Отчёт по прохождению внешнего курса

2 этап

Ярослав Антонович Меркулов

Содержание

Список иллюстраций

# 1 Цель работы

Пройти второй этап внешнего курса “Введение в Linux”.

# 2 Выполнение этапа

1. Все варианты подходят, теоретический вопрос (рис. 1).

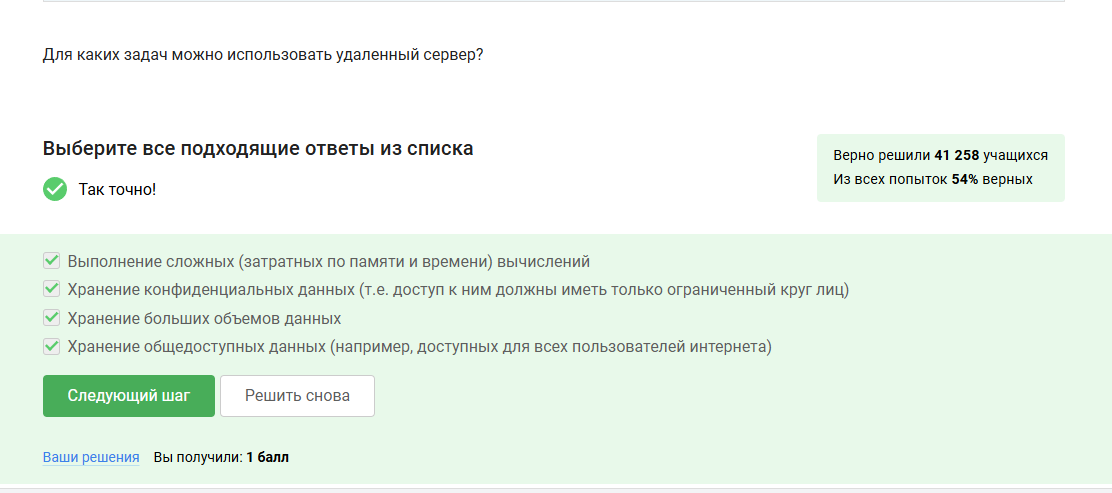


Рис. 1: Задание 1

1. Ключ, который можно безопасно пересылать по интернету, — это файл id\_rsa.pub. Это публичный ключ, его предназначение — обмениваться с серверами или другими пользователями для установления доверия.

Файл id\_rsa — приватный ключ, его нельзя передавать или показывать другим, чтобы не скомпрометировать безопасность вашей системы.(рис. 2).

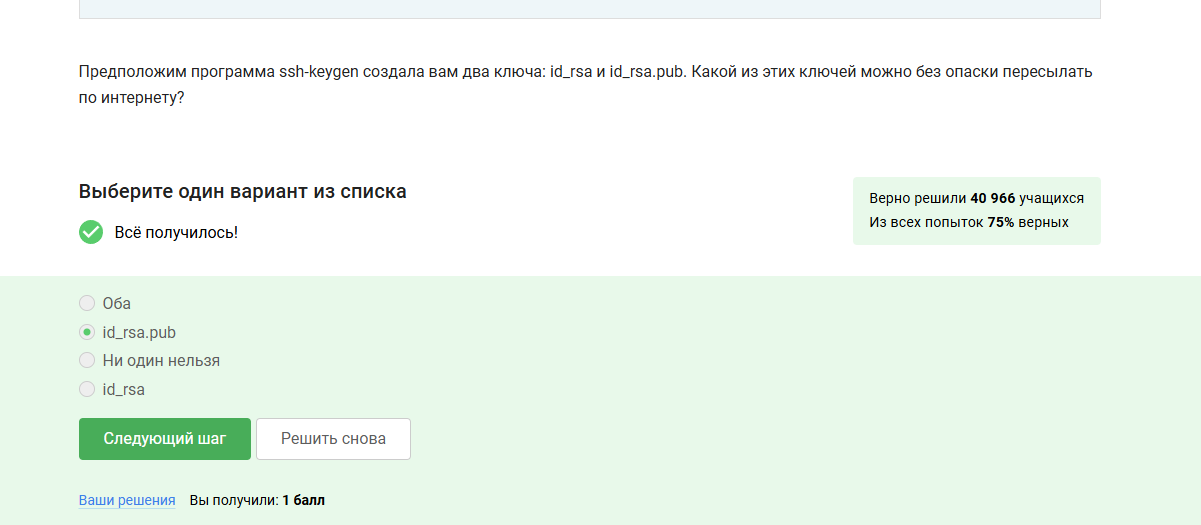


Рис. 2: Задание 2

1. -r — рекурсивное копирование, необходимо для папок и их содержимого. stepic — имя папки, которую нужно скопировать. username@server:~/ — целевая директория на сервере (домашняя директория пользователя).(рис. 3).

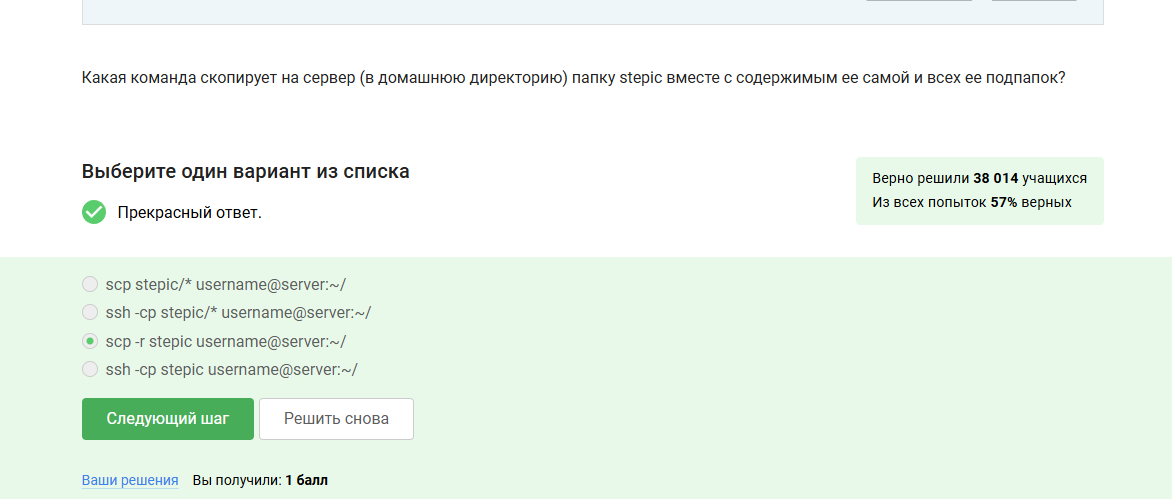


Рис. 3: Задание 3

1. sudo apt-get update

Объяснение:

Эта команда обновляет список доступных пакетов и их версий из репозиториев. После её выполнения система узнает о последних версиях пакетов и сможет найти нужный вам пакет program. (рис. 4).

