Отчёт по прохождению внешнего курса на Stepik (этап 1)

Введение в Linux

Дарья Эдуардовна Ибатулина

Содержание

# 1 Цель работы

Пройти курс “Введение в Linux” на платформе [stepik.org] и получить сертификат. Для этого необходимо просмотреть видеоролики и выполнить задания, чтобы закрепить полученный материал. Процесс выполнения заданий требуется записать в виде скринкаста.

# 2 Задание

Пройти 3 этапа курса, записывая скринкасты, получить сертификат, сделать отчёт.

# 3 Теоретическое введение

Linux — это семейство операционных систем (ОС), работающих на основе одноименного ядра. Нет одной операционной системы Linux, как, например, Windows или MacOS. Есть множество дистрибутивов (набор файлов, необходимых для установки ПО), выполняющих конкретные задачи.

Линус Торвальдс — первый разработчик и создатель Linux. Именно в честь него и была названа ОС. В 1991 году Линус начал работу над собственной ОС семейства Unix. Через три года появилась первая версия, доступная для скачивания. Но тогда она имела очень низкий спрос — ей пользовались буквально несколько человек. Только через 10 лет ОС Linux получила широкое распространение. Сообщество программистов подхватило идею свободного ПО, специалисты стали помогать развивать проект. В ходе курса я познакомлюсь с основными командами терминала, научусь работать с серверами, познаю текстовый редактор *vi/vim* и рисование графиков в *gnuplot*.

# 4 Выполнение первого этапа курса

## 4.1 Общая информация о курсе (1.1)

Сначала авторизуемся на платформе [stepik.org] (рис. [1](#fig:001)).

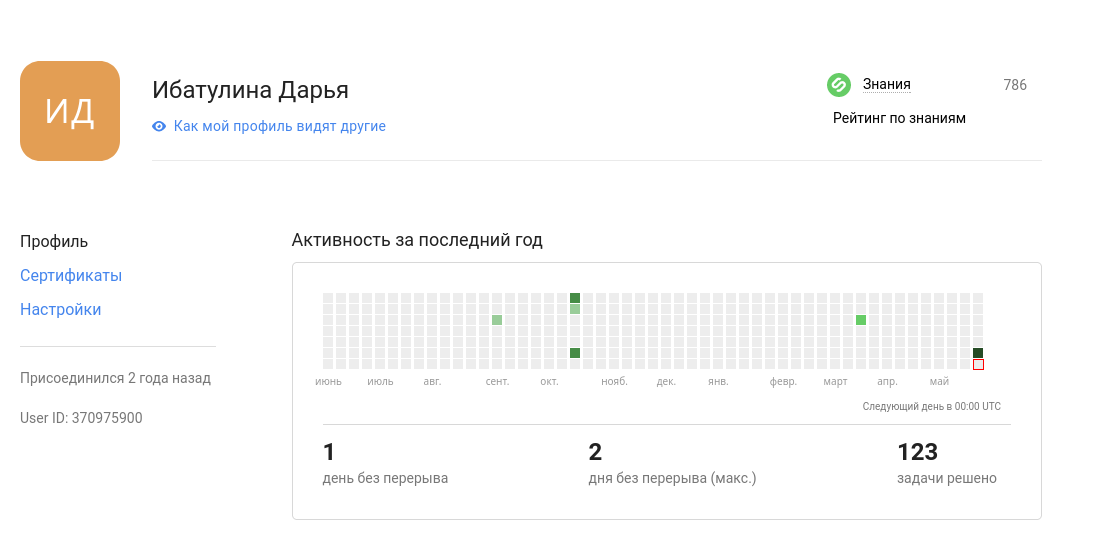


Figure 1: Авторизация на платформе

Затем читаем введение к курсу: основным его правилом является самостоятельное и честное выполнение заданий. Дедлайнов нет, можно выполнять курс в любое удобное время. И переходим к первому вопросу: Как называется курс? Ответ: Введение в Linux (рис. [2](#fig:002)).

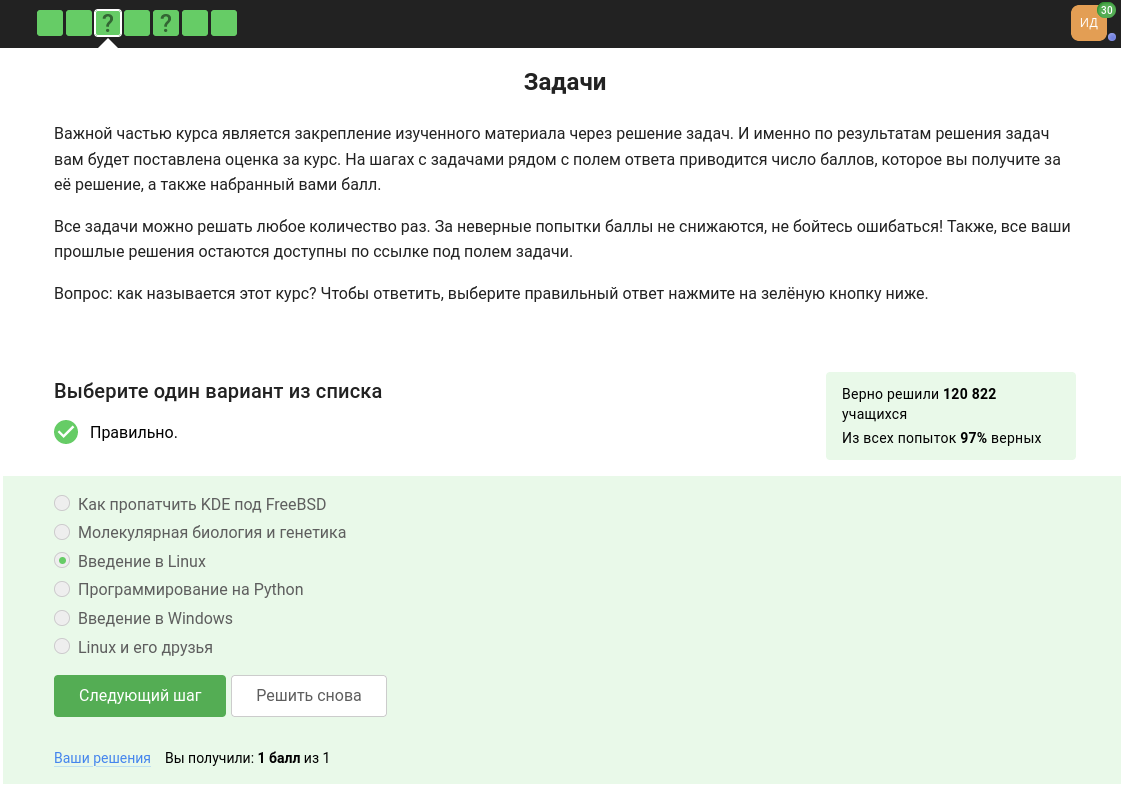


Figure 2: Задание 1.1 (1)

Теперь переходим к следующему заданию блока 1.1. Отмечаем все верные утверждения о курсе. Они касаются правил его прохождения, с которым я ознакомилась ранее (рис. [3](#fig:003)).

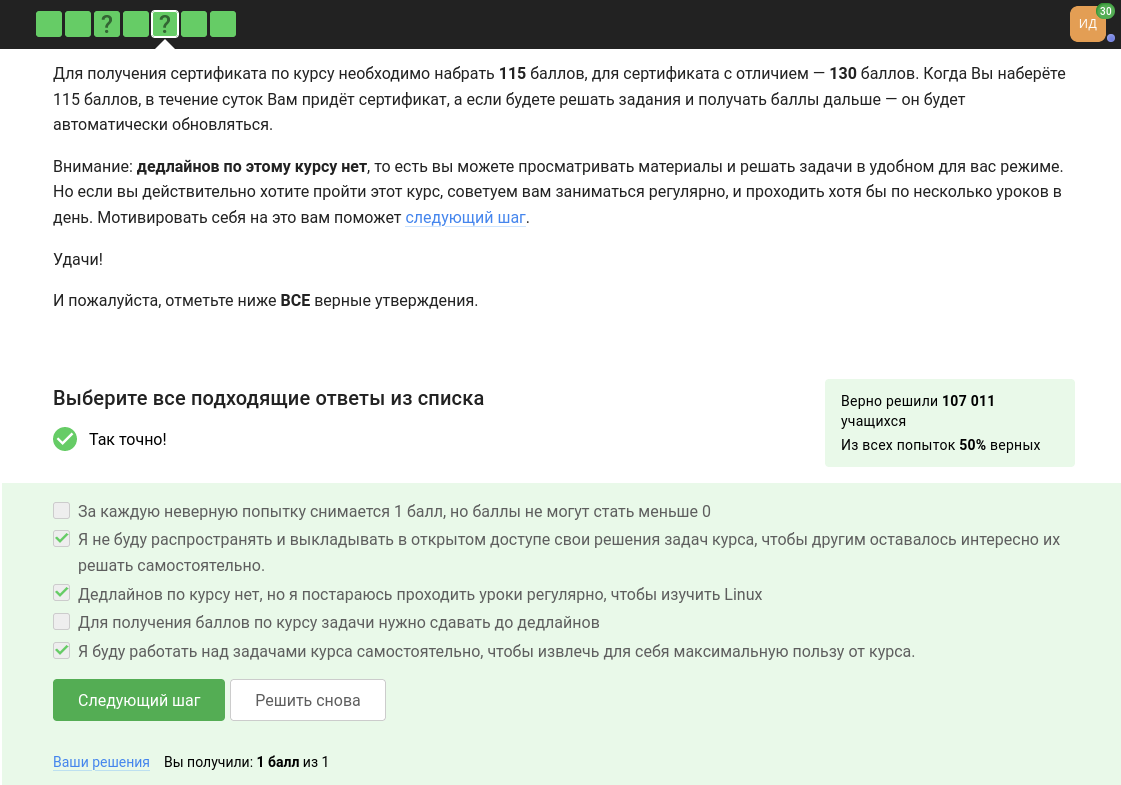


Figure 3: Задание 1.1 (2)

## 4.2 Как установить Linux (1.2)

Первый блок первого этапа выполнен, переходим ко второму. Здесь требуется отметить, какую ОС обычно я использую. Отмечаю, что Windows (служит основной ОС на моём компьютере) и Linux (с ним я научилась работать в ходе прохождения курсов “Операционные системы” и “Архитектура компьютера”) (рис. [4](#fig:004)).

