Отчёт по прохождению 2 этапа внешнего курса

Работа с серверами

Терентеевская Светлана Александровна

2026-05-10

Содержание I

1. Информация
2. Вводная часть
3. Прохождение внешнего курса
4. Выводы
5. Список литературы

Раздел 1

Информация

Докладчик

- Терентеевская Светлана Александровна

Докладчик

- Терентеевская Светлана Александровна
- Группа НКАбд-05-25, студент бакалавра

Докладчик

- Терентеевская Светлана Александровна
- Группа НКАбд-05-25, студент бакалавра
- Российский университет дружбы народов им. П. Лумумбы

Докладчик

- Терентеевская Светлана Александровна
- Группа НКАбд-05-25, студент бакалавра
- Российский университет дружбы народов им. П. Лумумбы
- 1032253498@rudn.ru

Докладчик

- Терентеевская Светлана Александровна
- Группа НКАбд-05-25, студент бакалавра
- Российский университет дружбы народов им. П. Лумумбы
- 1032253498@rudn.ru
- https://github.com/SvetlanaTerenteevskaya

Раздел 2

Вводная часть

Цель работы

Пройти 2 этапа курса «Введение в Linux» на платформе Stepik и ознакомиться с функционалом Linux.

Задание

Просмотреть видео и на основе полученной информации пройти тестовые и интерактивнные задания.

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.

Раздел 3

Прохождение внешнего курса

Прохождение внешнего курса

Удаленный сервер - это компьютер, который находится в дата-центре, а доступ к нему можно получить через Интернет. Он позволяет хранить общедоступные данные, конфиденциальные данные. Также может выполнять затратные по времени и памяти вычисления и хранить большие объемы данных (рис. 1).

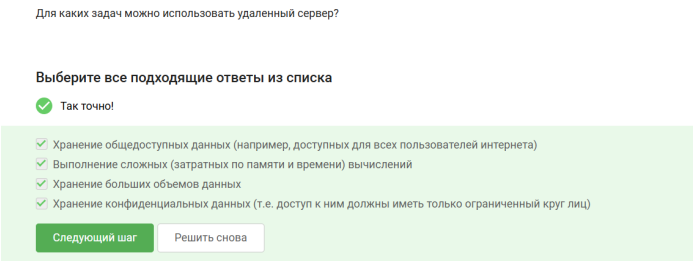

Рисунок 1: 1 задание

Только id_rsa.pub, потому что он является открытым (рис. 2).

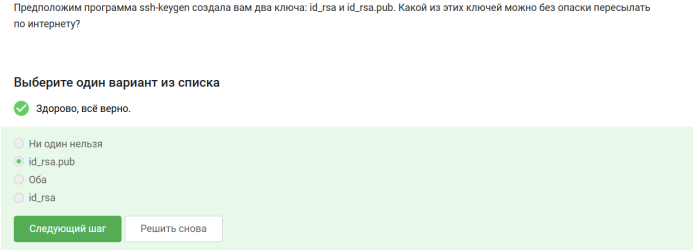

Рисунок 2: 2 задание

-r рекурсивно копирует все папки и подпапки (рис. 3).

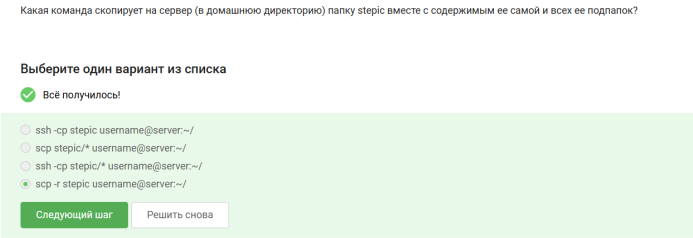

Рисунок 3: 3 задание

Необходимо проверить интернет соединения, чтоб понять, может ли устройство соединиться с сервером, а затем проверить, знает ли оно вообще о существовании такой программы (рис. 4).

