

Отчёт по лабораторной работе №4

Операционные системы

Лисенков Егор Романович

Содержание

Цель работы	1
Начало работы	1
1 раздел	1
2 РАЗДЕЛ	14
3 РАЗДЕЛ	27
Выводы	52

Цель работы

Освоить основы операционной системы Linux и приобрести навыки, необходимые для эффективной работы с ней.

Начало работы

1 раздел

Обозначил, что понимаю суть и название курса.

The screenshot shows a Stepik course interface. The left sidebar lists sections: '1 Введение' (1 Introduction) and '2 Работа на сервере' (2 Work on the server). The main content area shows '1.1 Общая информация о курсе' (1.1 General information about the course). It includes a progress bar (2/125), a summary (7 из 7 шагов проходимо / 2 из 2 баллов получено), and a 'Задачи' (Tasks) section. A note states: 'Важной частью курса является закрепление изученного материала через решение задач. И именно по результатам решения задач вам будет поставлена оценка за курс.' Below this, it says: 'Все задачи можно решать любое количество раз. За неверные попытки баллы не снижаются, не бойтесь ошибаться! Также, все ваши прошлые решения остаются доступны по ссылке под полем задачи.' A question asks: 'Вопрос: как называется этот курс? Чтобы ответить, выберите правильный ответ нажмите на зелёную кнопку ниже.' A green button says 'Отличное решение!' (Great answer!). A statistics box shows: 'Всего решены 120 822 учащихся' (120,822 students solved) and 'Из всех попыток 97% верных' (97% of attempts were correct). The bottom navigation buttons are 'Следующий шаг' (Next step) and 'Решить снова' (Solve again).

Понял правила курса.

Критерии прохождения курса по Linux

Рядом с каждым тестом и задачей указано количество баллов, которое вы получите за правильное решение. Ваш общий прогресс также отображается в оглавлении курса, там же видны и сроки сдачи каждой из задач.

Суммарная стоимость всех задач составляет 140 баллов.

Для получения сертификата по курсу необходимо набрать 115 баллов, для сертификата с отличием — 130 баллов. Когда Вы наберёте 115 баллов, в течение суток Вам придет сертификат, а если будете решать задания и получать баллы дальше — он будет автоматически обновляться.

Внимание: **дедлайн по этому курсу нет**, то есть вы можете просматривать материалы и решать задания в удобном для вас режиме. Но если вы действительно хотите пройти этот курс, советуем вам заниматься регулярно, и проходить хотя бы по несколько уроков в день. Мотивировать себя на это вам поможет следующий шаг:

Удач!

И пожалуйста, отметьте ниже **ВСЕ** верные утверждения.

Выберите все подходящие ответы из списка

Так точно!

Верно решили 107 011 учеников
Из всех попыток 56% верных

Дедлайн по курсу нет, но я стараюсь проходить уроки регулярно, чтобы изучить Linux

За каждую неверную попытку снимаются 1 балл, но баллы не могут стать меньше 0

Для получения баллов по курсу задачи нужно сдавать до дедлайнов

Я не буду распространять и выкладывать в открытом доступе свои решения задач курса, чтобы другим становилось интересно их решать самостоятельно

Я буду работать над задачами курса самостоятельно, чтобы извлечь для себя максимальную пользу от курса.

Указал свой выбор.

1.2 Как установить Linux 5 из 10 шагов пройдено 1 из 3 баллов получено

Какую операционную систему вы обычно используете? В таких типах заданий (с галочками/чекбоксами/checkbox) вы можете выбирать несколько вариантов ответа (от 0 до всех!)

Выберите все подходящие ответы из списка

Здорово, все верно.

Верно решили 95 278 учеников
Из всех попыток 100% верных

Linux

OS X

Windows

Другое

Следующий шаг Решить снова

Ваш результат Вы получили 1 балл из 1

1513 102 Шаг 6 Следующий шаг >

79 Комментарии 11 Решений Самые популярные

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

Оставить комментарий Показать обсуждение (79)

Знаю что такое виртуальная машина.

1.2 Как установить Linux 6 из 10 шагов пройдено 2 из 3 баллов получено

Что такое виртуальная машина? Выберите наиболее подходящий ответ! В таком типе заданий (с радиокнопками/radio button) ответ всегда ровно один!

Выберите один вариант из списка

Отлично!

Верно решили 91 808 учеников
Из всех попыток 97% верных

Специальная программа для запуска одной ОС на другой ОС

Монитор

Вид операционной системы (ОС)

Автомобиль будущего

Следующий шаг Решить снова

Ваш результат Вы получили 1 балл из 1

1513 102 Шаг 8 Следующий шаг >

29 Комментарии 11 Решений Самые популярные

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

Показал, что я могу запустить Линукс.

The screenshot shows a Stepik course titled "Введение в Linux". The current step is "1.2 Как установить Linux". A question asks if the user has installed Linux on their computer. The user has selected "Да" (Yes). The interface includes a sidebar with navigation links and a footer with statistics.

Для выполнения этого задания я создал документ в OpenOffice/LibreOffice Writer и написал в нём строку “Hello, Linux!” с использованием шрифта FreeMono (или Arial или Times New Roman, если FreeMono не был доступен). Затем я сохранил этот документ в формате XML (Microsoft Word 2003 XML) или в формате FODT (OpenDocument Text: Flat XML), как указано в задании. Это позволяет сохранить структуру документа в виде XML-кода или XML-подобного формата, что обеспечивает совместимость с другими программами и облегчает обмен файлами.

Это задание позволило мне ознакомиться с основными функциями текстового редактора OpenOffice/LibreOffice Writer, а также познакомиться с процессом сохранения документов в различных форматах, в том числе в форматах XML и FODT. Теперь я готов к дальнейшему изучению Linux и использованию его в своих проектах и задачах.

The screenshot shows a Stepik course titled "Введение в Linux". The current step is "1.3 Осваиваем Linux". A task requires writing "Hello, Linux!" in FreeMono font. The text is already entered. The interface includes a sidebar with navigation links and a footer with statistics.

Установочные пакеты в Linux, особенно в дистрибутиве Ubuntu, обычно имеют расширение “.deb”. Это формат пакетов, который используется в Debian и его производных дистрибутивах, включая Ubuntu. Такие пакеты содержат программное обеспечение, а также информацию о зависимостях и инструкции по установке, что делает процесс установки и управления программами на Ubuntu более удобным и эффективным.

The screenshot shows a Stepik course interface for 'Основы Linux'. The current step is '1.3 Основы Linux' (Step 6 of 10). The question asks: 'Какое расширение имеют установленные пакеты в Linux (Ubuntu)?' (What extension do installed packages have in Linux (Ubuntu)?). The options are: ubuntu, deb, txt, exe, and dmg. The user has selected 'ubuntu'. Below the question, it says 'Верно решите 79 897 ученика' (Correctly solved by 79,897 students) and 'Из всех попыток 74% верны' (Of all attempts, 74% are correct). At the bottom, there are buttons for 'Следующий шаг' (Next step) and 'Решить снова' (Solve again).

Вкладка “About” (О программе) в плеере VLC содержит информацию о разработчиках программы. В данном случае, первая фамилия, указанная в списке авторов, - Aimar.

The screenshot shows a Stepik course interface for 'Основы Linux'. The current step is '1.3 Основы Linux' (Step 8 of 10). The question asks: 'Поставьте себе в систему плеер VLC (любым способом: через Software Center или скачиванием установленного пакета с сайта VLC). Запустите, откроется Help –> About (либо Shift+F1) и напишите ниже первое фамилию (без имени) из вкладки Authors. Обратите внимание, что в англоязычных текстах обычно имя стоит на первом месте (first name), а фамилия на втором (last name).'. Below the question, it says 'Напишите текст' (Type text) and 'Верно решите 50 037 ученика' (Correctly solved by 50,037 students) and 'Из всех попыток 62% верны' (Of all attempts, 62% are correct). There is a text input field containing 'Aimar'. At the bottom, there are buttons for 'Следующий шаг' (Next step) and 'Решить снова' (Solve again).

1. Для обновления ссылок в Software Center - НЕТ. Update Manager не предназначен для обновления ссылок в Software Center. Его основная задача - управление обновлениями операционной системы и установленного программного обеспечения.
2. Для установки новых программ - НЕТ. Update Manager не используется для установки новых программ. Его функциональность ограничивается управлением обновлениями.
3. Для обновления установленных программ - ДА. Основная цель Update Manager - обновление уже установленных программ и компонентов операционной системы, обеспечивая пользователям последние версии и исправления для программ и пакетов, установленных на их компьютерах.

1.3 Осваиваем Linux 10 из 10 шагов пройдено 6 из 6 баллов получено

Для чего можно использовать приложение Update Manager?

Выберите все подходящие ответы из списка

Правильно, молодец!

Верно решили 60 759 учащихся
Из всех попыток 37% верных

Для удаления установленных программ
 Для обновления ссылок в Software Center
 Для установки новых программ
 Для обновления всей системы до новой версии
 Для обновления установленных программ

Следующий шаг Решить снова

Ваш результат Вы получите: 1 балл из 1

1323 Шаг 10 Следующий шаг >

27 Комментарии 1 Решений Самые популярные

Будьте вежливы и соблюдайте наши принципы сообщества. Пожалуйста, не оставляйте решения и подсказки в комментариях; для этого есть отдельный форум.

Составить комментарий

Ну тут я использовал знания русского языка.

1.4 Terminal: основы 3 из 13 шагов пройдено 1 из 5 баллов получено

Выберите все синонимы для "командной строки".

Потрясающий ответ!

Верно решили 69 413 учащихся
Из всех попыток 80% верных

Ассоль
 Терминал
 Термин
 Консоль

Следующий шаг Решить снова

Ваш результат Вы получите: 1 балл из 1

2059 Шаг 3 Следующий шаг >

27 Комментарии 1 Решений Самые популярные

Будьте вежливы и соблюдайте наши принципы сообщества. Пожалуйста, не оставляйте решения и подсказки в комментариях; для этого есть отдельный форум.

Составить комментарий

Команда `pwd` используется для вывода текущей рабочей директории в командной строке. Она отображает полный путь к текущей директории, в которой пользователь находится в данный момент. Название команды `pwd` является сокращением от “print working directory” (печать текущей директории). Она показывает, где находится пользователь относительно корневой директории файловой системы.

1.4 Terminal: основы 5 из 13 шагов пройдено 2 из 5 баллов получено

Какая команда напечатает в какой директории мы сейчас находимся?

Выберите один вариант из списка

Хорошие новости, верно!

Верно решили 48 373 учащихся
Из всех попыток 91% верных

Только `pwd`
или `PWD`, но не `Print`
 Только `Print`

Следующий шаг Решить снова

Ваш результат Вы получите: 1 балл из 1

2059 Шаг 5 Следующий шаг >

17 Комментарии 1 Решений Самые популярные

Будьте вежливы и соблюдайте наши принципы сообщества. Пожалуйста, не оставляйте решения и подсказки в комментариях; для этого есть отдельный форум.

Составить комментарий

Да, все предложенные команды являются полностью эквивалентными команде `ls -A --human-readable -l /some/directory`.

1. `ls --almost-all --human-readable -l /some/directory --almost-all` является альтернативным именем для `-A`, и они выполняют одну и ту же функцию, которая показывает все файлы и папки, включая скрытые, кроме ссылок на текущий и родительский каталоги.
2. `ls --human-readable -A -l /some/directory` - порядок опций не важен, поэтому `-A` и `-l` можно расположить в любом порядке.
3. `ls -lAh /some/directory` - опции могут быть объединены вместе без пробелов между ними, поэтому `-lAh` эквивалентно `-l -A -h`.
4. `ls -h -A -l /some/directory` - аналогично предыдущему, но опции перечислены в другом порядке.
5. `ls -Ahl /some/directory` - опции также можно объединить без пробелов в любом порядке, поэтому `-Ahl` эквивалентно `-A -h -l`.

Все эти команды выполнены с одним и тем же набором опций, что и `ls -A --human-readable -l /some/directory`, и поэтому они полностью эквивалентны.

The screenshot shows a Stepik course interface with a terminal window titled "1.4 Terminal: основы". The terminal displays the following text:

```
Укажите, какие из следующих команд полностью эквивалентны команде: ls -A --human-readable -l /some/directory
```

Подсказка: для правильного ответа на этот вопрос вам может потребоваться справка о команде ls. Напоминаем, что её можно получить с помощью команды man.