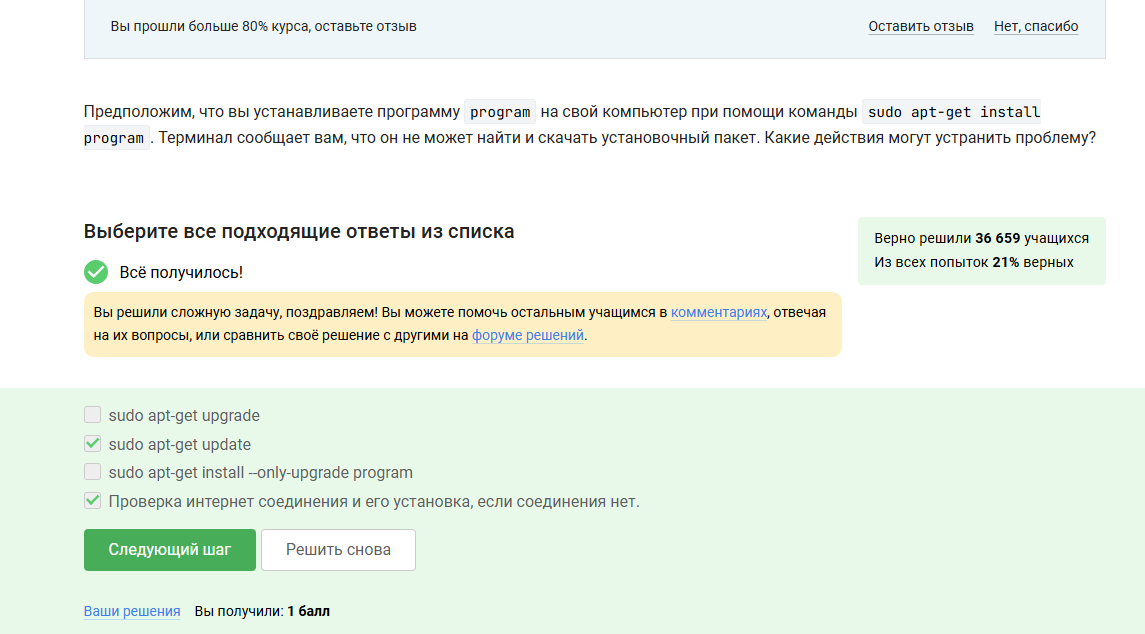


Рис. 4: Задание 4

1. Filezilla можно использовать для всех названных действий, исключая запуск программ на сервере(рис. 5).

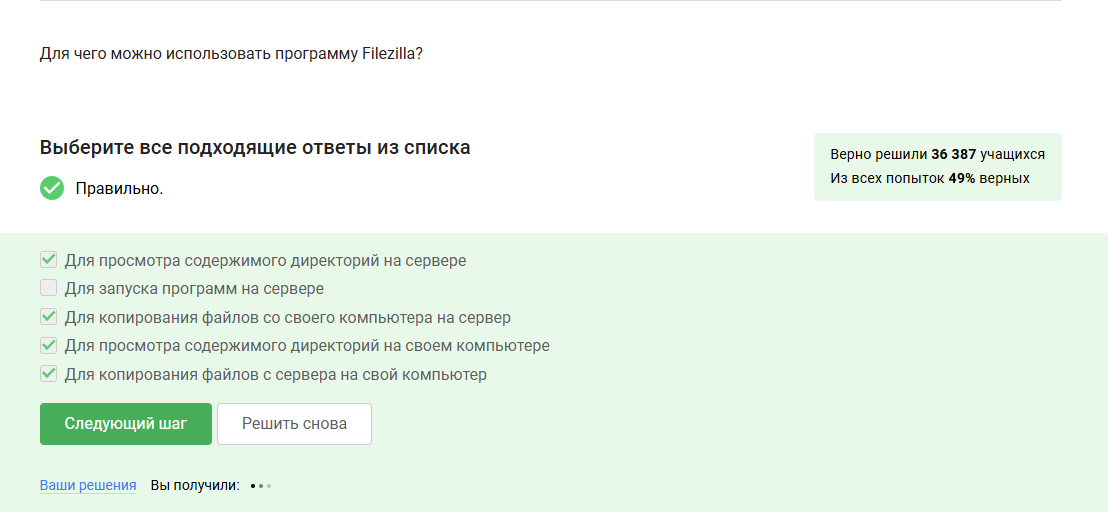


Рис. 5: Задание 5

1. Теоретический вопрос(рис. 6).

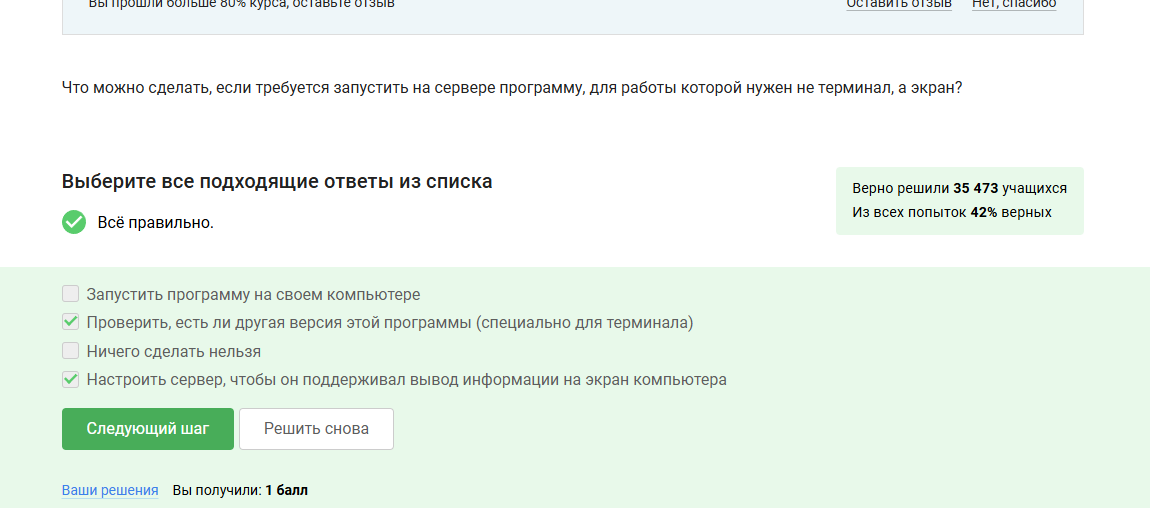


Рис. 6: Задание 6

1. Теоретический вопрос. program ?! не является вызовом справки.(рис. 7).

