Отчёт по прохождению внешнего курса на Stepik (этап 2)

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. Данный отчёт содержит прохождение второго этапа курса.

# 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 лабораторной работы

## 4.1 Знакомство с сервером (2.1)

Приступаю к выполнению первого задания данного блока. Требуется отметить, для каких задач можно использовать удаленный сервер. Просмотрев видеолекцию, отвечаю, что он используется для выполнения сложных (затратных по памяти и времени) вычислений, хранения общедоступных данных (например, доступных для всех пользователей интернета), хранения конфиденциальных данных (т.е. доступ к ним должны иметь только ограниченный круг лиц) и хранения больших объемов данных, т.е. все варианты ответов верны (рис. [1](#fig:001)).

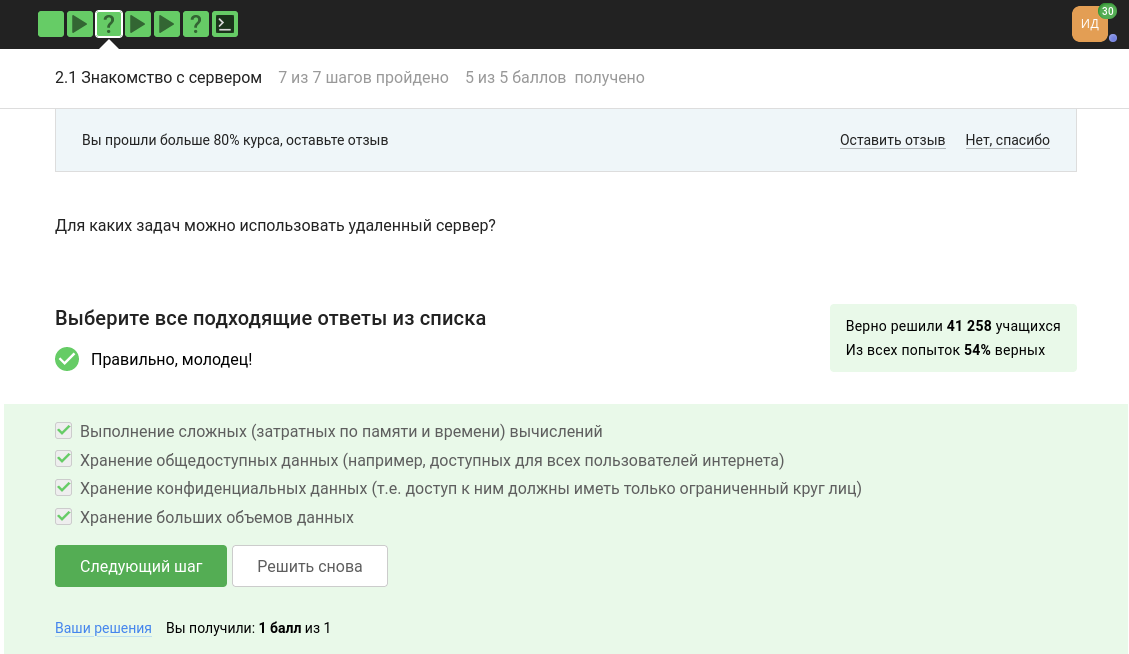


Figure 1: Задание 2.1 (1)

Далее, требуется выбрать, какой из этих ключей можно без опаски пересылать по интернету: id\_rsa и id\_rsa.pub. Первый ключ - приватный, его нельзя никому передавать и показывать, а второй - публичный, то есть его мы и выберем в качестве правильного варианта ответа (рис. [2](#fig:002)).

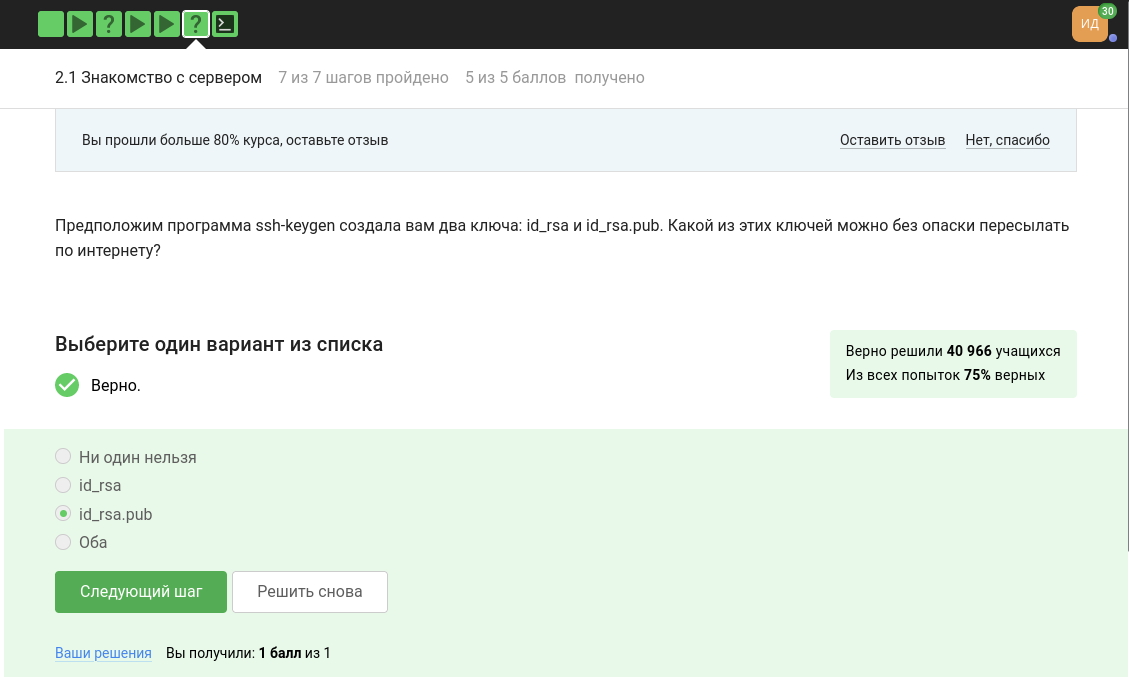


Figure 2: Задание 2.1 (2)

Теперь необходимо поработать с сервером. задание звучит так (рис. [3](#fig:003)):

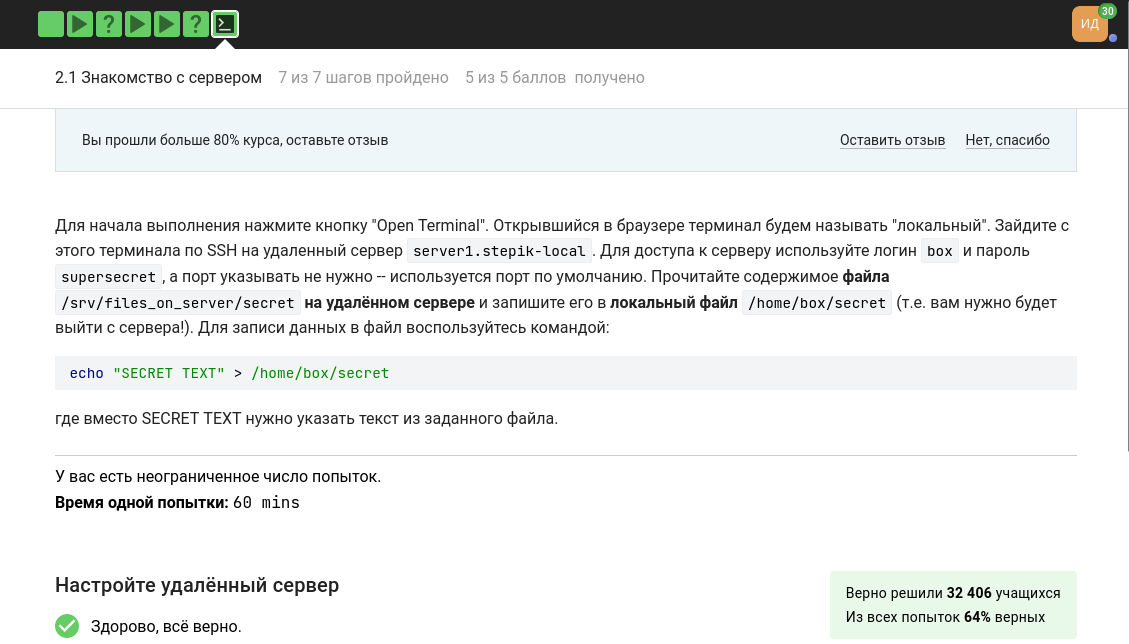


Figure 3: Задание 2.1 (3)

Вводим в онлайн-терминал следующие команды по очереди:

ssh server1.stepik-local # заходим на сервер, используем логин box, а пароль supersecret, порт не указываем  
cat /srv/files\_on\_server/secret # выводим содержимое файла на удалённом сервере  
# I hacked ssh! - искомая фраза, которую выдал терминал. Её нужно запоминть или записать куда-то.  
exit # выходим с сервера  
echo "I hacked ssh" > /home/box/secret # записываем искомую фразу в \*\*локальный\*\* файл

## 4.2 Обмен файлами (2.2)

Первое задание звучит так: какая команда скопирует на сервер (в домашнюю директорию) папку stepic вместе с содержимым ее самой и всех ее подпапок? Ответ: это выполнит команда scp -r stepic username@server:~/, где опция -r указывает на то, что мы копируем именно директорию вместе со всем её содержимым, stepic - имя директории, scp - команда копирования с сервера, username@server - логин и пароль для доступа к серверу, :~/ - домашняя директория (двоеточие отвечает за то, что мы указываем после него, куда мы копируем) (рис. [4](#fig:004)).

