

Отчет о прохождении З этапа внешних курсов

Продвинутые темы

Выполнил: Чубаев Кирилл Евгеньевич, НММбд-01-24

Содержание

1	Цель работы.....	1
2	Задание	1
3	Теоретическое введение.....	1
4	Выполнение лабораторной работы.....	1
5	Сертификат.....	24
6	Выводы.....	24
	Список литературы	24

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

3 Этап: (рис. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41).

3.1 Текстовый редактор vim 12 из 13 шагов пройдено 7 из 10 баллов получено

Какую клавишу(и) нужно нажать на клавиатуре, чтобы выйти из редактора vim? Считайте, что вы только что открыли файл и вам сразу понадобилось выйти из редактора.

Выберите один вариант из списка

Верно.

Верно решили 32 523 учащихся
Из всех попыток 69% верных

• ':', затем 'q', затем 'Enter'
• ':', затем 'q'
• 'Esc'
• 'q'
• 'Ctrl', затем 'x'

Следующий шаг | Решить снова

Ваше решение Вы получили 1 балл из 1

1030 402 Шаг 5

3.1 Текстовый редактор vim 3.2 Скрипты на bash: основы 3.3 Скрипты на bash: ветвь... 3.4 Скрипты на bash: разн... 3.5 Продвинутый поиск и... 3.6 Строки графики в gnu... 3.7 Разное

Комментарии | Решения Самые популярные

Figure 1: Задание 1

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 изменили, 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.

При перемещении в vim "по словам" есть небольшая разница в том, используем мы маленькую (`w`, `e`, `b`) или большую (`W`, `E`, `B`) букву.

Первые перемещают нас по "словам" (`word`), а вторые по "большим словам" (`WORD`). Посмотрите справку по этим перемещениям и разберитесь в чем заключается разница между `word` и `WORD`.

А для того, чтобы убедиться, что вы разобрались, отметьте ниже все верные утверждения про следующую строку:

`Strange_ TEXT is_here. 2=2 YES!`

Примечание: во всех утверждениях имеется ввиду, что мы находимся в редакторе vim, включен нормальный режим работы и курсор находится в самом начале строки.

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 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`.

Выберите все подходящие ответы из списка

Абсолютно точно.

В этой строке 9 "слов" (`word`)

Нажимая только на `W`, нельзя переместить курсор на `"`

Чтобы попасть в конец строки, нужно одинаковое число нажатий, что на `W`, что на `w`

В этой строке 5 "слов" (`word`)

После 10 нажатий на `W` курсор окажется там же, где бы он был после 10 нажатий на `w`

В этой строке 12 "слов" (`word`)

Верно решено 25 385 учащихся
Из всех попыток 20% верных

Вы решали сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свой решение с другими на [Форуме решений](#).

Следующий шаг | **Решить снова**

Ваше решение Вы получили 1 балл из 1

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`,

Предположим, что в текстовом файле записана одна единственная строка:

`one two three four five`

и вам нужно преобразовать её в строку

`three four four four five`

Какие(й) из предложенных ниже наборов нажатий клавиш выполнит такое редактирование? В этих наборах нажатие на клавишу Esc обозначается как «`<Esc>`» (т. е. знаки «`c`» и «`z`» не несут отдельного смысла).

Примечание: во всех утверждениях имеется ввиду, что мы находимся в редакторе vim, включен нормальный режим работы и курсор находится в самом начале строки.

Выберите все подходящие ответы из списка

Верно.

Верно решено 23 655 учащихся
Из всех попыток 10% верных

Вы решали сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свой решение с другими на [Форуме решений](#).

Следующий шаг | **Решить снова**

Ваше решение Вы получили 1 балл из 1

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

3.1 Текстовый редактор vim 12 из 13 шагов пройдено 7 из 10 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 самое 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

✓ Верно. Так держать!

:%s/Windows/Linux

Следующий шаг Решить снова

Ваше решение Вы получили: 2 балла из 2

1030 402 Шаг 11 Следующий шаг >

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 %.

3.1 Текстовый редактор vim 12 из 13 шагов пройдено 7 из 10 баллов получено

Мы совсем не рассказали вам про третий режим работы vim – режим выделения (Visual). Предлагаем вам ознакомиться с ним самостоятельно. Например, это можно сделать во время прохождения упражнений в vimtutor, который мы настоятельно 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

✓ Хорошая работа.

Вы решите сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [Форуме решений](#).

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

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

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

Чтобы выйти из режима выделения, нужно ввести :q

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

✓ Режим выделения открывается из нормального режима по нажатию 'V'

Следующий шаг Решить снова

Ваше решение Вы получили: 2 балла из 2

1030 402 Шаг 12 Следующий шаг >

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.

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

The screenshot shows a Stepik course interface for a Linux introduction. The sidebar lists chapters 1.0 through 3.7. The main content area is titled '3.2 Скрипты на bash: основы' and shows step 3. The question asks: 'Надеемся, что вы разобрались, что одну оболочку (например, sh) можно запустить из другой оболочки (например, из bash). Предположим, что вы открыли терминал и у вас в нем запущена оболочка bash. Вы набираете в ней команды A1, A2, A3, а затем запускаете оболочку sh. В этой оболочке вы набираете команды B1, B2, B3 и запускаете оболочку bash. И, наконец в этой последней оболочке вы набираете команды C1, C2, C3. Если теперь вы попробуете при помощи стрелочек вверх/вниз перемещаться по истории набранных команд, то команды из какого набора(ов) будут появляться?' Below the question is a list of five options with radio buttons. A green checkmark indicates option 3 is correct: 'Только из набора С'. A green box at the top right says 'Верно решили 30 266 учащихся Из всех попыток 65% верных'. At the bottom are buttons for 'Следующий шаг' and 'Решить снова'.