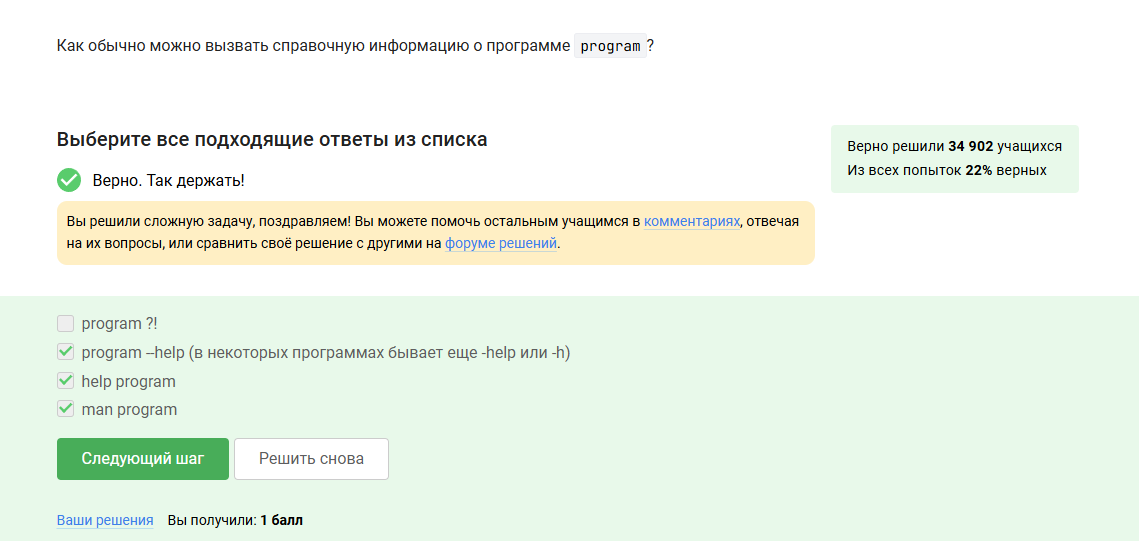


Рис. 7: Задание 7

1. bam,sam,bam\_mapped,sam\_mapped and fastq (форматы, которые поддерживаются)(рис. 8).

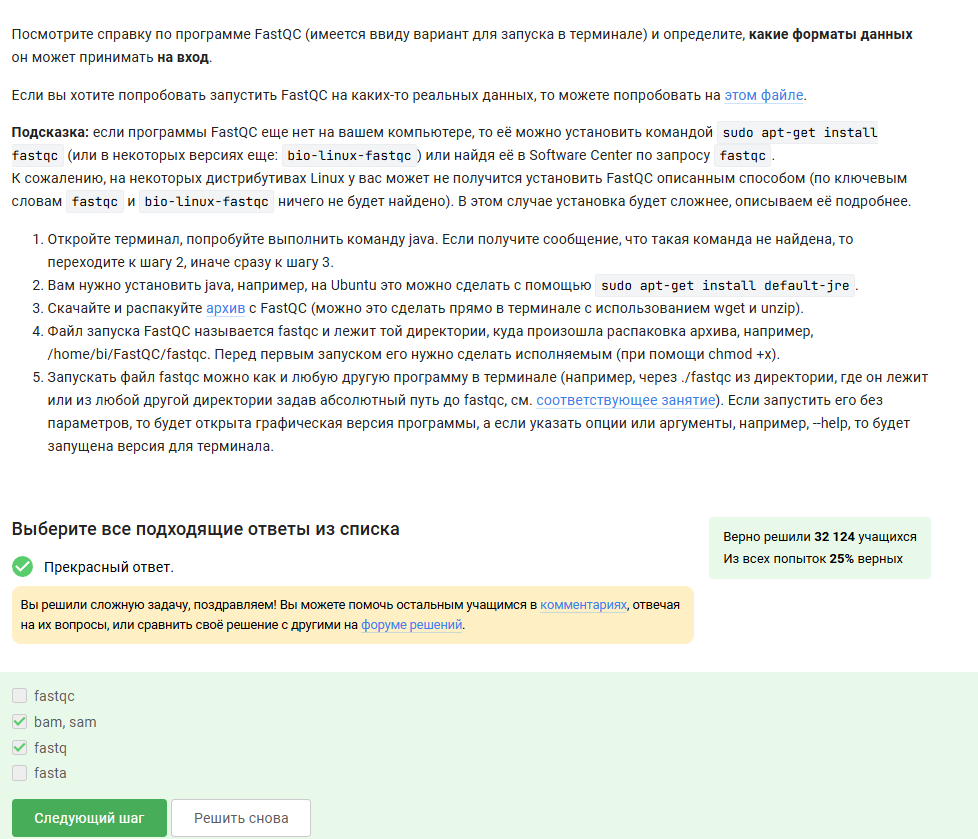


Рис. 8: Задание 8

1. Читаем справку и вводим нужную команду(рис. 9).