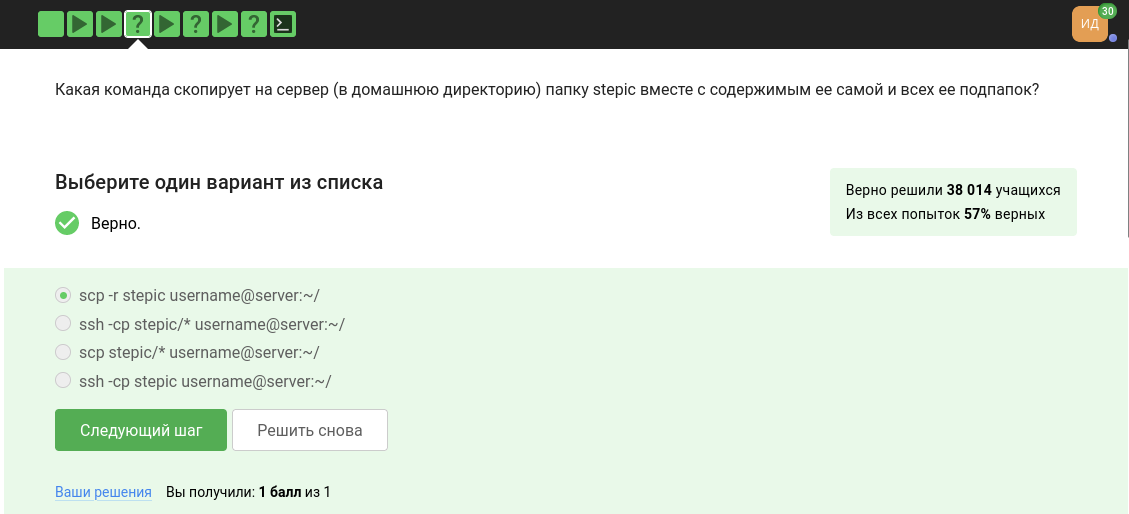


Figure 4: Задание 2.2 (1)

Далее, нам нужно указать, какие действия могут устранить проблему, если при установке программы с помощью команды sudo apt-get install program терминал сообщает, что он не может найти и загрузить установочный пакет. Из предложенных вариантов ответа нам подойдут 1 и 3:

1. sudo apt-get update;
2. sudo apt-get upgrade;
3. Проверка интернет соединения и его установка, если соединения нет;
4. sudo apt-get install –only-upgrade program.

В видео сказано, что первая команда обновит ссылки на пакеты, и тогда терминал их найдёт и обновит программу. Также, самое простое решение - проверить, есть ли проблемы с Интернетом (рис. [5](#fig:005)).

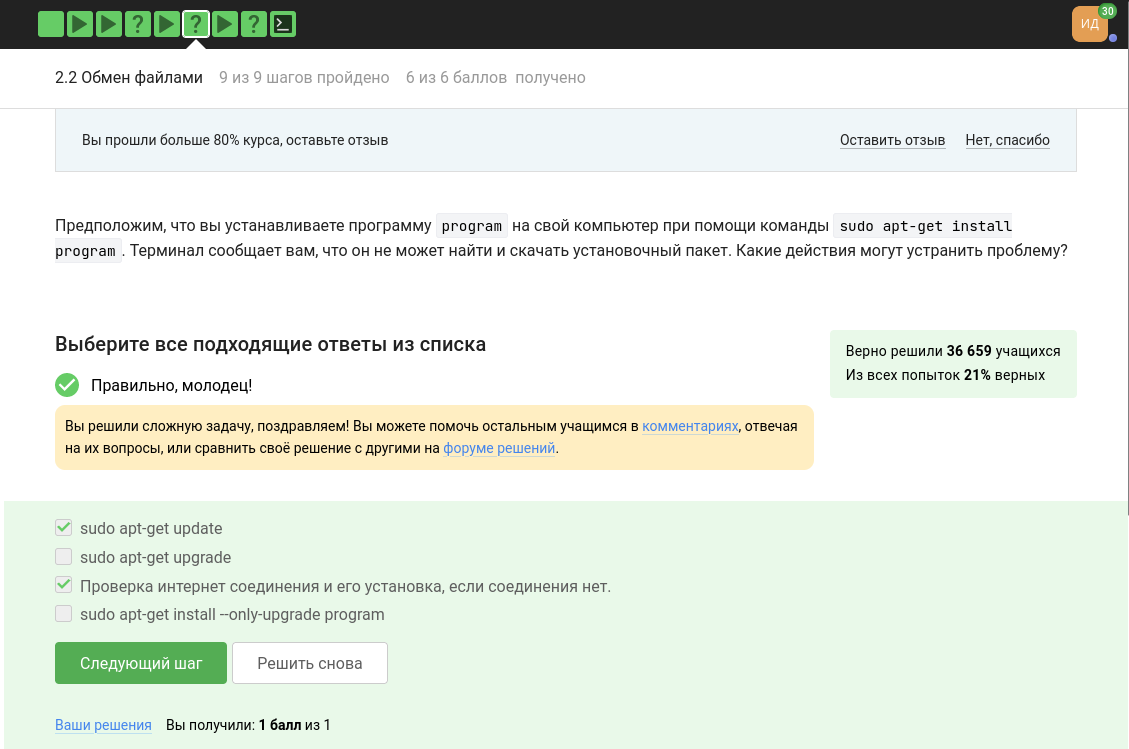


Figure 5: Задание 2.2 (2)

Для чего можно использовать программу Filezilla? - так выглядит следующее задание блока. Посмотрев видеолекцию, отвечаем, что она используется для копирования файлов со своего компьютера на сервер, просмотра содержимого директорий на своем компьютере и копирования файлов с сервера на свой компьютер (рис. [6](#fig:006)).

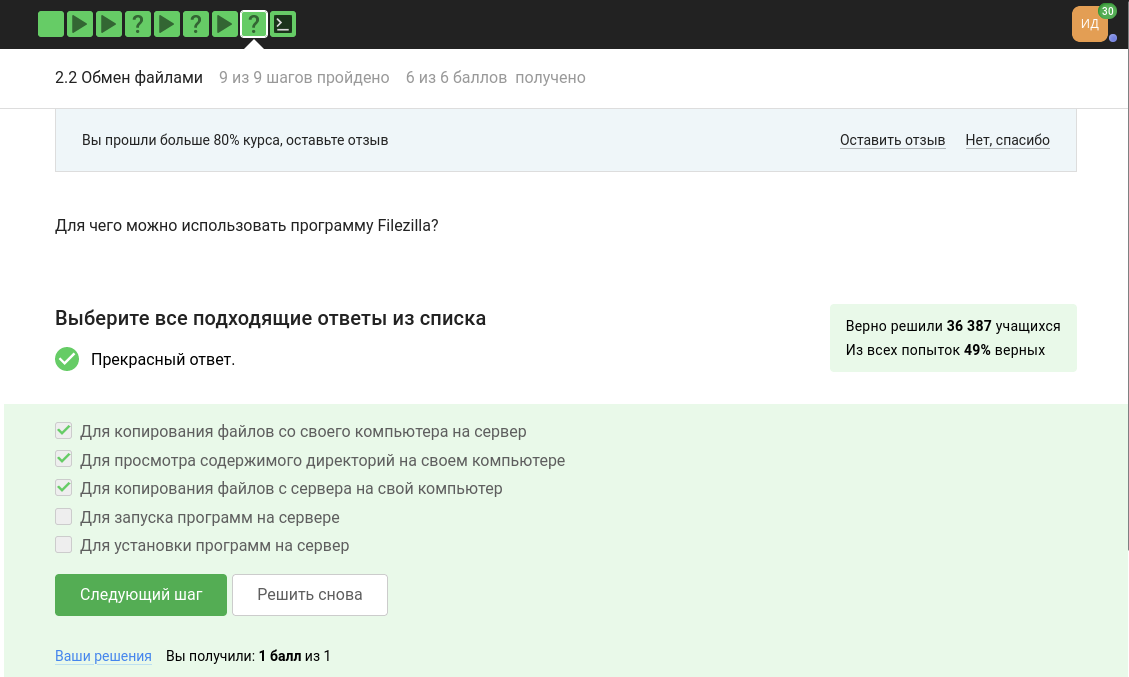


Figure 6: Задание 2.2 (3)

Последнее задание с онлайн-терминалом. Звучит оно так (рис. [7](#fig:007)):

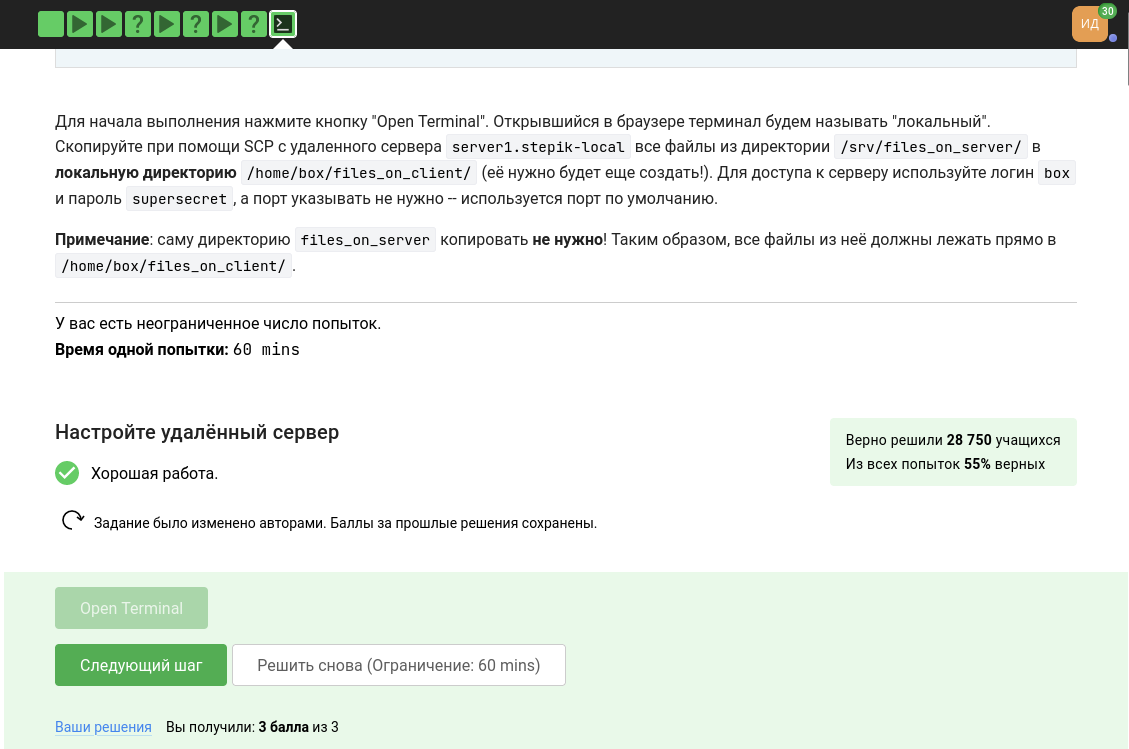


Figure 7: Задание 2.2 (4)

Для его выполнения введём в терминал следующие команды по очереди:

mkdir files\_on\_client # создаём директорию, в которой будут лежать файлы  
scp -r box@server1.stepik-local:/srv/files\_on\_server/\* /home/box/files\_on\_client/  
# копируем файлы с удалённого сервера в локальный репозиторий

## 4.3 Запуск приложений (2.3)

Первое задание выглядит так: что можно сделать, если требуется запустить на сервере программу, для работы которой нужен не терминал, а экран? Варианты проверки наличия версии этой программы специально для терминала и настройка сервера так, чтобы он поддерживал вывод информации на экран компьютера являются правильными (рис. [8](#fig:008)).

