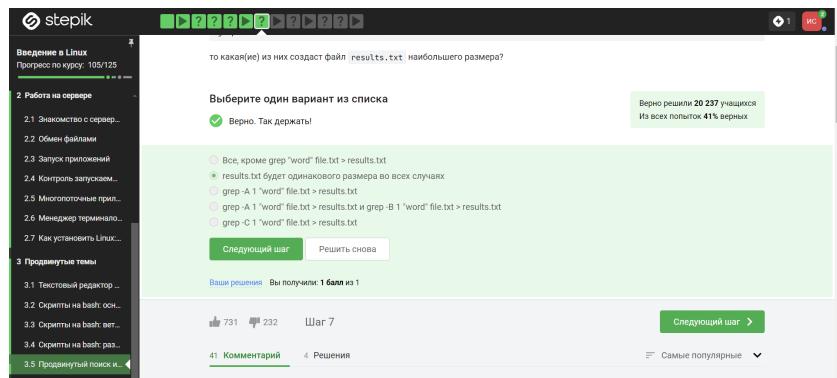


Рис. 4.24: 24 Задание

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

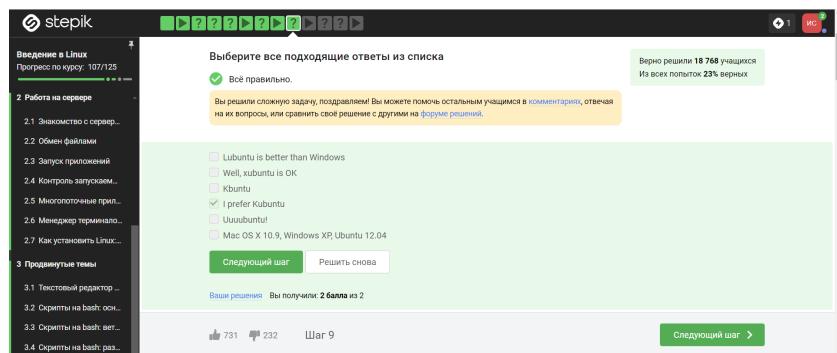


Рис. 4.25: 25 Задание

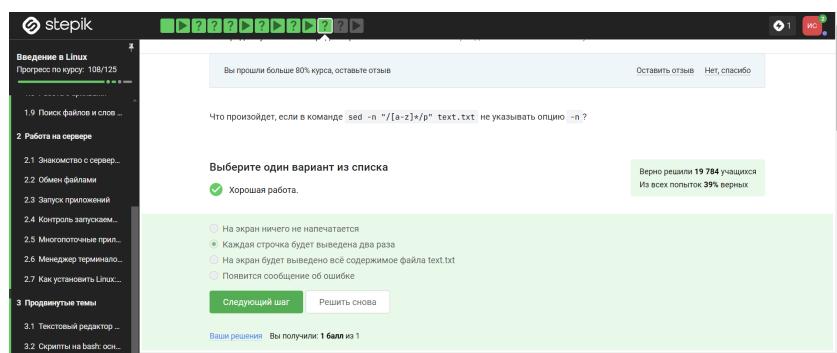


Рис. 4.26: 26 Задание

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.

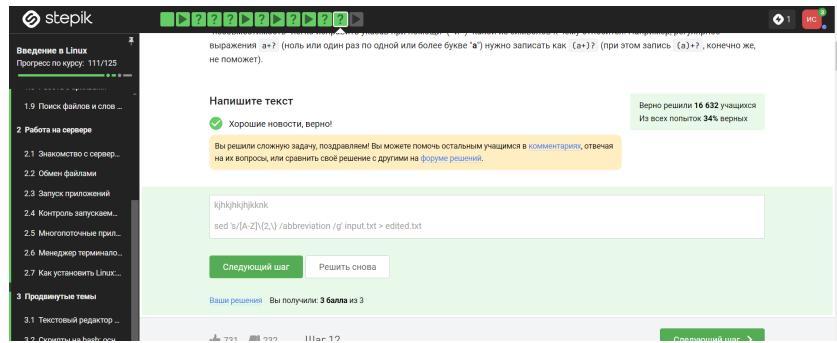


Рис. 4.27: 27 Задание

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.