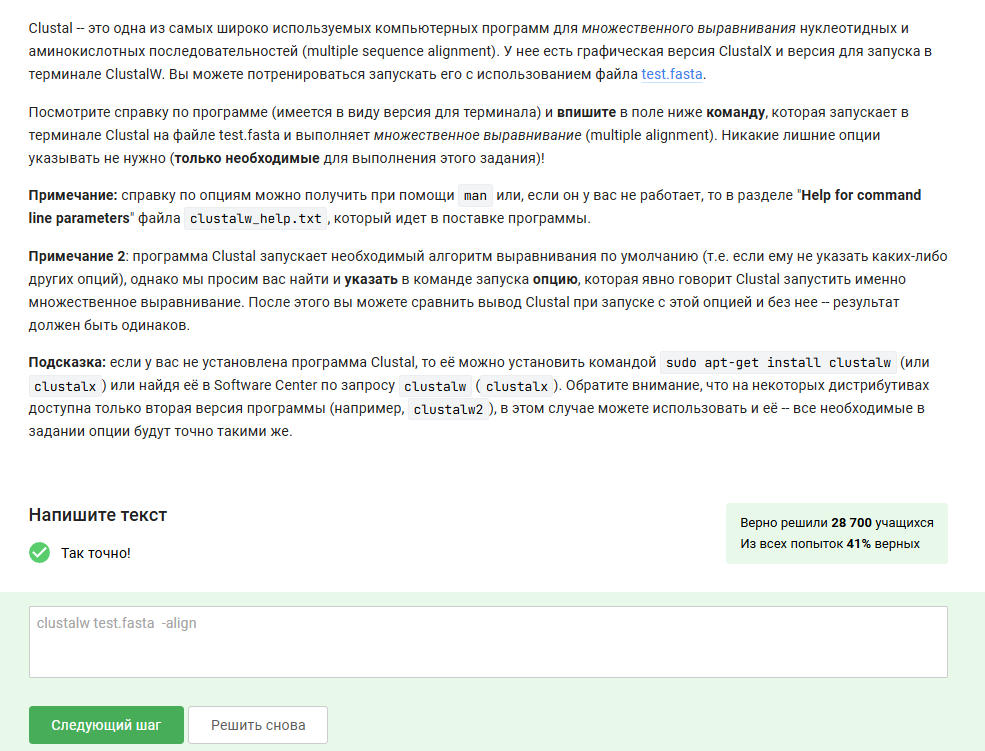


Рис. 9: Задание 9

1. Давайте разберёмся по шагам, что произошло:

Вы запустили program1, program2 и program3 в фоновом режиме — скорее всего, командой с &, например:

program1 & # запущен в фонеprogram2 & # запущен в фонеprogram3 & # запущен в фоне Выполнили fg %1 — переводим на передний план (фронт-энд) программу, которая была запущена как job №1 (предположим, это program1). Нажали Ctrl+C — послали сигнал прерывания текущей программы (program1), она завершилась. Выполнили fg %2 — снова переводим на передний план программу №2 (program2). Нажали Ctrl+Z — приостановили выполнение программы №2 (program2) (она переходит в состояние “заморожена” или “остановлена”). Выполнили команду jobs. Что показывает команда jobs? Она выводит список текущих задач (jobs), которые находятся в состоянии “запущены в фоне” или “приостановлены”.

После всех действий:

program1 завершилась (после Ctrl+C) — она исчезла из списка jobs. program2 была приостановлена (Ctrl+Z) — она всё ещё есть в списке jobs. program3, который был запущен ранее и не был тронут, всё ещё работает в фоне. Итог: В списке jobs будут отображены только те программы, которые ещё активны или приостановлены, то есть:

program2 (приостановлена) program3 (в фоновом режиме)(рис. 10).

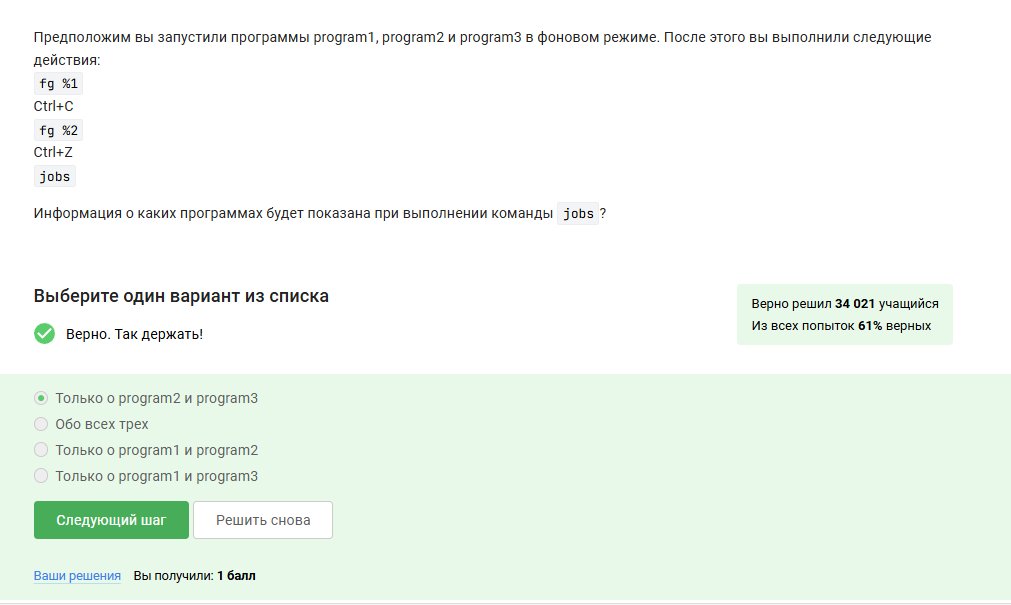


Рис. 10: Задание 10

1. Теоретический вопрос(рис. 11).

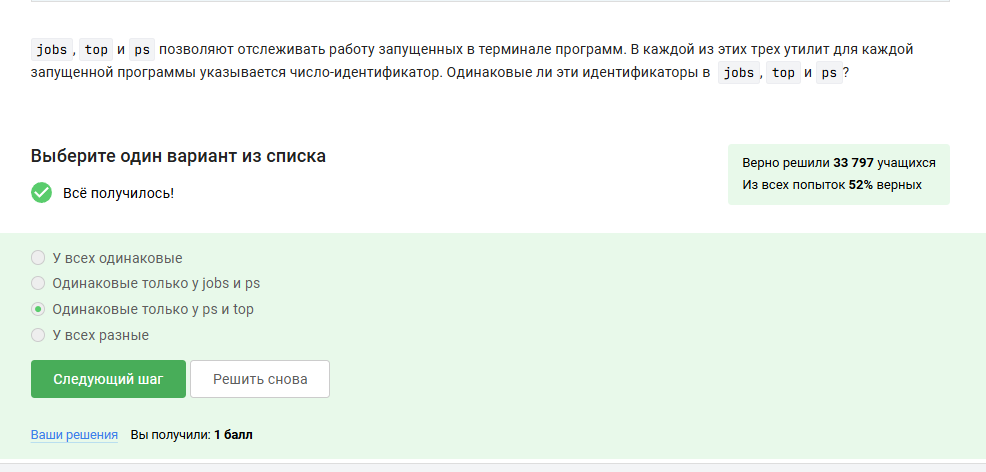


Рис. 11: Задание 11

1. kill -9 — принудительно завершает процесс без возможности обработки сигнала.(рис. 12).