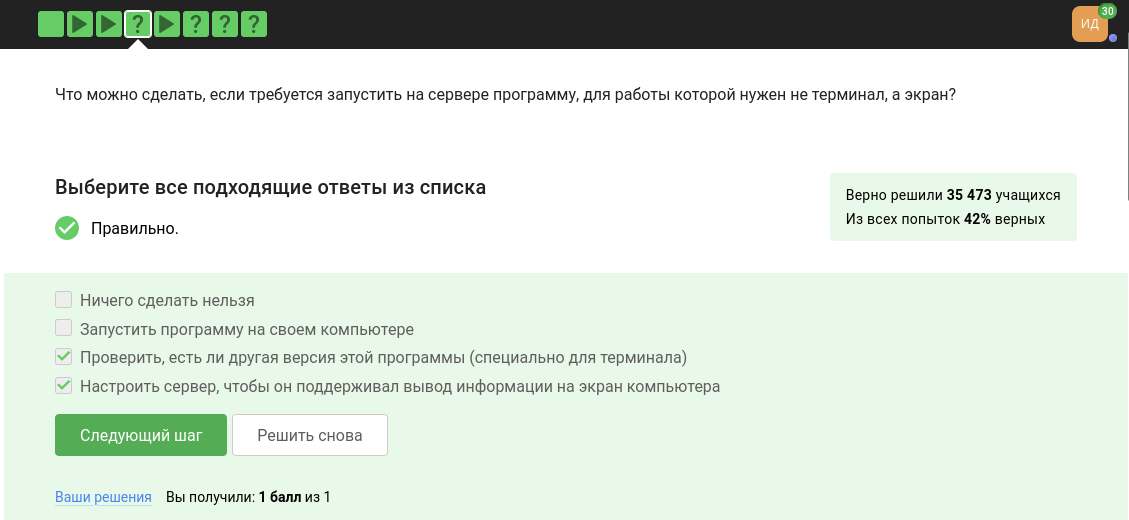


Figure 8: Задание 2.3 (1)

Из видеолекции мы узнаём, что для доступа к справочной информации какой-либо программы использовать можно либо команду man, help (–help, -help, -h) (рис. [9](#fig:009)).

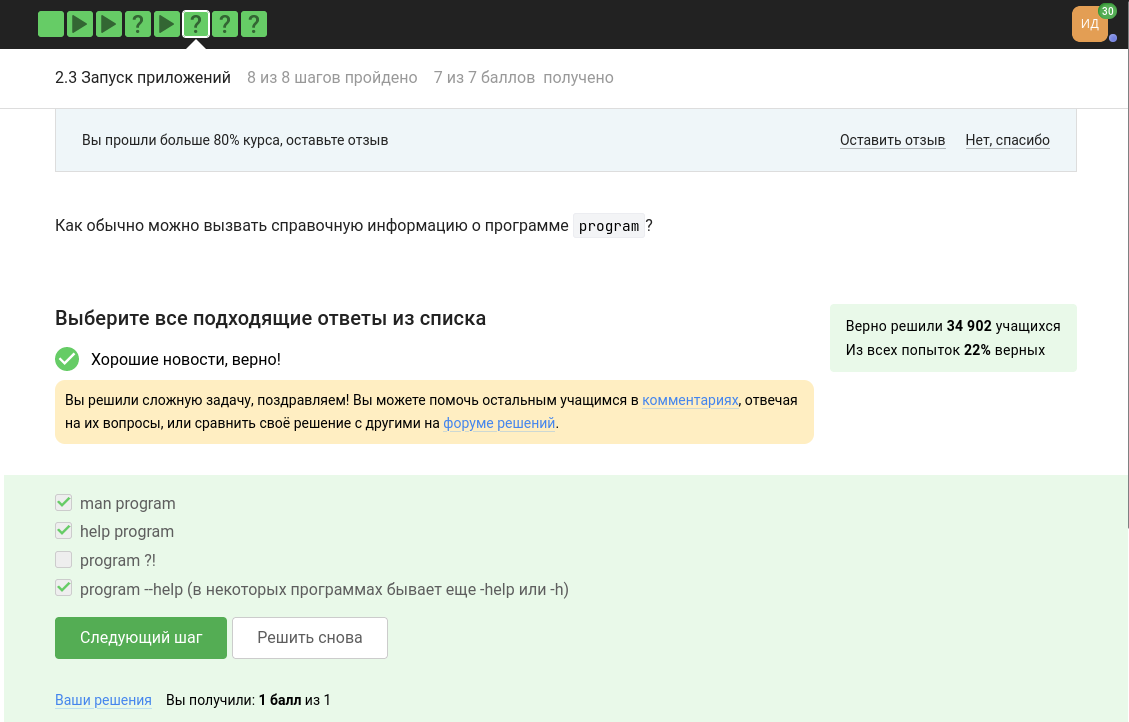


Figure 9: Задание 2.3 (2)

Нужно узнать, какие форматы данных программа FastQC, про которую говорили в лекции, может принимать на вход. Открываем хелп по ней, перенаправляемся в браузер и читаем (рис. [10](#fig:010)), а затем выбираем подходящие ответы из списка (рис. [11](#fig:011)):

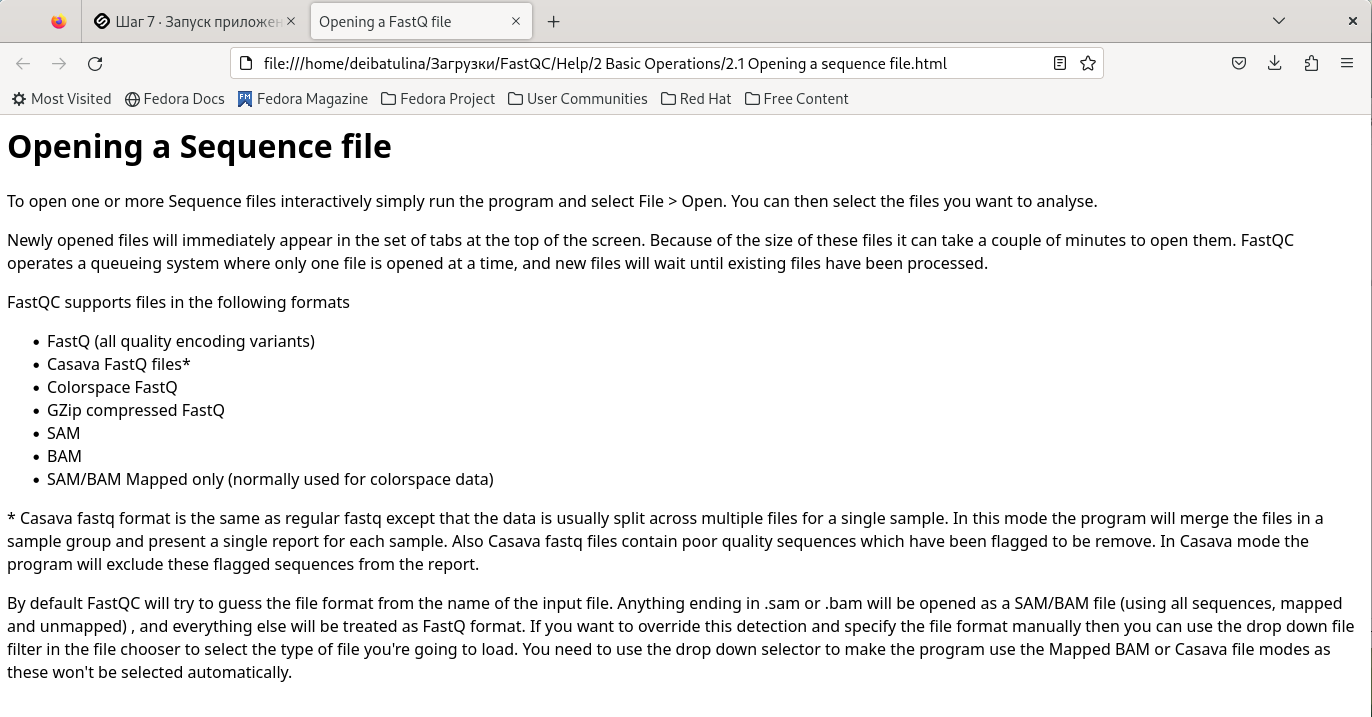


Figure 10: Хелп по программе FastQC

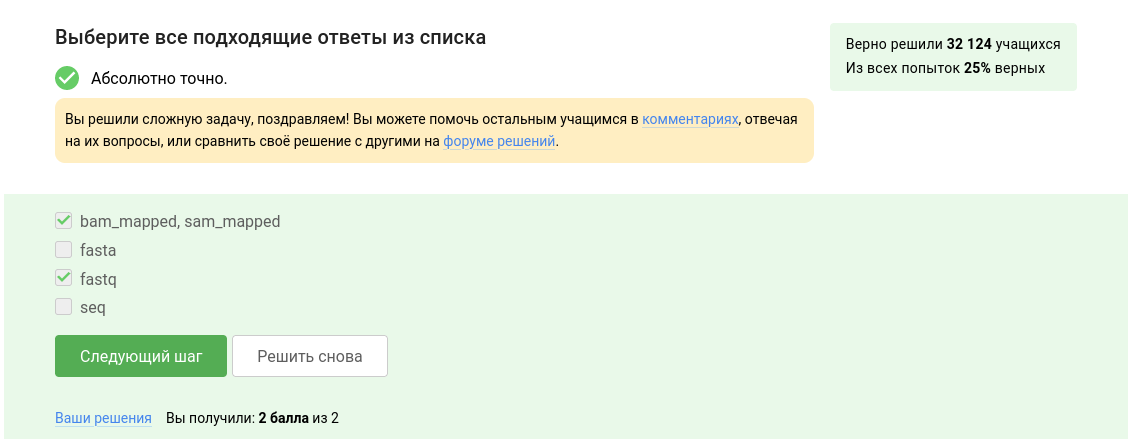


Figure 11: Задание 2.3 (3)