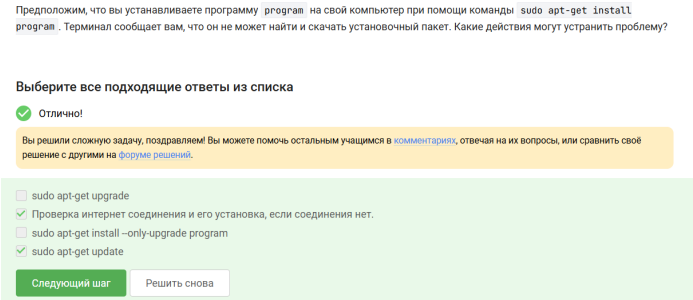

Рисунок 4: 4 задание

Filezilla - FTP/SFTP-клиент и его главная задача - передавать файлы в обе стороны, также он показывает удаленную файловую систему как обычный файловый менеджер и работает только с файлами, а не с выполнением программ на сервере (рис. 5).

Рисунок 5: 5 задание

Запуск программы на своем компьютере не решит проблему «запуск программы на сервере», поэтому необходимо настроить сервер, чтоб он поддерживал вывод информации на экран компьютера и проверить, есть ли другая версия этой программы (рис. 6).

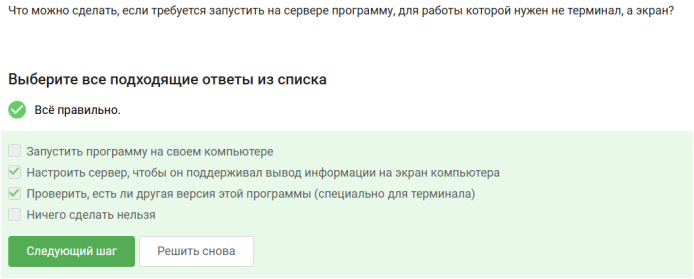

Рисунок 6: 6 задание

Все варинаты кроме program ?! верны. Не существует такой команды program ?! (рис. 7).

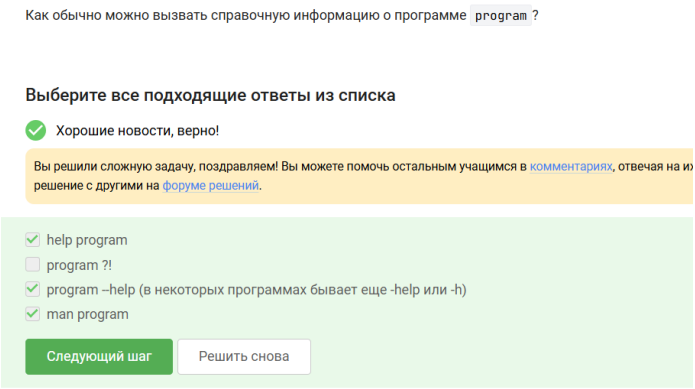

Рисунок 7: 7 задание

Из справки программы FastQC следует, что она принимает форматы: bam, sam, bam_mapped, sam_mapped, fastq. Остальные не поддерживаются, а fastqc-это название программы, а не формат (рис. 8).

Посмотрите справку по программе FastQC (имеется ввиду вариант для запуска в терминале) и определите, **какие форматы данных** он может принимать **на вход**.

Если вы хотите попробовать запустить FastQC на каких-то реальных данных, то можете попробовать на этом файле.

Подсказка: если программы FastQC еще нет на вашем компьютере, то её можно установить командой `sudo apt-get install fastqc` (или в некоторых версиях еще: `bio-linux-fastqc`) или найдя её в Software Center по запросу `fastqc`.
К сожалению, на некоторых дистрибутивах Linux у вас может не получится установить FastQC описанным способом (по ключевым словам `fastqc` и `bio-linux-fastqc` ничего не будет найдено). В этом случае установка будет сложнее, описываем её подробнее.