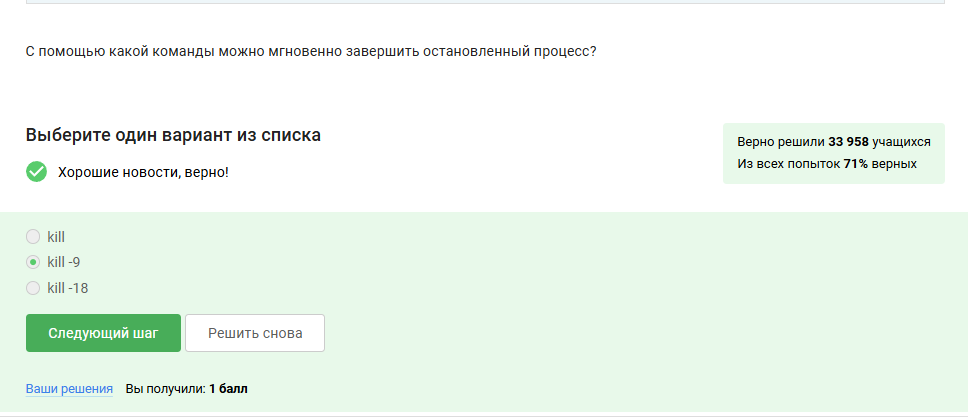


Рис. 12: Задание 12

1. Сначала процесс надо продолжить, только после этого он будет завершен(рис. 13).

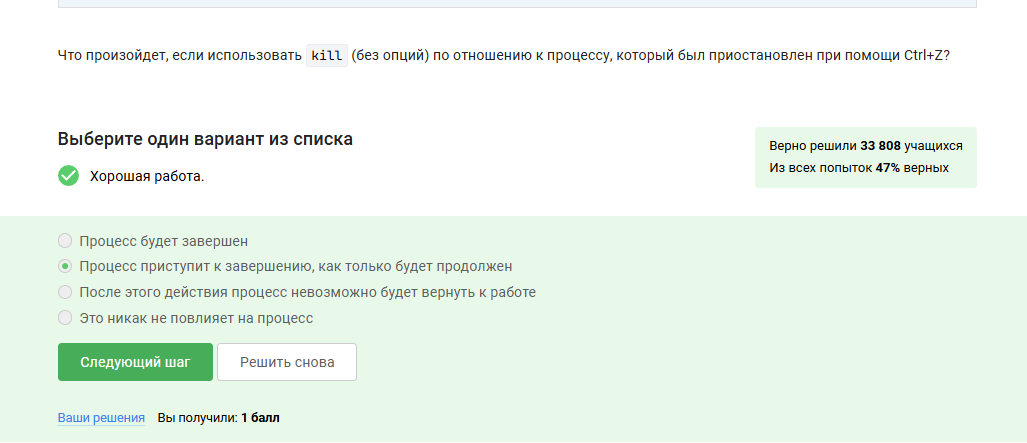


Рис. 13: Задание 13

1. Когда процесс или поток приостанавливается (сигналом SIGSTOP или через Ctrl+Z), он переходит в состояние “заморожен” (stopped). В этом состоянии он не выполняется и не занимает процессорное время. Поэтому, в состоянии “приостановки”, использование CPU для этого процесса равно 0%.(рис. 14).

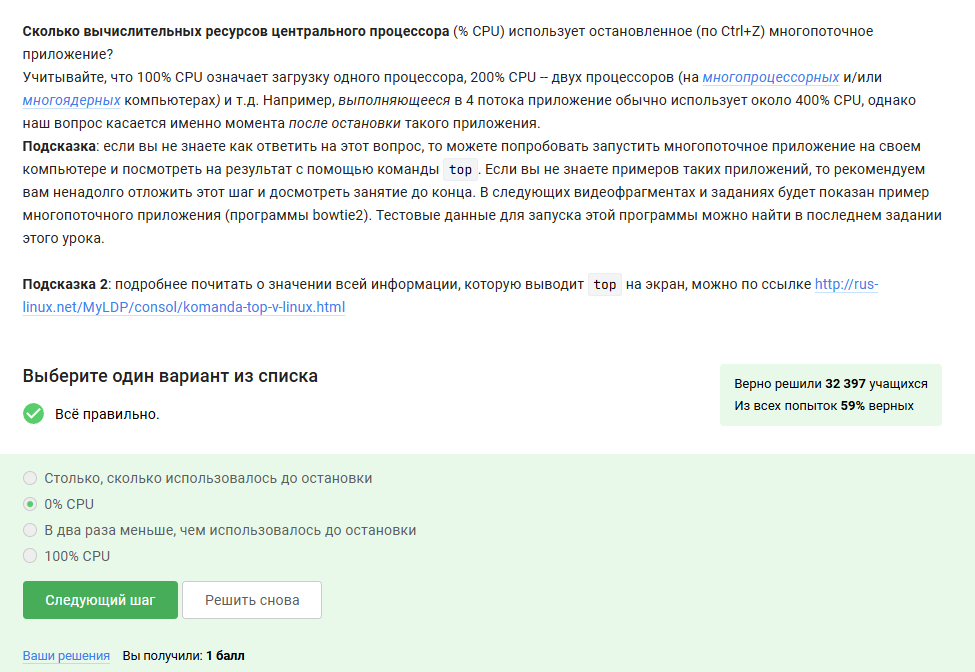


Рис. 14: Задание 14

1. Приостановка процесса с помощью Ctrl+Z (после сигнала SIGSTOP) не освобождает и не уменьшает его использование памяти. Процесс остается в памяти операционной системы, его ресурсы (включая память) сохраняются. Единственное изменение — процесс переходит в состояние “остановлен” (stopped), но его память остается выделенной.(рис. 15).