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.

The screenshot shows a Stepik course interface for a Linux introduction. The sidebar lists chapters 1.0 through 3.7. The main content area is titled '3.2 Скрипты на bash: основы' and shows step 3. It displays a code fragment: '#!/bin/bash\ncd /home/bi/\ntouch file1.txt\ncd /home/bi/Desktop/'. The question asks: 'Как будет выглядеть абсолютный путь до созданного файла file1.txt по окончанию работы скрипта?' Below the question is a list of four options with radio buttons. A green checkmark indicates option 1 is correct: '/home/bi/file1.txt'. A green box at the top right says 'Верно решили 29 905 учащихся Из всех попыток 76% верных'. At the bottom are buttons for 'Следующий шаг' and 'Решить снова'.

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.

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [variables1.sh](#), [variables2.sh](#).

Какие из представленных ниже строк **могут** быть именами переменных в bash? Выберите **все** подходящие варианты!

Подсказка: если все варианты ответов являются неверными, то не отмечайте ни один из них и нажимайте кнопку "Отправить"/"Submit".

Выберите все подходящие ответы из списка

variable

vari/able

variable_123

var i able

var@iable

variable123

_variable

Всё получилось!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

Верно решили 27 188 учащихся
Из всех попыток 25% верных

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

Figure 8: Задание 8

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

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [arguments.sh](#).

Напишите скрипт на bash, который принимает на вход два аргумента и выводит на экран строку следующего вида:

Arguments are: \$1:первый_аргумент \$2:второй_аргумент

Например, если ваш скрипт называется `./script.sh`, то при запуске его `./script.sh one two` на экране должно появиться:

Arguments are: \$1=one \$2=two

Подсказка: в случае проблем с решением задачи, обратите внимание на [наши рекомендации по написанию скриптов](#).

Напишите программу. Тестируется через stdin → stdout

Правильно.

Теперь вам доступен [форум решений](#), где вы можете сравнить свое решение с другими или спросить совета.

Верно решили 25 053 учащихся
Из всех попыток 41% верных

```
1 #!/bin/bash
2 var1=$1
3 var2=$2
4
5 echo "Arguments are: \$1=\$var1 \$2=\$var2"
6
7
8
9
10
```

Следующий шаг Решить снова

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.

Вы можете скопировать и изучить скрипты, которые мы показали в видеофрагменте: [branching1.sh](#), [branching2.sh](#).

Предположим, вы пишете скрипт на bash и хотите использовать в нем конструкцию `if`, в следующем фрагменте:

```
if ([ ... ])
then
  echo "True"
fi
```

Вы можете написать вместо "...": `[[$1]]` и не забыть про пробел после `[[` и перед `]]` любое из перечисленных ниже условий. Сразу мы просим вас выбрать только то из них, при котором `echo` напечатает на экране `True`: вне зависимости от того, с какими параметрами был запущен ваш скрипт и какие в нем есть переменные.

Например, условие: `0 -eq $ # подадут`, т.к. ноль всегда равен нулю вне зависимости от аргументов и переменных внутри скрипта и он всегда будет напечатан `True`. В то же время условие `$var1 -eq 0 -ne подадут`, т.к. в переменной `var1`, как может быть записана ноль (тогда будет напечатано `False`), так это может и не быть (тогда ничего напечатано не будет).

Помощник: если вы познакомите профессора варианты ответа с собой в комментах, обратите внимание на то, что содержание символов `$` тексты могут меняться при копировании – не забудьте отредактировать их в соответствии с изображением на экране. Это связано с особенностями копирования `$` в некоторых видах задания на Stepik.

Выберите все подходящие ответы из списка

-n \$0
 \$# -ge 0
 -z "\$"
 -n \$1
 \$# != 0
 \$var1 == 0
 \$var1 == 3 && \$var1 != \$var2

[Следующий шаг](#) [Решить снова](#)

База решений Вы получите: 1 балл из 1

1078 132 Шаг 3

[Следующий шаг >](#)

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.

Вы можете скопировать и изучить скрипты, которые мы показали в видеофрагменте: [branching1.sh](#), [branching2.sh](#).

Помогите на фрагменте bash-скрипта:

```
if ([[ $var5 < 5 ]]
then
  echo $var5
else
  echo 5
fi
```

Какие строки и в какой последовательности они выведут на экран, если начнешь этот скрипт запустив задачу переменную `var5`, а затем запустив еще раз, но уже с `var5`?

Выберите один вариант из списка

Сначала 5, потом \$var5
 Сначала \$var5, потом 5
 Сначала 5, потом four
 Сначала four, потом one

[Следующий шаг](#) [Решить снова](#)

База решений Вы получите: 1 балл из 1

21 Комментарий 4 Решения

Самые популярные

Будьте вежливы и соблюдайте наши правила сообщества. Пожалуйста, не оставляйте решения и подсказки в комментариях, для этого есть специальный форум.

[Поставить комментарий](#)

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.