1. Откройте терминал, попробуйте выполнить команду java. Если получите сообщение, что такая команда не найдена, то переходите к шагу 2, иначе сразу к шагу 3.
2. Вам нужно установить java, например, на Ubuntu это можно сделать с помощью `sudo apt-get install default-jre`.
3. Скачайте и распакуйте архив с FastQC (можно это сделать прямо в терминале с использованием wget и unzip).
4. Файл запуска FastQC называется fastqc и лежит той директории, куда произошла распаковка архива, например, /home/bi/FastQC/fastqc. Перед первым запуском его нужно сделать исполняемым (при помощи chmod +x).
5. Запускать файл fastqc можно как и любую другую программу в терминале (например, через ./fastqc из директории, где он лежит или из любой другой директории задав абсолютный путь до fastqc, см. соответствующее занятие). Если запустить его без параметров, то будет открыта графическая версия программы, а если указать опции или аргументы, например, -help, то будет запущена версия для терминала.

Выберите все подходящие ответы из списка

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.

- [ ] seq
- [x] bam, sam
- [ ] fastqc
- [ ] fasta

Рисунок 8: 8 задание

-infile=test.fasta - указывает входной файл. -align - опция, которая явно запускает множественное выравнивание (рис. 9).

Посмотрите справку по программе (имеется в виду версия для терминала) и **впишите** в поле ниже **команду**, которая запускает в терминале Clustal на файле test.fasta и выполняет *множественное выравнивание* (multiple alignment). Никакие лишние опции указывать не нужно (**только необходимые** для выполнения этого задания)!

Примечание: справку по опциям можно получить при помощи `man` или, если он у вас не работает, то в разделе "**Help for command line parameters**" файла `clustalw_help.txt`, который идет в поставке программы.

Примечание 2: программа Clustal запускает необходимый алгоритм выравнивания по умолчанию (т.е. если ему не указать каких-либо других опций), однако мы просим вас найти и **указать** в команде запуска **опцию**, которая явно говорит Clustal запустить именно множественное выравнивание. После этого вы можете сравнить вывод Clustal при запуске с этой опцией и без нее -- результат должен быть одинаков.

Подсказка: если у вас не установлена программа Clustal, то её можно установить командой `sudo apt-get install clustalw` (или `clustalx`) или найдя её в Software Center по запросу `clustalw` (`clustalx`). Обратите внимание, что на некоторых дистрибутивах доступна только вторая версия программы (например, `clustalw2`), в этом случае можете использовать и её -- все необходимые в задании опции будут точно такими же.

Напишите текст

Хорошая работа.

clustalw-infile=test.fasta -align

Следующий шаг | Решить снова

Рисунок 9: 9 задание

Ctrl+C завершает процесс, а Ctrl+Z приостанавливает (рис. 10).

Предположим вы запустили программы program1, program2 и program3 в фоновом режиме. После этого вы выполнили следующие действия:

`fg %1`
Ctrl+C
`fg %2`
Ctrl+Z
`jobs`

Информация о каких программах будет показана при выполнении команды `jobs` ?

Выберите один вариант из списка

Отлично!

- Только о program1 и program2
- Только о program2 и program3
- Только о program1 и program3
- Обо всех трех

Рисунок 10: 10 задание

ps и top показывают PID, jobs - номер задачи, который не совпадает с PID (рис. 11).

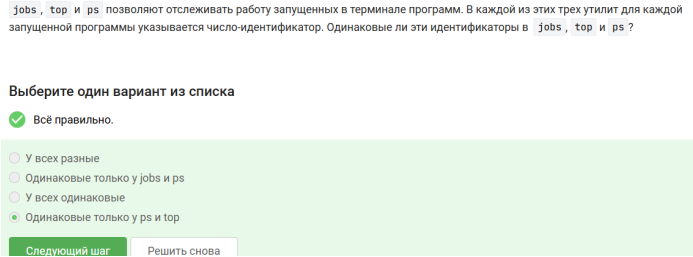

Рисунок 11: 11 задание

При kill -9 ядро мгновенно завершает процесс без возможности игнорирования. kill -18 продолжает остановленный процесс, а просто kill процесс может его перехватить или завершиться «мягко» (рис. 12).