Для выполнения следующего задания (рис. [12](#fig:012)) необходимо скачать приложенный к заданию файл, установить программу Clustalw. Затем воспользовать хелпом по программе и найти команду, которая запускает в терминале Clustal на файле test.fasta и выполняет множественное выравнивание (multiple alignment). Это команда:

clustalw test.fasta -align

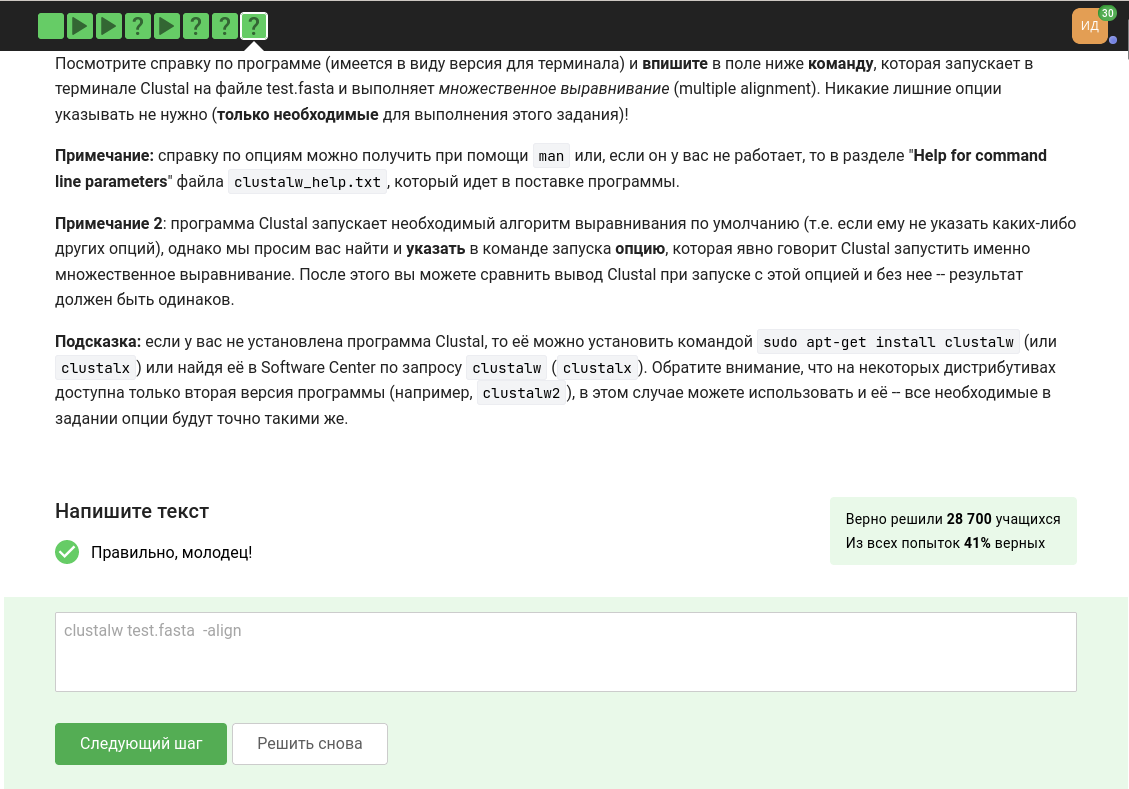


Figure 12: Задание 2.3 (4)

## 4.4 Контроль запускаемых процессов (2.4)

Первое задание выглядит так: Предположим вы запустили программы program1, program2 и program3 в фоновом режиме. После этого вы выполнили следующие действия: 1. fg %1 - отправили программу с идентификатором 1 в фоновый режим 2. Ctrl+С - остановили этот процесс 3. fg %2 - отправили программу с идентификатором 2 в фоновый режим 4. Ctrl+Z - приостановили этот процесс 5. jobs - вывели на экран список запущенных процессов

При выполнении последней команды система покажет нам, что лишь процессы 2 и 3 запущены. Процесс 1 не будет отображаться, поскольку его мы полностью остановили (рис. [13](#fig:013)).

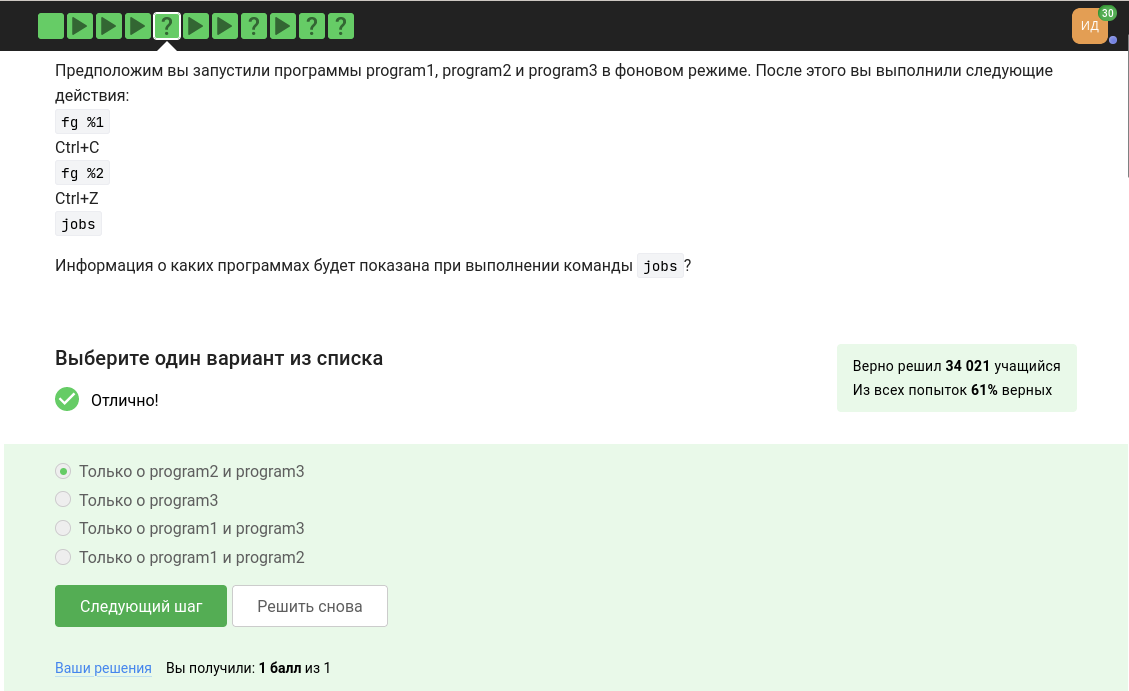


Figure 13: Задание 2.4 (1)

Из видеолекции мы узнаём, что PID - идентификаторы запущенных процессов одинаковы только у ps и top (рис. [14](#fig:014)).

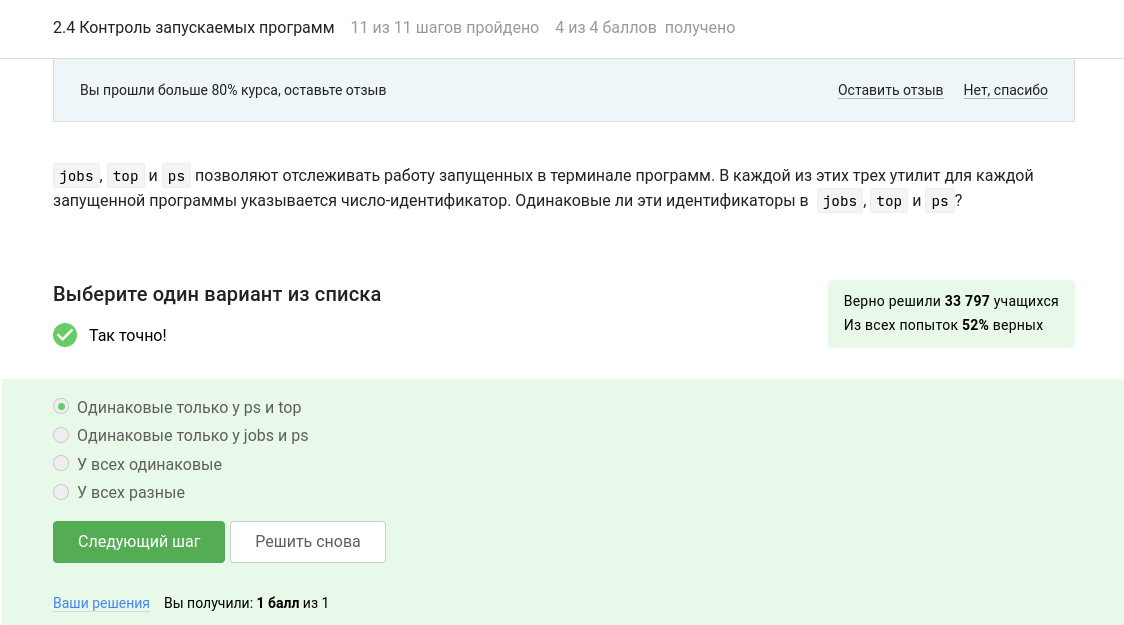


Figure 14: Задание 2.4 (2)

С помощью команды kill -9 можно мгновенно завершить остановленный процесс (рис. [15](#fig:015)). kill без опций остановит запущенный процесс, а про команду kill с опцией 18 не говорилось ничего.