The screenshot shows a terminal window on the Stepik platform. The title bar says "Введение в Linux" and "Проект по курсу: 125/137". The main area contains a terminal session:

```
Напишите скрипт на bash, который принимает на вход один аргумент (целое число от 0 до бесконечности), который будет обозначать
число студентов в классе. В зависимости от значения числа нужно вывести разные сообщения.

Соответствие входа и выхода должно быть таким:

0 --- No students
1 --- 1 student
2 --- 2 students
3 --- 3 students
...
n --- n students
5 и больше --- A lot of students

Примечание 1) выводить можно только строку справа, т.е. ">" выводить не нужно.
Примечание 2) в строке с описанием задачи написано слово "let" с маленькой буквой
(например, что он будет иметь значение и значение от 0 до бесконечности), то этого делать не нужно!

Пример №1: если ваш скрипт называется ./script.sh , то при запуске его как ./script.sh 1 на экране должно появиться
1 student

Пример №2: если ваш скрипт называется ./script.sh , то при запуске его как ./script.sh 5 на экране должно появиться
A lot of students

Подсказка: в случае проблем с решением задания, обратите внимание на видеопрезентацию по написанию скриптов.

Напишите программу. Тестируется через stdin → stdout
```

Below the terminal, there is a code editor with the following script content:

```
[1]#!/bin/bash
#ваш скрипт
if [ $1 -lt 0 ]
then
    echo "Error"
else
    echo "There are $1 students"
fi
echo "Goodbye"
```

On the right side of the screen, there is a progress bar indicating "Всего решено 23 310 учащихся" and "Из всех попыток 38% верных".

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

The screenshot shows a terminal window on the Stepik platform. The title bar says "Введение в Linux" and "Проект по курсу: 125/137". The main area contains a terminal session:

```
3.0 Скрипты на bash: ветвление и циклы 0 из 9 шагов пройдено 10 из 10 баллов: получено
Вы можете скопировать и изучить скрипты, которые мы показали в видеофрагментах: lesson1.sh, lesson2.sh.

Помогите на выполнение вашего скрипта:

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

Below the terminal, there is a code editor with the following script content:

```
#!/bin/bash
#ваш скрипт
if [ $1 -lt 3 ]
then
    echo "Start"
elif [ $1 -gt 3 ]
then
    echo "Finish"
else
    echo "Error"
fi
```

On the right side of the screen, there is a progress bar indicating "Всего решено 24 582 учащихся" and "Из всех попыток 45% верных".

Figure 13: Задание 13

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

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

The screenshot shows the Stepik platform interface for a Linux task. The top navigation bar includes 'Вход в Linux' and 'Проект по курсу: 125/137'. The main area has a title '3. Скрипты на bash: введение и циклы' with a progress bar showing '9 из 9 шагов пройдено' and '10 из 10 файлов выполнено'. A sidebar on the left lists various topics under '1 Введение' and '2 Работа со скриптом'. The central workspace contains two code snippets labeled 'N1' and 'N2'. N1 contains a script that reads 'enter your name:' and 'enter your age:', then prints 'Your group is child' if the name starts with 'E'. N2 contains a script that reads 'enter your name:' and 'enter your age:', then prints 'Hello Petrachev, your group is youth' if the name is 'Petrachev' and the age is between 17 and 25. Below the code is a note: 'После этого скрипт спрашивает сообщение "enter your name:" и если начинается по первой (бесконечный цикл). Если в какой-то момент работы скрипта будет введен пустое имя или возраст 0, то скрипт должен завершить на звуке "肄" и вывести свою работу (авторизацию).'. A green button at the bottom right says 'Напишите программу. Тестируется через stdin → stdout'.

Figure 14: Задание 14

This screenshot shows the same Stepik task interface as Figure 14, but with a different code submission. The code in N1 is identical to Figure 14. The code in N2 is a more complex version that handles multiple cases for name and age, including handling for empty input and age 0. It also includes a break statement. Below the code is a note: 'Вы можете скопировать задание, поделиться им с друзьями в комментариях, отредактировать его или удалить, или оставить свой отзыв на форуме проекта.' A green button at the bottom right says 'Напишите программу. Тестируется через stdin → stdout'.

Figure 15: 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 [[ $stdout != 1 ]] ;do
    echo "enter your age: " #To пусть вводит возраст
    read age #Считываем возраст
    if [[ ($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

```

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [math1.sh](#), [math2.sh](#).

Какие(ая) из предложенных ниже инструкций увеличат значение переменной `a` на значение переменной `b`? Например, если в `a` было записано 10, то в `b` было 5, то в `a` должно записаться 15.

Выберите все подходящие варианты!

Примечание: если вы планируете проверять варианты ответов у себя в терминале, обратите внимание на то, что содержащие символ `$` тексты могут изменяться при копировании – не забудьте отредактировать их в соответствии с изображением на экране. Это связано с особенностями написания `$` в некоторых видах заданий на Stepik.

Подсказка: обратите особое внимание на кавычки и пробелы, они могут как принципиально изменить команду, так и ни на что не повлиять (в зависимости от команды и контекста)!