Подсказка 2: в вопросах с выбором из нескольких вариантов может возникнуть ситуация, когда **все** предложенные варианты являются неверными (варианты каждый раз выбираются случайным образом из большого набора ответов, где есть как верные, так и ложные). В этом случае вы просто не должны **отметить не один** из них (ведь мы просим указывать **только верные варианты!**) и нажать кнопку "Отправить"/"Submit". Возможна и обратная ситуация, т.е. все предложенные варианты верны. В этом случае отмечайте их все и нажимайте "Отправить"/"Submit".

Выберите все поддающиеся ответы из списка

Так точно!

Вы решите стоящую задачу, подозреваем Вы можете помочь остальным учащимся в [комментариях](#), ставя на них звездочки, или спросить свой вспомогательный вопрос [здесь](#).

Вопрос решен: 55 396 учеников
Из всех попыток 1% засчитан

Список вариантов:

- ls -almost-all --human-readable -l /some/directory
- ls -A --human-readable -l /some/directory
- ls -lAh /some/directory
- ls -h -A -l /some/directory
- ls -Ahl /some/directory

[Следующий шаг](#) [Решить задачу](#)

Ваш результат: Вы получите: 1 балл из 1

[Следующий шаг >](#)

1. `ls ~/Downloads` - Выведет содержимое директории Downloads, которая находится в домашней директории пользователя. Символ `~` обозначает домашнюю директорию текущего пользователя.
2. `ls ../Downloads` - Выведет содержимое директории Downloads, которая находится в родительской директории относительно текущей (то есть в директории `/home/bi`).
3. `ls /home/bi/Downloads` - Выведет содержимое директории Downloads, используя абсолютный путь к ней.
4. `ls ../../Downloads` - Выведет содержимое директории Downloads, используя относительный путь, указывая на родительскую директорию относительно текущей, а затем переходя в директорию Downloads.

Каждая из этих команд позволяет вывести содержимое директории Downloads, находясь в `/home/bi/Documents`, при этом не показывая содержимое других директорий.

The screenshot shows a terminal window titled "1.4 Terminal: основы" with the status "1 из 13 шагов пройдено 4 из 5 баллов получено". The question asks to choose the correct answer from a list of four options: "ls -l", "rm -r", "mkdir -r", and "mv". The correct answer, "rm -r", is selected with a radio button. A green box at the top right indicates "Верно решено 54 471 учащихся Из всех попыток 18% верных". Below the list are two buttons: "Следующий шаг" and "Решить снова".

rm - это команда для удаления файлов или директорий.

-r - это опция команды **rm**, которая указывает на рекурсивное удаление. Это означает, что команда **rm** будет удалять все файлы и поддиректории в указанной директории, включая саму эту директорию. Именно благодаря этой опции **rm** способна удалить целые директории и их содержимое.

The screenshot shows a terminal window titled "1.4 Terminal: основы" with the status "13 из 13 шагов пройдено 5 из 5 баллов получено". The question asks which command is used for deleting a directory. The correct answer, "rm -r", is selected with a radio button. A green box at the top right indicates "Верно решили 58 022 учащихся Из всех попыток 83% верных". Below the list are two buttons: "Следующий шаг" and "Решить снова". At the bottom, there are statistics: "2066" users, "451" solutions, and "Шаг 12".

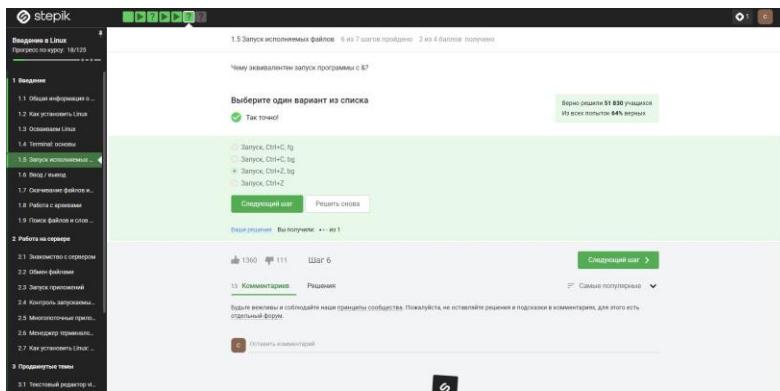
Когда вы вводите команду **firefox** в терминале, он запускает браузер Mozilla Firefox в новом процессе. Затем, когда вы вводите **exit** в этом же терминале, это завершает работу текущего терминального сеанса, но не оказывает влияния на процессы, запущенные из этого сеанса, такие как Firefox. Следовательно, закрытие терминала не приведет к закрытию Firefox.

The screenshot shows a terminal window titled "1.5 Запуск исполняемых файлов" with the status "3 из 7 шагов пройдено 1 из 4 баллов получено". The question asks what happens if you run **firefox** in the terminal and then exit. The correct answer, "Некто не закроется", is selected with a radio button. A green box at the top right indicates "Верно решили 54 317 учащихся Из всех попыток 82% верных". Below the list are two buttons: "Следующий шаг" and "Решить снова". At the bottom, there are statistics: "1360" users, "111" solutions, and "Шаг 3".

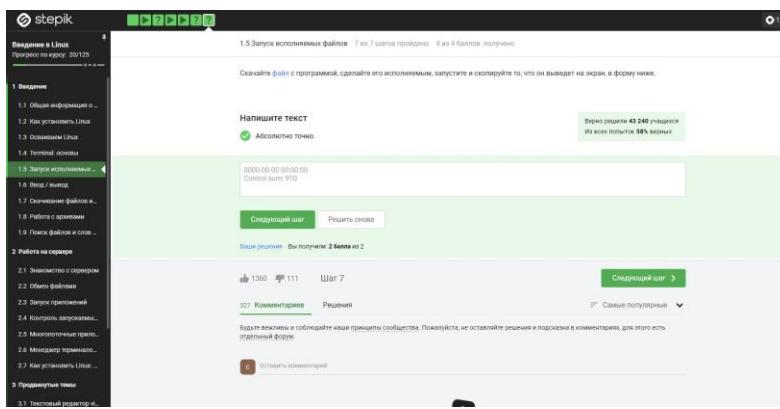
Запуск: После ввода команды с символом &, программа запускается в фоновом режиме, аналогично тому, как если бы она была запущена без блокировки терминала.

Ctrl+Z: Это сочетание клавиш используется для приостановки выполнения текущего процесса в терминале. Если программа была запущена без символа &, то при использовании комбинации клавиш Ctrl+Z выполнение программы приостанавливается. В случае, если программа была запущена с символом &, то процесс можно приостановить, нажав Ctrl+Z.

bg: После приостановки выполнения процесса с помощью Ctrl+Z, команда bg позволяет возобновить выполнение этого процесса в фоновом режиме.



Произвёл запуск и увидел.



Когда программа запускается в терминале, её стандартный вывод (stdout) и поток ошибок (stderr) по умолчанию направляются на терминал. Это означает, что любые сообщения об ошибках, которые генерирует программа, будут выведены на экран, чтобы пользователь мог видеть их и реагировать соответственно.

1. `program 2>> file.txt`: Эта команда направляет поток ошибок программы `program` в файл `file.txt`, причем оператор `>>` используется для добавления вывода в файл, сохраняя при этом уже существующие данные в файле, или создавая новый файл, если он не существует.
2. `program 2> file.txt`: Эта команда также направляет поток ошибок программы `program` в файл `file.txt`, но оператор `>` перенаправляет вывод, полностью перезаписывая файл, если он уже существует, а не добавляя новые данные к уже существующим. Таким образом, при первом выполнении команды файл `file.txt` будет создан и заполнен потоком ошибок программы `program`, но при последующих выполнениях файл будет перезаписываться, а не дополняться.

Когда программы объединяются в конвейер, стандартный вывод (`stdout`) первой программы передается на вход (`stdin`) второй программы. Однако поток ошибок (`stderr`) каждой программы остается незатронутым и продолжает направляться на стандартный вывод терминала. Таким образом, если не производится явное перенаправление потока ошибок в конвейер, сообщения об ошибках будут выводиться на экран.

1.6 Вход / выход. 7 из 7 шагов пройдено. 4 из 4 баллов получено

Куда деваются сообщения об ошибках (т. е. вывод в stderr) от тех программ, которые объединены в конвейер (pipe)?

Подсказка: чтобы узнать правильный ответ, вы можете смоделировать описанную ситуацию. Для этого скопируйте показанную на странице программу `interactor.py`, скопируйте файл `names.txt` со списком имен (запись часть из них с ошибкой, т. е. с маленькой буквой) и пускайте конвейер, аналогичный тому, что был показан на видео. Запустите конвейер и посмотрите на результат! Обратите внимание, что указанная программа работает корректно только с именами, набранными **латинскими буквами**.

Выберите один вариант из списка

Записываются в файл err.txt
 Выводятся на экран
 Записываются в файл pipe_err

Следующий шаг Решить снова

Ваш результат Вы получили: 2 балла из 2

Шаг 7

47 Комментариев 1 Решений

Самые популярные

`cd /home/alex/`: Команда `cd` переходит в директорию `/home/alex/`.

`wget -P /home/alex/Pictures -O 1.jpg http://example.com/example.jpg`: Эта команда скачивает картинку по указанному URL (`http://example.com/example.jpg`) и сохраняет её в директорию `/home/alex/Pictures` под именем `1.jpg`. Однако, поскольку в команде указан флаг `-O 1.jpg`, который задает имя файла, картинка будет сохранена как `1.jpg` в указанной директории.

1.7 Скачивание файлов из интернета. 3 из 7 шагов пройдено. 1 из 4 баллов получено

В каком файле на диске окажется картинка, если для ее скачивания были выполнены следующие команды?

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

Выберите один вариант из списка

Правильно, молодец!

Верно решили 46 350 учащихся
Из всех попыток 36% верных

/home/alex/.jpg
 /home/alex/Pictures/example.jpg
 /home/alex/example.jpg
 /home/alex/Pictures/1.jpg

Следующий шаг Решить снова

Ваш результат Вы получили: 1 балл из 1

Шаг 3

1063 90 Шаг 3

Следующий шаг >

34 Комментариев 1 Решений

Самые популярные

`-q` или `-quiet`: Эти опции подавляют вывод сообщений о выполнении команды `wget`, что означает, что команда будет работать в тихом режиме и не будет выводить никаких лишних сообщений на экран.

1.7 Скачивание файлов из интернета. 5 из 7 шагов пройдено. 3 из 4 баллов получено

Какую опцию нужно указать команде `wget`, чтобы она не выводила никаких сообщений на экран (`Resolving... Connecting to...`)?

Подсказка: для ответа на этот вопрос вам понадобится справка по команде `wget`, которую легко можно получить, набрав `man wget`.

Выберите один вариант из списка

Всё получилось!

Верно решили 45 653 учащихся
Из всех попыток 65% верных

-n или -no-verbose
 -v или -verbose
 -q или -quiet

Следующий шаг Решить снова

Ваш результат Вы получили: 2 балла из 2

Шаг 5

1053 90 Шаг 5

Следующий шаг >

34 Комментариев 1 Решений

Самые популярные

`wget -r`: Эта опция указывает `wget` рекурсивно следовать по ссылкам на web-странице для загрузки связанных файлов.

-l 1: Эта опция ограничивает глубину рекурсии до 1, то есть wget будет загружать только файлы, связанные с начальной страницей.

-A jpg: Эта опция указывает wget загружать только файлы с расширением .jpg.

Таким образом, скачиваются файлы с расширением .jpg и все HTML-файлы на глубине

1. Однако, поскольку wget был запущен с фильтром -A jpg, все скачанные HTML-файлы будут удалены после завершения загрузки, так как они не соответствуют этому фильтру.

1.7 Скачивание файлов из интернета 7 из 7 шагов пройдено 4 из 4 баллов получено
Попытка некоторой веб-страницы есть ссылки на картинки в формате jpg и jpg, а также ссылки на другие страницы сайта (внешние темы файлы). Какие файлы будут скачаны на компьютере, если запустить wget -r -l 1 -A jpg, и передать в качестве аргумента ссылку на эту веб-страницу? Выберите наиболее полный ответ
Подсказка для правильного ответа на этот вопрос: вам может не хватить стартовой информации от команды wget, т.к. в嫂 введенны опции -A, есть некоторые исключения. Рекомендуем посмотреть соответствующий раздел в полном описании утилиты wget на сайте разработчиков

Выберите один вариант из списка
Отличное решение!