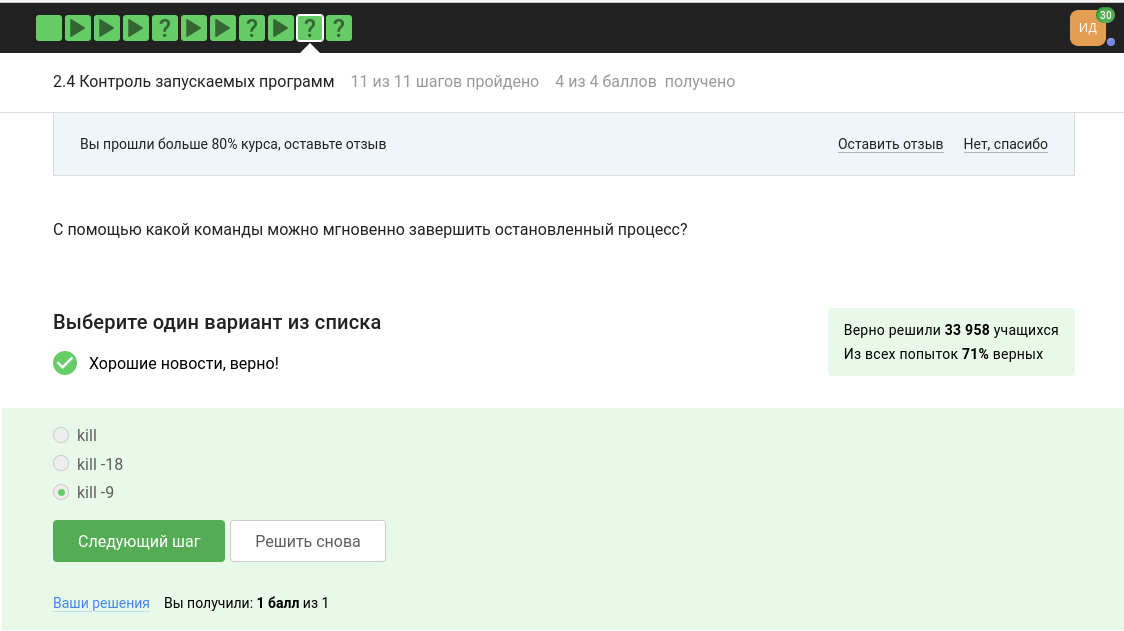


Figure 15: Задание 2.4 (3)

Для выполнения следующего задания нужно ответить на вопрос: Что произойдет, если использовать kill (без опций) по отношению к процессу, который был приостановлен при помощи Ctrl+Z? Команда kill без опций остановит запущенный процесс, поэтому сначала нужно этот процесс возобновить (продолжить). (рис. [16](#fig:016)).

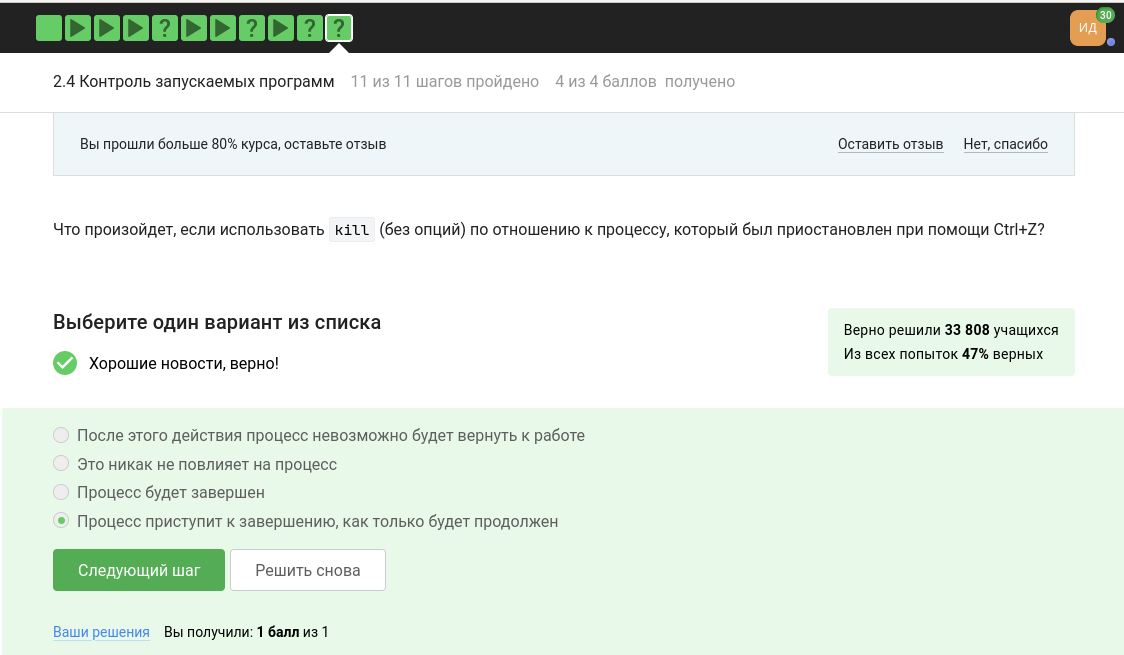


Figure 16: Задание 2.4 (4)

## 4.5 Многопоточные приложения (2.5)

Поскольку приложение остановлено, оно не использует ресурсы процессора (рис. [17](#fig:017)).

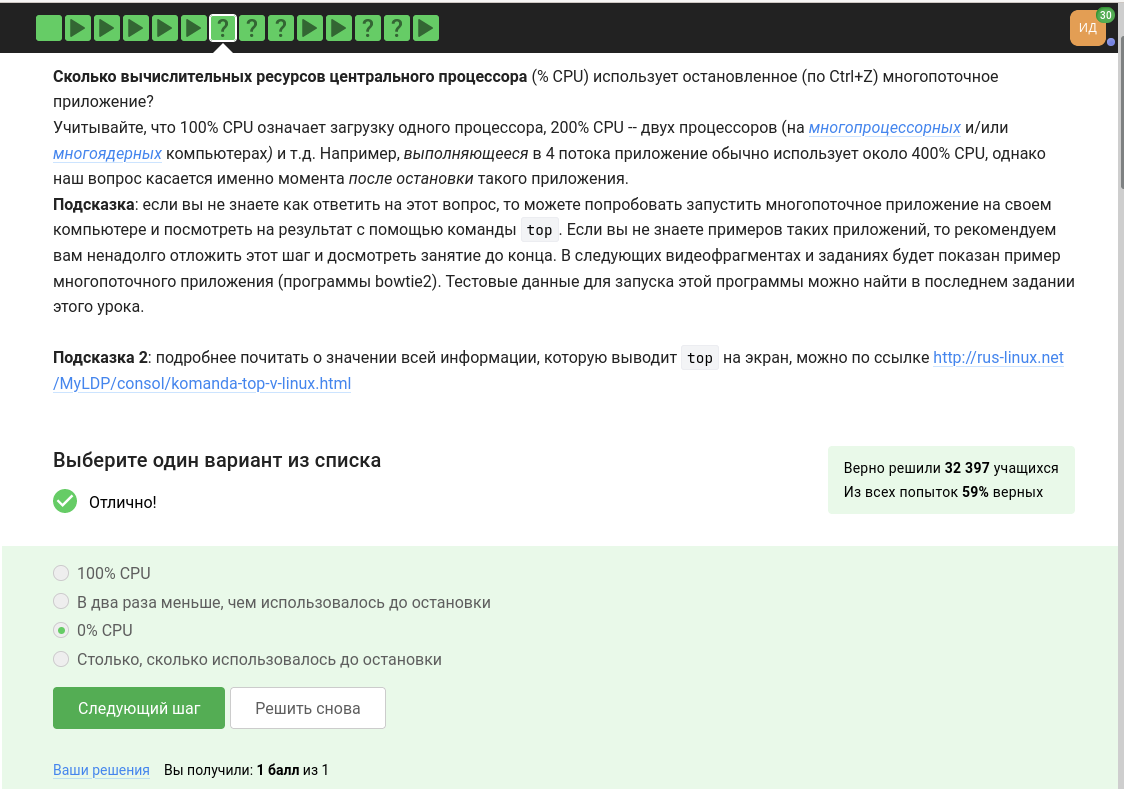


Figure 17: Задание 2.5 (1)

А вот памяти оно потребляет столько же, сколько и в момент остановки (рис. [18](#fig:018)).