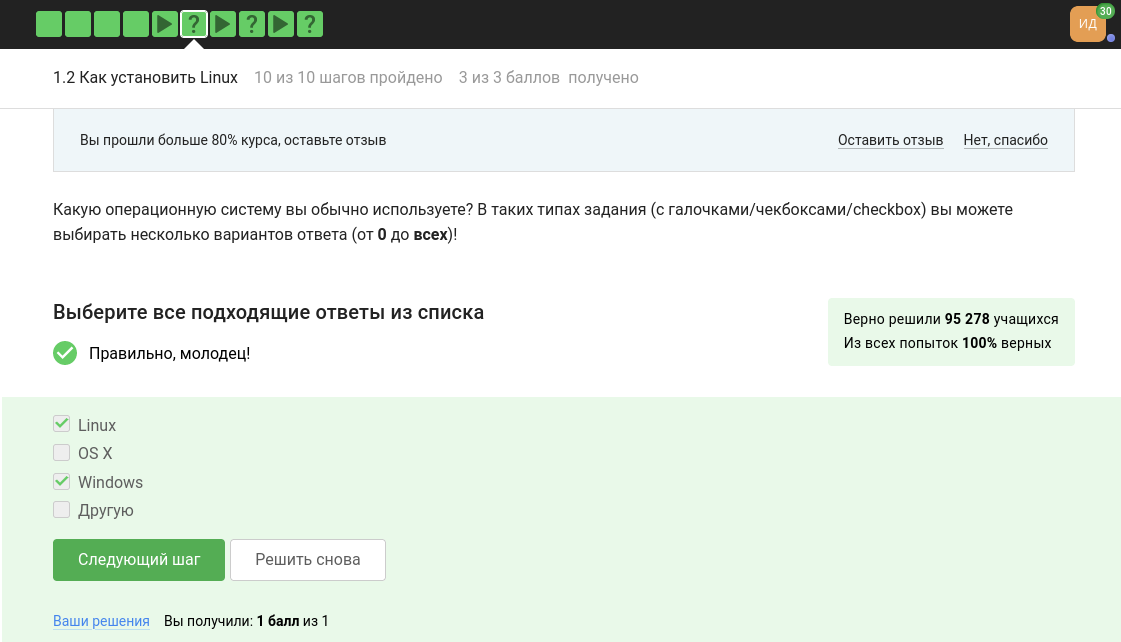


Figure 4: Задание 1.2 (1)

Переходим к заданию 2 этого же блока. В нём нас спрашивают, что такое виртуальная машина. Отвечаю, что это специальная программа, предназначенная для запуска одной ОС на другой (у меня запускается Linux в виртуальной машине на Windows) (рис. [5](#fig:005)).

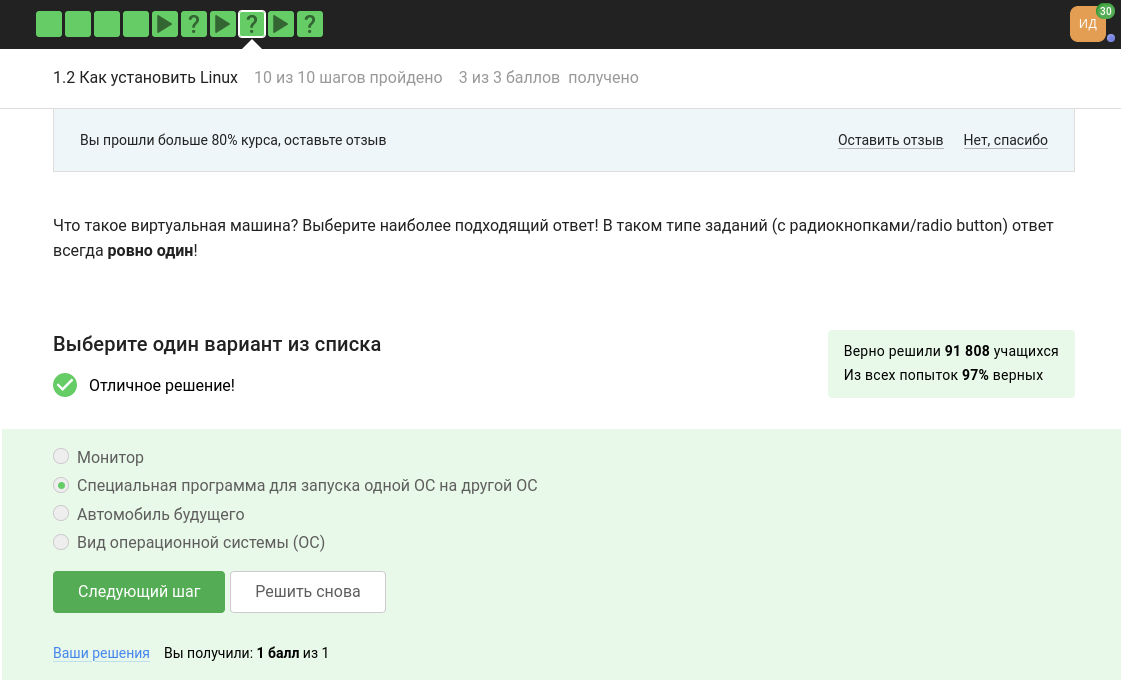


Figure 5: Задание 1.2 (2)

Далее отвечаем, что смогли запустить Линукс на своём компьютере (рис. [6](#fig:006)). Этот отчёт я делаю с Линукса.

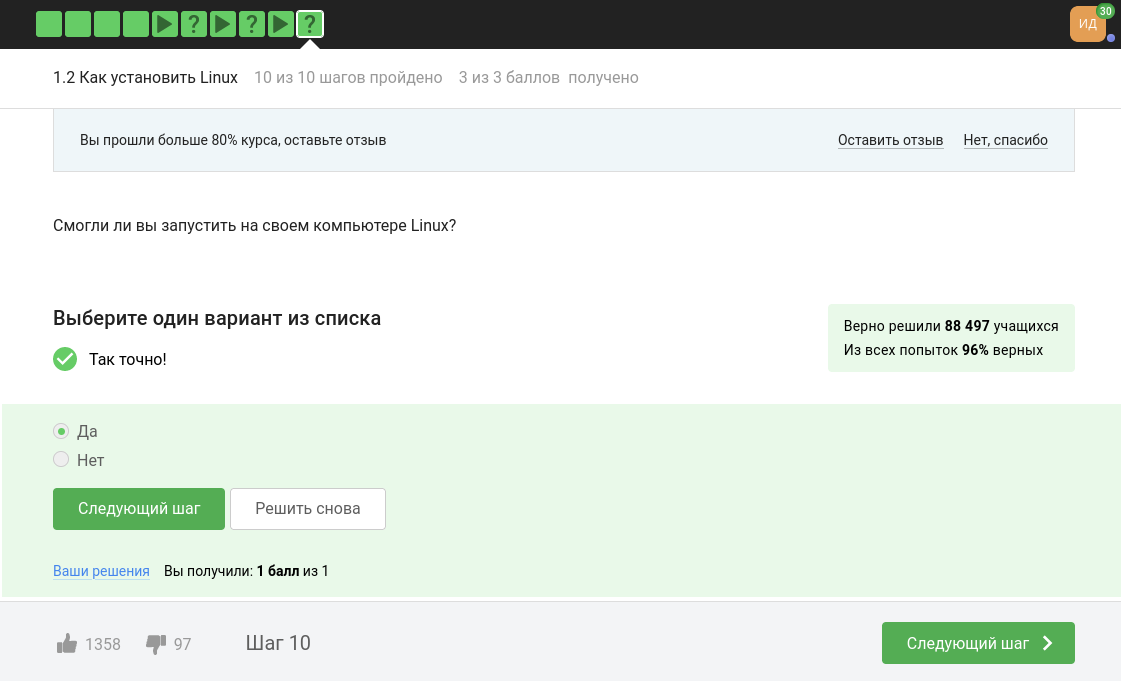


Figure 6: Задание 1.2 (3)

## 4.3 Осваиваем Linux (1.3)

Переходим к следующему разделу. Здесь требуется создать документ в OpenOffice/LibreOffice Writer (аналог Microsoft Word) и написать в нём шрифтом FreeMono одну-единственную строчку:

Hello, Linux!

Я это сделала (рис. [7](#fig:007)), а затем сохранила документ в формате xml и отправила. Ответ засчитали.

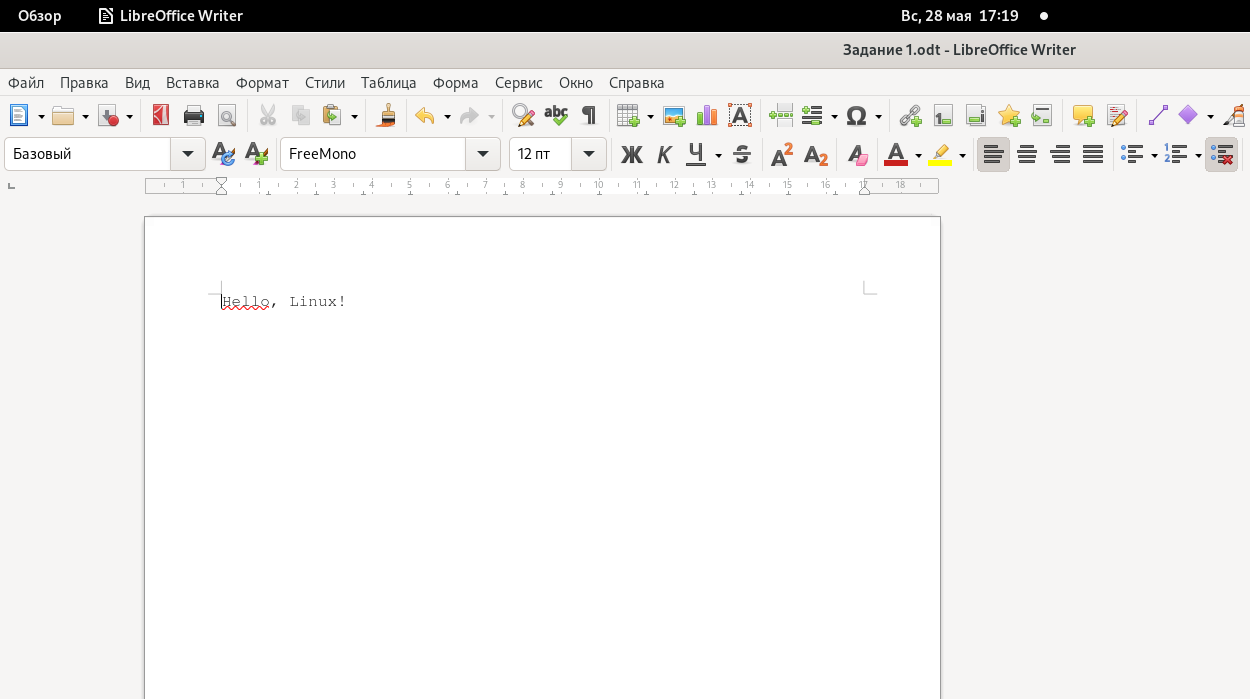


Figure 7: Задание 1.3 (1)

Теперь нужно сказать, какое расширение имеют установочные пакеты в Linux (Ubuntu). Они имеют расширение *deb* (рис. [8](#fig:008)).