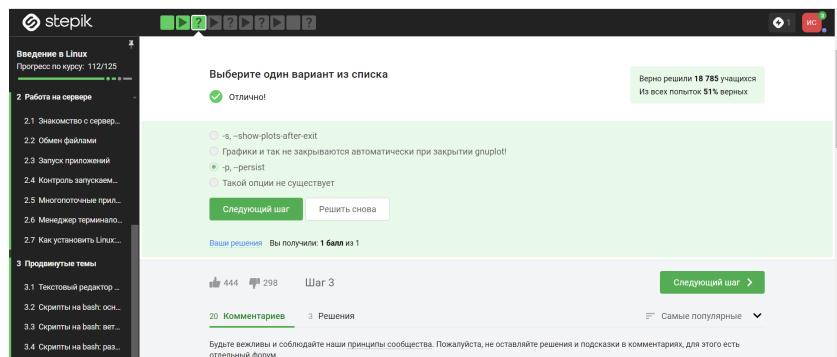


Рис. 4.28: 28 Задание

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

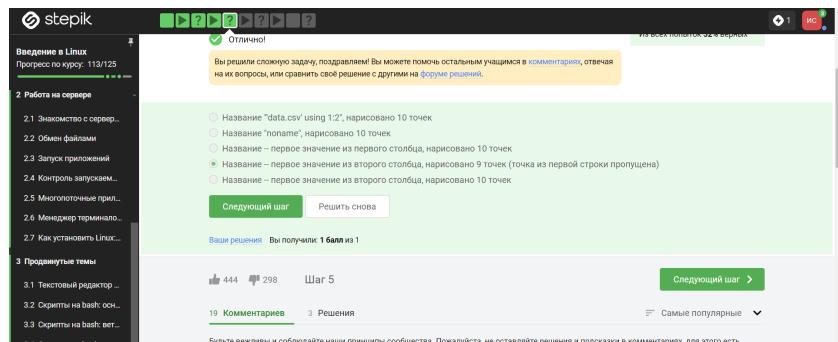


Рис. 4.29: 29 Задание

`plot 'data.csv' using 1:2` даст ошибку:
 warning: Skipping data file with no valid points ^ x range is invalid
 Скорее всего причиной такого поведения является тот факт, что формат CSV содержит строки, где столбцы разделены запятой? Содержимое файла:

1,21
 2,22
 3,23
 4,24
 5,25
 6,26
 7,27
 8,28
 9,29
 10,30

Рис. 4.30: 30 Задание

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.

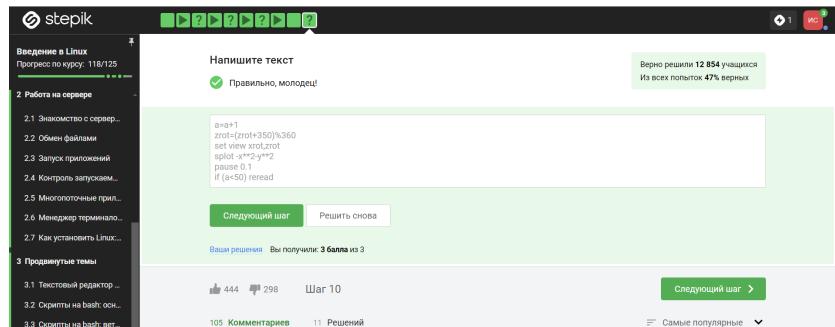


Рис. 4.31: Задание

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. Стока “`pause 0.2`” ставит выполнение на паузу на определенный промежуток времени. В задании сказали перерисовывать чаще, значит пауза должна быть меньше.

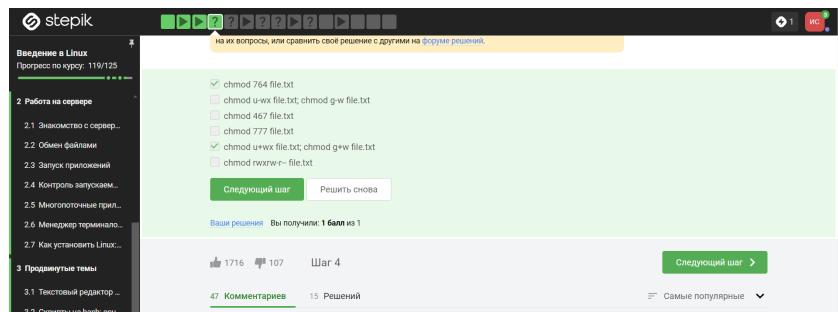


Рис. 4.32: Задание

- 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.

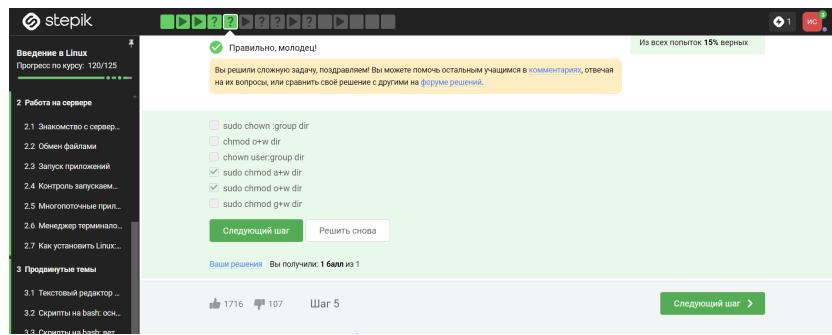


Рис. 4.33: 33 Задание

Решений два типа:

- Сменить права гостей, добавив W
- Сделать владельцем нужную группу или пользователя, в зависимости от того, у кого из них уже есть права на W
- Помнить, что root - владелец и остальные для него - others.