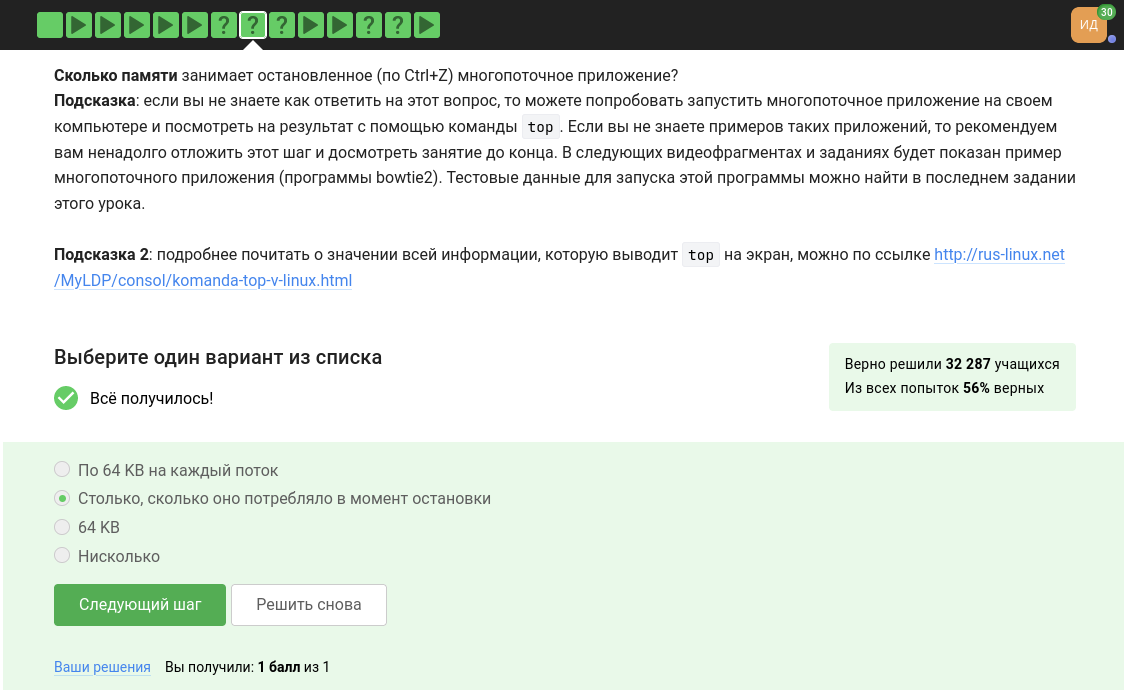


Figure 18: Задание 2.5 (2)

Из лекции мы узнаём, что принудительно завершить один из потоков запущенного многопоточного приложения никак нельзя (рис. [19](#fig:019)).

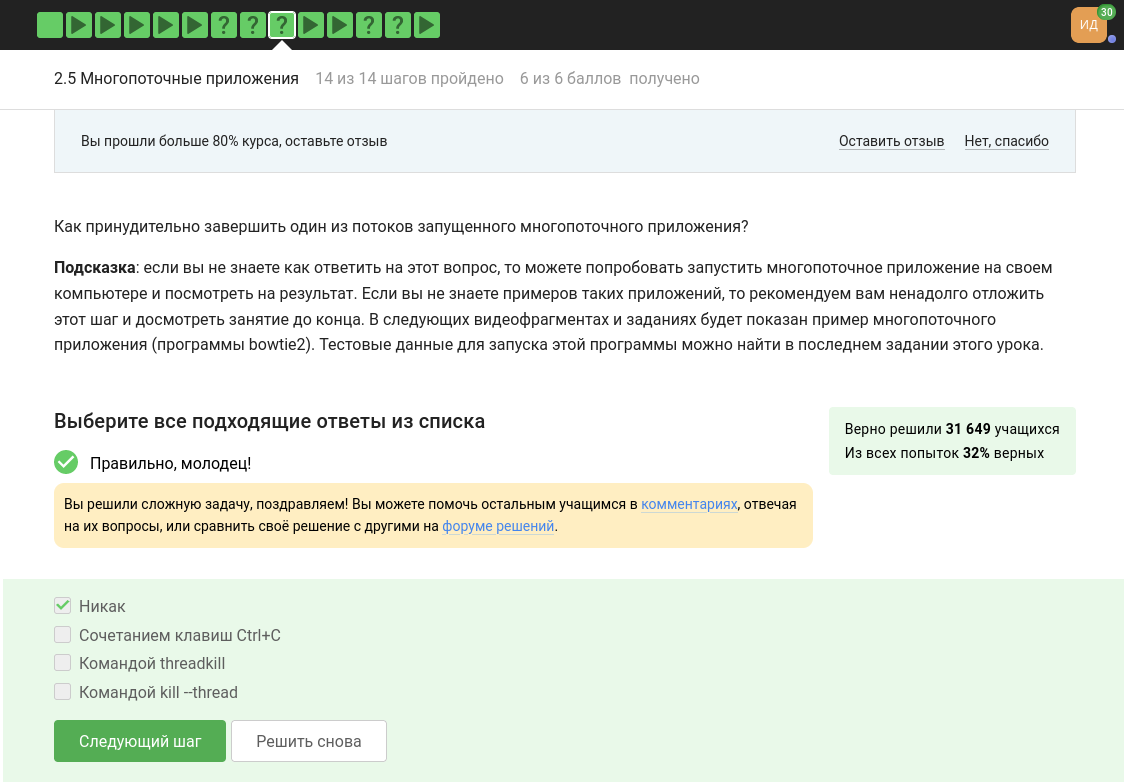


Figure 19: Задание 2.5 (3)

В следующем задании идёт речь о программе bowtie2. Запуск bowtie2 состоит из двух шагов - сначала запускаем подпрограмму bowtie2-build, а затем подпрограмму bowtie2. Из этих шагов можно выполнить в несколько потоков обе подпрограммы, как указано в документации. Приведу выдержку из документации к этой программе:

--threads <int> By default bowtie2-build is using only one thread. Increasing the number of threads will speed up the index building considerably in most cases.

Однако, в курсе ответ *Оба* является неверным, скорее всего потому, что курс устарел. Но, отметив вариант *Только bowtie2* как правильный, получаем свой балл (рис. [20](#fig:020)).

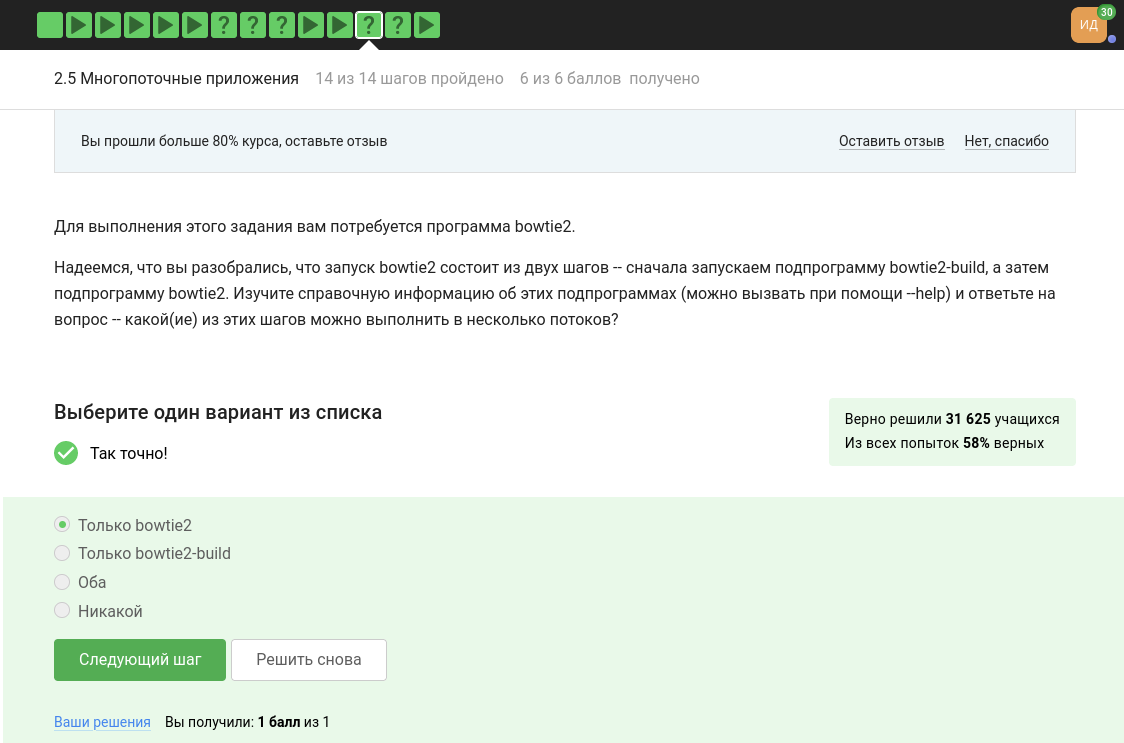


Figure 20: Задание 2.5 (4)

Задание звучит так (рис. [21](#fig:021)):

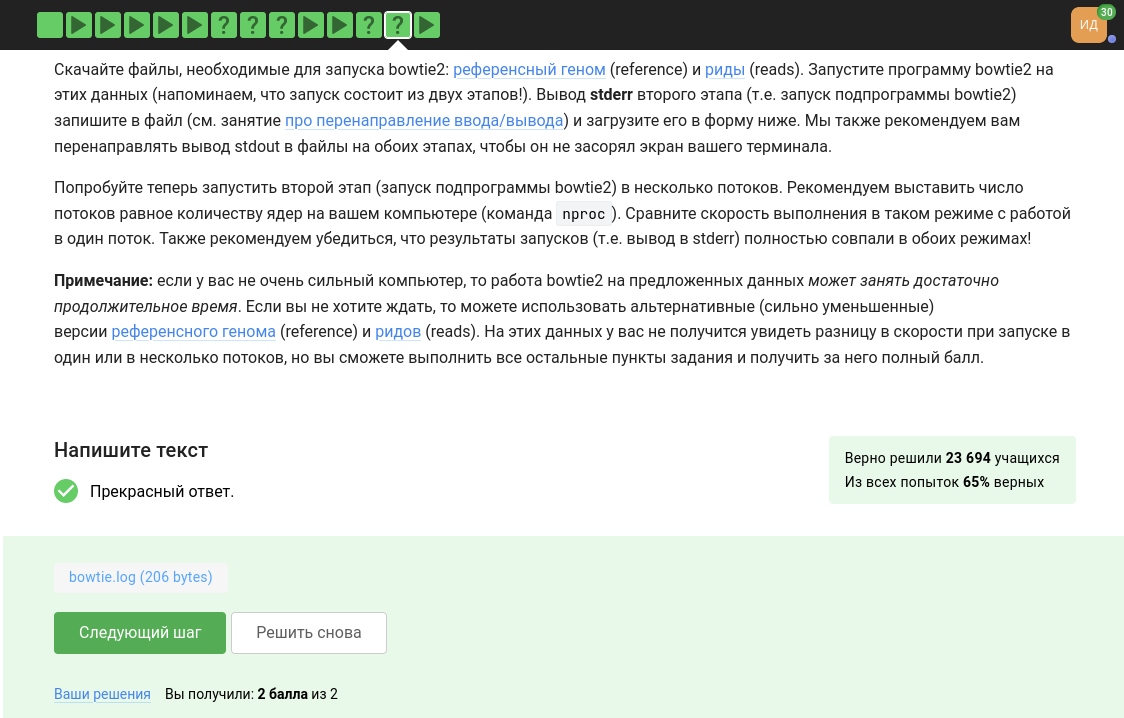


Figure 21: Задание 2.5 (5)

Для того, чтобы выполнить его, сделаем следующее:

1. Скачаем архив bowtie2 версия для 64-разрядного Linux из первого слайда (т.к. у меня 64-разрядный процессор).
2. Скачиваем референсный геном и риды со слайда с этим заданием (нужные файлы).
3. Распаковываем архив bowtie2-2.1.0-linux-x86\_64.zip с помощью команды unzip (т.к. расширение у архива *zip*).
4. Переместим скачанные файлы reference.fasta и архив reads.fastq.gz в директорию с распакованным bowtie2-2.1.0.
5. Распаковываем архив reads.fastq.gz с помощью команды gunzip (т.к. архив имеет расширение gz).
6. Запускаем программу (команда ./bowtie2-build - первый шаг) с нужным файлом и индексом (см. уроки и описание задания).
7. Запускаем программу ./bowtie2 - второй шаг с индексом и вторым файлом (см. уроки и описание задания). Лог записываем в один файл, ошибки в другой, пользуясь перенаправлением ввода-вывода из предыдущих уроков.
8. Добавляем в форму файл с ошибками (т.к. именно его требуется загрузить).