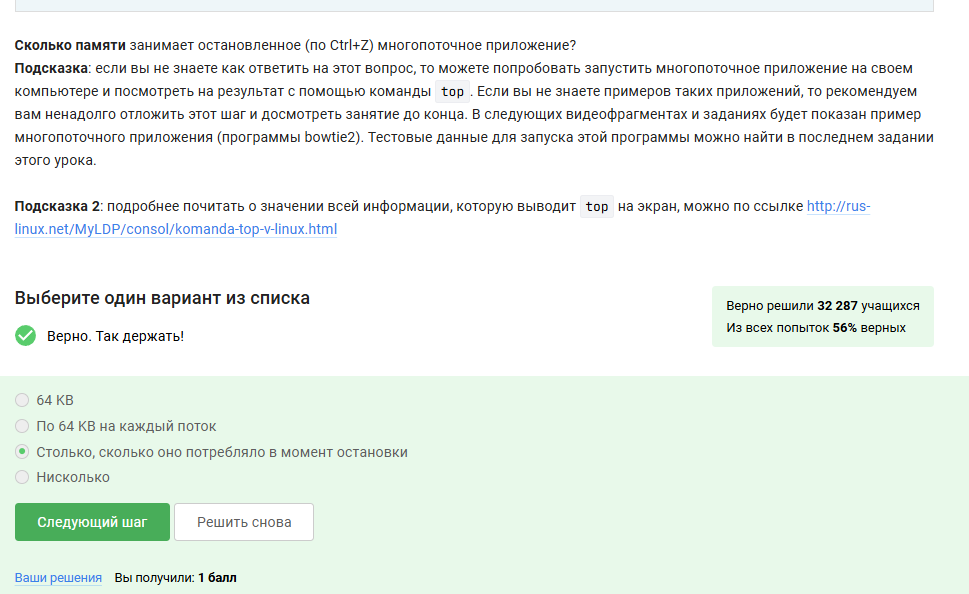


Рис. 15: Задание 15

1. Нельзя завершить отдельный поток(рис. 16).

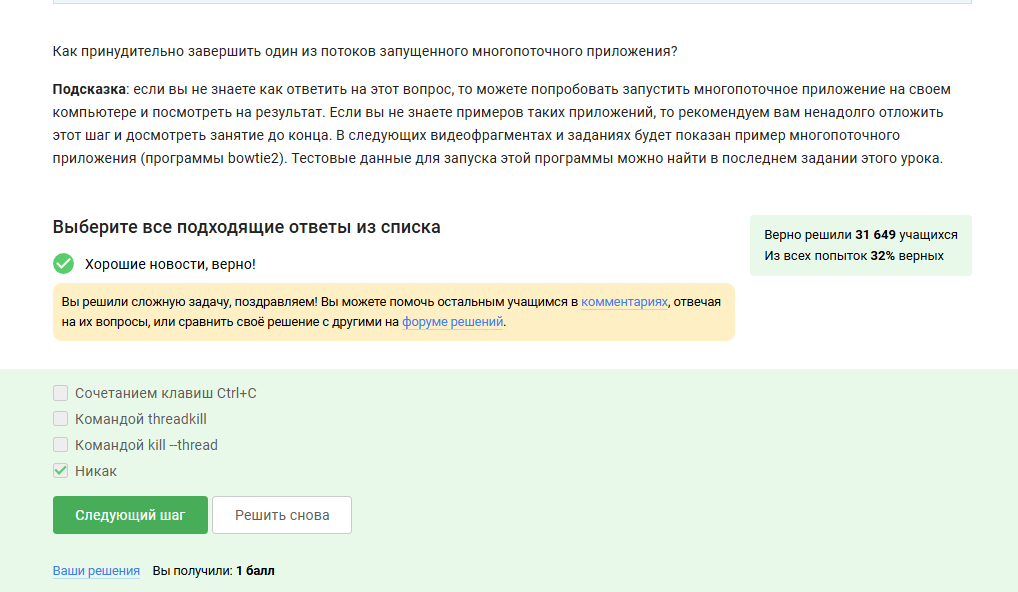


Рис. 16: Задание 16

1. Теоретический вопрос(рис. 17).

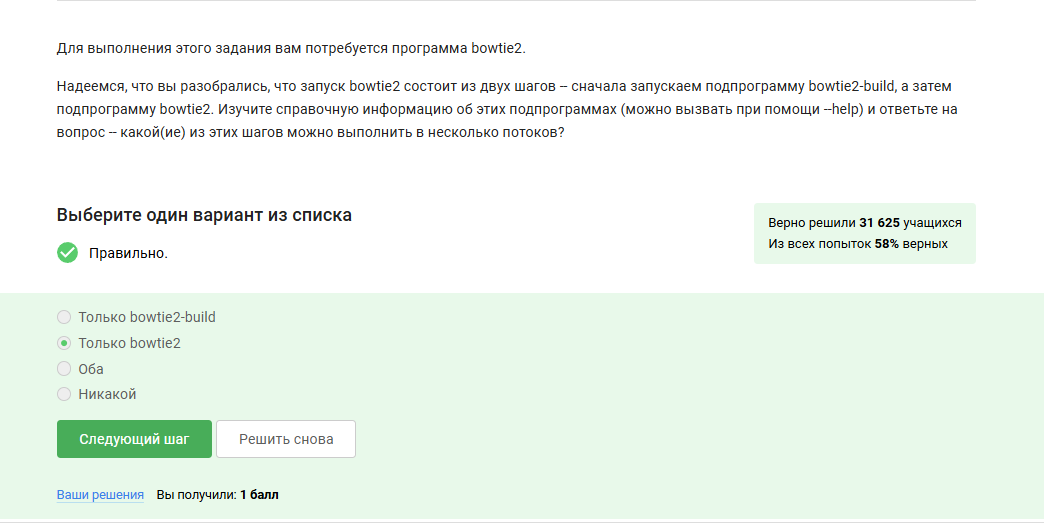


Рис. 17: Задание 17

1. Работа в программе bowtie2

(рис. 18).