Выберите все подходящие ответы из списка

Прекрасный ответ.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

Верно решили 22 116 учащихся
Из всех попыток 20% верных

2.1 Знакомство с сервером
2.2 Обмен файлами
2.3 Запуск приложений
2.4 Контроль запускаемых...
2.5 Многопоточные прило...

a=\$a+\$b
 let a=\$a+\$b
 let a=a+b
 let "a=\$a+\$b"
 let a = a + b

Следующий шаг Решить снова

Figure 16: 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)

The screenshot shows a Stepik course interface for a Linux introduction. The main content area displays a task titled "3.4 Скрипты на bash: разное". It includes a code snippet in a terminal window:

```
#!/bin/bash
cd /home/bi/
echo "pwd"
```

A note below the code says: "Что в этом случае выведет команда echo на экран?" (What will the echo command output on the screen?). A feedback box indicates: "Выбери один вариант из списка" (Choose one option from the list) and "Хорошая работа." (Good job). A green bar at the bottom shows statistics: "Верно решили 23 677 учащихся" (23,677 students solved correctly) and "Из всех попыток 51% верных" (51% of attempts were correct).

Figure 17: Задание 16

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

The screenshot shows a Stepik course interface for a Linux introduction. The main content area displays a task titled "3.4 Скрипты на bash: разное". It includes a note: "Мы рассказали, что можно проверить код возврата внешней программы прямо в конструкции if при помощи `if '\$program options \$arg1...`" (We explained that you can check the exit code of an external program directly in an if construct using `if '\$program options \$arg1...'`). Another note says: "Однако как быть, если хочется всё-таки запустить программу program, которая пишет что-то в stdout и потом выполнить какие-то действия если ее код возврата равен 0? Выберите все верные утверждения или правильно работающие конструкции if." (However, what if you want to run the program program, which writes something to stdout and then performs some actions? Select all correct statements or choose the correct if constructs). A note at the bottom says: "Примечание: во всех вариантах ответов, где есть кавычки, используется именно косая кавычка (‘), а не обычная (‘) или двойная (“)."

A feedback box indicates: "Выбери все подходящие ответы из списка" (Select all correct answers) and "Здорово, всё верно." (Great, everything is correct). A green bar at the bottom shows statistics: "Верно решили 21 426 учащихся" (21,426 students solved correctly) and "Из всех попыток 20% верных" (20% of attempts were correct).

Figure 18: Задание 16_2

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

3.4 Скрипты на bash: разное 10 из 10 шагов пройдено 14 из 14 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [functions1.sh](#), [functions2.sh](#).

Посмотрите на функцию из bash-скрипта:

```
counter () # takes one argument
{
    local let "c1=$1"
    let "c2+=${!1}+2"
}
```

Впишите в форму ниже строку, которую выведет на экран команда `echo "counters are $c1 and $c2"` если она находится в скрипте после **десяти вызовов** функции `counter` с параметрами сначала 1, затем 2, затем 3 и т.д., последний вызов с параметром 10.

Подсказка: этот пример можно решить в уме, но если система проверки не принимает ваше решение, то возможно вы что-то упустили (возможно что-то совсем небольшое/невидимое ☺). В этом случае имеет смысл написать небольшой скрипт на bash, который проделает ровно то, что указано в задании и поисковом запросе свой ответ с тем, что он виден на экран.

Напишите текст

Верно решили 20 009 учащихся
Из всех попыток 28% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравните свой решение с другими на [форуме решений](#).

[Следующий шаг](#) [Решить снова](#)

Figure 19: 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.

3.4 Скрипты на bash: разное 10 из 10 шагов пройдено 14 из 14 баллов получено

Напишите скрипт на bash, который будет искать наибольший общий делитель (НОД, greatest common divisor, GCD) двух чисел. При запуске ваш скрипт не должен ничего писать на экран, а просто ждет ввода двух натуральных чисел через пробел (для этого можно использовать `read` и указать ему две переменные – см. пример в видеофрагменте). После ввода чисел скрипт считает их НОД и выводит на экран сообщение **"GCD is <полученное значение>"**, например, для чисел 15 и 25 будет "GCD is 5". После этого скрипт опять выходит в режим ожидания двух натуральных чисел. Если в какой-то момент работы пользователя ввел вместо этого пустую строку, то нужно написать на экран `bye` и закончить свою работу.

Вычисление НОД несложно реализовать с помощью [алгоритма Евклида](#). Вам нужно написать функцию `gcd`, которая принимает на вход два аргумента (назовем их **M** и **N**). Если аргументы равны, то мы нашли НОД – он равен **M** (или **N**), нужно выводить соответствующее сообщение на экран (см. выше). Иначе нужно сравнивать аргументы между собой. Если **M** больше **N**, то запускаем ту же функцию `gcd`, но в качестве первого аргумента передаем (**M-N**), а в качестве второго **N**. Если же наоборот, **M** меньше **N**, то запускаем функцию `gcd` с первым аргументом **M**, в втором (**N-M**).

Пример корректной работы скрипта:

```
./script.sh
10 15
GCD is 5
7 5
GCD is 1
bye
```

Примечание: в вызове функции из себя самой нет ничего страшного или неправильного, т.ч. смело вызывайте `gcd` прямо внутри `gcd`!

Примечание 2: для завершения работы функции в произвольном месте, можно использовать инструкцию `return` (все инструкции функции после `return` выполняться не будут). В отличии от `exit` эта команда завершит только функцию, а не выполнение всего скрипта целиком. Однако в данной задаче можно обойтись и без использования `return`!

Подсказка: в случае проблем с решением задачи, обратите внимание на [наши рекомендации по написанию скриптов](#).