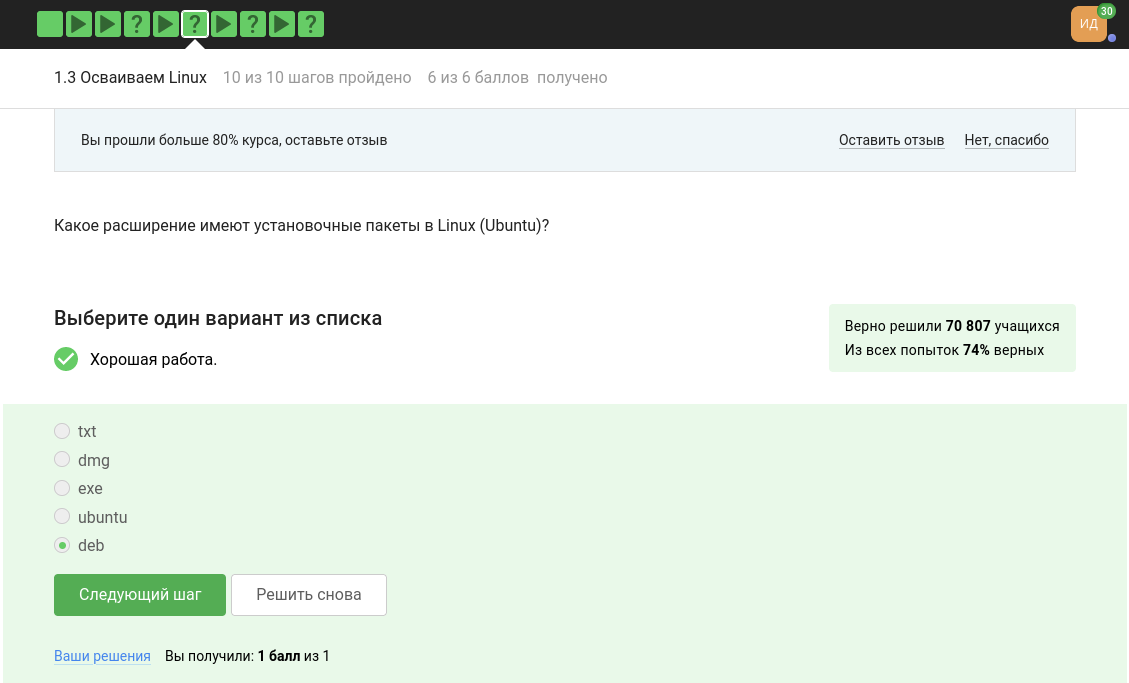


Figure 8: Задание 1.3 (2)

В следующем задании нужно поставитьсебе в систему плеер VLC (любым способом: через Software Center или скачиванием установочного пакета с сайта VLC). А затем запустить, открыть Help → About (или Shift+F1) и написать в поле ответа первую фамилию (без имени!) из вкладки Authors. Это Denis-Courmont (рис. [9](#fig:009)).

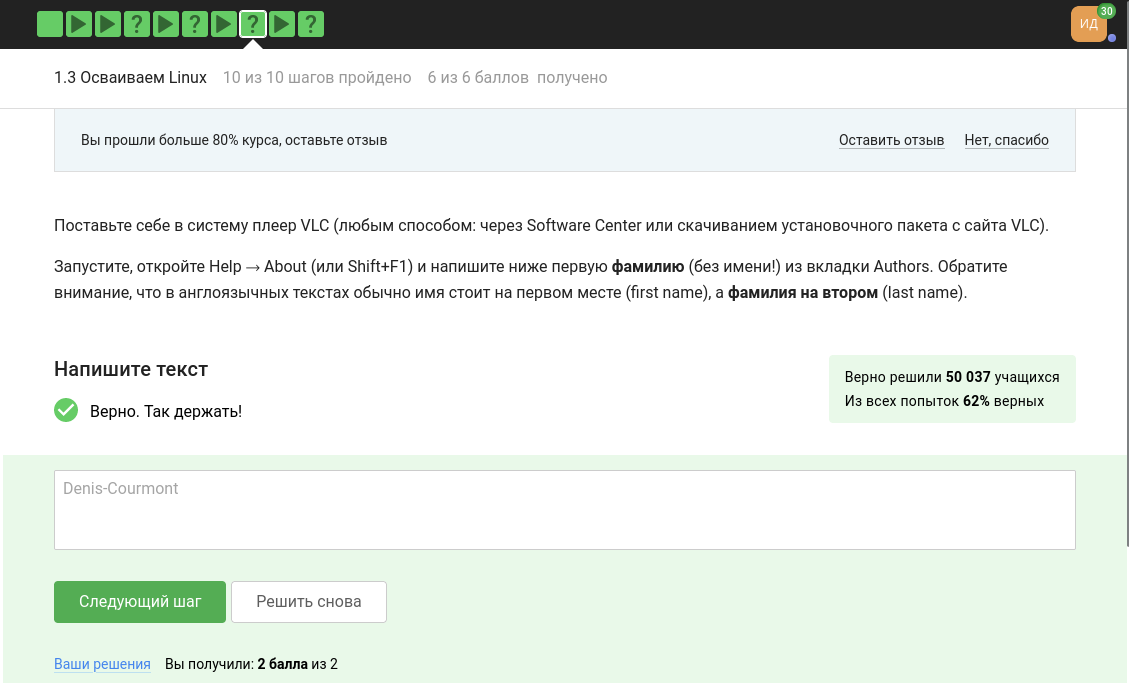


Figure 9: Задание 1.3 (3)

В последнем задании этого блока нужно выбрать, для чего используется Update Manager. Как очевидно из лекции к этому блоку, он используется для обновления ссылок в Software Center, для обновления установленных программ и для обновления всей системы до новой версии (рис. [10](#fig:010)).

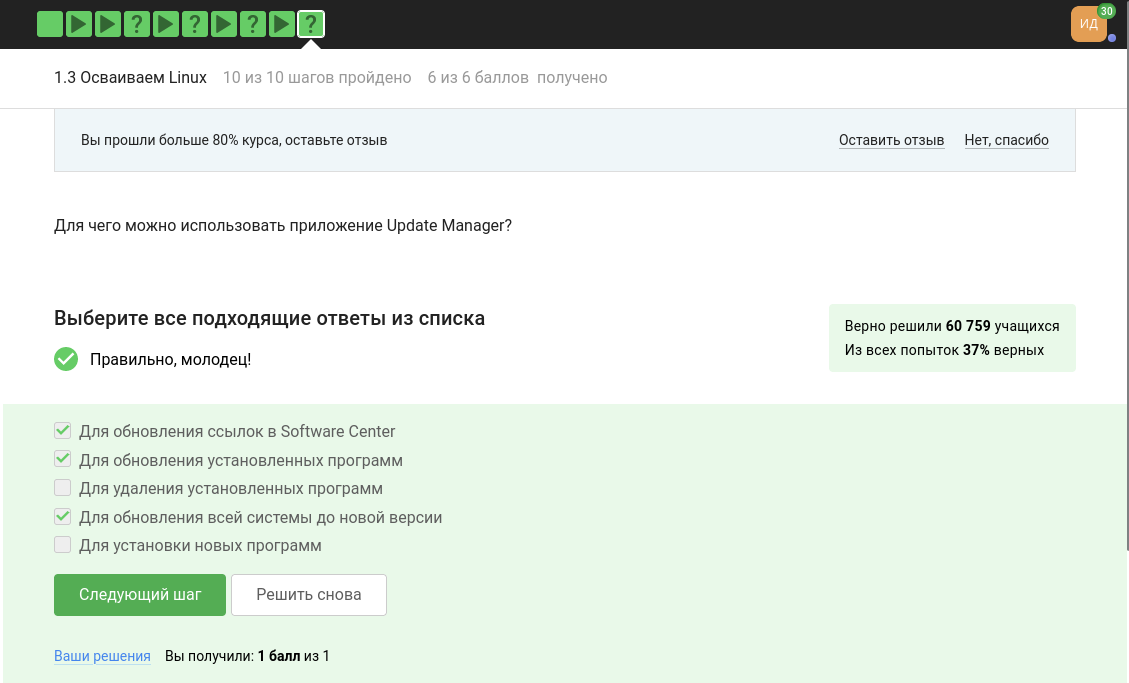


Figure 10: Задание 1.3 (4)

## 4.4 Terminal: основы (1.4)

В первом задании этого блока нужно выбрать все синонимы для “командной строки” (рис. [11](#fig:011)). Командную строку часто именуют консолью или терминалом.

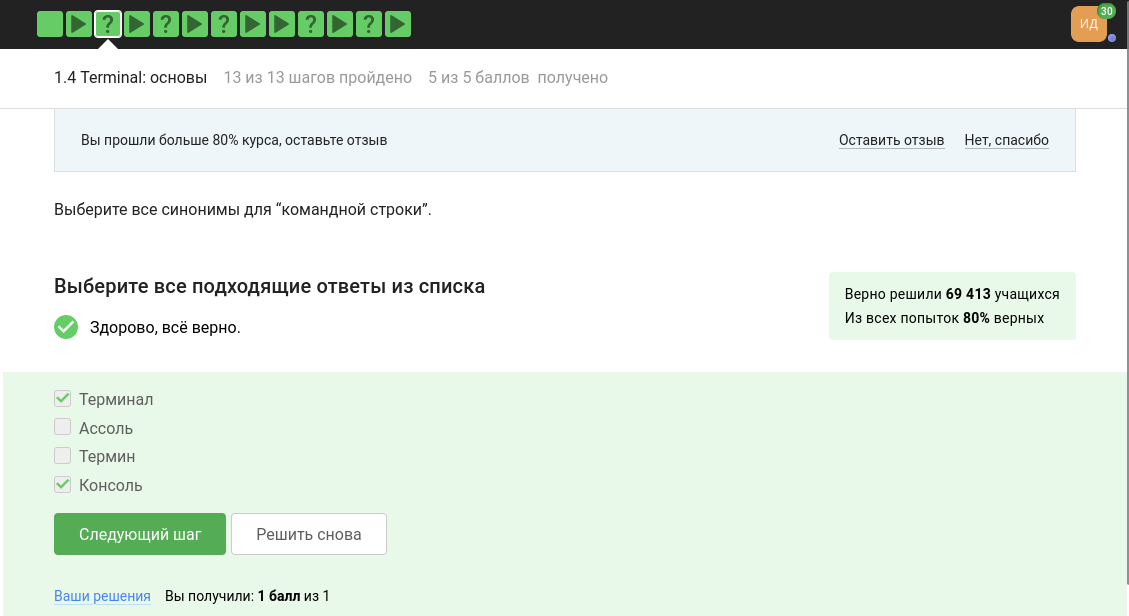


Figure 11: Задание 1.4 (1)

Далее, мы уже начинаем изучать базовые команды терминала. Так, в следующем вопросе речь идёт о команде *pwd - print working directory*. Она напечатает в какой директории мы сейчас находимся. Печатаем её маленькими буквами, остальные варианты будут неверны (рис. [12](#fig:012)).