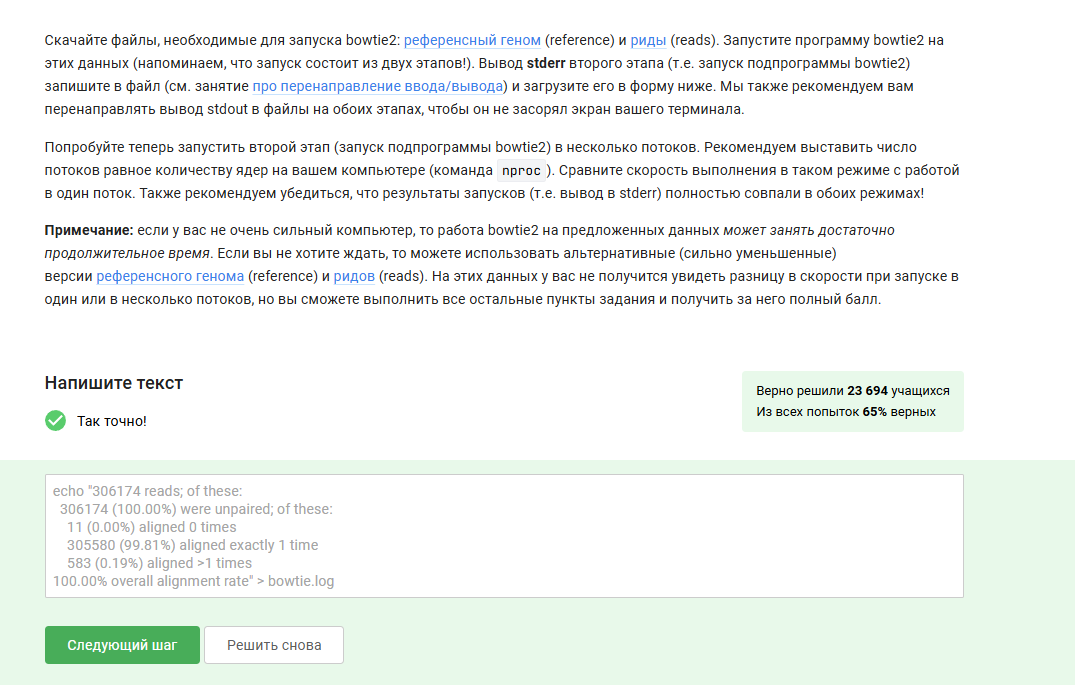


Рис. 18: Задание 18

1. Теоретический вопрос(рис. 19).

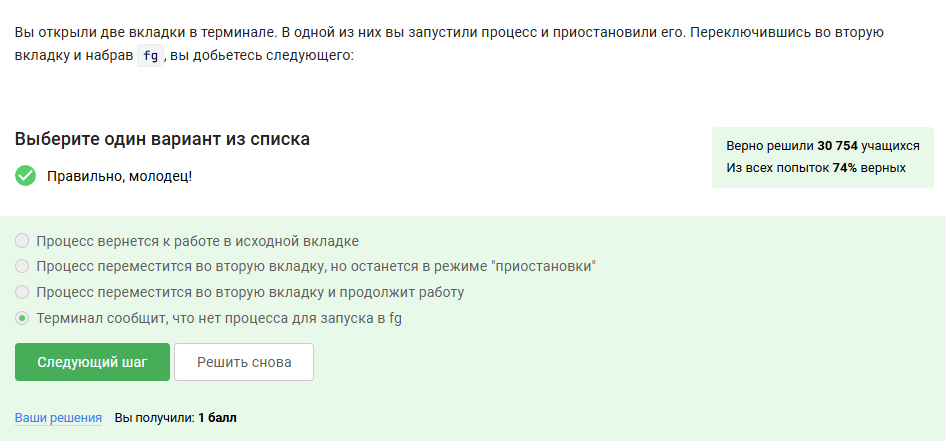


Рис. 19: Задание 19

1. Теоретический вопрос(рис. 20).

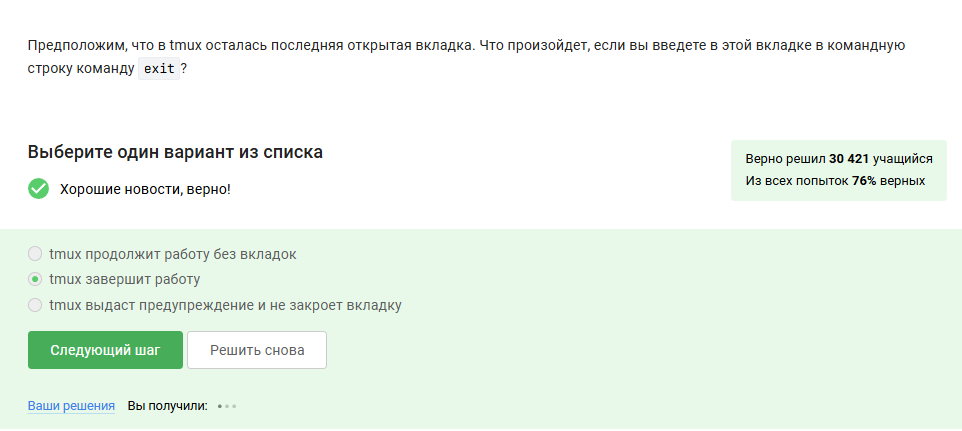


Рис. 20: Задание 20

1. Закрытие терминала не закрывает tmux. Tmux продолжит работать в фоновом режиме.(рис. 21).

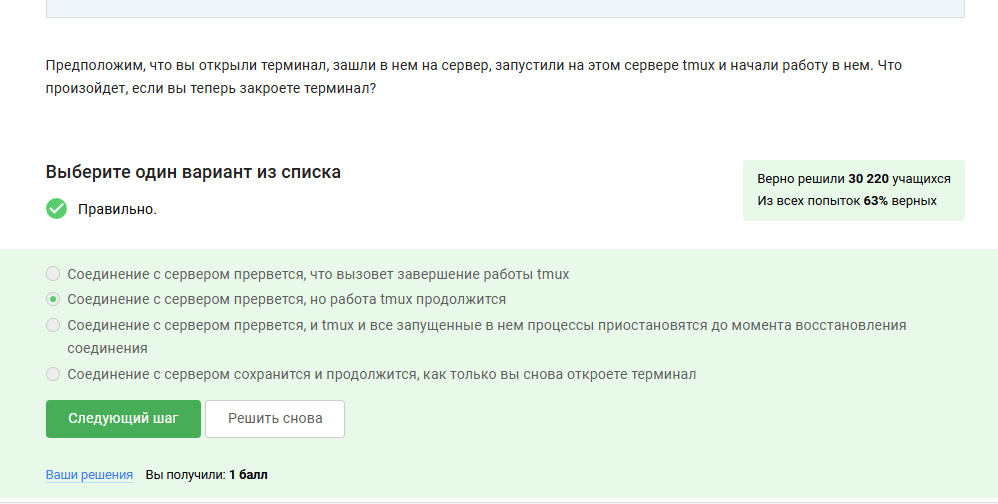


Рис. 21: Задание 21

1. В tmux каждая вкладка (или окно) — это отдельная сессия терминала. Когда вы закрываете вкладку (kill-window или через X), по умолчанию все процессы, запущенные внутри этой вкладки, получают сигнал завершения (SIGHUP или другой сигнал), и обычно завершаются. Если процесс был запущен в фоновом режиме (например, с помощью &), он всё равно связан с текущей сессией окна и завершится при закрытии окна, если не был отделён.(рис. 22).