Верно решено 43 677 участников
из всех попыток 49% верных

Будут скачаны jpg и html файлы, но все jpg будут удалены
Не будут скачаны ни jpg, ни html картинки, только html файлы
Будут скачаны только jpg файлы
Не будут скачаны ничего
Будут скачаны jpg и html страницы, но все jpg будут удалены
Будут скачаны только картинки (jpg и png), но все jpg будут удалены

Следующий шаг > Решение

Ваш результат: Вы получите 1 балл из 1

Шаг 7

41. Комментарии > Решений Самые популярные

gzip: Это архиватор, который работает с файлами в формате gzip (.gz). Он используется для сжатия файлов, при этом сохраняя только один файл и не сохраняя структуру директорий. По умолчанию, после распаковки архива gzip, сам архив удаляется.

zip: Это архиватор, который работает с файлами в формате zip (.zip). В отличие от gzip, zip сохраняет не только один файл, но и структуру директорий, что позволяет архивировать и извлекать целые папки с файлами. По умолчанию, архив zip не удаляется после его распаковки, оставляя его на месте.

1.8 Работа с архивами 3 из 7 шагов пройдено 1 из 2 баллов получено
Чем отличаются архиваторы gzip и zip?
Примечание: имеется ввиду запуск этих программ с параметрами по умолчанию (без использования дополнительных опций).

Выберите один вариант из списка
Хорошие новости, верно!

Верно решено 44 111 участников
из всех попыток 78% верных

gzip удаляет архив после его распаковки
zip и gzip никак не отличаются
zip удаляет архив после его распаковки
gzip сжимает лучше, чем zip
zip сжимает лучше, чем gzip

Следующий шаг > Решение

Ваш результат: Вы получите: 1 из 1

Шаг 3

29 Комментарии > Решений Самые популярные

zip: Zip поддерживает архивирование файлов и директорий в формате .zip.

tar: Tar может создавать архивы, включающие в себя файлы и директории, сохраняя при этом их иерархию.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The current step is '1.8 Работа с архивами' (Step 8 of 7). The question asks: 'Какие из перечисленных программ-архиваторов могут создать архив из директории с файлами?' (Which of the listed archive programs can create an archive from a directory of files?). The correct answer is 'tar'. The interface includes a sidebar with course navigation, a progress bar, and a footer with social sharing and comment sections.

-с: Создание нового архива.

-j: Использование алгоритма сжатия bzip2 для сжатия файлов.

-v: Отображение подробной информации о процессе создания архива.

-f: Указание имени архива.

This screenshot shows the same Stepik course interface. The current step is '1.8 Работа с архивами' (Step 8 of 7). The question asks: 'Какой набор опций нужно указать программе "tar", чтобы запаковать файлы в "my_archive.tar.bz2"?' (What set of options should be specified to the "tar" program to pack files into "my_archive.tar.bz2"?). The correct answer is '-cvf'. The interface includes a sidebar, a progress bar, and a footer.

Маска *.jpg ищет все файлы, имя которых оканчивается на .jpg, и не будет включать файлы с именем Alexey.jpeg, так как они не соответствуют данному шаблону.

The screenshot shows the Stepik course interface again. The current step is 'Шаг 3 – Поиск файлов и слое в файлах – Stepik'. The question asks: 'Какая маска команды "find" не найдет файл Alexey.jpeg?' (What mask for the "find" command will not find the file "Alexey.jpeg"?). The correct answer is 'Alex*'. The interface includes a sidebar, a progress bar, and a footer.

Ищем по-нижнему регистру

1.9 Поиск файлов и слов в файлах – 5 из 5 шагов проходено – 2 из 4 баллов получено

Предположим, что в файле `text.txt` записаны строки, показанные среди вариантов ответа. Отметьте только те из них, которые выводят на экран команда `grep "world" text.txt`.

Выберите все подходящие ответы из списка

✓ Задорово, все верно.

Вы решите сложную задачу: поделитесь вы можете помочь остальным участникам в [комментариях](#), спечечка на их вопросы, или сравнить свое решение с другими на [другие решения](#).

Верно решено 49 642 участника
Из всех попыток 37% верных

2.1 Элементы с сервером
2.2 Обмен файлами
2.3 Запуск приложений
2.4 Копирование, запускаемы...
2.5 Многоточечные прис...
2.6 Минидиректория...
2.7 Как установить Linux...
3 Программные темы
3.1 Текстовый редактор M...
3.2 Сервисы на Unixе: поч...
3.3 Сервисы на Unixе: веб...
3.4 Сервисы на Unixе: раз...
3.5 Программный поиск ...
3.6 Основы графики в gnu...

Следующий шаг Решить снова

Ваше решение: Вы получите: 1 балла из 1

Следующий шаг >

Программа отработала корректно

1.9 Поиск файлов и слов в файлах – 6 из 6 шагов проходено – 4 из 4 баллов получено

Скачайте [poem.txt](#) с произведениями Шекспира. Вам нужно сконвертировать файл, в котором будут все строки из этих произведений, содержащие "love", и загрузить этот файл в форму.

Пароль: для того, чтобы результаты поиска записались сразу в файл, можно воспользоваться перенаправлением вывода (см. занятие [Ввод/Вывод](#)).

Напишите текст

✓ Верно. Так держать!

Верно решено 34 497 участников
Из всех попыток 57% верных

Следующий шаг Решить снова

Ваше решение: Вы получите: 2 балла из 2

Следующий шаг >

1. Выполнение сложных (затратных по памяти и времени) вычислений:
Удаленные серверы с более мощным аппаратным обеспечением и большими вычислительными ресурсами могут использоваться для выполнения сложных вычислений, которые требуют большого объема памяти и времени.
2. Хранение общедоступных данных (например, доступных для всех пользователей интернета): Удаленные серверы могут использоваться для хранения данных, которые должны быть доступны всем пользователям интернета, например, содержимое веб-сайтов, общедоступные файлы и т. д.
3. Хранение больших объемов данных: Удаленные серверы обычно обладают большим объемом хранилища, что позволяет использовать их для хранения больших объемов данных, например, для архивирования, резервного копирования и обработки данных.
4. Хранение конфиденциальных данных (т.е. доступ к ним должны иметь только ограниченный круг лиц): Удаленные серверы могут быть настроены для хранения конфиденциальных данных, к которым должен иметь доступ только ограниченный круг лиц. Это может включать в себя данные о клиентах, финансовые данные и другую чувствительную информацию.

2 РАЗДЕЛ

The screenshot shows a Stepik course interface. The left sidebar lists topics such as 'Введение в Linux', 'Работа с файлами', 'Работа с архивами', 'Поиск файлов и слов...', 'Работа на сервере', 'Экономство с сервером', 'Обмен файлами', 'Запуск приложений', 'Конфигурации...', 'Менеджер привилегий...', 'Как установить Linux...', 'Практические темы', 'Тестовый редактор...', 'Скрипты на bash: основы...', 'Скрипты на bash: ветвь...', 'Скрипты на bash: циклы...', 'Программный поиск в...'. The main content area displays a question: 'Для каких задач можно использовать удаленный сервер?'. Below it is a list of options: 'Выполнение сложных (затратных по памяти и времени) вычислений', 'Хранение общедоступных данных (например, доступных для всех пользователей интернета)', 'Хранение больших объемов данных', 'Хранение конфиденциальных данных (т.е. доступ к ним должны иметь только ограниченный круг лиц)'. A green checkmark indicates that the last option is correct. At the bottom, there are buttons for 'Следующий шаг' and 'Решите снова'. A progress bar shows 1142 из 133 шагов.

Ключ `id_rsa.pub` является открытым ключом, который используется для проверки подписи, а не для создания подписи. Это означает, что он не содержит приватной информации и может быть безопасно распространен и использован для аутентификации на удаленных серверах.

The screenshot shows a Stepik course interface. The left sidebar lists topics such as 'Введение в Linux', 'Работа с файлами', 'Работа с архивами', 'Поиск файлов и слов...', 'Работа на сервере', 'Экономство с сервером', 'Обмен файлами', 'Запуск приложений', 'Конфигурации...', 'Менеджер привилегий...', 'Как установить Linux...', 'Практические темы', 'Тестовый редактор...', 'Скрипты на bash: основы...', 'Скрипты на bash: ветвь...', 'Скрипты на bash: циклы...', 'Программный поиск в...'. The main content area displays a question: 'Предположим программа ssh-keygen создала вам два ключа: id_rsa и id_rsa.pub. Какой из этих ключей можно без опаски пересыпалть по интернету?'. Below it is a list of options: 'Оба', 'id_rsa', 'id_rsa.pub', 'Ни один нельзя'. A green checkmark indicates that 'id_rsa.pub' is the correct answer. At the bottom, there are buttons for 'Следующий шаг' and 'Решите снова'. A progress bar shows 1142 из 133 шагов.

- **scp:** Эта команда используется для копирования файлов между компьютерами по SSH.
- **-r:** Этот флаг указывает `scp` на рекурсивное копирование, что означает, что она будет копировать папку `stepic` вместе с ее содержимым и всеми подпапками.
- **stepic:** Это имя папки, которую нужно скопировать.
- **username@server:** Это имя пользователя и адрес сервера, куда нужно скопировать файлы.
- **~/:** Это указывает на домашнюю директорию пользователя на сервере, куда будут скопированы файлы.

2.2 Обмен файлами 4 из 8 шагов пройдено 1 из 3 баллов получен

Какая команда скопирует на сервер (в домашнюю директорию) папку stepic вместе с содержимым ее самой и всех ее подпапок?

Работа на сервере Выберите один вариант из списка

Отлично!

ssh -cp stepic username@server:~/

scp -r stepic username@server:~/

ssh -cp stepic/* username@server:~/

scp stepic/* username@server:~/

Верно решили 38 014 участника Из всех попыток 57% верны

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

- Проверка интернет-соединения и его установка, если соединения нет:** Первым шагом следует убедиться, что компьютер подключен к интернету. Если интернет-соединение отсутствует, необходимо его настроить или восстановить.
- Обновление списка пакетов:** Команда `sudo apt-get update` обновляет список доступных пакетов из источников АРТ. Это может помочь в случае, если информация о доступных пакетах устарела или была изменена.

2.2 Обмен файлами Выберите все подходящие ответы из списка

Предположим, что вы устанавливаете программу `program` на свой компьютер при помощи команды `sudo apt-get install program`. Терминал сообщает вам, что он не может найти и скачать установочный пакет. Какие действия могут устранить проблему?

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 36 659 участников Из всех попыток 21% верны

Проверка интернет соединения и его установка, если соединения нет.

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install --only-upgrade program

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

1042 195 Шаг 6 Следующий шаг

- Для копирования файлов со своего компьютера на сервер:** Filezilla предоставляет пользователю возможность загружать файлы с локального компьютера на удаленный сервер.

2. **Для копирования файлов с сервера на свой компьютер:** Также Filezilla позволяет пользователю скачивать файлы с удаленного сервера на локальный компьютер.

Filezilla также обеспечивает возможность просмотра содержимого директорий как на локальном компьютере, так и на удаленном сервере, но основное её назначение - управление файлами и их передача между компьютером пользователя и сервером.

The screenshot shows a Stepik course interface. The left sidebar lists chapters: 'Введение в Linux' (40/125), '1 Ввод / вывод', '1.7 Скачивание файлов и...', '1.8 Работа с архивами', '1.9 Поиск файлов и слов...', '2 Работа на сервере' (selected), '2.1 Знакомство с сервером', '2.2 Обмен файлами' (selected), '2.3 Запуск приложений', '2.4 Контроль запускаемых...', '2.5 Многопоточные прило...', '2.6 Менеджер терминал...', '2.7 Как установить Linux...'. The main content area shows a question: 'Для чего можно использовать программу Filezilla?'. Below it is a list of options: 'Для просмотра содержимого директорий на своем компьютере' (checked), 'Для копирования файлов со своего компьютера на сервер' (checked), 'Для установки программ на сервер' (unchecked), 'Для копирования файлов с сервера на свой компьютер' (checked), 'Для запуска программ на сервере' (unchecked). A green button 'Хорошая работа.' is visible. At the bottom, there are buttons 'Следующий шаг' and 'Решить снова'. On the right, a box says 'Верно решили 36 387 уча...' and 'Из всех попыток 49% вер...'. The footer shows 'Ваши решения' with 'Вы получили: ••• из 1', '1042', '195', 'Шаг 8', and another 'Следующий шаг' button.