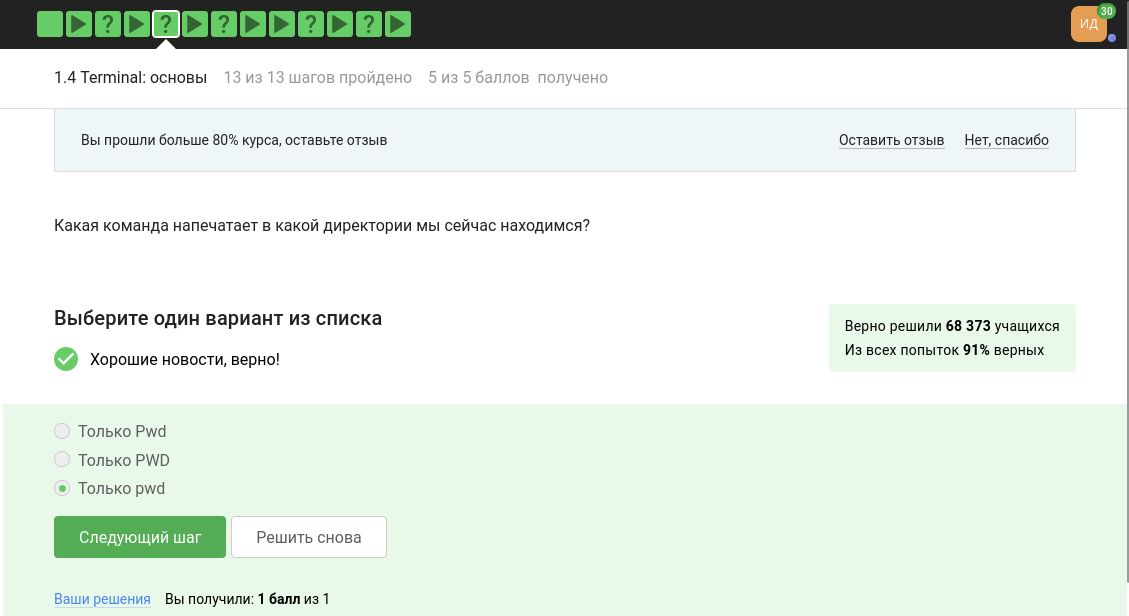


Figure 12: Задание 1.4 (2)

Затем нам нужно указать, какие команды полностью эквивалентны команде: *ls -A –human-readable -l /some/directory*. Из данного списка это команды *ls –almost-all –human-readable -l /some/directory* и *ls –human-readable -A -l /some/directory* (рис. [13](#fig:013)).

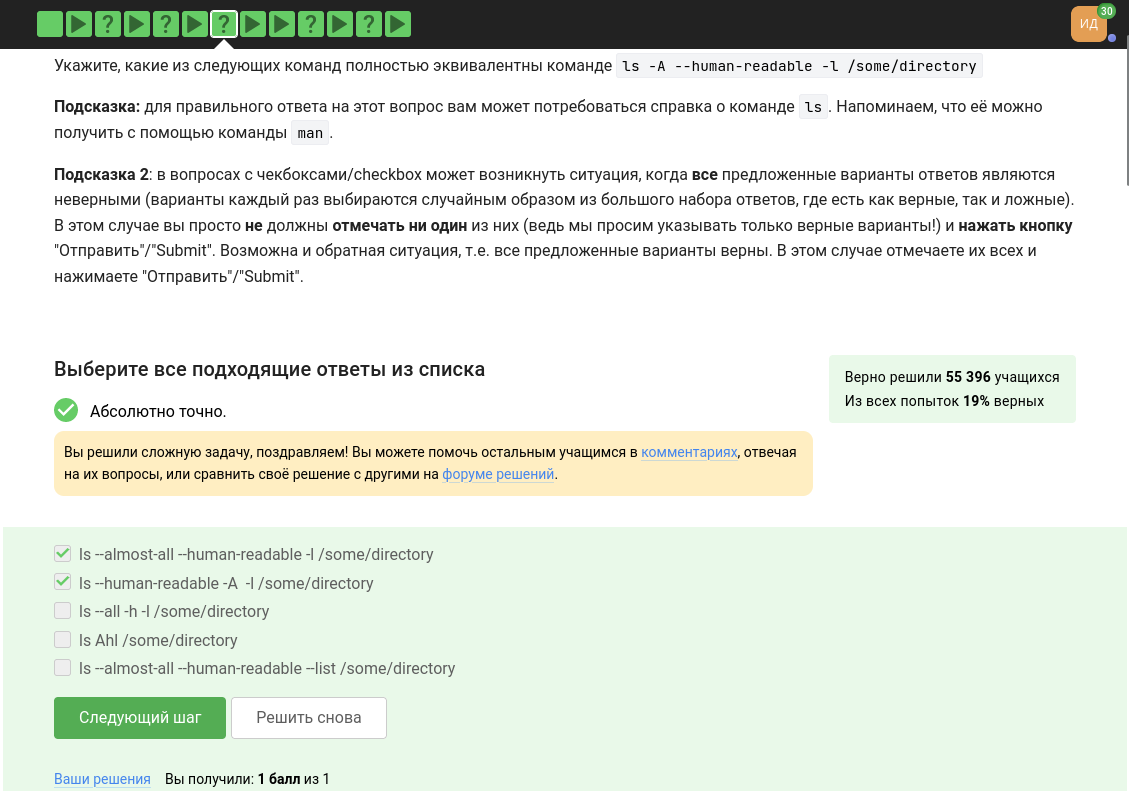


Figure 13: Задание 1.4 (3)

Выполнить следующее задание можно, проверив все ниже приведённые команды у себя в терминале. Получаем, что в ответ записываем единственную из этого списка команду: *s /home/bi/Downloads* (рис. [14](#fig:014)).

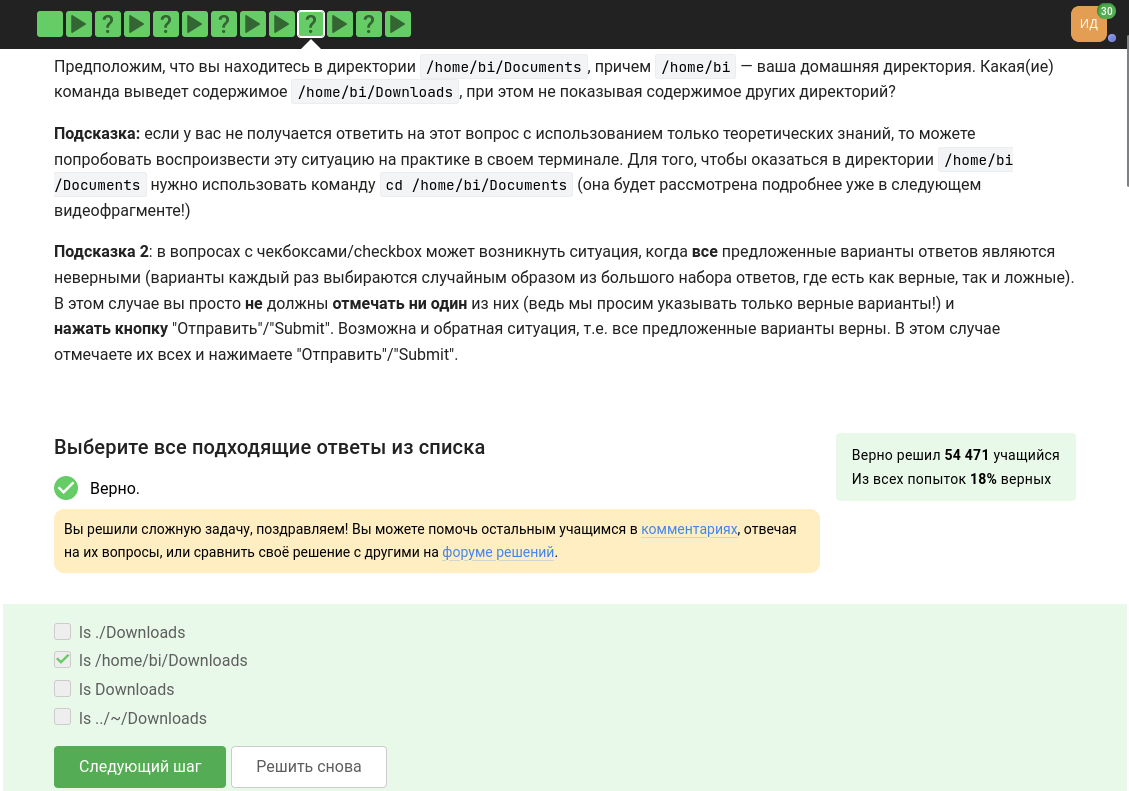


Figure 14: Задание 1.4 (4)

Ещё из курса Операционных систем мы знаем, что для удаления директорий используется команда *rm -r* (рис. [15](#fig:015)).