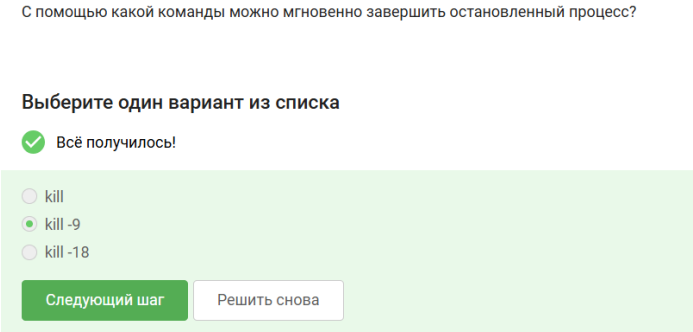

Рисунок 12: 12 задание

Приостановленный процесс не получает сигналы, пока не будет продолжен, после kill процесс останется остановленным. А когда его продолжат он начнет завершаться (рис. 13).

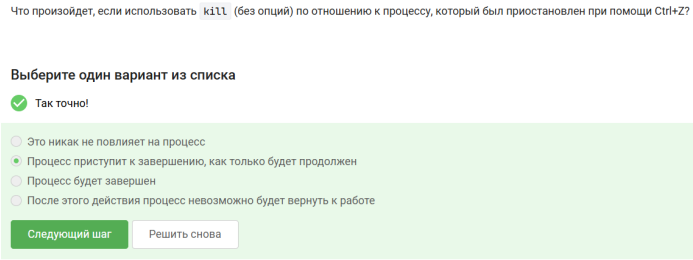

Рисунок 13: 13 задание

Запущенная программа потребляет ресурсы CPU, а остановленная нет (рис. 14).

Сколько вычислительных ресурсов центрального процессора (% CPU) использует остановленное (по Ctrl+Z) многопоточное приложение?
Учитывайте, что 100% CPU означает загрузку одного процессора, 200% CPU – двух процессоров (на *многопроцессорных* и/или *многоядерных* компьютерах) и т.д. Например, *выполняющееся* в 4 потока приложение обычно использует около 400% CPU, однако наш вопрос касается именно момента *после остановки* такого приложения.
Подсказка: если вы не знаете как ответить на этот вопрос, то можете попробовать запустить многопоточное приложение на своем компьютере и посмотреть на результат с помощью команды `top`. Если вы не знаете примеров таких приложений, то рекомендуем вам ненадолго отложить этот шаг и досмотреть занятие до конца. В следующих видеофрагментах и заданиях будет показан пример многопоточного приложения (программы bowtie2). Тестовые данные для запуска этой программы можно найти в последнем задании этого урока.

Подсказка 2: подробнее почитать о значении всей информации, которую выводит `top` на экран, можно по ссылке http://rus-linux.net/MyLDP/consol/komanda-top-v-linux.html

Выберите один вариант из списка

Здорово, всё верно.

- 100% CPU
- В два раза меньше, чем использовалось до остановки
- Столько, сколько использовалось до остановки
- 0% CPU

Следующий шаг | Решить снова

Рисунок 14: 14 задание

Приостановленное приложение не выполняет новых действий, поэтому не занимает вычислительные ресурсы компьютера (CPU 0%). При этом, в оперативной памяти оно сохранится, поэтому оно будет занимать столько же оперативной памяти, сколько до постановки на паузу (рис. 15).

Сколько памяти занимает остановленное (по Ctrl+Z) многопоточное приложение?

Подсказка: если вы не знаете как ответить на этот вопрос, то можете попробовать запустить многопоточное приложение на своем компьютере и посмотреть на результат с помощью команды `top`. Если вы не знаете примеров таких приложений, то рекомендуем вам ненадолго отложить этот шаг и досмотреть занятие до конца. В следующих видеофрагментах и заданиях будет показан пример многопоточного приложения (программы bowtie2). Тестовые данные для запуска этой программы можно найти в последнем задании этого урока.

Подсказка 2: подробнее почитать о значении всей информации, которую выводит `top` на экран, можно по ссылке http://rus-linux.net/MyLDP/consol/komanda-top-v-linux.html

Выберите один вариант из списка

Здорово, всё верно.

- По 64 KB на каждый поток
- Нисколько
- 64 KB
- Столько, сколько оно потребляло в момент остановки

Следующий шаг | Решить снова

Рисунок 15: 15 задание

Завершить один из потоков запущенного многопоточного приложения никак нельзя (рис. 16).

Как принудительно завершить один из потоков запущенного многопоточного приложения?