1. **Настроить сервер, чтобы он поддерживал вывод информации на экран компьютера:** Это может быть реализовано различными способами, включая использование удаленного рабочего стола или графического сервера (например, X11), который позволит программе выводить графический интерфейс на удаленном компьютере.
2. **Проверить, есть ли другая версия этой программы (специально для терминала):** Возможно, для программы существует версия, предназначенная для работы в терминале (CLI), которая может быть запущена на сервере без необходимости графического интерфейса. Если такая версия существует, её можно использовать для выполнения задач на сервере.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 1, 2, and 3. Chapter 2, 'Работа на сервере', is selected. Step 2.3, 'Запуск приложений', is completed with 4/8 steps passed and 1/7 points earned. A question asks what can be done if a program needs a terminal to run. The correct answer is 'Настроить сервер, чтобы он поддерживал вывод информации на экран компьютера'. The user's response was marked as correct. Below, a list of options includes 'man program' and 'help program'. The user has 866 likes and 661 dislikes.

1. **man program:** Команда `man` (от “manual”) предназначена для отображения справочной информации о программе в терминале. После выполнения команды `man program` будет отображена справочная страница, содержащая описание программы, её опций и примеры использования.
2. **help program:** Эта команда предназначена для отображения встроенной справочной информации о командах в командной оболочке (например, в `bash`). Однако, она может быть не доступна для всех программ, а только для команд, встроенных в оболочку.
3. **program -help:** Многие программы поддерживают опцию `--help` для вывода краткой справочной информации о программе и её опциях. В некоторых случаях также могут использоваться варианты опций, такие как `-help`, `-h`, `-N` и т. д.

The screenshot shows the same Stepik course interface. Step 2.3 is now 6/8 steps passed and 2/7 points earned. A question asks how to get help for a program. The correct answer is 'man program'. The user's response was marked as correct. Below, a list of options includes 'man program', 'help program', and 'program -help'. The user has 866 likes and 661 dislikes.

Для определения форматов данных, которые программа FastQC может принимать на вход, следует обратиться к её справке. Для этого выполним команду `./fastqc --help` из каталога, в котором находится исполняемый файл fastqc.

По результатам обзора справки, форматы данных, которые программа FastQC может принимать на вход, не указаны напрямую. FastQC предназначен для анализа качества последовательностей, полученных из различных источников, и обычно принимает файлы в форматах FASTQ и BAM. Однако, без доступа к точным данным из справки программы, нельзя дать точный ответ на вопрос о поддерживаемых форматах данных.

The screenshot shows a Stepik course interface with a sidebar containing a navigation menu and a main content area. The main content area displays a question from a course titled 'Введение в Linux'. The question asks: 'Посмотрите справку по программе FastQC (имеется вида варант для запуска в терминале) и определите, какие форматы данных он может принимать на вход.' Below the question is a detailed answer in Russian, which includes instructions for installing FastQC on Ubuntu, details about its command-line interface, and specific steps for running it with different file types like FASTQ and BAM. At the bottom of the answer, there is a list of correct answers: 'bam_mapped, sam_mapped', 'fastq', 'fasta', and 'seq'. A green button at the bottom right says 'Следующий шаг' (Next step).

clustalw: имя исполняемого файла Clustal для версии терминала.

test.fasta: имя входного файла с последовательностями для выравнивания.

-align: опция, указывающая Clustal на выполнение множественного выравнивания.

The screenshot shows a Stepik course interface with a sidebar containing a navigation menu and a main content area. The main content area displays a question from a course titled 'Введение в Linux'. The question asks: 'Clustal – это одна из самых широко используемых компьютерных программ для множественного выравнивания нуклеотидных и аминокислотных последовательностей (multiple sequence alignment). У нее есть графическая версия ClustalX и версия для запуска в терминале ClustalW. Вы можете потренироваться запускать его с использованием файла test.fasta.' Below the question is a detailed answer in Russian, which provides information about ClustalW, its graphical counterpart ClustalX, and how to run it with the specified input file. It also includes a note about the '-align' option. At the bottom of the answer, there is a text input field containing the command 'clustalw test.fasta -align'. A green button at the bottom right says 'Напишите текст' (Write text).

Команда `jobs` отображает список заданий, запущенных в фоновом режиме в текущем сеансе терминала. После выполнения предложенной последовательности команд:

1. `fg %1`: Эта команда возобновляет выполнение задания №1 в переднем плане (foreground), то есть программа program1. Однако, мы немедленно прерываем выполнение этой программы с помощью `Ctrl+C`.

2. `fg %2`: Затем команда пытается возобновить выполнение задания №2 в переднем плане, то есть `program2`. После этого выполнения этого задания мы приостанавливаем его с помощью `Ctrl+Z`.

Таким образом, после выполнения этих команд, информация о программах `program1` и `program2` будет потеряна, а в списке `jobs` останется только информация о `program3`.

4.4 Контроль запускаемых программ 5 из 11 шагов пройдено 1 из 4 баллов получен

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

Абсолютно точно.

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

Верно решил 34 021 учащегося
Из всех попыток 61% верны

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

812 лайков 59 неудачных Шаг 5 Следующий шаг

Идентификаторы, которые указываются в утилитах `jobs`, `top` и `ps`, имеют разное назначение и принцип работы:

1. **jobs**: Эта команда отображает список заданий, запущенных в текущем сеансе терминала. Идентификаторы в `jobs` представляют собой порядковые номера заданий в рамках текущего сеанса терминала. Они могут быть не уникальными и могут изменяться при завершении или приостановке заданий.
2. **top**: Эта утилита отображает список процессов, запущенных в системе. Идентификаторы процессов в `top` являются уникальными идентификаторами (PID - Process ID), присвоенными операционной системой каждому процессу. Они уникальны в пределах всей системы и не изменяются в течение жизненного цикла процесса.
3. **ps**: Утилита `ps` также отображает список процессов, запущенных в системе. Идентификаторы процессов в `ps` также являются уникальными идентификаторами (PID), присвоенными операционной системой каждому процессу. Они уникальны в пределах всей системы и не изменяются в течение жизненного цикла процесса.

Таким образом, идентификаторы в jobs не совпадают с идентификаторами в top и ps, потому что jobs отображает информацию о заданиях в терминале, в то время как top и ps показывают информацию о процессах в системе.

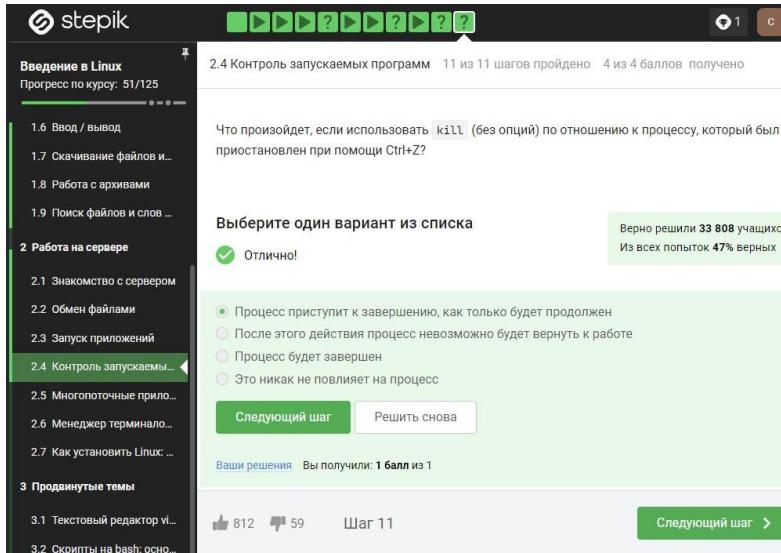
The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 1.6 through 3.2. Chapter 2.4, titled 'Контроль запускаемых программ', is selected. The main content area displays a question: 'jobs, top и ps позволяют отслеживать работу запущенных в терминале программ. В каждой из этих трех утилит для каждой запущенной программы указывается число-идентификатор. Одинаковые ли эти идентификаторы в jobs, top и ps?' Below the question is a list of four options with radio buttons. A green checkmark indicates the correct answer: 'Одноковые только у ps и top'. A green button at the bottom right says 'Следующий шаг'.

Команда kill используется для отправки сигналов процессам в UNIX-подобных системах, включая Linux. Сигнал с номером 9, известный как SIGKILL, используется для немедленного завершения процесса. Когда процесс получает сигнал SIGKILL, он не имеет возможности обработать его или выполнить какие-либо дополнительные операции завершения, что приводит к немедленному завершению процесса.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 1.6 through 3.2. Chapter 2.4, titled 'Контроль запускаемых программ', is selected. The main content area displays a question: 'С помощью какой команды можно мгновенно завершить остановленный процесс?' Below the question is a list of three options with radio buttons. A green checkmark indicates the correct answer: 'kill -9'. A green button at the bottom right says 'Следующий шаг'.

Когда процесс приостанавливается с помощью комбинации клавиш Ctrl+Z, он становится в состояние приостановки (Suspended). Если после этого на него будет применена команда kill без каких-либо опций, то процесс приступит к завершению,

как только будет возобновлен, т.е. когда вы введете команду `fg` или `bg`, чтобы возобновить выполнение процесса.



2.4 Контроль запускаемых программ 11 из 11 шагов пройдено 4 из 4 баллов получено

Что произойдет, если использовать `kill` (без опций) по отношению к процессу, который был приостановлен при помощи `Ctrl+Z`?

Выберите один вариант из списка

Верно решили 33 808 учащихся
Из всех попыток 47% верных

Отлично!

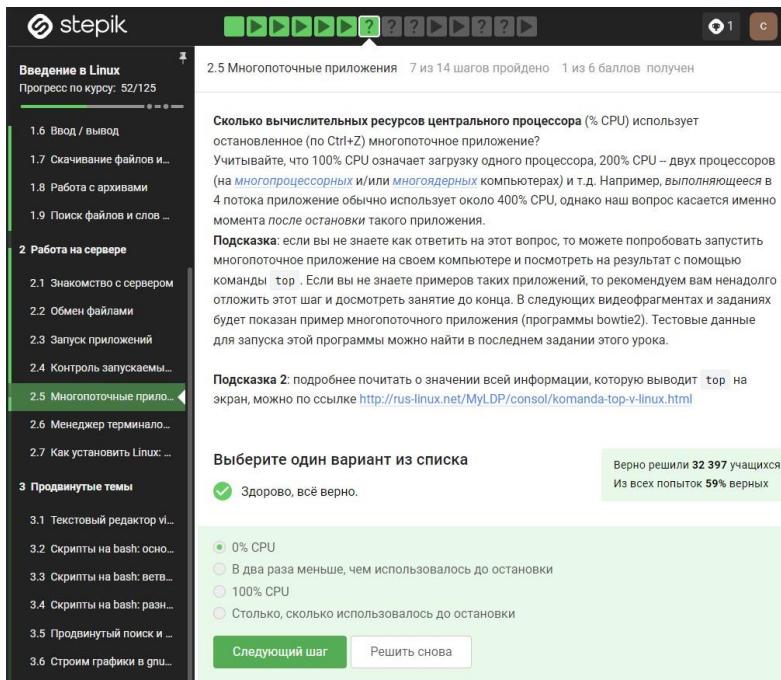
Процесс приступит к завершению, как только будет продолжен
 После этого действия процесс невозможно будет вернуть к работе
 Процесс будет завершен
 Это никак не повлияет на процесс

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

812 59 Шаг 11 Следующий шаг >

Когда многопоточное приложение приостанавливается посредством `Ctrl+Z`, все его потоки также останавливаются, что приводит к тому, что они больше не используют процессорное время. Поэтому процент использования CPU для этого приложения будет равен 0%.



2.5 Многопоточные приложения 7 из 14 шагов пройдено 1 из 6 баллов получен

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 `bwtie2`). 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

Верно решили 32 397 учащихся
Из всех попыток 59% верных

Здорово, всё верно.

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

Следующий шаг Решить снова

Приложение, остановленное посредством `Ctrl+Z`, не освобождает память, которую использовало в момент приостановки. Таким образом, количество занимаемой памяти останется таким же, как в момент остановки приложения.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 1.6 through 3.6. Chapter 2.5, 'Многопоточные приложения', is selected. The main content area displays a question: 'Сколько памяти занимает остановленное (по Ctrl+Z) многопоточное приложение?' with a detailed hint about using the 'top' command. Below is a multiple-choice question with one correct answer checked: 'Столько, сколько оно потребляло в момент остановки'. A green button 'Следующий шаг' is at the bottom.

Если многопоточное приложение запущено, то завершить только один из его потоков принудительно довольно сложно. Это связано с тем, что потоки взаимодействуют с общими ресурсами и данными, и завершение одного потока может нарушить работу приложения в целом. Обычно потоки в рамках приложения завершаются вместе с его завершением. Если однако потребуется остановить выполнение приложения, то можно воспользоваться командой `kill` с соответствующим идентификатором процесса.