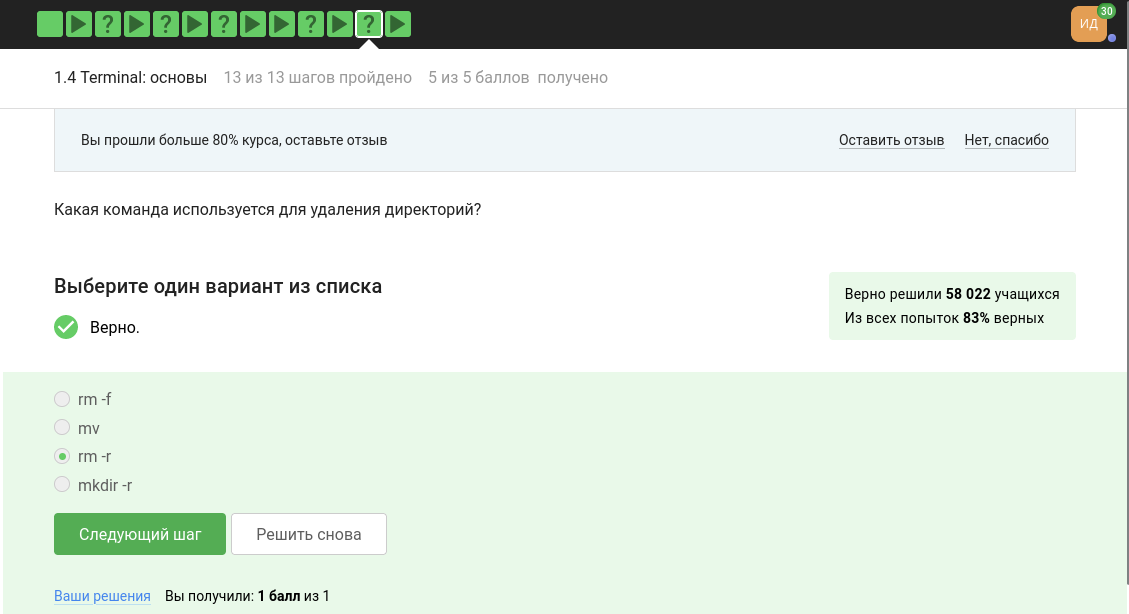


Figure 15: Задание 1.4 (5)

## 4.5 Запуск исполняемых файлов (1.5)

Чтобы запустить какую-либо программу через терминал, достаточно ввести её название. Вот и на это вопрос мы отвечаем, что, если ввести в терминал команду firefox (для запуска одноименного браузера), а затем ввести туда же команду exit, то никто не закроется (рис. [16](#fig:016)). Команда exit в Linux используется для выхода из командной оболочки, в которой она запущена в данный момент, а так как мы запустили процесс в терминале, то никто не закроется.

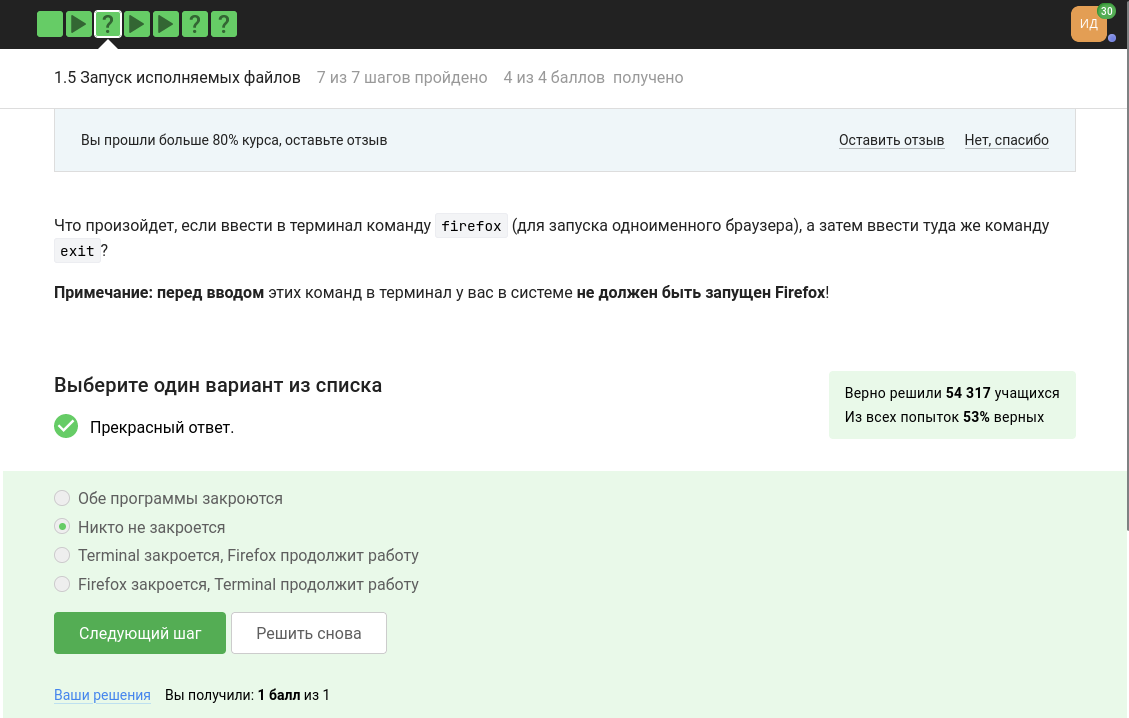


Figure 16: Задание 1.5 (1)

Запуск программы с & (фоновый режим) эквивалентен запуску Ctrl+Z (приостановка программы), bg (её переход в фоновый режим) (рис. [17](#fig:017)).

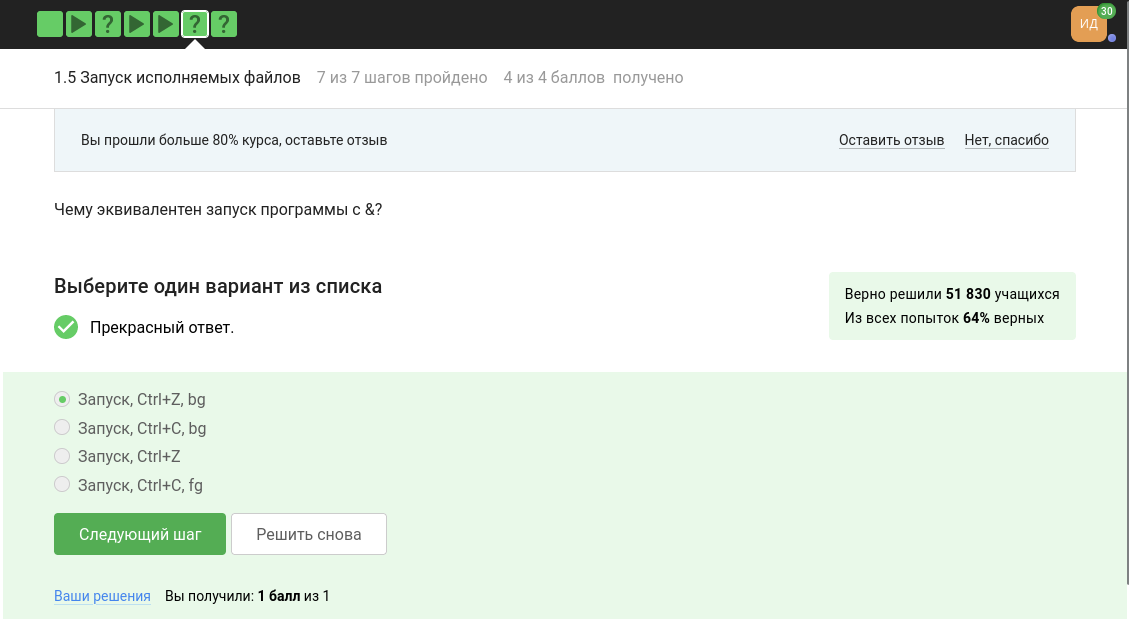


Figure 17: Задание 1.5 (2)

Последнее задание блока звучит так: Скачайте файл с программой, сделайте его исполняемым (chmod +x), запустите (./*имя\_программы*) и скопируйте то, что он выведет на экран, в форму ниже. (рис. [18](#fig:018)).

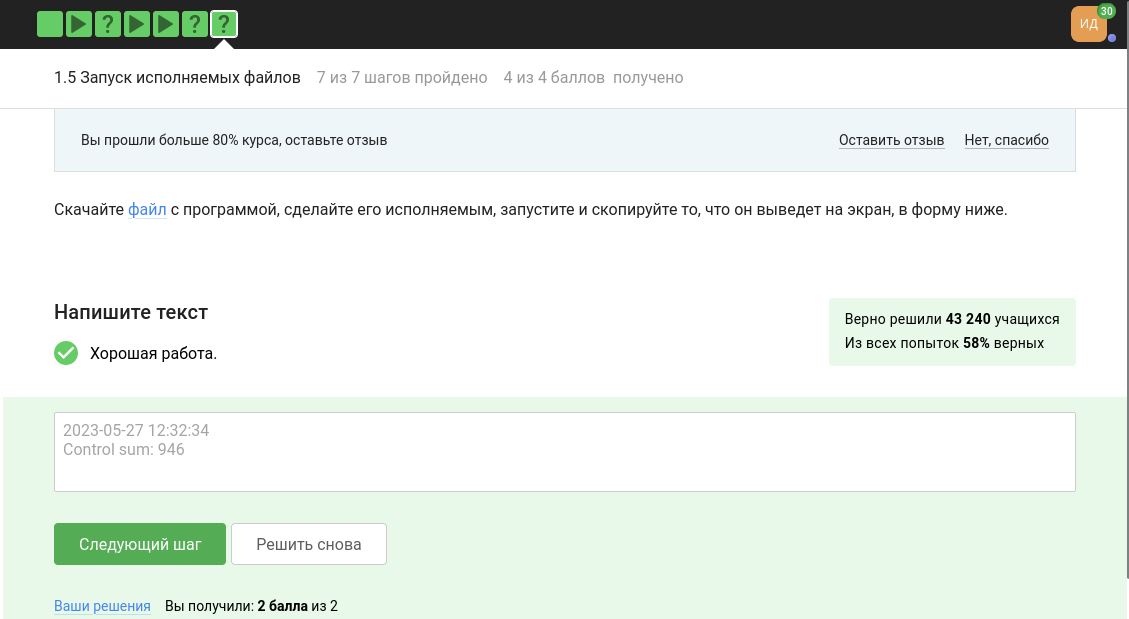


Figure 18: Задание 1.5 (3)

## 4.6 Ввод/вывод (1.6)

По умолчанию поток ошибок выводится на экран, как мы узнаем из просмотренной лекции (рис. [19](#fig:019)).

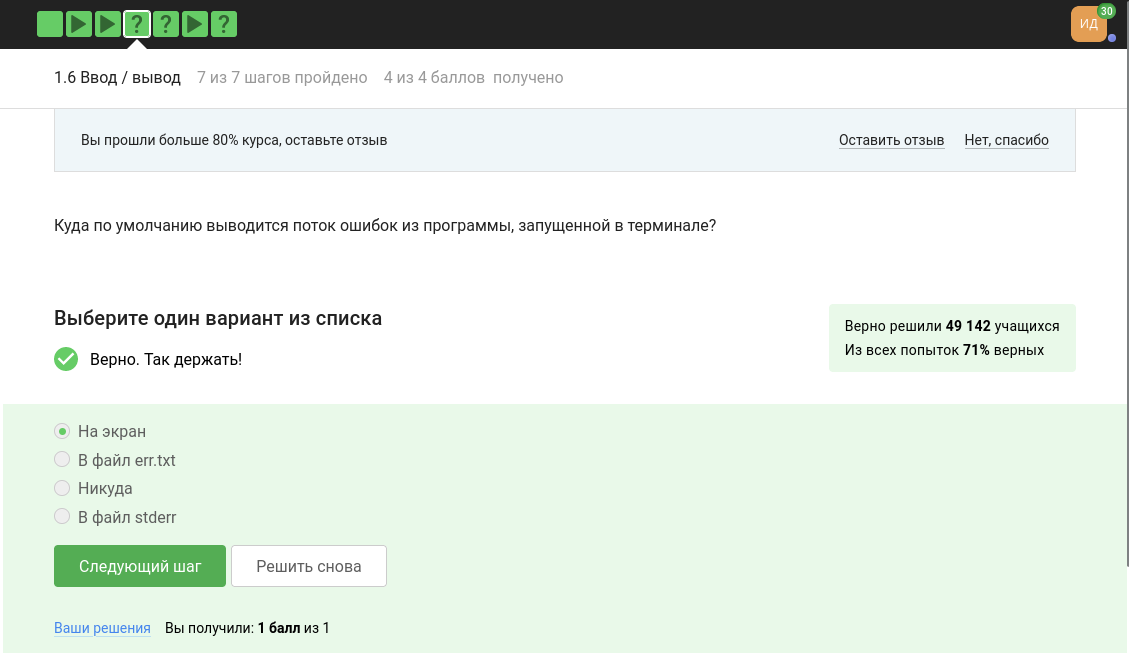


Figure 19: Задание 1.6 (1)

Чтобы создать файл file.txt и записать в него поток ошибок программы program, нужно выполнить команду *program 2>> file.txt* - дозапись в файл, если он существует или *program 2> file.txt* - создание файла, если его не существует (рис. [20](#fig:020)).