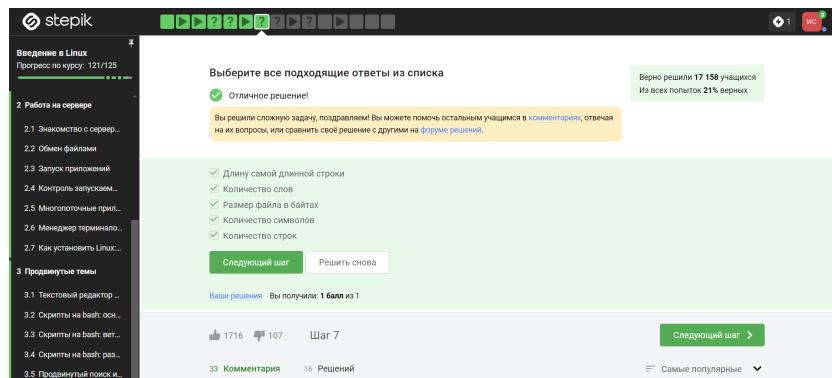


Рис. 4.34: 34 Задание

- 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

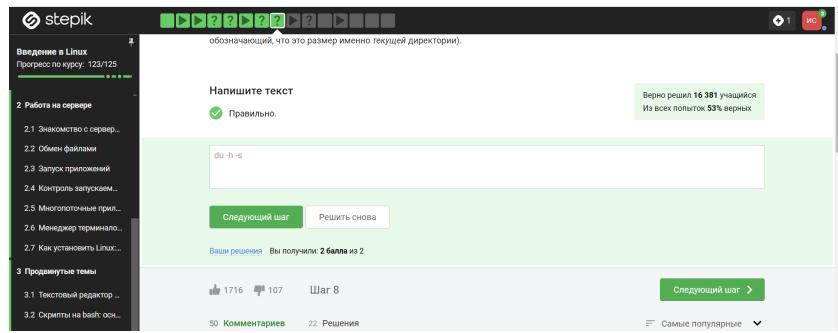


Рис. 4.35: 35 Задание

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

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

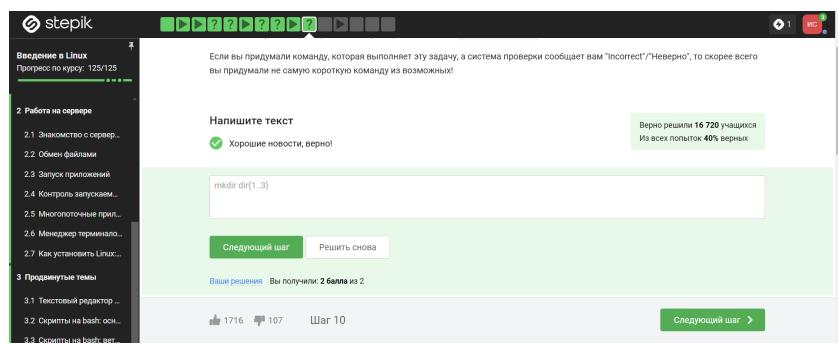


Рис. 4.36: 36 Задание

Команда создаст три директории от dir1 до dir3.

5 Сертификат

<https://stepik.org/cert/2466269>

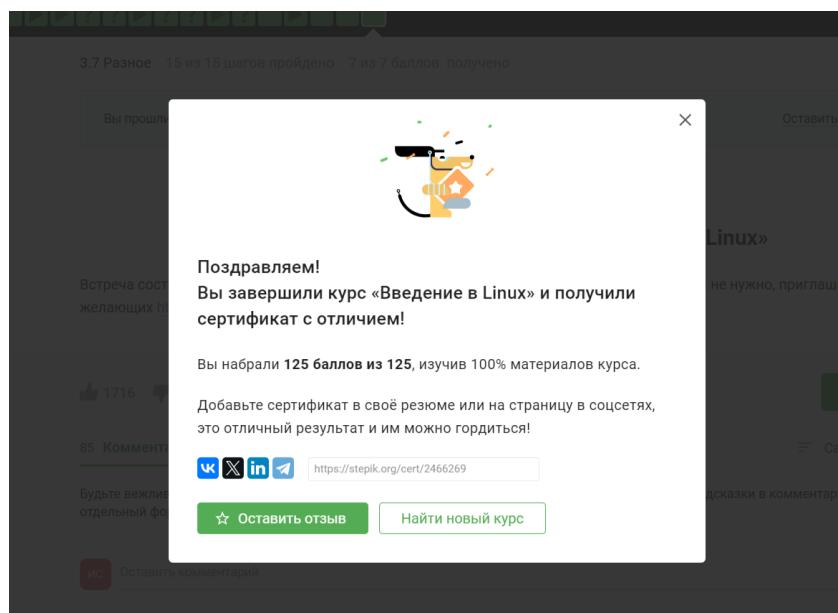


Рис. 5.1: Сертификат

6 Вывод

Я просмотрел курс и освежил в памяти навыки работы с архивами, скачивание файлов, команды grep и тп.

Список литературы

1. Введение в Linux