The screenshot shows the same Stepik course interface. Chapter 2.5 is still selected. The question now asks how to forcefully terminate one thread of a multi-threaded application. It includes a hint about using the 'top' command. Below is a multiple-choice question with one correct answer checked: 'Никак'. A yellow box highlights a note: 'Вы решили сложную задачу, поздравляем! Вы можете помочь остальным ученикам в комментариях, отвечая на их вопросы, или сравнить свое решение с другими на форуме решений.' A green button 'Следующий шаг' is at the bottom.

Из двух подпрограмм bowtie2 и bowtie2-build, только bowtie2 позволяет выполняться в несколько потоков одновременно. Программа bowtie2-build, используемая для предварительной индексации референсного генома, обычно выполняется последовательно и не поддерживает многопоточность.

2.5 Многопоточные приложения 12 из 14 шагов пройдено 4 из 6 баллов получено

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

Верно решили 31 625 учащихся
Из всех попыток 58% верных

Правильно, молодец!

Только bowtie2-build
Оба
Никакой
Только bowtie2

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

795 451 Шаг 12 Следующий шаг >

Это сообщение представляет собой результат работы программы bowtie2 и содержит информацию о выравнивании прочтений на референсный геном. Вот что означают эти цифры:

- “306174 reads” - всего обработано 306174 прочтений.
- “306174 (100.00%) were unpaired” - все прочтения были неспаренными.
- “11 (0.00%) aligned 0 times” - 11 прочтений не были выровнены ни на один раз.
- “305580 (99.81%) aligned exactly 1 time” - 305580 прочтений были выровнены ровно один раз.
- “583 (0.19%) aligned >1 times” - 583 прочтения были выровнены более одного раза.
- “100.00% overall alignment rate” - общая степень выравнивания составляет 100%.

Эти данные предоставляют информацию о качестве и количестве выравнивания прочтений на геном.

2.5 Многопоточные приложения 13 из 14 шагов пройдено 6 из 6 баллов получено

Скачайте файлы, необходимые для запуска `boinc2`: референсный геном (`reference`) и риды (`reads`). Запустите программу `boinc2` на этих данных (например, если запуск состоит из двух этапов). Вызовите `stffer` второго этапа (т.е. запуск подпрограммы `boinc2`) запишите в файл (см. заметку про перенаправление входа/выхода) и загрузите его в форму ниже. Мы также рекомендуем вам перенаправлять вывод stdout в файлы на оба этапа, чтобы он не заисрел экран вашего терминала.

Порядок работы: запустите второй этап (запуск подпрограммы `boinc2`) в несколько потоков. Рекомендуем выставить число потоков равное количеству ядер на вашем компьютере (команда `lscpu`, пункт 1). Оставьте скорость выполнения в таком режиме с работой в один поток. Также рекомендуем убедиться, что результаты запусков (т.е. вывод в `stffer`) полностью совпадут в обоих режимах!

Примечание: если у вас не очень сильный компьютер, то работа `boinc2` на предоставленных данных может занять достаточно продолжительное время. Если вы не хотите ждать, то можете использовать альтернативные (хотя и уменьшенные) версии референсного генома (`reference`) и риды (`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
Все попытки!

Верно решил 23 494 ученика
Из всех попыток 69% верных

Следующий шаг Решить снова

Ваше решение Вы получили 2 балла из 2

Когда вы запускаете команду `fg` во второй вкладке терминала, она пытается вернуть в активное состояние процесс, который был приостановлен в первой вкладке. Однако, так как вторая вкладка не имеет доступа к процессам, созданным в первой вкладке, терминал выдаст сообщение об ошибке, указывая на отсутствие процесса для возобновления.

Шаг 5 - Менеджер терминалов tmux - Stepik

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
Хорошая работа.

Верно решил 30 754 ученика
Из всех попыток 74% верных

Следующий шаг Решить снова

Ваше решение Вы получили 1 балл из 1

1011 112 Шаг 5

13 Комментарии 2 Решения

Самые популярные

Когда вы вводите команду `exit` в последней открытой вкладке tmux, это приведет к завершению сеанса tmux, так как больше не останется активных окон.

2.6 Менеджер терминалов tmux 15 из 19 шагов пройдено 2 из 7 баллов получено

Предположим, что в tmux осталась последняя открытая вкладка. Что произойдет, если вы введите в этой вкладке в командную строку команду `exit ?`

Выберите один вариант из списка
Отличное решение!

Верно решил 30 421 ученик
Из всех попыток 76% верных

tmux продолжит работу без вкладок
 tmux выдаст предупреждение и не закроет вкладку
 tmux завершит работу

Следующий шаг Решить снова

Ваше решение Вы получили 1 балл из 1

1011 112 Шаг 10

Следующий шаг >

Когда вы закрываете терминал, это приведет к разрыву соединения с сервером, но работа внутри tmux на сервере будет продолжена, поскольку tmux работает в фоновом режиме и сохраняет состояние сессии даже после разрыва соединения.

The screenshot shows a Stepik course interface for 'Введение в Linux'. The sidebar lists chapters 1.6 through 3.3. The main content area is titled '2.6 Менеджер терминалов tmux' with a progress bar at 16 из 19 шагов пройдено. A question asks: 'Предположим, что вы открыли терминал, зашли в него на сервер, запустили на этом сервере tmux и начали работу в нем. Что произойдет, если вы теперь закроете терминал?'. Below it, a list of options is shown, with the fourth one selected: 'Соединение с сервером прервется, но работа tmux продолжится'. A green button 'Следующий шаг' is visible.

Когда вы закрываете вкладку в tmux комбинацией клавиш **Ctrl+B, X**, tmux закрывает эту вкладку и все процессы, запущенные в ней, завершаются.

This screenshot shows the same Stepik course interface as the previous one, but the current step is '2.6 Менеджер терминалов tmux'. The question asks: 'Что произойдет, если запустить процесс в фоновом режиме в одной из вкладок tmux, а затем принудительно закрыть эту вкладку (Ctrl+B, X)?'. The correct answer is 'Вкладка закроется и процесс перейдет во вкладку, ближайшую из открытых (если есть, то слева, иначе справа)'. A green button 'Следующий шаг' is visible.

Ctrl+B и , - это комбинация клавиш для переименования текущей вкладки в tmux.

2.6 Менеджер терминалов tmux 18 из 19 шагов пройдено 5 из 7 баллов получено

Ввод / вывод

7 Старт на самостоятельное изучение tmux.
4 из 4 баллов получено

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

Всё правильно.

Верно решили 29 445 учащихся
Из всех попыток 54% верных

Ctrl+B и t
Ctrl+B и , (запятая)
Ctrl+B и g
Ctrl+B и ~ (тильда)
Ctrl+B и . (точка)

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

1011 112 Шаг 18

Следующий шаг >

23 Комментария 4 Решения Самые популярные

- Вкладку можно разделить и горизонтально, и вертикально:** tmux позволяет разделять вкладки как по горизонтали, так и по вертикали, что дает возможность создавать различные макеты для параллельной работы.
- Можно закрыть одну из “частей” вкладки:** tmux позволяет закрывать “части” вкладок, созданные при разделении, чтобы освободить место для других процессов или визуализации.
- Команды-“разделения” действуют только в текущей вкладке tmux:** Каждая команда, связанная с разделением вкладок, применяется только к текущей активной вкладке tmux, что позволяет контролировать разделение и компоновку вкладок в каждом tmux-окне отдельно.
- Если разделенную горизонтально вкладку разделить еще и вертикально, то получится 3 “части”:** tmux позволяет комбинировать разделение по вертикали и по горизонтали для создания более сложных макетов, включая три или более “части” в одной вкладке.

Неправильное утверждение о перемещении по половинкам “разделенной” вкладки с помощью стрелок не верно, потому что в tmux для перемещения между “панелями” или “частями” вкладки используются другие комбинации клавиш, а не стрелки.

stekip

Введение в Linux
Прогресс по курсу: 64/125

1. Ввод / вывод
1.1 Сканирование файлов и...
1.2 Работа с архивами
1.3 Поиск файлов и слов...
2 Работа на сервере
2.1 Экономист с сервером
2.2 Обмен файлами
2.3 Запуск приложений
2.4 Контроль затратами...
2.5 Мониторинг пропуск...
2.6 Менеджер терминала
2.7 Как установить Linux...
3 Практические темы
3.1 Текстовый редактор vi...
3.2 Скрипты на bash: основы...
3.3 Основы на bash: сетев...

запустить процесс в виртуальном подынамике вкладки, а продолжить работу в нихнем и одновременно следить за тем, что происходит с процессом. Для "горизонтального" разделения используется (Ctrl+H-1), а для "вертикального" – (Ctrl+H-5).

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

Выберите все подходящие ответы из списка

Верно.

Вы решите 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 **24 656** учащихся
Из всех попыток **23%** верных

✓ По подынамикам "разделенные" вкладки можно перемещаться при помощи (Ctrl+B в стрелочках)
✓ Вкладку можно разделять только горизонтально или только вертикально, на попытку ввести вторую команду "разделения" она реагирует уже не будет
✓ Вкладку можно разделять и горизонтально, и вертикально, и даже на несколько раз – просто используем нужные команды "разделения" необязательное количество раз
✓ Можно закрыть одну из "частей" вкладки выполнив (Ctrl+N в х)
✓ Команда "разделения" действует только в текущей вкладке тих, а не во всех вкладках одновременно
✓ Если разделенную горизонтально вкладку разделить еще и вертикально ("и: нажать один раз Ctrl+H-5"), то получится 3 "части" две маленькие и одна большая

[Следующий шаг](#) [Решить снова](#)

3 РАЗДЕЛ

В редакторе Vim команда для выхода из редактора - это :q. Если вы внесли изменения в файл и хотите выйти без сохранения, вы можете использовать команду :q!. После ввода команды нужно нажать клавишу Enter.

**stepik**





Введение в Linux
Прогресс по курсу: 65/125

1.6 Ввод / вывод
1.7 Скачивание файлов и...
1.8 Работа с архивами
1.9 Поиск файлов и слов ...
2 Работа на сервере
2.1 Знакомство с сервером
2.2 Обмен файлами
2.3 Запуск приложений
2.4 Контроль запускаемы...
2.5 Многоточечные прило...
2.6 Менеджер терминало...
2.7 Как установить Linux: ...
3 Продвинутые темы
3.1 Текстовый редактор vi...◀
3.2 Скрипты на bash: осно...
3.3 Скрипты на bash: ветв...

3.1 Текстовый редактор vim 5 из 12 шагов пройдено 1 из 7 баллов получен

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

 Абсолютно точно.

 ":"; затем "q"
 ":"; затем "q", затем "Enter"
 "Q"
 "Ctrl", затем "x"
 "Esc"

Следующий шаг **Решить снова**

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

 1121  432 Шаг 5 **Следующий шаг >**

23 **Комментария** 4 **Решения** **= Самые популярные** ▾

- Разница между **w** и **W** в том, что **w** перемещает курсор по “словам” (word), разделенным пробелами или другими символами пунктуации, а **W** перемещает курсор по “большим словам” (WORD), разделенным пробелами. Таким образом, **W** игнорирует все символы, кроме пробелов.
 - После 10 нажатий на **W** курсор действительно окажется на том же месте, где был бы после 10 нажатий на **w**, если строки не содержат “больших слов”.

3.1 Текстовый редактор vim 7 из 12 шагов проходено 2 из 7 баллов получено

При перемещении в vim "то слово" 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, 2x? YES!`

Выберите все подходящие ответы из списка

Хорошая работа.

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](#).

В этой строке 12 слов (word)

Чтобы попасть в конец строки, нужно однократное нажатие на W, что на W

Чтобы попасть в конец строки, нужно совершить меньший нажатий на W, чем на w

В этой строке 9 "больших слов" (WORD)

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

Чтобы попасть в конец строки, нужно совершить больше нажатий на W, чем на w

Следующий шаг **Решить снова**

- В первом наборе клавиш `d2wwywPp` сначала удаляются два слова (`two` и `three`), затем курсор перемещается к слову “`four`”, оно копируется (`yw`), а затем вставляется дважды (`Pp`).
- Во втором наборе клавиш `d2w$$bifour four <<Esc>>` сначала удаляются два слова (`one` и `two`), затем курсор перемещается к концу строки (\$\$), после чего вставляется “`four four four`” перед словом “`five`”.

one two three four five
и вам нужно преобразовать её в строку
three four four four five

Какие(ой) из предложенных ниже наборов нажатий клавиш выполнят такое редактирование? В этих наборах нажатие на клавишу Esc обозначается как <Esc> (т.е. знаки "<" и ">" не несут отдельного смысла).