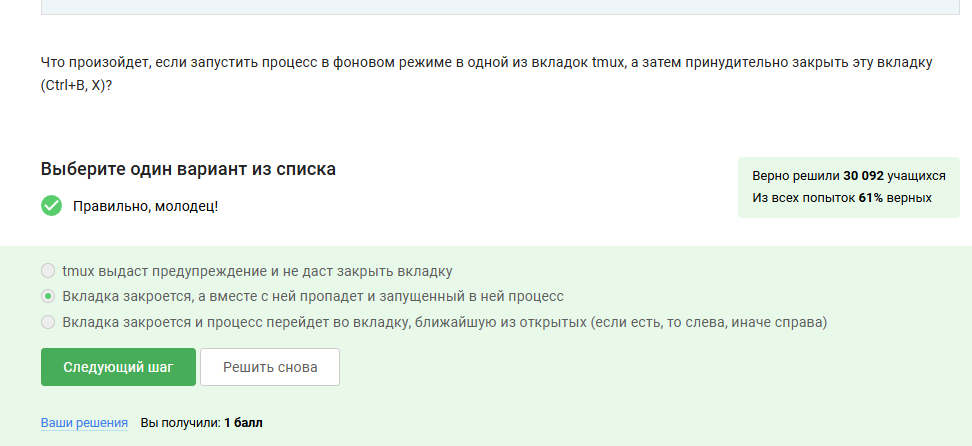


Рис. 22: Задание 22

1. Теоретический вопрос(рис. 23).

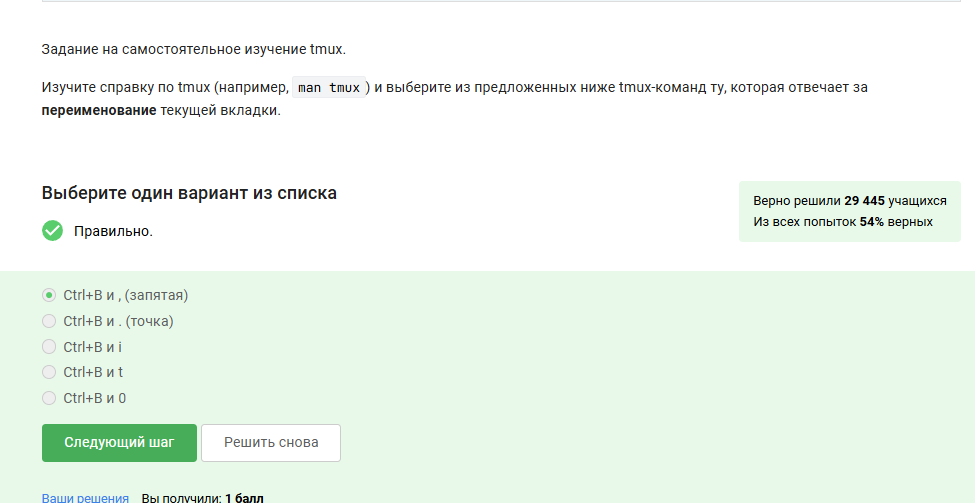


Рис. 23: Задание 23

1. Много теоретический аспектов(рис. 24).

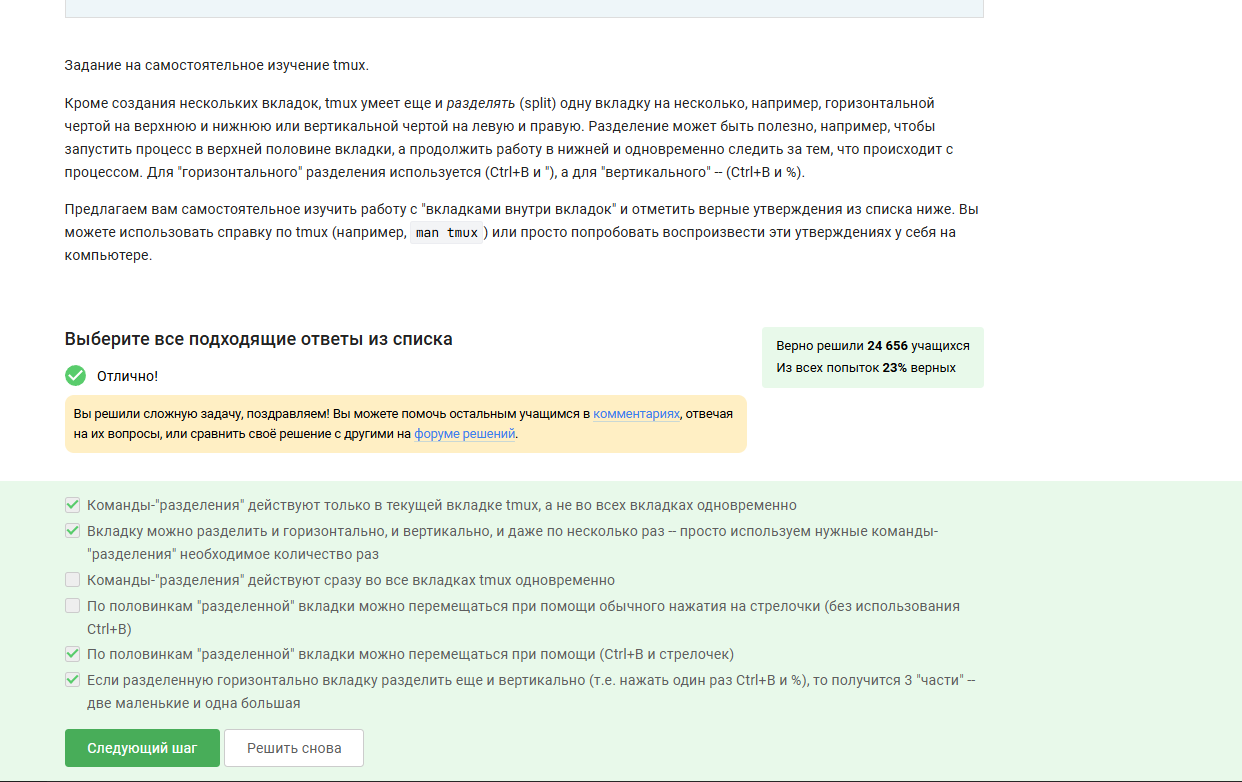


Рис. 24: Задание 24

# 3 Выводы

Были получены знания о Линуксе: серверах, запуске и работе приложений (bowtie2), потоках. Были выполнены тесты.