Figure 20: Задание 18

The screenshot shows a Stepik course interface for a Linux course. The left sidebar lists chapters and sub-chapters. The main area displays a code editor with the following C-like pseudocode:

```

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

```

Below the code editor are two buttons: "Следующий шаг" (Next step) and "Решить снова" (Solve again). A note at the bottom says: "Ваше решение: Вы попали в 4 балла из 4".

Figure 21: 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.

The screenshot shows a Stepik course interface for a Linux course. The left sidebar lists chapters and sub-chapters. The main area displays a code editor with the following C-like pseudocode:

```

10 * 1
2 ** 10
exit

```

Below the code editor is a note: "то на экране будет:" (so on the screen will be:).

```

11
120
bye

```

Below that is another note: "Если же на вход поступит следующий файл:" (If however on input comes the following file:).

```

-2
exit

```

Below that is a note: "т.к. вторая команда была некорректной (в ней всего один аргумент, т.к. нет пробелов между числом и операцией в единственная допустимая команда из одного аргумента это "exit")." (because the second command was incorrect (there is only one argument, i.e. no spaces between the number and the operation, in the single allowed command from one argument this is "exit").)

At the bottom, there is a note: "Подсказка в случае проблем с решением задачи, обратите внимание на наши рекомендации по написанию скриптов." (Tip in case of problems with solving tasks, pay attention to our recommendations for writing scripts.)

Figure 22: Задание 19

The screenshot shows a Stepik task interface for "Введение в Linux". The task title is "Напишите программу. Тестируется через stdin -> stdout". The code area contains a shell script:

```

1 #!/bin/bash
2 #title {{ True }}
3
4 read strinck1 eval imainch1
5 if [[ $strinck1 == "exit" ]]
6 then
7   echo "exit"
8   break
9 else [[ "$strinck1" =~ ^([a-z]+) && "$imainch1" =~ ^([a-z]+)$ ]]
10 then
11   echo "error"
12   break
13 fi
14
15 case $strinck1 in
16   "+") let result=$imainch1+$strinck1;;
17   "-") let result=$imainch1-$strinck1;;
18   "/"") let result=$imainch1/$strinck1;;
19   "*") let result=$imainch1*$strinck1;;
20   "%") let result=$imainch1%$strinck1;;
21   "<") let result=$imainch1<$strinck1;;
22   ">") let result=$imainch1>$strinck1;;
23   "<=") let result=$imainch1<=$strinck1;;
24   ">=") let result=$imainch1>=$strinck1;;
25   "<>") let result=$imainch1<>$strinck1;;
26   "<=") let result=$imainch1<=$strinck1;;
27   ">=") let result=$imainch1>=$strinck1;;
28   "<>") let result=$imainch1<>$strinck1;;
29   esac
30 echo "$result"
31 fi
32 done
33
34
35
36
37

```

Below the code are two buttons: "Следующий шаг" and "Решить снова". A summary bar at the bottom says "Ваше решение: Вы получили: 5 баллов из 5".

Figure 23: Задание 19

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 task interface for "Введение в Linux". The task title is "3.5 Продвинутый поиск и редактирование". The code area contains a command:

```
Пусть в директории /home/b1 лежат файлы Star_Wars.avi, star_trek_OST.mp3, STARS.txt, stardust.mpeg, Eddard_Stark_biography.txt.
```

Instructions: Отметьте все файлы, которые **найдет** команда `find /home/b1 -iname "star*"`, но **НЕ найдет** команда `find /home/b1 -name "star*"`.

Below the command is a list of files with checkboxes:

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

Below the list are two buttons: "Следующий шаг" and "Решить снова". A summary bar at the bottom says "Ваше решение: Вы получили: 1 балл из 1".

Figure 24: 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 на понимание работы опций `-path` и `-name` команды `find`. Отметьте все верные утверждения из перечисленных ниже.

Выберите все подходящие ответы из списка

Абсолютно точно.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

Если заменить в команде поиска `-name`, на `-path`, то результат поиска всегда останется неизменным

Опция `-path` аналогична `-name`, но игнорирует размер букв (строчные/прописные) в имени файла

Если заменить в команде поиска `-name`, на `-path`, то результат поиска иногда может остаться таким же

В некоторых случаях `find` с `-name` найдет меньше файлов, чем `find` с таким же запросом, но с `-path`

Опция `-path` используется только для поиска директорий, а `-name` только для поиска файлов

Следующий шаг **Решить снова**

Ваши решения Вы получили: **1 балл из 1**

648 лайков 213 комментариев Шаг 4

Миниатюры Решения

Figure 25: 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);
`expression` - это выражение, которое определяет критерии поиска файлов.

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

Предположим, что в директории `/home/bi/` есть следующая структура файлов и поддиректорий:

```

/home/bi/
└── dir1
    ├── file1
    └── dir2
        ├── file2
        └── dir3
            └── file3

```

Какие(ой) из трех файлов (`file1`, `file2`, `file3`) будут найдены по команде `find /home/bi -mindepth 2 -maxdepth 3 -name "file*"`?

Выберите один вариант из списка

Хорошая работа.

Все кроме `file1`

Ни один файл найден не будет

Все кроме `file3`

Только `file1`

Все кроме `file2`

Следующий шаг **Решить снова**

Figure 26: 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 на понимание работы опций `-A`, `-B` и `-C` команды `grep`. Пусть у вас есть файл `file.txt` из 10 строк, причем в каждой строке есть слово `'word'`. Если вы выполните на этом файле команды:

```
grep "word" file.txt > results.txt
grep -A 1 "word" file.txt > results.txt
grep -B 1 "word" file.txt > results.txt
grep -C 1 "word" file.txt > results.txt
```

то какая(ие) из них создаст файл `results.txt` наибольшего размера?