Выберите все подходящие ответы из списка

Так точно!

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](#).

d2wwywPp

d2wifour four <<Esc>>

d2dwywPp

ddthree four four five<<Esc>>

xxxxxxxwywPp

d2w\$bfifour four <<Esc>>

Следующий шаг **Решить снова**

Вашим решением было получено 1 балл из 1

1127 433 Шаг 9 Следующий шаг >

- % означает, что замена будет производиться во всем файле.
- `s/Windows/Linux/` означает замену строки “`Windows`” на “`Linux`”.
- `g` означает, что замена будет производиться глобально в каждой строке, то есть все вхождения “`Windows`” в каждой строке будут заменены на “`Linux`”, а не только первое.

3.1 Текстовый редактор vim 11 из 12 шагов пройдено 5 из 7 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 "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:

Верно решил 24 631 учащийся
Из всех попыток 57% верных

:%s/Windows/Linux

Следующий шаг Решить снова

Ваше решение Вы получили: 2 балла из 2

1127 433 Шаг 11

72 Комментария 10 Решений

Самые популярные

Когда вы находитесь в нормальном режиме Vim и нажимаете клавишу "v", вы входите в режим выделения (Visual). В этом режиме вы можете выделять текст, используя клавиши перемещения или другие команды, такие как W, e, \$. Выделенный текст будет подсвечен, и внизу редактора будет отображаться надпись "VISUAL", указывающая на то, что вы находитесь в режиме выделения.

3.1 Текстовый редактор vim 12 из 12 шагов пройдено 7 из 7 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 viimtutor, 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 viimtutor 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
✓ Правильно, молодец!

Верно решили 23 497 учащихся
Из всех попыток 29% верных

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](#).

Режим выделения открывается из любого другого режима по нажатию "v"
 Режим выделения открывается из нормального режима по нажатию "V"
 Чтобы выйти из режима выделения, нужно ввести :q
 В режиме выделения можно использовать команды перемещения (например, W, e, \$, и др.)
 Режим выделения открывается при помощи команды :visual
 Когда вы находитесь в режиме выделения, внизу редактора горит надпись – VISUAL – (или – ВИЗУАЛЬНЫЙ РЕЖИМ –)

Следующий шаг Решить снова

Ваше решение Вы получили: -- из 2

1127 433 Шаг 12

Следующий шаг >

Когда вы используете стрелки вверх/вниз для перемещения по истории набранных команд в терминале, это обычно означает, что вы перемещаетесь по истории команд текущей оболочки. В вашем случае, когда вы запускаете оболочку sh из bash, а затем запускаете bash из sh, и набираете команды в каждой из этих оболочек, истории этих оболочек будут независимыми.

Таким образом, когда вы попытаетесь использовать стрелки вверх/вниз в последней оболочке (bash), вы будете перемещаться только по истории команд этой оболочки, то есть по набору C.

3.2 Скрипты на bash: основы 3 из 10 шагов пройдено 1 из 6 баллов получен

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?

Выберите один вариант из списка

✓ Всё правильно.

Верно решили 30 266 участников
Из всех попыток 65% верных

Только из набора B
 Только из набора C
 Из наборов B и C
 Никакие команды появляться не будут
 Только из набора A

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

1152 144 Шаг 3

44 Комментария 4 Решения Самые популярные

После выполнения скрипта, создание файла `file1.txt` произойдет в директории, указанной перед командой `touch`, которая будет `/home/bi/`, так как перед командой `cd` указан абсолютный путь `/home/bi/`.

Следовательно, абсолютный путь до созданного файла `file1.txt` будет `/home/bi/file1.txt`.

3.2 Скрипты на bash: основы 5 из 10 шагов пройдено 2 из 6 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: `script1.sh`, `script2.sh`.

Предположим, что вы находитесь в директории `/home/bi/Documents/` и запускаете в ней скрипт следующего содержания:

```
#!/bin/bash
cd /home/bi/
touch file1.txt
cd /home/bi/Desktop/
```

Как будет выглядеть абсолютный путь до созданного файла `file1.txt` по окончанию работы скрипта?

Выберите один вариант из списка

✓ Отлично!

Верно решили 29 905 участников
Из всех попыток 76% верных

/home/bi/file1.txt
 /home/bi/Desktop/file1.txt
 /home/bi/Documents/file1.txt
 Никак (файла file1.txt не будет существовать после завершения работы скрипта)

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл из 1

1152 144 Шаг 5

Следующий шаг >

Да, `variable_123` может быть именем переменной в `bash`. В `Bash` переменные могут содержать буквы (в верхнем или нижнем регистре), цифры и символ подчеркивания, и должны начинаться с буквы или символа подчеркивания.

The screenshot shows a user interface for an online programming exercise. At the top, there's a navigation bar with icons for back, forward, and search. Below it, a progress bar indicates "3.2 Скрипты на bash: основы" (Step 3.2: Bash Scripts: Basics), "7 из 10 шагов пройдено" (7 of 10 steps completed), and "3 из 6 баллов получено" (3 out of 6 points earned). A message says, "Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: variables1.sh, variables2.sh." Another message asks, "Какие из представленных ниже строк **могут** быть именами переменных в bash? Выберите **все** подходящие варианты!" A note says, "Подсказка: если все варианты ответов являются неверными, то не отмечайте ни один из них и нажимайте кнопку 'Отправить'/'Submit'." A green checkmark icon next to "Хорошая работа." (Good work.) indicates the answer is correct. A yellow box contains the text: "Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в комментариях, отвечая на их вопросы, или сравнить свой решение с другими на форуме решений." Below the question, a list of options is shown: variable_123, 12variable, vari@able, vari@ble, var@ble, variab\$le, vari-a-ble. There are two buttons at the bottom: "Следующий шаг" (Next step) and "Решить снова" (Solve again). A progress bar at the bottom shows "Ваше решение" (Your solution) and "Вы получили: 1 балл из 1" (You got: 1 point out of 1). Below the progress bar are like and dislike buttons with counts: 1152 likes and 144 dislikes. The current step is "Шаг 7".

В данном скрипте:

1. `#!/bin?/bash`: Эта строка указывает на путь к оболочке (в данном случае, к Bash). Обычно она начинает исполняемый скрипт и сообщает системе, какую оболочку нужно использовать для его выполнения. Ошибка в написании пути (неправильно написано `#!/bin?/bash`) может привести к неправильному выполнению скрипта или ошибке “Неверный интерпретатор”.
2. `var1=$1` и `var2=$2`: Здесь переменным `var1` и `var2` присваиваются значения первого и второго аргументов командной строки соответственно. Переменная `$1` содержит первый аргумент, а `$2` содержит второй.
3. `echo "Arguments are: \$1=$var1 \$2=$var2"`: Эта строка выводит сообщение в стандартный вывод. В ней используется экранирование символов `\$1` и `\$2`, чтобы они интерпретировались как текст, а не как переменные. В результате выводится строка “Arguments are: \$1=значение_первого_аргумента \$2=значение_второго_аргумента”, где значения переменных заменяются на фактические значения первого и второго аргументов, переданных скрипту при его запуске.

The screenshot shows a programming challenge interface. At the top, there are navigation icons for back, forward, and search. Below that, a message says: "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". Another message follows: "а при запуске ./script.sh three four будет: Arguments are: \$1=three \$2=four". A note at the bottom says: "Подсказка: в случае проблем с решением задачи, обратите внимание на наши рекомендации по написанию скриптов." A green box indicates "Напишите программу. Тестируется через stdin → stdout" and "Отличное решение!". To the right, it says "Верно решили 25 053 учащихся из всех попыток: 41% верных". Below this, a button says "Теперь вам доступен форум решений, где вы можете сравнить свое решение с другими или спросить совета." The main area contains a code editor with the following script:

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

Below the code editor are two buttons: "Следующий шаг" and "Решить снова". At the bottom, it says "Ваше решение" and "Вы получили: ••• из 3". There are also like and dislike buttons with counts of 1152 and 144 respectively, and a progress bar labeled "Шаг 10".

- **5 -ge 5:** Это условие проверяет, является ли число 5 больше или равным 5. В данном случае, 5 равно 5, поэтому условие верно.
- **-e \$0:** Это проверка существования файла с именем, указанным в переменной \$0. В \$0 хранится имя скрипта, который выполняется. Если файл существует, то выражение вернет true.
- **-s \$0:** Это проверка на непустоту файла с именем, указанным в переменной \$0. Если файл существует и он не пустой, то выражение вернет true.
- **! (4 -le 3):** Это отрицание условия “4 меньше или равно 3”. Поскольку это условие ложно, то его отрицание будет истинным.
- **\$# -ge 0:** Это проверка количества переданных в скрипт аргументов. Если количество аргументов больше или равно 0, то условие верно.
- **\$:** Это специальный символ, обозначающий имя переменной в bash. В данном контексте, он не имеет смысла как условие.
- **\$var1 ==:** Это некорректное выражение, так как оператор сравнения равенства (==) должен быть справа от значения переменной.
- **\$var2 ||:** Это некорректное выражение, так как логический оператор “или” (||) должен иметь значение справа от него.
- **\$var1 !=:** Это некорректное выражение, так как оператор сравнения неравенства (!=) должен быть справа от значения переменной.
- **\$var2:** Это имя переменной, но без дополнительного контекста, это не является условием.

Например, условие `$ -eq 0` **подходит**, т.к. ноль всегда равен нулю вне зависимости от аргументов и переменных внутри скрипта и на экран будет напечатано `True`. В то же время условие `$var1 -eq 0` **не подходит**, так как в переменной `var1` как может быть записан ноль (тогда будет напечатано `True`), так его может и не быть (тогда ничего напечатано не будет).

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 задач на Stepik.

Выберите все подходящие ответы из списка

✓ Отличное решение!

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

✓ 5 -ge 5
✓ -e \$0
✓ -s \$0
✓ ! (4 -le 3)
✓ \$x -ge 0
✓ \$var1 == \$var2 || \$var1 != \$var2

Следующий шаг Решить снова

Ваше решение: Вы получили 1 балл из 1

1180 164 Шаг 3 Следующий шаг >

Когда переменная `var` установлена в 3, скрипт выведет “four”, так как это значение не удовлетворяет ни одному из условий в if-elif-else блоке.

Когда переменная `var` установлена в 5, скрипт также выведет “four”. Важно понимать, что после выполнения первого подходящего условия в if-elif-else блоке, остальные условия проверяться не будут. Так как первое условие `[[$var -gt 5]]` вернет `false` для `var=5`, скрипт перейдет к следующему условию. Но так как в этом случае все остальные условия тоже не будут удовлетворены (поскольку `var=5` не меньше 3 и не равно 4), то будет выполнена ветвь else и выведено “four”.

3.3 Скрипты на bash: ветвления и циклы 5 из 9 шагов пройдено 2 из 10 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [branching2.sh](#), [branching3.sh](#).

Посмотрите на фрагмент bash-скрипта:

```
if [[ $var -gt 5 ]]
then
    echo "one"
elif [[ $var -lt 3 ]]
then
    echo "two"
elif [[ $var -eq 4 ]]
then
    echo "three"
else
    echo "four"
fi
```

Какие строки и в какой последовательности он выведет на экран, если сначала этот скрипт запустили задав переменную `var=3`, а затем запустили еще раз, но уже с `var=5`.

Выберите один вариант из списка

✓ Здорово, всё верно.

Верно решили 25 138 учащихся
Из всех попыток 64% верных

Сначала four, потом one
 Сначала four, потом four
 Сначала two, потом one
 Сначала two, потом four

Следующий шаг Решить снова

Этот скрипт на bash является примером использования оператора `case` для выбора действия на основе значения переменной.

1. `#!/bin/bash`: Это шебанг, который указывает на то, что скрипт должен быть интерпретирован с использованием оболочки bash.

2. `v=student`: Здесь переменной `v` присваивается значение “student”. Это общая часть, которая будет использоваться в каждом выводе.
3. `case $1 in`: Это начало блока `case`, который будет проверять значение первого аргумента (`$1`), переданного скрипту при запуске.
4. `0) res="No ${v}s";;`: Если значение `$1` равно 0, переменной `res` присваивается строка “No students”.
5. `1) res="$1 ${v}";;`: Если значение `$1` равно 1, переменной `res` присваивается строка “1 student”.
6. `[2-4]) res="$1 ${v}s";;`: Если значение `$1` находится в диапазоне от 2 до 4 включительно, переменной `res` присваивается строка вида “2 students”, “3 students” или “4 students”.
7. `*) res="A lot of ${v}s";;`: Если значение `$1` не соответствует ни одному из вышеперечисленных случаев, переменной `res` присваивается строка “A lot of students”.
8. `esac`: Завершает блок `case`.
9. `echo "$res"`: Выводит содержимое переменной `res` в стандартный вывод.