Получаем баллы за это задание и двигаемся дальше.

## 4.6 Менеджер терминалов tmux (2.6)

На этом занятии мы рассмотрим менеджер терминалов tmux. Это специальная программа, которая может объединять в себе несколько терминалов, каждый из которых содержит свои процессы. Важная особенность tmux заключается в том, что если запустить его на сервере, то он будет помнить историю запусков процессов и они продолжат выполнение даже если соединение с сервером будет потеряно. Кроме того, при новом соединении с сервером можно будет подключиться к уже запущенному tmux и продолжить работу с момента потери связи.

Задание 1: Вы открыли две вкладки в терминале. В одной из них вы запустили процесс и приостановили его. Переключившись во вторую вкладку и набрав fg, вы добьётесь следующего:

* Процесс вернется к работе в исходной вкладке
* Процесс переместится во вторую вкладку, но останется в режиме “приостановки”
* Терминал сообщит, что нет процесса для запуска в fg
* Процесс переместится во вторую вкладку и продолжит работу
* Проделав данные действия, я обнаружила, что терминал сообщит, что нет процесса для запуска в fg, поскольку вкладки здесь работают независимо друг от друга (как вкладки браузера) (рис. [22](#fig:023)).

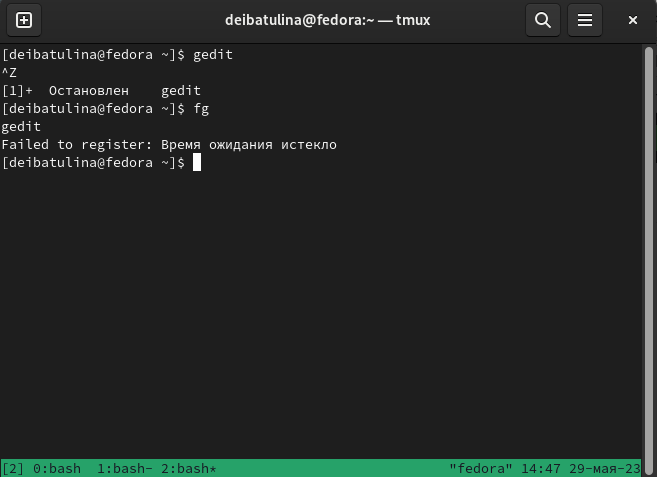


Figure 22: Задание 2.6 (1)

Задание 2: Предположим, что в tmux осталась последняя открытая вкладка. Что произойдет, если вы введете в этой вкладке в командную строку команду exit? Ответ: tmux завершит работу (рис. [23](#fig:024)).

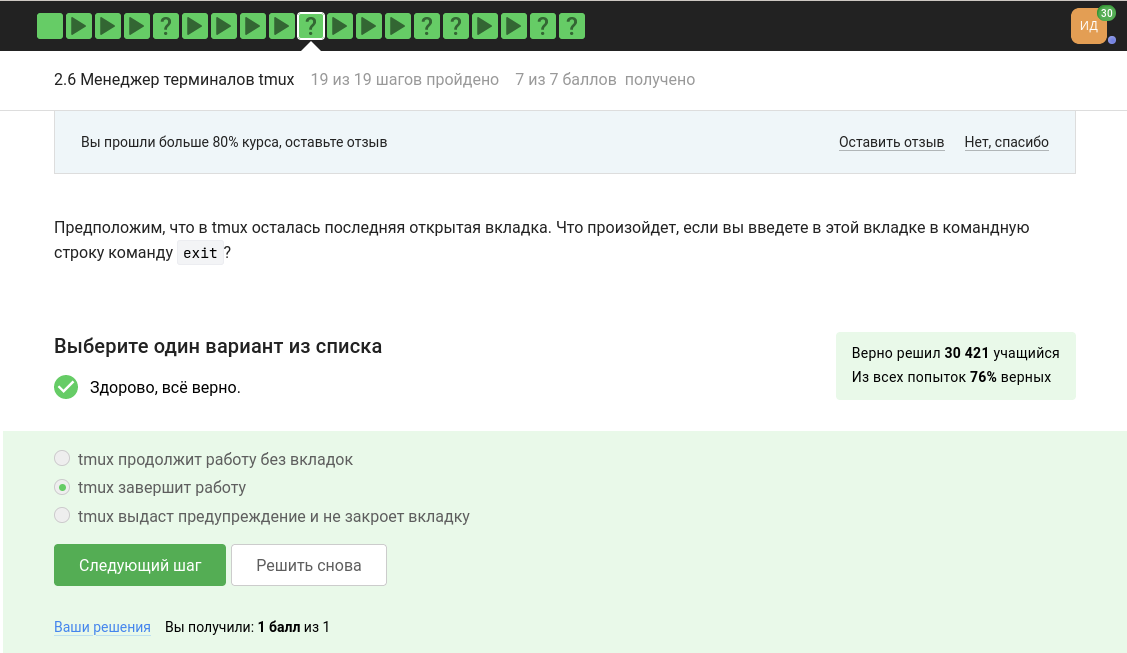


Figure 23: Задание 2.6 (2)

Следующее задание звучит так: Предположим, что вы открыли терминал, зашли в нем на сервер, запустили на этом сервере tmux и начали работу в нем. Что произойдет, если вы теперь закроете терминал? Если терминал закрыть, то соединение с сервером будет утеряно, однако tmux продолжит работу (рис. [24](#fig:025)).

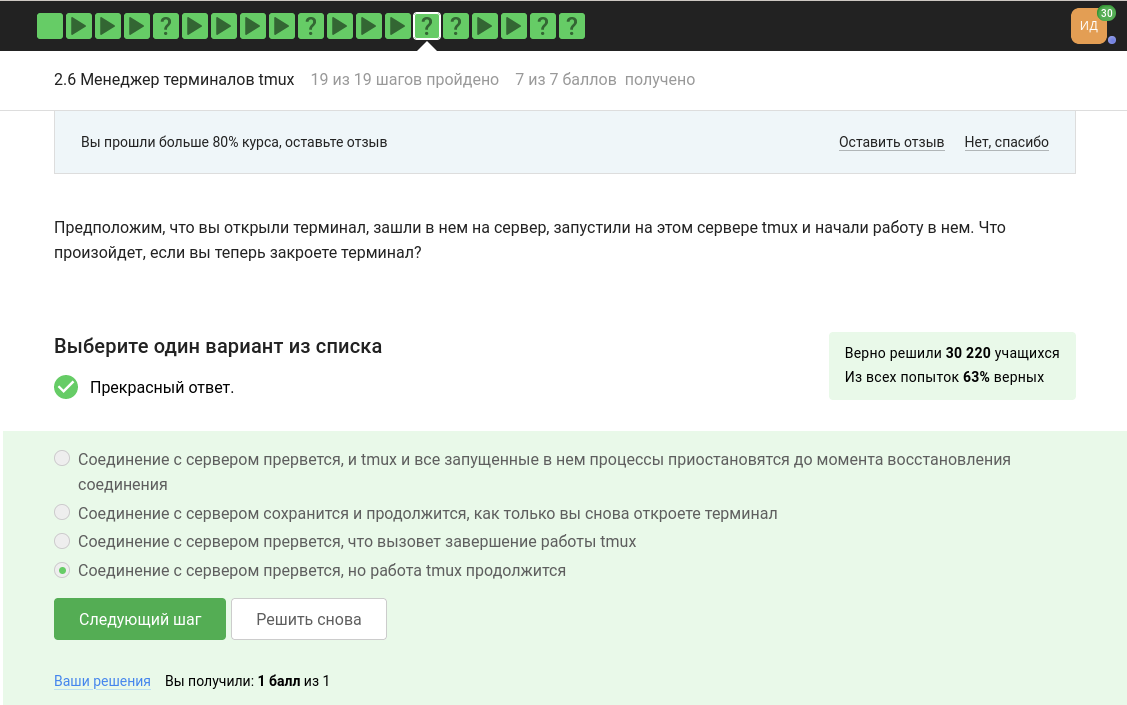


Figure 24: Задание 2.6 (3)

Задание: Что произойдет, если запустить процесс в фоновом режиме в одной из вкладок tmux, а затем принудительно закрыть эту вкладку (Ctrl+B, X)? Так как вкладки не связаны друг с другом, то закроется и вкладка, и запущенный в ней в фоновом режиме процесс (рис. [25](#fig:026)).