Подсказка: если вы не знаете как ответить на этот вопрос, то можете попробовать запустить многопоточное приложение на своем компьютере и посмотреть на результат. Если вы не знаете примеров таких приложений, то рекомендуем вам ненадолго отложить этот шаг и досмотреть занятие до конца. В следующих видеофрагментах и заданиях будет показан пример многопоточного приложения (программы bowtie2). Тестовые данные для запуска этой программы можно найти в последнем задании этого урока.

Выберите все подходящие ответы из списка

Правильно, молодец!

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.

- [ ] Командой threadkill
- [ ] Командой kill --thread
- [ ] Сочетанием клавиш Ctrl+C
- [x] Никак

Следующий шаг | Решить снова

Рисунок 16: 16 задание

Только bowtie2 можно выполнить в несколько потоков (рис. 17).

Для выполнения этого задания вам потребуется программа bowtie2.

Надеемся, что вы разобрались, что запуск bowtie2 состоит из двух шагов -- сначала запускаем подпрограмму bowtie2-build, а затем подпрограмму bowtie2. Изучите справочную информацию об этих подпрограммах (можно вызвать при помощи --help) и ответьте на вопрос -- какой(ие) из этих шагов можно выполнить в несколько потоков?

Выберите один вариант из списка

Правильно.

- (•) Только bowtie2
- ( ) Только bowtie2-build
- ( ) Оба
- ( ) Никакой

Следующий шаг | Решить снова

Рисунок 17: 17 задание

Я скачала необходимые для запуска bowtie2 файлы, запустила программу bowtie (рис. 18).

```
saterenteevskaya@saterenteevskaya:~$ gunzip reads.fastq.gz

gzip: reads.fastq.gz: not in gzip format
saterenteevskaya@saterenteevskaya:~$ file reads.fastq.gz
reads.fastq.gz: ASCII text
saterenteevskaya@saterenteevskaya:~$ mv reads.fastq.gz reads.fastq
saterenteevskaya@saterenteevskaya:~$ head reads.fastq
@EAS20_8_6_1_3_1914/1
CGCGTAACAAAAGTGTCTATAATCACGGCAGAAAAGTCCACATTGATTATTTGCACGGCGTCACACTTTGCTATGCCATAGCATTTTTATCCATAAGATT
+
HHHHHHHHHFHGGHHHHHHHHHHHHHHHHHHHHHEHHHHHHHHHHHHHHGHHHGHHHGHIHHHHHHHHHHHHHHHGCHHHHFHHHHHHHGGGCFHBFBCCF
@EAS20_8_6_1_4_703/1
GAGTAAAATTCGGAAAACATAAAGATACGGGTTCTGATAAAAACAGAATCCATTGCCGTGGAATGAGTCATTTACCACAGCATAATAATCCATAATGTTT
+
HHHHHHEHHHHHHEHHHHHHHAHHHEHHFGHGHEHHGHGHEH@EEA?HHFCHGEHEEEGGG;GGGCHGGGEGGEGIIGD?EFGE@GGCG1EDEG=GEEGE
@EAS20_8_6_1_5_1457/1
AATAATCGCCCTATAAACGGGTAATTATACTGACACGGGCGAAGGGGAATTTCCTCTCCGCCCGTGCATTCATCTAGGGGCAATTTAAAAAAGATGGCTA
saterenteevskaya@saterenteevskaya:~$ bowtie2-build reference.fasta reference
```