Таким образом, этот скрипт анализирует значение первого аргумента, переданного при запуске скрипта, и в зависимости от этого значения формирует строку вывода, указывающую на количество студентов.

3.3 Скрипты на bash: ветвления и циклы 6 из 9 шагов пройдено 5 из 10 баллов получено

Напишите скрипт на bash, который принимает на вход один аргумент (целое число от 0 до бесконечности), который будет обозначать число студентов в аудитории. 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
4 --> 4 students
5 и больше --> A lot of students
```

Примечание а: выводить нужно только строку справа, т.е. “->” выводить не нужно.
Примечание б: в последней строке слово “lot” с маленькой буквы!

Примечание 2: в этой и всех последующих задачах на написание скриптов, если не указано явно, что нужно проверять вход (например, что он будет именно числом и именно 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 задачи, обратите внимание на наши рекомендации по написанию скриптов.

Напишите программу. Тестируется через `stdin` → `stdout`

Проверено: Прекрасный ответ.

Верно решили 23 310 участников
Из всех попыток 38% верных

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

Напишите программу. Тестируется через stdin → stdout

✓ Прекрасный ответ.

Верно решили 23 310 учащихся
Из всех попыток 38% верных

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

```
1#!/bin/bash
2 v=student #задаем общую часть в каждом выводе - слово "student"
3 case $1 in
4 *) res="N ${v}s"; #res - это результат для вывода
5 *) res="$1 ${v}";;
6 [2-4]) res="$1 ${v}s";;
7 *) res="A lot of ${v}s";;
8 esac
9 echo "$res"
10
11
12
13
14
```

[Следующий шаг](#)

[Решить снова](#)

В данном скрипте используется цикл `for`, который перебирает элементы в списке строк “`a`”, “`,`”, “`b`”, “`,`”, “`c_d`”. В каждой итерации цикла сначала выводится слово “`start`”, а затем выполняется проверка условия `if`.

- Когда переменная `str` равна “`a`”, условие `$str > "c"` не выполняется, поэтому выводится “`start`”, а затем “`finish`”.
- Когда переменная `str` равна “`,`”, также выполняется вывод “`start`” и “`finish`”.
- Когда переменная `str` равна “`b`”, также выполняется вывод “`start`” и “`finish`”.
- Когда переменная `str` равна “`,`”, также выполняется вывод “`start`” и “`finish`”.
- Когда переменная `str` равна “`c_d`”, условие `$str > "c"` выполняется (поскольку “`c_d`” больше, чем “`c`”), и выполняется оператор `continue`, который прерывает текущую итерацию цикла и переходит к следующей строке.
- Когда переменная `str` равна “`“`”, условие `$str > "c"` не выполняется, поэтому выводится “`start`”, а затем “`finish`”.

Таким образом, слово “`start`” будет выведено 5 раз, а слово “`finish`” будет выведено 4 раза.

3.3 Скрипты на bash: ветвления и циклы 8 из 9 шагов пройдено 6 из 10 баллов: получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: `loops1.sh`, `loops2.sh`.

Посмотрите на фрагмент bash-скрипта:

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

Если запустить этот скрипт, то сколько раз на экран будет выведено слово “`start`”, а сколько раз слово “`finish`”?

Выберите один вариант из списка

✓ Здорово, всё верно.

Верно решили 24 582 учащихся
Из всех попыток 49% верных

5 раз “`start`” и ни разу “`finish`”
 5 раз “`start`” и 2 раза “`finish`”
 5 раз “`start`” и 4 раза “`finish`”
 3 раза “`start`” и 3 раза “`finish`”

[Следующий шаг](#)

[Решить снова](#)

Этот скрипт представляет собой пример использования циклов `while`, условных операторов `if-elif-else` и операторов сравнения в bash.

1. На первых трех строках определяются переменные `child`, `adult`, и `stdout`, которые устанавливаются в определенные значения.
2. Далее следует основной цикл `while`, который будет выполняться, пока значение переменной `stdout` не станет равным 1.
3. Внутри цикла выводится приглашение ввести имя ("enter your name: ") и считывается введенное значение в переменную `name`.
4. Затем выполняется проверка условий:
 - Если введенное имя пусто или равно 0, то выводится "bye", переменная `stdout` устанавливается в 1, и программа завершает работу.
 - Если введенное имя не пусто, то вводится возраст ("enter your age: "), который сохраняется в переменную `age`.
5. После этого идет вложенный цикл `while`, который также выполняется, пока значение переменной `stdout` не станет равным 1.
6. Внутри вложенного цикла проверяется возраст:
 - Если возраст равен 0 или пуст, то выводится "bye", `stdout` устанавливается в 1, и программа завершает работу.
 - Если возраст меньше или равен значению переменной `child`, выводится соответствующее сообщение о принадлежности к группе "child".
 - Если возраст больше значения переменной `adult`, выводится сообщение о принадлежности к группе "adult".
 - В остальных случаях проверяется, попадает ли возраст в диапазон от 17 до 25 лет. Если да, выводится сообщение о принадлежности к группе "youth".
7. После каждой итерации вложенного цикла выполняется оператор `break`, который прерывает выполнение вложенного цикла и переходит к следующей итерации основного цикла.
8. Если введено имя, но не введен возраст, программа просто вернется к вводу возраста.
9. Если введен возраст, программа завершит работу, только если возраст равен 0 или пуст.



Напишите скрипт на bash, который будет определять в какую возрастную группу попадают пользователи. При запуске скрипта должен вывести сообщение "enter your name:" и ждать от пользователя ввода имени (используйте `read`, чтобы прочитать его). Когда имя введено, то скрипт должен написать "enter your age:" и ждать ввода возраста (для этого нужен `read`). Когда возраст введен, скрипт пишет на экран "«имя», your group is <группа>", где <группа> определяется на основе возраста по следующим правилам:

- младше либо равно 16: "child";
- от 17 до 25 (включительно): "youth";
- старше 25: "adult".

После этого скрипт опять выводит сообщение "enter your name:" и всё начинается по новой (бесконечный цикл). Если в какой-то момент работы скрипта будет введено **пустое имя** или **возраст 0**, то скрипт должен написать на экран "bye" и закончить свою работу (выход из цикла).

Примеры корректной работы скрипта:

Nº1

```
./script.sh
enter your name:
Egor
enter your age:
16
Egor, your group is child
enter your name:
Elena
enter your age:
0
bye
```

Nº2

```
./script.sh
enter your name:
Elena Petrovna
enter your age:
25
Elena Petrovna, your group is youth
enter your name:
bye
```



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

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](#).

```
1 child=16
2 adult=25
3 stdout=0
4
5 while [[ $stdout != 1 ]]
6   do
7     echo "enter your name: "
8     read name
9     if [[ (-z $name) || ($name = 0) ]] ;then
10       echo "bye"
11       stdout=1
12     elif [[ (-z $name) ]]; then
13       while [[ $stdout != 1 ]] ;do
14         echo "enter your age: "
15         read age
16         if [[ ($age -eq 0) || (-z $age) ]] ;then
17           echo "bye"
18           stdout=1
19         elif [[ $age -le $child ]] ;then
20           echo "$name, your group is child"
21         elif [[ $age -ge $adult ]] ;then
22           echo "$name, your group is adult" ;else
23             if [[ ($age -ge 17) && ($age -le 25) ]] ;then
24               echo "$name, your group is youth" ;fi
25             fi ;break
26       done ;#fi
27     done
28
29
30
31
32
```

[Следующий шаг](#)

[Решить снова](#)

Эти варианты представляют собой арифметические операции сложения значений переменных *a* и *b* и записи результата в переменную *a*.



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

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

Всё правильно.

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

Верно решили 22 116 участников
Из всех попыток 20% верных.

`a+=b`
 `let "a=$a+b"`
 `let a=$a+$b`
 `let "a = a + b"`
 `let a+=b`

[Следующий шаг](#) [Решить снова](#)

Ваше решение Вы получили: 1 балл из 1

896 255 Шаг 3

[Следующий шаг >](#)

Команда `echo` выведет на экран строку, которая будет результатом выполнения команды `pwd`, которая возвращает текущий рабочий каталог. В данном случае скрипт сначала переходит в каталог `/home/bi/`, а затем выводит его полный путь с помощью команды `pwd`.



3.4 Скрипты на bash: разное 5 из 10 шагов пройдено 2 из 14 баллов получено

Вы прошли больше 80% курса, оставьте отзыв

[Оставить отзыв](#) Нет, спасибо

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

Пусть вы находитесь в директории `/home/bi/Documents/` и запускаете в ней скрипт следующего содержания:

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

Что в этом случае выведет команда `echo` на экран?

Выберите один вариант из списка

Верно.

Верно решили 23 677 участников
Из всех попыток 51% верных

`/home/bi/Documents`
 `pwd`
 `/home/bi`
 `'pwd'`
 Код возврата команды `pwd` (0 в случае успешного выполнения и не 0 в случае ошибок)

[Следующий шаг](#) [Решить снова](#)

Ваше решение Вы получили: 1 балл из 1

896 255 Шаг 5

[Следующий шаг >](#)

В случае, когда нужно сначала выполнить программу и затем проверить её код возврата, можно использовать следующую конструкцию:

```
program
if [[ $? -eq 0 ]]
then
    # Ваши действия, если код возврата программы равен 0
fi
```

Если же программа пишет что-то в `stdout`, но вам нужно все-таки проверить её код возврата, можно перенаправить вывод в файл и затем прочитать этот файл в

условии if. Однако, это может быть неэффективным, особенно если вывод программы большой или если вы не хотите сохранять вывод в файл. Вот пример:

```
if program > output.txt ; then
    # Ваши действия, если код возврата программы равен 0
fi
```

В этом случае программа `program` будет запущена, её вывод будет перенаправлен в файл `output.txt`, и затем код возврата будет проверен в условии if.

The screenshot shows a user interface for a programming challenge. At the top, there's a toolbar with icons for file operations like Open, Save, and Print. Below it, a progress bar indicates "3.4 Скрипты на bash: разное" has been completed with 6 из 10 шагов пройдено and 3 из 14 баллов получено. A message box says "Вы прошли больше 80% курса, оставьте отзыв" with "Оставить отзыв" and "Нет, спасибо".

The main area contains text explaining how to check the return code of an external program within an if statement. It mentions that if the program exits with 0, the code in the then block will run. It also notes that if the program exits with something other than 0, the code in the then block won't run, but the exit code will still be checked.

Below this, there's a note about using single quotes for command substitution and double quotes for variable expansion.

A poll section asks "Выберите все подходящие ответы из списка" with a green checkmark and the text "Всё получилось!". It shows statistics: "Верно решили 21 426 учащихся" and "Из всех попыток 20% верных".

The poll options are:

- Сначала запустить program, затем if [[\$? -eq 0]]
- Ничего делать нельзя
- Сначала var=program, затем if [[\$var -eq 0]]
- if [[program = -eq 0]]
- if 'program > some_file.txt'

At the bottom of the poll area are two buttons: "Следующий шаг" and "Решить снова".

Строка, 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, будет:

`counters are 55 and 110`

Пояснение: - Первый вызов функции `counter` с параметром 1 увеличит `c1` на 1, а `c2` на 2 (т.к. `c2` умножается на 2 при каждом вызове). - Второй вызов функции `counter` с параметром 2 увеличит `c1` на 2, а `c2` на 4. - И так далее, до последнего вызова функции `counter` с параметром 10, который увеличит `c1` на 10, а `c2` на 20.

Итак, `c1` будет равен сумме чисел от 1 до 10, что равно 55, а `c2` будет равен удвоенной сумме этих чисел, т.е. 110.



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 "$1+=${!1}"
    let "$2+=${!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 посимвольно сверить свой ответ с тем, что он выдаст на экран.

Напишите текст

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 **20 009** учащихся
Из всех попыток **28%** верных

counters are and 110

Этот скрипт на языке оболочки Bash вычисляет наибольший общий делитель (НОД) двух чисел, введенных пользователем во время выполнения скрипта.

Основная часть скрипта находится внутри бесконечного цикла `while`, который будет выполняться до тех пор, пока условие [`true`] истинно.

- Сначала скрипт считывает два числа `n1` и `n2`, введенные пользователем с клавиатуры.
- Далее идет проверка на пустоту переменной `n1` при помощи условного оператора `if [-z $n1]`. Если `n1` пусто (т.е. пользователь ничего не ввел и просто нажал Enter), то выводится сообщение “`bye`”, и цикл завершается с помощью команды `break`.
- Если `n1` не пусто, тогда создается функция `gcd` (наибольший общий делитель), которая используется для вычисления НОД двух чисел.
- Функция `gcd` использует алгоритм Евклида для вычисления НОД. Он состоит в последовательных делениях, пока не получится нулевой остаток.
- После завершения работы функции выводится сообщение “`GCD is $n1`”, где `'$n1'` содержит значение НОД двух введенных чисел.

Итак, скрипт будет продолжать запрашивать ввод двух чисел и вычислять их НОД до тех пор, пока пользователь не введет пустую строку (нажмет Enter).



3.4 Скрипты на bash: разное 8 из 10 шагов пройдено 5 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 <посчитанное значение>", например, для чисел 15 и 25 это будет "GCD is 5". После этого скрипт опять входит в режим ожидания двух натуральных чисел. Если в какой-то момент работы пользователь 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 вывести соответствующее сообщение на экран (см. выше). Иначе нужно сравнять 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**, а вторым (**N**,**M**).

Пример корректной работы скрипта:

```
./script.sh  
10 15  
GCD is 5  
7 3  
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.