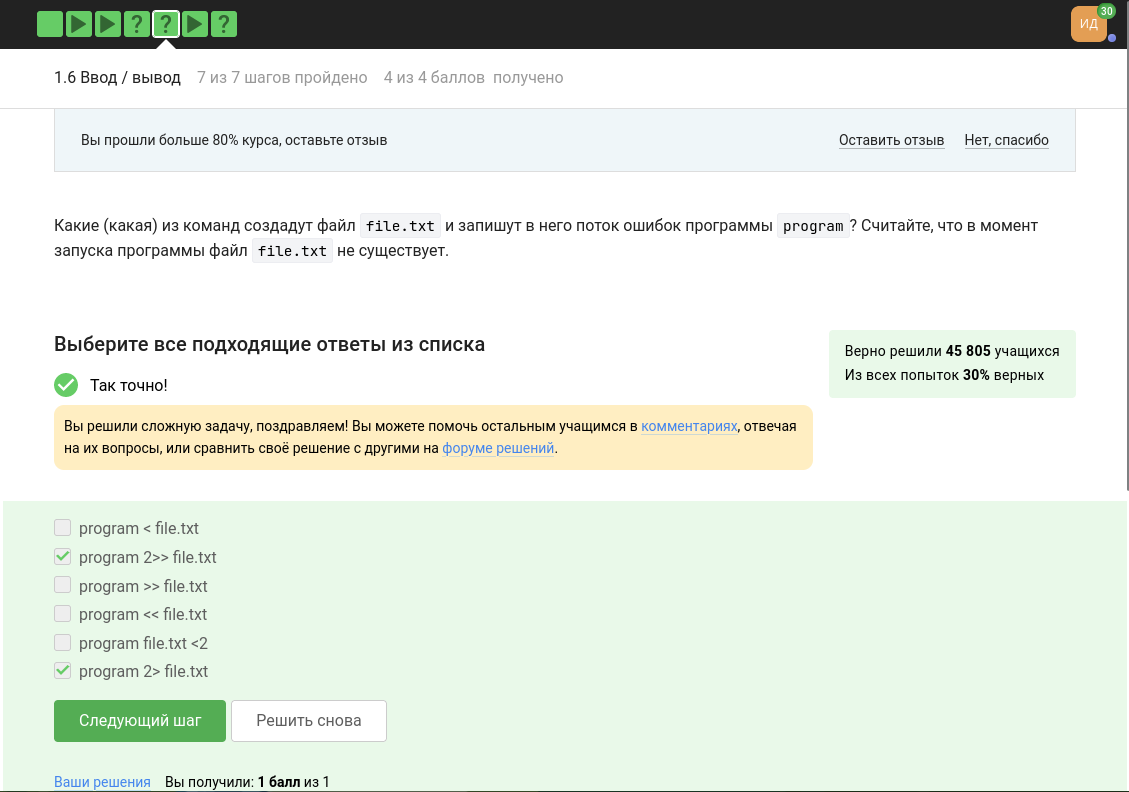


Figure 20: Задание 1.6 (2)

Последнее задание блока звучит так: Куда деваются сообщения об ошибках (т.е. вывод в stderr) от тех программ, которые объединены в конвейер (pipe)? Для этого скачаем показанную на занятии программу interacter.py, создадим файл names.txt со списком имен (записав часть из них с ошибкой, т.е. с маленькой буквы) и построим конвейер, аналогичный тому, что был показан на видео. Запустим конвейер и посмотрим на результат! (рис. [21](#fig:021)). Мы видим, что сообщения об ошибках выводятся на экран.

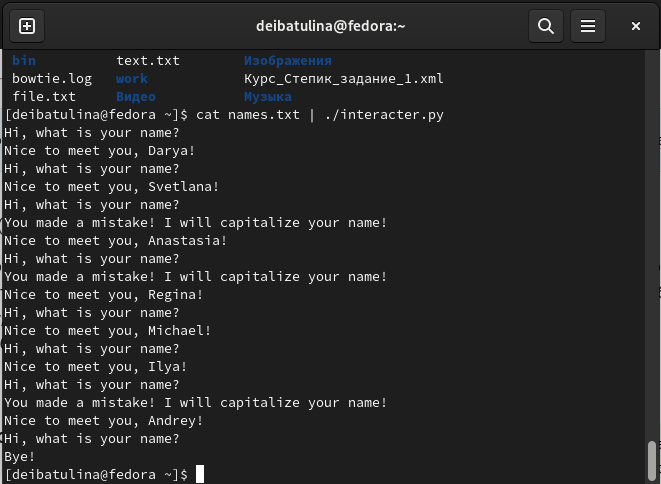


Figure 21: Задание 1.6 (3)

## 4.7 Скачивание файлов из Интернета (1.7)

cd /home/alex/  
wget -P /home/alex/Pictures -O 1.jpg http://example.com/example.jpg

При выполнении следующих команд картинка окажется в директории */home/alex/1.jpg* (рис. [22](#fig:022)).

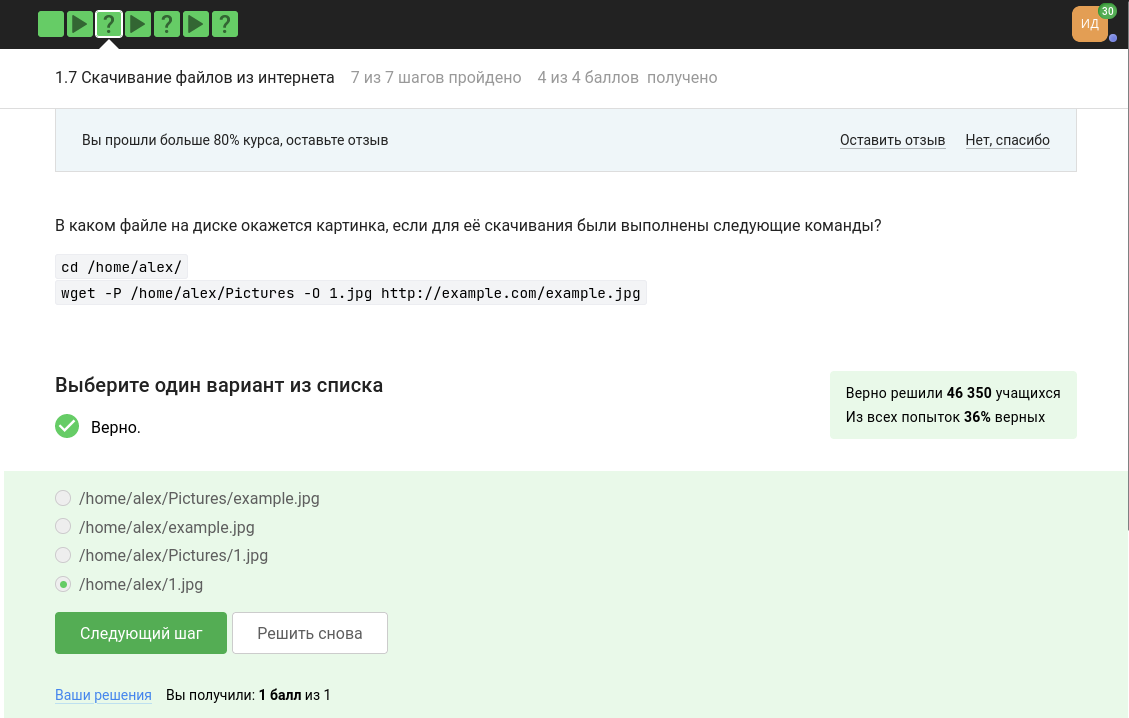


Figure 22: Задание 1.7 (1)

Изучив man по команде wget, мы понимаем, что, чтобы она не выводила никаких сообщений на экран (Resolving.., Connecting to.. и т.д.) необходимо передать опцию *-q* или *–quiet* (рис. [23](#fig:023)).

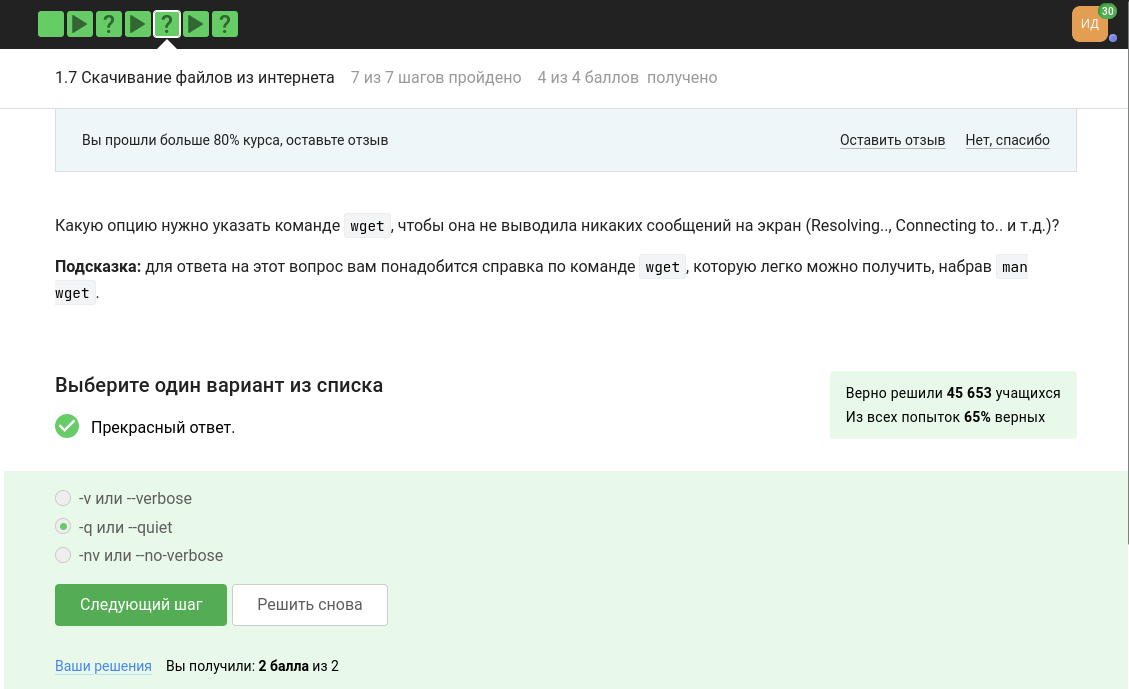


Figure 23: Задание 1.7 (2)

Если запустить wget -r -l 1 -A jpg и передать в качестве аргумента ссылку на эту web-страницу, то будут скачаны jpg и html файлы, но все html будут удалены (рис. [24](#fig:024)). Данная команда означает, что мы скачаем все файлы по ссылкам первого уровня, и только формата jpg.