Выберите один вариант из списка

Отлично!

Верно решили 29 237 учащихся
Из всех попыток 41% верных

grep -A 1 "word" file.txt > results.txt
 Все, кроме grep "word" file.txt > results.txt
 results.txt будет одинакового размера во всех случаях
 grep -C 1 "word" file.txt > results.txt
 grep -A 1 "word" file.txt > results.txt и grep -B 1 "word" file.txt > results.txt

Следующий шаг **Решить снова**

Ваши решения Вы получили: 1 балл из 1

Figure 27: 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

Предположим, что в файле `text.txt` записаны строки, показанные среди вариантов ответа. Отметьте только те из них, которые выведут на экран команда `grep -E "xktxKLJ?u\buntu\$" text.txt`.

Выберите все подходящие ответы из списка

Хорошие новости, 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](#).

Верно решили 18 768 учащихся
Из всех попыток 23% верных

Mac OS X 10.9, Windows XP, Ubuntu 12.04
 Well, xubuntu is OK
 I prefer Kubuntu
 Kubuntu
 Uuuubuntu!
 Linux is not always Ubuntu

Следующий шаг **Решить снова**

Ваши решения Вы получили: 2 балла из 2

Figure 28: Задание 24

The screenshot shows a Stepik task interface. On the left, there's a question: "Выберите все подходящие ответы из списка". It contains several options, with one checked: "Linux is not always Ubuntu". A green checkmark indicates this is a correct answer. Below the question are two buttons: "Следующий шаг" (Next step) and "Решить снова" (Solve again). At the bottom, it says "Ваши решения Вы получили: 2 балла из 2". On the right, there's a terminal window showing a session where the user runs commands like `touch text.txt`, `gedit text.txt`, and `grep` to search for "Linux" in the file. The terminal output includes lines like "Linux is not always Ubuntu" and "I prefer Kubuntu".

Figure 29: Задание 24

Объяснение на втором скриншоте.

This screenshot shows another Stepik task. The question asks: "Что произойдет, если в команде `sed -n "/[a-z]*p" text.txt` не указывать опцию `-n`?" (What will happen if you don't specify the `-n` option in the command `sed -n "/[a-z]*p" text.txt`?). The correct answer is "Каждая строчка будет выведена два раза" (Each line will be printed twice), which is marked with a green checkmark. Below the question are buttons for "Следующий шаг" and "Решить снова". At the bottom, it says "Ваши решения Вы получили: 1 балл из 1". The sidebar on the left lists course modules: "Введение в Linux" (Introduction to Linux) and "Работа на сервере" (Working on a server). The main content area shows a comment section with 648 likes and 213 dislikes, and a "Шаг 11" button.

Figure 30: 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.

The screenshot shows a Stepik task interface for "Введение в Linux". The main area displays a terminal window with the following command:

```
sed -e 's/ABBA/ABBA/g' input.txt > edited.txt
```

Below the terminal, there is explanatory text about the 'abbreviatuра' command and its usage.

Figure 31: 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.

The screenshot shows a Stepik task interface for "Введение в Linux". The main area displays a configuration dialog for gnuplot:

Выберите один вариант из списка

Верно. Так держать!

Графики и так не закрываются автоматически при закрытии gnuplot!
 -p, -persist
 Такой опции не существует
 -s, -show-plots-after-exit

Below the dialog, there is explanatory text and a summary of user responses.

Figure 32: Задание 27

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

3.6 Строим графики в gnuplot 10 из 10 шагов пройдено 7 из 7 баллов получено

Предположим у вас есть файл `data.csv` с двумя столбцами по 10 чисел в каждом. В первой строке не записаны названия столбцов, т.е. ряды данных начинаются прямо с первой строки. Вы запускаете gnuplot и вводите в него две команды:

```
set key autotitle columnhead
plot 'data.csv' using 1:2
```

Какое в этом случае будет **название** **у построенного ряда данных** и сколько будет нарисовано **точек** на графике?

Выберите один вариант из списка

Отличное решение!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свой ответ с другими на [форуме решений](#).

(*) Название – первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)
 Название "data.csv" using 1:2", нарисовано 10 точек
 Название – первое значение из второго столбца, нарисовано 10 точек
 Название – первое значение из первого столбца, нарисовано 9 точек (точка из первой строки пропущена)
 Название "нолите", нарисовано 10 точек

[Следующий шаг](#) [Решить снова](#)