Рисунок 18: 18 задание

Записала данные в файл (рис. 19).

```
Headers:
    len: 100000
    bwtLen: 100001
    sz: 25000
    bwtSz: 25001
    lineRate: 6
    offRate: 4
    offMask: 0xfffffff0
    ftabChars: 10
    eftabLen: 20
    eftabSz: 80
    ftabLen: 1048577
    ftabSz: 4194308
    offsLen: 6251
    offsSz: 25004
    lineSz: 64
    sideSz: 64
    sideBwtSz: 48
    sideBwtLen: 192
    numSides: 521
    numLines: 521
    ebwtTotLen: 33344
    ebwtTotSz: 33344
    color: 0
    reverse: 1
Total time for backward call to driver() for mirror index: 00:00:00
Renaming reference_index.3.bt2.tmp to reference_index.3.bt2
Renaming reference_index.4.bt2.tmp to reference_index.4.bt2
Renaming reference_index.1.bt2.tmp to reference_index.1.bt2
Renaming reference_index.2.bt2.tmp to reference_index.2.bt2
Renaming reference_index.rev.1.bt2.tmp to reference_index.rev.1.bt2
Renaming reference_index.rev.2.bt2.tmp to reference_index.rev.2.bt2
saterenteevskaya@saterenteevskaya:~$ bowtie2 -x reference_index -U reads.fastq 2> bowtie2.stderr.txt > bowtie.stdout.txt
saterenteevskaya@saterenteevskaya:~$ ls -l bowtie2.stderr.txt
-rw-r--r--. 1 saterenteevskaya saterenteevskaya 206 апр 27 15:58 bowtie2.stderr.txt
saterenteevskaya@saterenteevskaya:~$
```

Рисунок 19: 18 задание

Вставила в форму на степике (рис. 20).

Скачайте файлы, необходимые для запуска bowtie2: референсный геном (reference) и риды (reads). Запустите программу bowtie2 на этих данных (напоминаем, что запуск состоит из двух этапов!). Вывод **stderr** второго этапа (т.е. запуск подпрограммы bowtie2) запишите в файл (см. занятие про перенаправление ввода/вывода) и загрузите его в форму ниже. Мы также рекомендуем вам перенаправлять вывод stdout в файлы на обоих этапах, чтобы он не засорял экран вашего терминала.

Попробуйте теперь запустить второй этап (запуск подпрограммы bowtie2) в несколько потоков. Рекомендуем выставить число потоков равное количеству ядер на вашем компьютере (команда `nproc`). Сравните скорость выполнения в таком режиме с работой в один поток. Также рекомендуем убедиться, что результаты запусков (т.е. вывод в stderr) полностью совпали в обоих режимах!

Примечание: если у вас не очень сильный компьютер, то работа bowtie2 на предложенных данных *может занять достаточно продолжительное время*. Если вы не хотите ждать, то можете использовать альтернативные (сильно уменьшенные) версии референсного генома (reference) и ридов (reads). На этих данных у вас не получится увидеть разницу в скорости при запуске в один или в несколько потоков, но вы сможете выполнить все остальные пункты задания и получить за него полный балл.

Напишите текст

Верно. Так держать!

bowtie2.stderr.txt (206 bytes)

Следующий шаг | Решить снова

Рисунок 20: 18 задание

fg работает только в той оболочке, где процесс был запущен и остановлен. Во второй вкладке - другой экземпляр оболочки, у него нет доступа к списку задач первой вкладки, поэтому терминал сообщит, что нет процесса для fg (рис. 21).

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

Выберите один вариант из списка

Абсолютно точно.

- 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
- Процесс вернется к работе в исходной вкладке

Следующий шаг | Решить снова

Рисунок 21: 19 задание

exit завершает работу tmux (рис. 22).

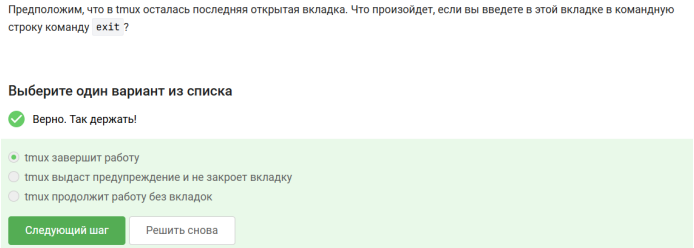

Рисунок 22: 20 задание

Мы заходили на сервер с терминала, который и закрыли, а tmux будет продолжать свою работу на сервере (рис. 23).

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?

Выберите один вариант из списка

Так точно!