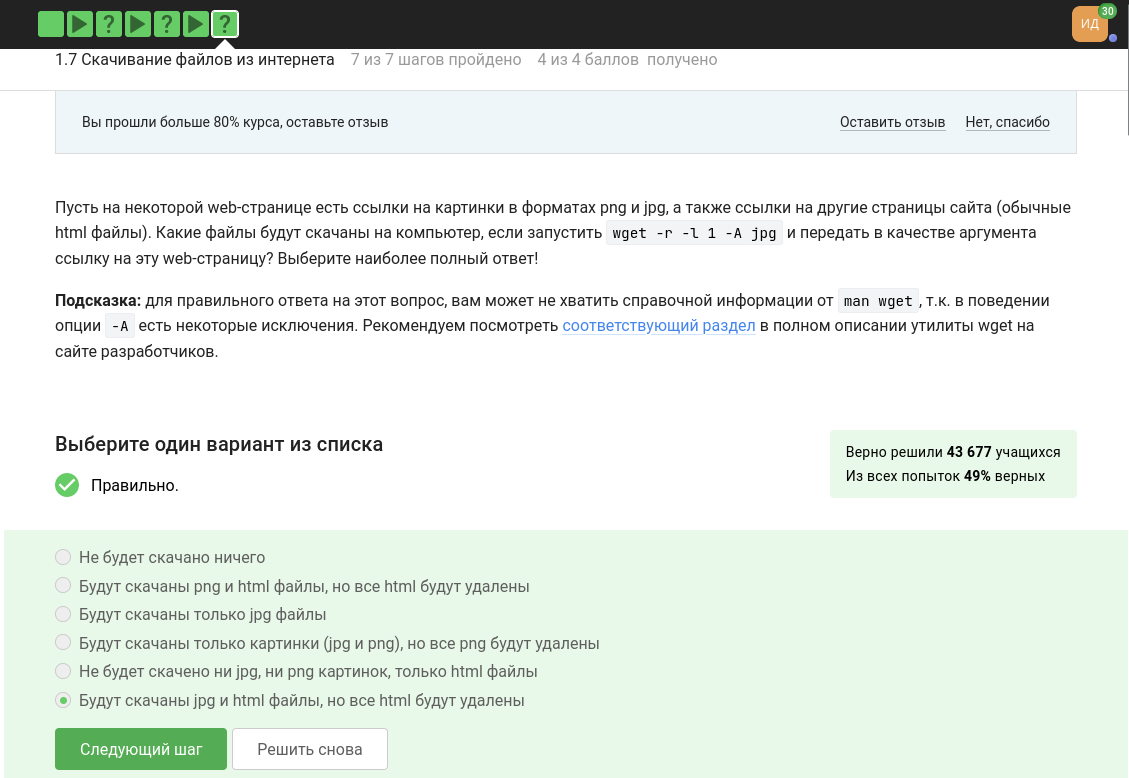


Figure 24: Задание 1.7 (3)

## 4.8 Работа с архивами (1.8)

Архиваторы gzip и zip отличаются тем, что gzip удаляет архив после его распаковки (рис. [25](#fig:025)).

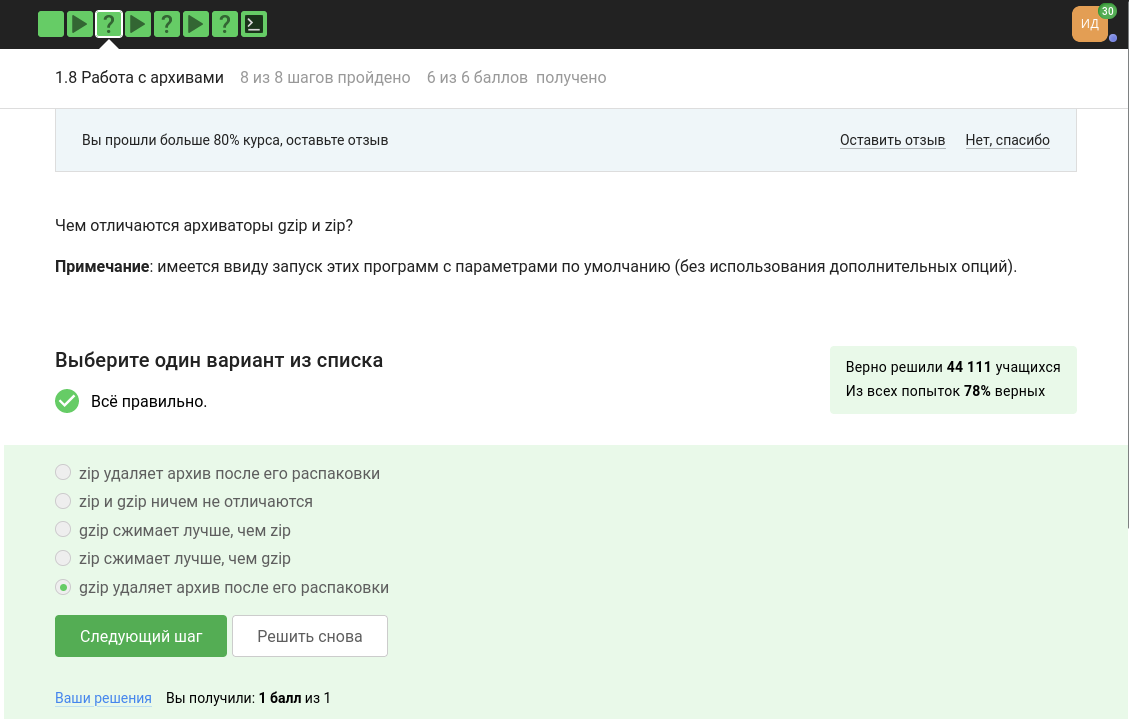


Figure 25: Задание 1.8 (1)

Только архиваторы zip и tar могут создать архив из директории с файлами (рис. [26](#fig:026)).

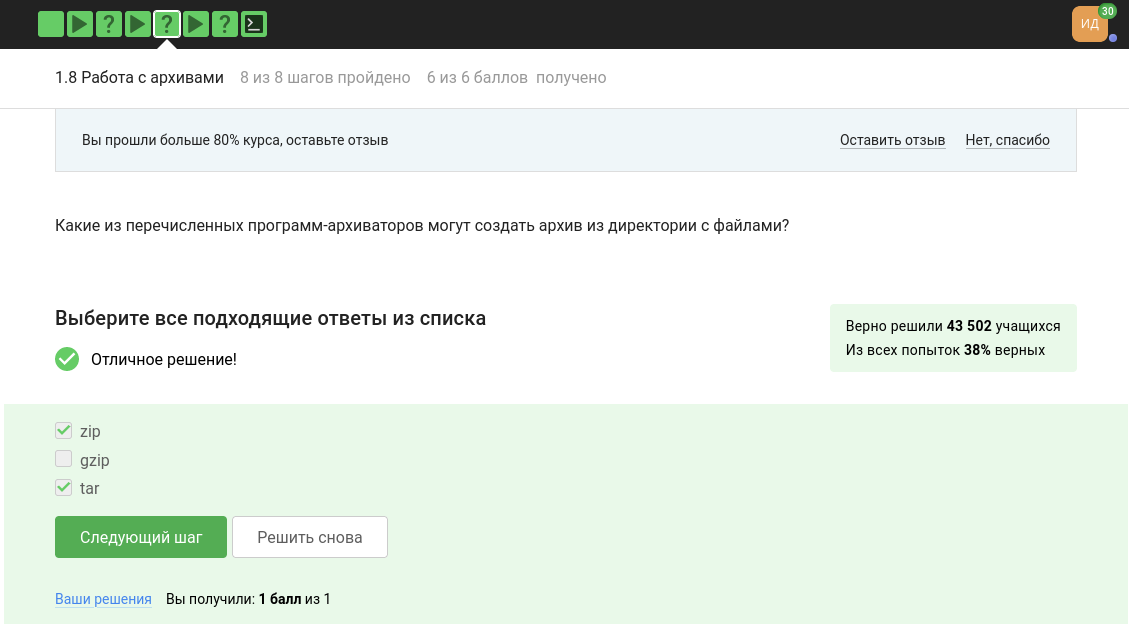


Figure 26: Задание 1.8 (2)

Программе tar, чтобы запаковать файлы в *my\_archive.tar.bz2* необходимо передать опции -cjf (f обозначает работу с файловой системой, с -создать (create), j - архив) (рис. [27](#fig:027)).