Figure 33: Задание 28

`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
```

3.6 Строим графики в gnuplot 10 из 10 шагов пройдено 7 из 7 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [plot.gnu](#), [plot_advanced.gnu](#), [plot_advanced2.gnu](#). Все три скрипта основаны на [этой заметке](#), данные также взяты оттуда.

Предположим, что вы пишите gnuplot-скрипт и у вас в нем есть три переменные `x1`, `x2`, `x3`, в которых записаны координаты важных точек по оси ОХ (по возрастанию). Вы хотите, чтобы на этой оси было только три деления (т.е. три черточки) в этих самых координатах, а подпись этих делений были оформлены в виде "point **номер точки**, value <значение соответствующей переменной>".

Например, для `x1=0, x2=10, x3=20`, это были бы надписи "point 1, value 0" в точке с координатой 0 по горизонтали, "point 2, value 10" в точке с координатой 10 и "point 3, value 20" в точке с координатой 20.

Или, например, `x1=100, x2=150, x3=250`, это были бы надписи "point 1, value 100" в точке с координатой 100, "point 2, value 150" в точке с координатой 150 и "point 3, value 250" в точке с координатой 250.

Впишите в форму ниже **одну команду** (т.е. одну строку), которую нужно добавить в скрипт, для выполнения этой задачи.

Примечание: проверять, что переменные `x1`, `x2`, `x3` идут по возрастанию или что они являются числами **не нужно**!

Примечание 2: в видеофрагменте на предыдущем шаге звучал термин конкатенация, который важен для выполнения данного задания. Под конкатенацией обычно понимают "склеивание" двух строк в одну длинную строку, например, конкатенация строк "Данные из файла" и "data.csv" даст строку "Данные из файла data.csv".

Подсказка: настоятельно рекомендуем изучить примеры скриптов – в них есть большая часть решения!

Напишите текст

Здорово, всё верно.

Верно решили 13 935 учащихся
Из всех попыток 44% верных

`set xtics ('point 1, value ' x1, 'point 2, value ' x2, 'point 3, value ' x3)`

Figure 34: Задание 29

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.

The screenshot shows a Stepik course interface. On the left, a sidebar lists chapters and sub-chapters for 'Введение в Linux'. The main area displays a task titled 'Строим графики в gnuplot'. The task text asks to change instructions in a file named 'move.got' to create a rotating graphic. It provides three bullet points explaining changes: 1. The graph rotates around the horizontal plane. 2. The image rotates in the opposite direction. 3. Rotation speed is doubled. Below the text is a code editor containing a snippet of gnuplot script. A message box at the bottom right indicates that 12,854 students solved the task correctly. Buttons for 'Следующий шаг' (Next step) and 'Решить снова' (Solve again) are visible at the bottom.

Figure 35: Задание 30

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.

Введение в Linux
Прогресс по курсу: 125/137

3.7 Разное 15 из 15 шагов пройдено 7 из 7 баллов получено

Какая команда(ы) установят файлу `file.txt` права доступа `rwxr--r--`, если изначально у него были права `r--r--r--`. Укажите **все** верные варианты ответа!

Примечание: запись вида `команда1; команда2; команда3` означает, что в терминале последовательно выполнились все три команды (сначала `команда1`, затем `команда2` и, наконец, `команда3`).

Выберите все подходящие ответы из списка

Так точно!

Вы решили сложную задачку, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свой ответ с другими на [форуме решений](#).

chmod o-wx file.txt; chmod g-x file.txt; chmod a+wx file.txt
 chmod u-wx file.txt; chmod g+w file.txt
 chmod 764 file.txt
 chmod rxrw-r-- file.txt
 chmod 467 file.txt
 chmod u+wx file.txt; chmod g+w file.txt

[Следующий шаг](#) [Решить снова](#)

Ваше решение Вы получили: 1 балл из 1

1529 99 Шаг 4 [Следующий шаг >](#)

44 Комментария 11 Решений [Самые популярные](#)

Figure 36: Задание 31

- **г - 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.**