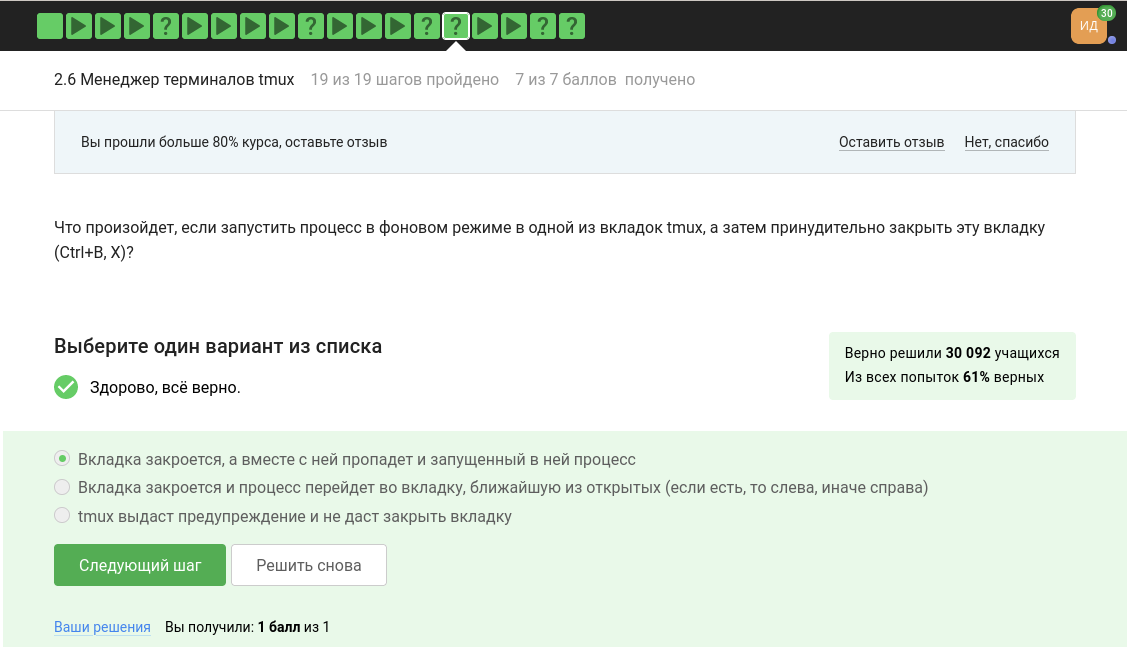


Figure 25: Задание 2.6 (4)

Задание звучит так (рис. [26](#fig:027)):

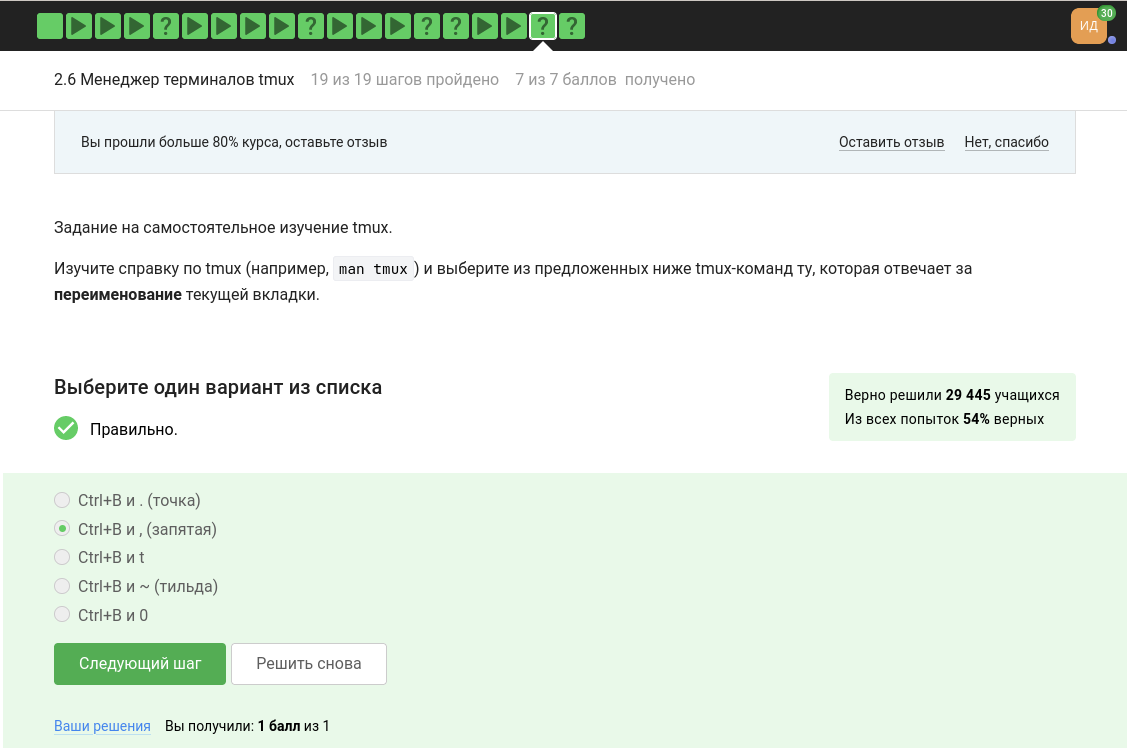


Figure 26: Задание 2.6 (5)

В справке написано, что ответ: ctrl+b и , (рис. [27](#fig:028)).

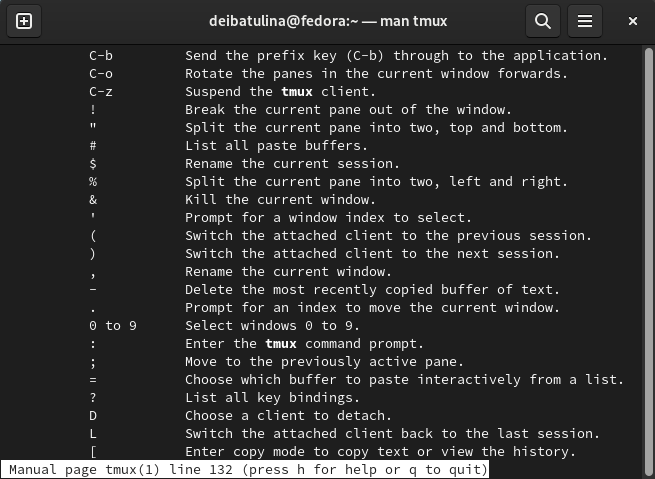


Figure 27: Справка по tmux

В задании необходимо поработать со справкой по tmux и отметить все верные утверждения (рис. [28](#fig:029)).

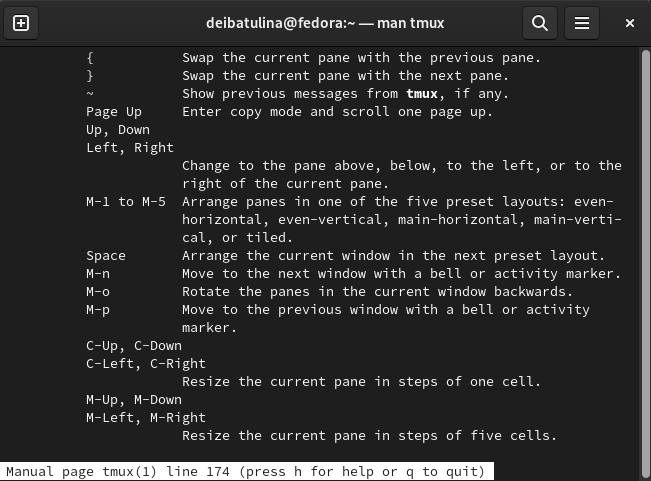


Figure 28: Справка по tmux

Таким образом, выбираем утверждения 1, 4, 6 (рис. [29](#fig:030)).

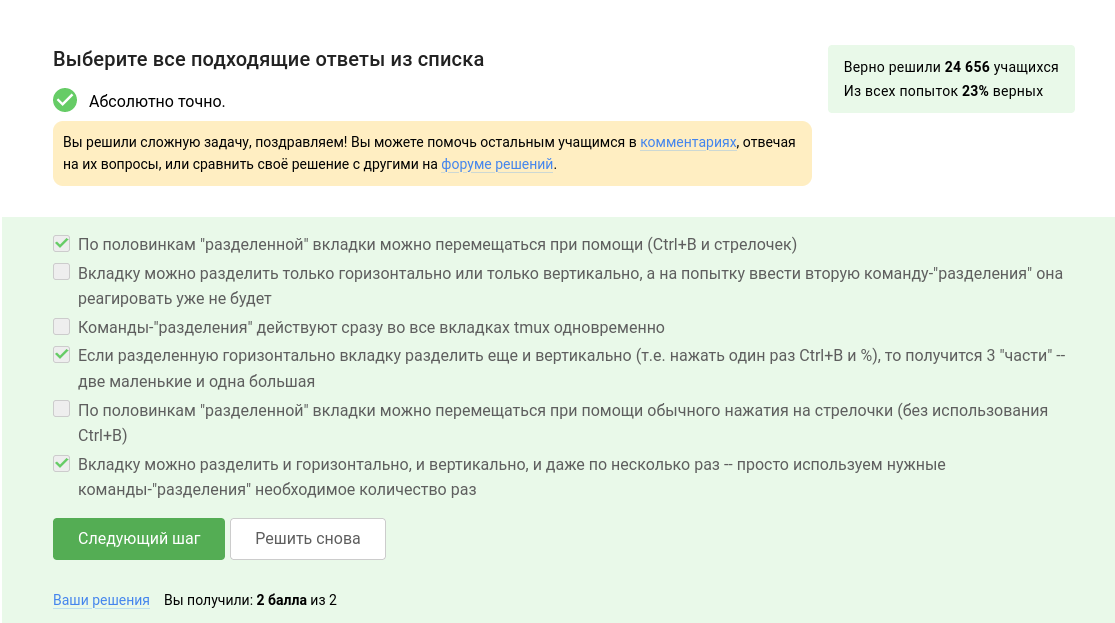


Figure 29: Задание 2.6 (6)

# 5 Выводы

В результате выполнения второго этапа курса я научилась работать с tmux, просматривать запущенные процессы, познакомилась с работой на сервере и запуском многопоточных приложений.

# Список литературы