- Соединение с сервером сохранится и продолжится, как только вы снова откроете терминал
- Соединение с сервером прервется, и tmux и все запущенные в нем процессы приостановятся до момента восстановления соединения
- Соединение с сервером прервется, что вызовет завершение работы tmux
- Соединение с сервером прервется, но работа tmux продолжится

Следующий шаг | Решить снова

Рисунок 23: 21 задание

Ещё будет предупреждение о том, что работа завершится. Запущенный процесс во вкладке, конечно же, при её закрытии, пропадёт (рис. 24).

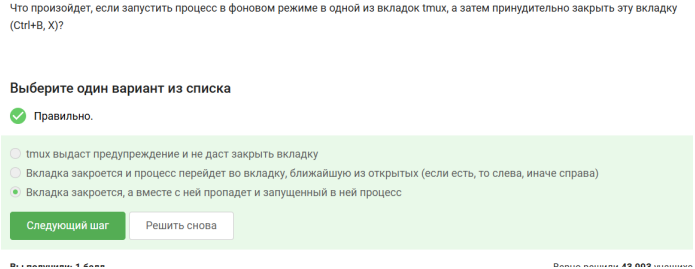

Рисунок 24: 22 задание

За переименование текущей вкладки отвечает Ctrl+B и запятая (рис. 25).

Задание на самостоятельное изучение tmux.

Изучите справку по tmux (например, `man tmux`) и выберите из предложенных ниже tmux-команд ту, которая отвечает за **переименование** текущей вкладки.

Выберите один вариант из списка

Здорово, всё верно.

- Ctrl+B и r
- Ctrl+B и . (точка)
- Ctrl+B и 0
- Ctrl+B и i
- Ctrl+B и , (запятая)

Следующий шаг | Решить снова

Рисунок 25: 23 задание

Ctrl+b c - создать новое окно;
Ctrl+b w - выбрать окно из списка;
Ctrl+b 0-9 - открыть окно по его номеру;
Ctrl+b , - переименовать текущее окно;
Ctrl+b % - разделить текущую панель по горизонтали;
Ctrl+b ” - разделить текущую панель по вертикали;
Ctrl+b стрелка - перейти на панель, находящуюся в стороне, куда указывает стрелка;
Ctrl+b Ctrl+стрелка - изменить размер текущей панели;
Ctrl+b o - перейти на следующую панель;
Ctrl+b ; - переключаться между текущей и предыдущей панелью;
Ctrl+b x - закрыть текущую панель;
Ctrl+b [- войти в режим копирования (подробнее ниже);
Ctrl+b] - вставить из внутреннего буфера обмена tmux;
Ctrl+b d - отключится от текущей сессии;
Ctrl+b : - открыть командную строку.

Можно закрыть одно из делений вкладки выполнив команды Ctrl+B и X, по половинам “разделенной” вкладки можно перемещаться при помощи Ctrl+B и стрелок - как описано в задании выше, делить экран можно только в текущей вкладке tmux, а не во всех вкладках одновременно (рис. 26).

Предлагаем вам самостоятельное изучить работу с "вкладками внутри вкладок" и отметить верные утверждения из списка ниже. Вы можете использовать справку по tmux (например, `man tmux`) или просто попробовать воспроизвести эти утверждениях у себя на компьютере.

Выберите все подходящие ответы из списка

Верно.

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.

- [ ] Вкладку можно разделить только горизонтально или только вертикально, а на попытку ввести вторую команду-"разделения" она реагировать уже не будет
- [x] Можно закрыть одну из "частей" вкладки выполнив (Ctrl+B и x)
- [ ] По половинкам "разделенной" вкладки можно перемещаться при помощи обычного нажатия на стрелочки (без использования Ctrl+B)
- [x] Команды-"разделения" действуют только в текущей вкладке tmux, а не во всех вкладках одновременно
- [ ] Если набрать в одной из "частей" вкладки команду exit, то вся вкладка закроется
- [ ] Если разделенную горизонтально вкладку разделить еще и вертикально (т.е. нажать один раз Ctrl+B и %), то получится 4 одинаковые "части"

Следующий шаг | Решить снова

Рисунок 26: 24 задание

Раздел 4

Выводы

Выводы

Я прошла 2 этап курса «Введение в Linux» на платформе Stepik ознакомилась с функционалом операционной системы Linux.

Раздел 5

Список литературы

Список литературы

Введение в Linux