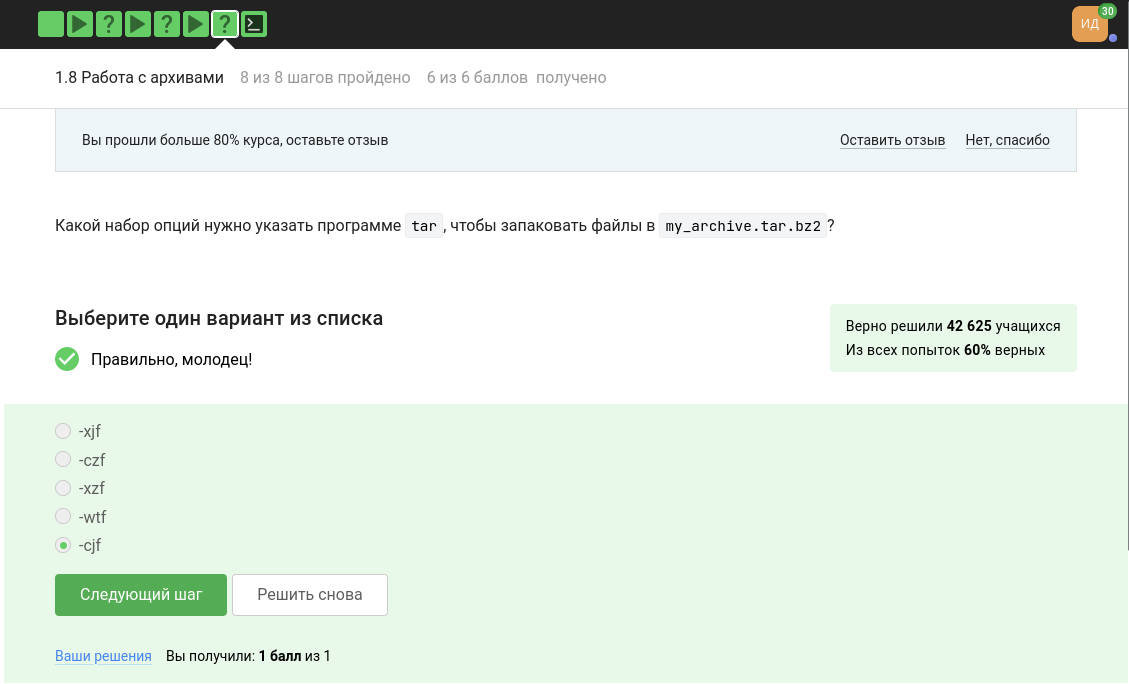


Figure 27: Задание 1.8 (3)

## 4.9 Работа с архивами (1.9)

Маска команды find (*.jpg - другое расширение,* .?\* - ? означает ровно 1 символ, *alexey.* - команда *find* чувствительна к регистру) НЕ найдет файл Alexey.jpeg (рис. [28](#fig:028)).

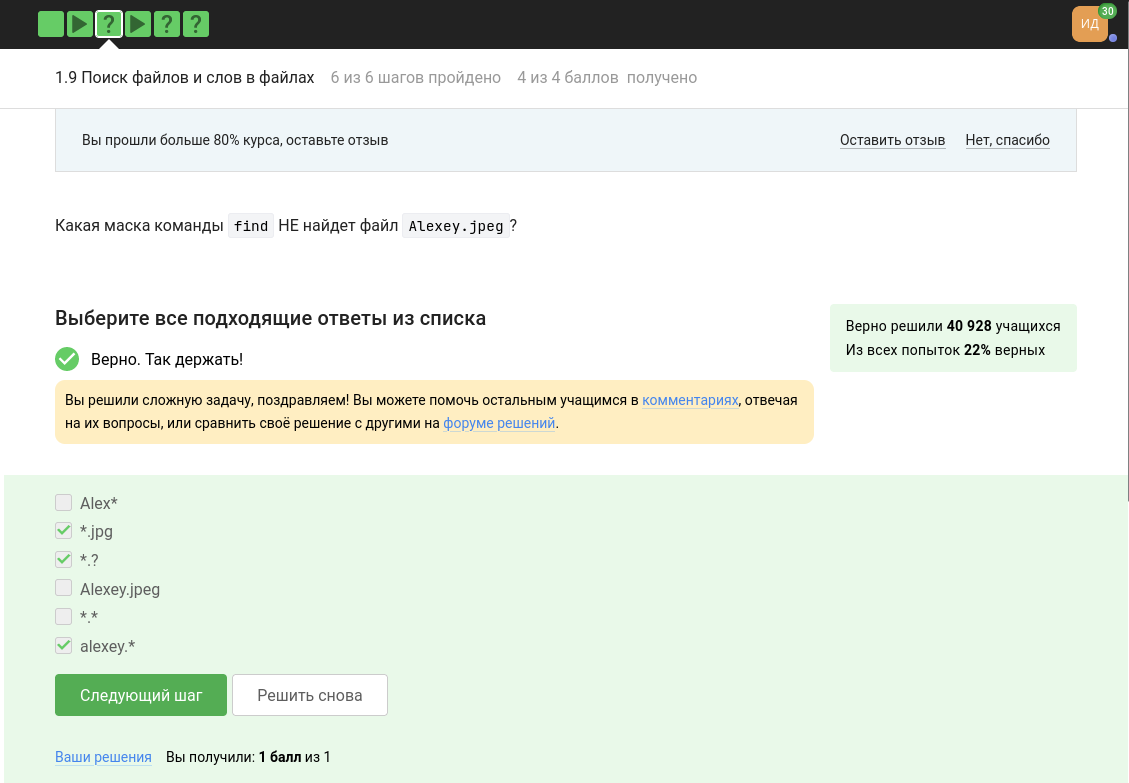


Figure 28: Задание 1.9 (1)

Команда *find* в данном случае вывдет только те строки, где world написан с маленькой буквы (рис. [29](#fig:029)).

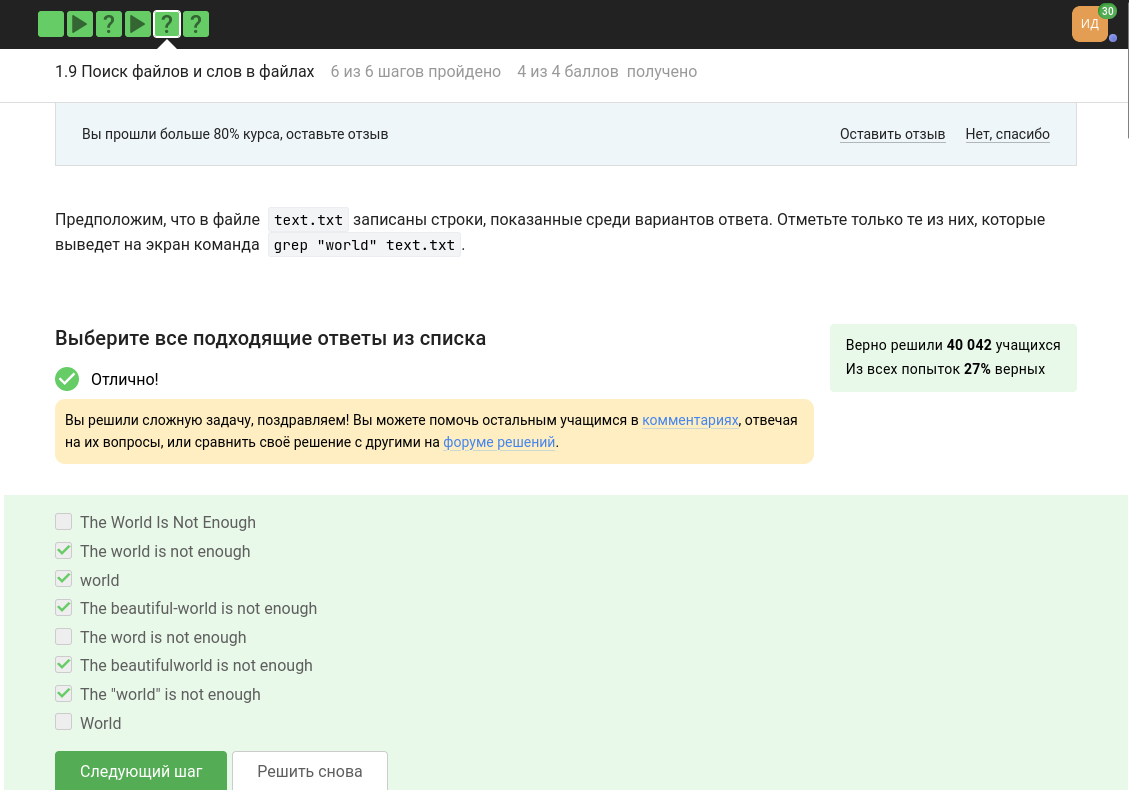


Figure 29: Задание 1.9 (2)

Для генерации файла со строчками произведений Шекспира, содержащих слово *love*, ввожу в терминал следующий код:

tar -xzvf shakespeare.tar.gz  
-F, --fixed-strings PATTERNS are strings  
-r, --recursive like --directories=recurse  
grep -Fr "love" Shakespeare/\* > ~/Документы/stepik/fileSH.txt.txt

И отправляю сгенерированный файл в форму (рис. [30](#fig:030)).

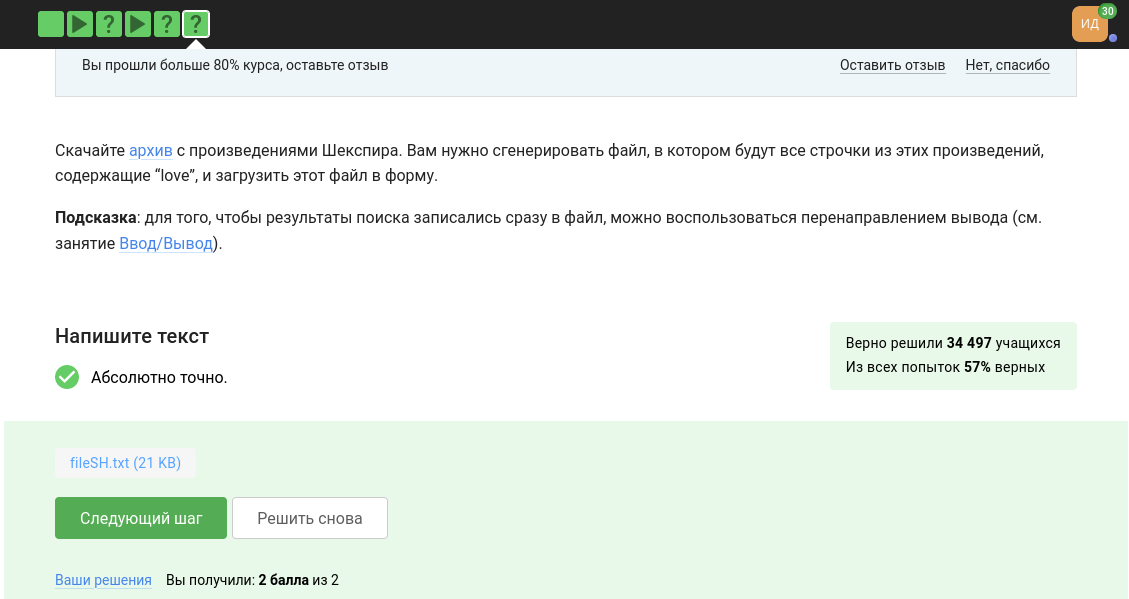


Figure 30: Задание 1.9 (3)

# 5 Выводы

В ходе выполнения первого этапа курса я научилась минимальным навыкам работы с ОС Linux (работа с консолью, запуск исполняемых файлов, архивация и распаковка архивов, перенаправление ввода и вывода, поиск файлов и скачивание файлов из Интернета).

# Список литературы