Предупреждение вы использовали команду `sudo` для создания директории `dir`. По умолчанию для `dir` были выставлены права доступа `umask 002` (владелец root, группа root). Таким образом никто кроме пользователя root не может ничего записывать в эту директорию, например, не может создавать файлы в ней.

После выполнения какой команды из группы `root` все-таки сможет создать файл внутри `dir`? Укажите все верные варианты ответов!

Примечание: считается, что все команды выполняются от имени root, если явно не указано, что команда выполнена с rootом.

Примечание: вы выбрали группу с директориями, а не с файлами (это ошибочно).

Дело в том, что если создать при помощи `sudo` файл с правами `rw-r--r--` в директории, которая наследует права владельца, то вспомогательная ситуация. С одной стороны пользователь может писать в этот файл (т.к. для записи нужно иметь права), с другой стороны файлы внутри этой директории не могут быть изменены (т.к. для изменения нужно иметь права на директорию (т.к. права "w" у файла есть только для root)). При этом некоторые "умные" редакторы, например, vim позволяют даже редактировать эти файлы, но сделает она это своеобразно: через удаление строки и создание копии узла с измененными правами (уделить это можно, но изменить это нельзя, т.к. изменения не склоняются). Итого получается, что несмотря на права на директорию, нельзя изменять файлы в ней.

В случае же, когда речь идет о директории созданной root, ситуация будет противоположная: сможет смотреть ее содержимое (у него есть право "У") но удалить и создать файлы в ней не сможет (права "w" у него нет).

Важно отметить, что директории и файлы это абсолютно разные вещи. Создание директории "dir" в директории root позволяет проконтролировать ее состав. Право "w" у директории дает возможность удалить/добавлять новые "записи", т.к. удалить/создавать файлы подразумевается в нем.

Но если директория root еще не имеет прав на директорию, то вспомогательная ситуация не меняется. Там, где есть право "w" у директории (или директорий) с таким атрибутом сможет удалить только их владелец или зависимые от них, установленные в директории, в которых эти файлы (или директории) лежат.

Определение способа следующего курса Алексея Антирофу за помощь в оформлении [Примечания 21](#)

Выберите все подходящие ответы из списка

проверенный ответ

Вы решите сложную задачу, поздравляем! Вы можете помочь остальным ученикам в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

chosen user group dir
 sudo chmod +rw dr
 sudo chmod user dr
 sudo chmod +rw dr
 sudo chmod +w dr
 chmod +w dr

[Следующий шаг](#) [Решить снова](#)

Верно решили 14 682 учащихся
Из всех попыток 15% верных

Figure 37: Задание 32

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.

3.7 Разное 15 из 15 шагов пройдено 7 из 7 баллов получено

Отметьте какие характеристики файла можно посчитать с использованием команды `wc`.

Выберите все подходящие ответы из списка

Так точно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным ученикам в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

Длину самой длинной строки
 Размер файла в байтах
 Количество строк
 Количество слов
 Количество предложений

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: **1 балл из 1**

1529 99 Шаг 7

100% Кликабельные 100% Решено

Сложность выполнения

Figure 38: 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

Stepik interface showing a task from the 'Введение в Linux' course. The task requires writing a command to show disk usage in human-readable format. The user has entered 'du -h -s'. The feedback says it's a good answer.

Figure 39: 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

Stepik interface showing a task from the 'Введение в Linux' course. The task requires writing a command to create three directories named dir1, dir2, and dir3. The user has entered 'mkdir dir{1..3}'. The feedback says it's a perfect answer.

Figure 40: Задание 35

Команда создаст три директории от dir1 до dir3.

5 Сертификат

Figure 41: Сертификат

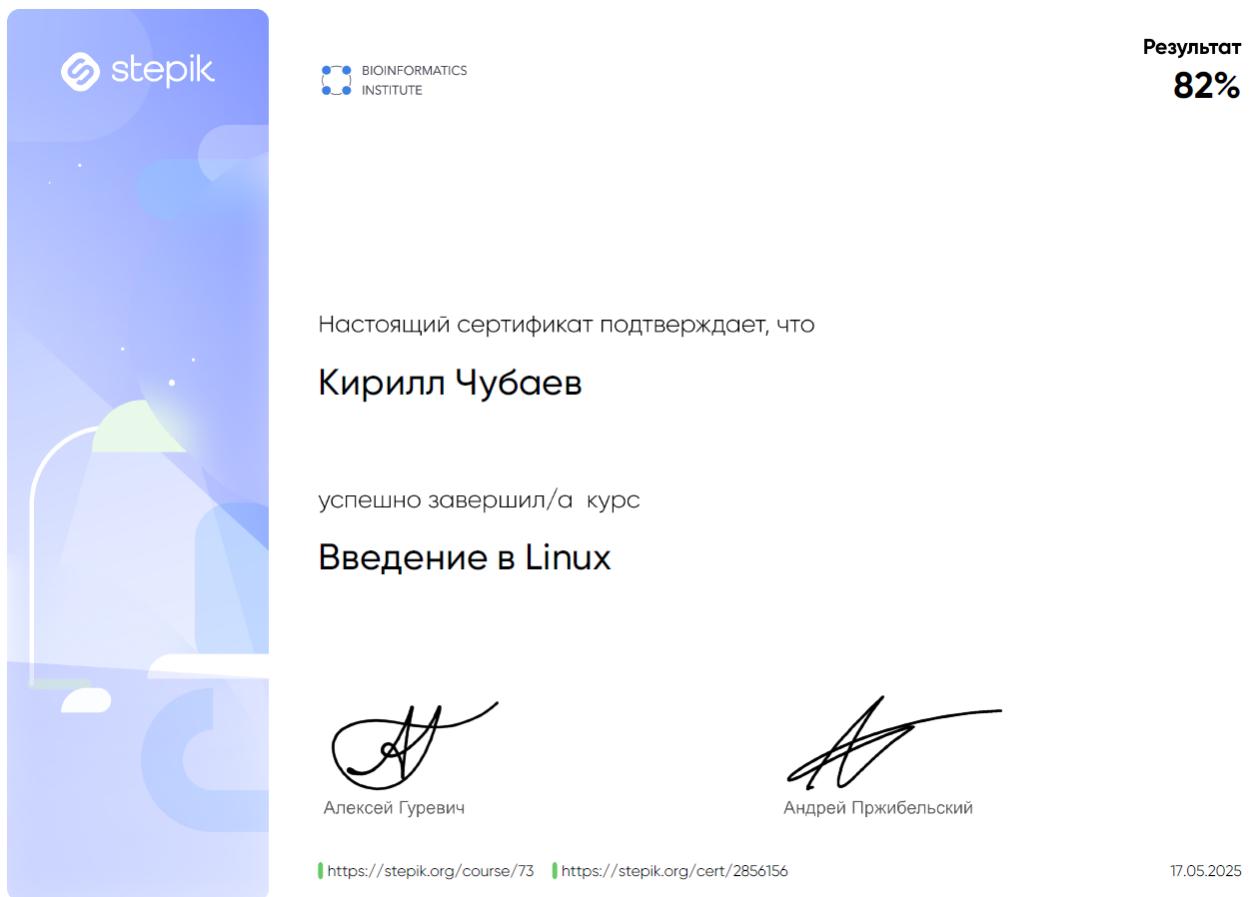


Figure 41: Сертификат

6 Выводы

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

Список литературы

1. Введение в Linux