Напишите программу. Тестируется через `stdin` → `stdout`



Всё получилось!

Верно решили 18 148 учащихся
Из всех попыток 35% верных

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

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

Следующий шаг

Решить снова

Этот скрипт работает следующим образом: - В бесконечном цикле `while true` происходит чтение ввода пользователя с помощью команды `read`. - После этого проверяется, является ли первый операнд строкой "exit". Если это так, то выводится "bye" и цикл завершается. - Затем проверяется, содержат ли оба операнда целые числа. Если нет, выводится "error" и цикл завершается. - Если оба операнда являются целыми числами, то в зависимости от оператора (+, -, *, /, %, **) производится соответствующая математическая операция. - Результат операции выводится на экран. - Если оператор не соответствует ни одной из допустимых операций, выводится "error" и цикл завершается.



Напишите калькулятор на bash. При запуске ваш скрипт должен ожидать ввода пользователям команды (при этом на экран выводить ничего не нужно). Команды могут быть трех типов:

1. Слово "exit". В этом случае скрипт должен вывести на экран слово "bye" и завершить работу.
2. Три аргумента через пробел – первый операнд (целое число), операция (одна из "+", "-", "*", "/", "%", "**") и второй operand (целое число). В этом случае нужно произвести указанную операцию над заданными числами и вывести результат на экран. После этого переходим в режим ожидания новой команды.
3. Любая другая команда из одного аргумента или из трех аргументов, но с операцией не из списка. В этом случае нужно вывести на экран слово "error" и завершить работу.

Чтобы проверить работу скрипта, вы можете записать сразу несколько команд в файл и передать его скрипту на stdin (т.е. выполнить `./script.sh < input.txt`). В этом случае он должен вывести сразу все ответы на экран.

Например, если входной файл будет следующего содержания:

```
10 + 1  
2 ** 10  
exit
```

то на экране будет:

```
11  
1024  
exit
```

Если же на вход поступит следующий файл:

```
3 - 5  
2/10  
exit
```

то на экране будет:

```
-2  
error
```

т.к. вторая команда была некорректной (в ней всего один аргумент, т.к. нет пробелов между числами и операцией, а единственная допустимая команда из одного аргумента это "exit").

Напишите программу. Тестируется через stdin → stdout

Абсолютно точно.

Верно решили 16 980 учащихся
Из всех попыток 36% верных

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

```
1 # put your shell (bash) code here  
2  
3 #!/bin/bash  
4 while [[ $True ]]  
5 do  
6     read birinchi amal ikkinchi  
7     if [[ $birinchi == "exit" ]]  
8     then  
9         echo "bye"  
10        break  
11    elif [[ "$birinchi" =~ ^[0-9]+\$ && "$ikkinchi" =~ ^[0-9]+\$ ]]  
12    then  
13        echo "error"  
14        break  
15    else  
16        case $amal in  
17            "+") let "result = birinchi + ikkinchi";;  
18            "-") let "result = birinchi - ikkinchi";;  
19            "/") let "result = birinchi / ikkinchi";;  
20            "*") let "result = birinchi * ikkinchi";;  
21            "%") let "result = birinchi % ikkinchi";;  
22            "**") let "result = birinchi ** ikkinchi";;  
23        *) echo "error" ; break ;;  
24        esac  
25        echo "$result"  
26    fi  
27 done  
28  
29  
30
```

Команда `find /home/bi -iname "star*"` найдет все файлы, начинающиеся с "star" без учета регистра, поэтому она найдет файлы, такие как "Star_Wars.avi", "star_trek_OST.mp3", "STARS.txt" и "stardust.mpeg".

С другой стороны, команда `find /home/bi -name "star*"` найдет только те файлы, начинающиеся с "star" с учетом регистра, поэтому она пропустит файлы "STARS.txt" и "Star_Wars.avi".

Таким образом, файлы, которые найдет первая команда, но не найдет вторая, это "STARS.txt" и "Star_Wars.avi".

3.5 Продвинутый поиск и редактирование 3 из 13 шагов пройдено 1 из 10 баллов получен

Пусть в директории /home/b1 лежат файлы Star_Wars.avi, star_trekOST.mp3, STARS.txt, stardust.mpeg, Eddard_Stark.biography.txt.

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

Выберите все подходящие ответы из списка

Правильно, молодец!

✓ starstardust.mpeg
✓ STARS.txt
✓ Star_Wars.avi
✗ Eddard_Stark.biography.txt
✗ star_trekOST.mp3

Верно решили 20 547 участника
Из всех попыток 36% верных

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

714 228 Шаг 3 Следующий шаг >

1. Если заменить в команде поиска `-name` на `-path`, то результат поиска иногда может остаться таким же. **- Верно.** Это связано с тем, что обе опции (`-name` и `-path`) используются для сопоставления имени файла, но `-name` ищет только в имени файла, тогда как `-path` ищет в полном пути к файлу. Если имена файлов не содержат пути, замена `-name` на `-path` не повлияет на результат поиска.
2. В некоторых случаях `find` с `-name` найдет меньше файлов, чем `find` с таким же запросом, но с `-path`. **- Верно.** Это может произойти, если искомые файлы находятся во вложенных директориях, и их имена включают в себя определенную строку, которая не является именем самого файла, но присутствует в пути к файлу. В этом случае `-path` найдет больше файлов, чем `-name`.

3.5 Продвинутый поиск и редактирование 4 из 13 шагов пройдено 2 из 10 баллов получено

Задание на понимание работы опций `-path` и `-name` команды `find`. Отметьте **все верные** утверждения из перечисленных ниже.

Выберите все подходящие ответы из списка

Так точно!

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](#).

✓ Опция `-path` аналогична `-name`, но игнорирует размер букв (строчные/прописные) в имени файла.
✓ В некоторых случаях `find` с `-name` найдет меньше файлов, чем `find` с таким же запросом, но с `-path`
✗ Если заменить в команде поиска `-name` на `-path`, то результат поиска всегда останется неизменным
✓ Если заменить в команде поиска `-name` на `-path`, то результат поиска иногда может оставаться таким же
✓ В некоторых случаях `find` с `-name` найдет больше файлов, чем `find` с таким же запросом, но с `-path`

Верно решили 18 450 участников
Из всех попыток 22% верных

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

714 228 Шаг 4 Следующий шаг >

Команда `find /home/b1 -mindepth 2 -maxdepth 3 -name "file*"` будет искать файлы в директории `/home/b1/` и ее поддиректориях с глубиной от 2 до 3 уровней и с именем, начинающимся на "file".

1. **file1:** Будет найден, так как он находится на уровне глубины 2 и его имя начинается с "file".

2. **file2**: Будет найден, так как он находится на уровне глубины 3 и его имя начинается с “file”.
3. **file3**: Не будет найден, так как он находится на уровне глубины 4, который выходит за рамки заданного диапазона глубины.

3.5 Продвинутый поиск и редактирование 5 из 13 шагов пройдено 3 из 10 баллов получено

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

Правильно, молодец!

Верно решили 20 711 учащихся
Из всех попыток 41% верных

Только `file2`
 Все три файла
 Все кроме `file3`
 Только `file3`
 Все кроме `file2`

Следующий шаг **Решить снова**

Ваше решение: Вы получили 1 балл из 1

Во всех четырех случаях создание файла `results.txt` приведет к одинаковому размеру, потому что ни одна из команд не изменяет количество символов, записываемых в файл.

- `grep "word" file.txt > results.txt` запишет все строки, содержащие “word”, в `results.txt`.
- `grep -A 1 "word" file.txt > results.txt` запишет все строки, содержащие “word”, а также одну следующую строку после каждой строки с “word” в `results.txt`.
- `grep -B 1 "word" file.txt > results.txt` запишет все строки, содержащие “word”, а также одну предыдущую строку перед каждой строкой с “word” в `results.txt`.
- `grep -C 1 "word" file.txt > results.txt` запишет все строки, содержащие “word”, а также одну предыдущую и одну следующую строку вместе с каждой строкой с “word” в `results.txt`.

Таким образом, все четыре команды создадут файл `results.txt` с одинаковым размером.

3.5 Продвинутый поиск и редактирование 7 из 13 шагов пройдено 4 из 10 баллов получено

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

Правильно, молодец!

Верно решили 20 237 учащихся
Из всех попыток 41% верных

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
 grep -B 1 "word" file.txt > results.txt

Следующий шаг Решить снова

Ваше решение Вы получили 1 балл из 1

714 228 Шаг 7 Следующий шаг >

Команда grep -E "[xk1XKL]?[uU]buntu\$" text.txt будет искать строки, которые заканчиваются на "ubuntu" или "Ubuntu", при условии, что перед "ubuntu" или "Ubuntu" может быть один из символов "x", "k", "l", "X", "K", "L".

3.5 Продвинутый поиск и редактирование 9 из 13 шагов пройдено 6 из 10 баллов получено

Предположим, что в файле text.txt записаны строки, показанные среди вариантов ответа. Отметьте только те из них, которые выведет на экран команда grep -E "[xk1XKL]?[uU]buntu\$" text.txt.

Выберите все подходящие ответы из списка

Всё получилось!

Вы решили сложную задачку, 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

Lubuntu is better than Ubuntu
 Mac OS X, Windows, Ubuntu
 The best OS is Kubuntu
 Hmm, XXLubuntu
 Linux is not always Ubuntu
 I prefer Kubuntu

Следующий шаг Решить снова

Ваше решение Вы получили 2 балла из 2

714 228 Шаг 9 Следующий шаг >

Если в команде sed "/[a-z]*p" text.txt не указывать опцию -n, то по умолчанию sed будет выводить каждую строку после ее обработки, вне зависимости от того, была ли она изменена или нет.

Команда sed "/[a-z]*p" text.txt будет искать строки, которые содержат ноль или более символов в нижнем регистре, и затем печатать их. Однако, поскольку мы не указали -n, sed также будет выводить оригинальные строки из файла, которые не были изменены. Это приведет к тому, что каждая строка будет выведена дважды - сначала в исходном виде, а затем после обработки.

Поэтому каждая строка будет выведена два раза.

The screenshot shows a step in a challenge titled "3.5 Продвинутый поиск и редактирование". It asks: "Что произойдет, если в команде `sed -n "/[a-z]*\w/" p text.txt` не указывать опцию `-n`?" Below the question is a list of four options:

- Будут выведены все строки файла `text.txt`, в которых есть только большие буквы латинского алфавита
- На экран ничего не напечатается
- Появится сообщение об ошибке
- Каждая строка будет выведена два раза

Below the list are two buttons: "Следующий шаг" and "Решить снова". A green box indicates "Верно решили 19 784 участника" and "Из всех попыток 39% верных". At the bottom, it says "Ваше решение Вы получили: 1 балл из 1".

Эта команда `sed` использует регулярное выражение для поиска последовательности символов в верхнем регистре длиной два символа или более, за которыми следует пробел. Затем эта последовательность символов заменяется на строку “abbreviation”.

Давайте разберем каждый элемент команды подробнее:

1. `sed`: Это утилита Unix/Linux для потокового редактирования текста.
2. `'s/.../.../g'`: Это команда `sed` для замены. Формат команды замены выглядит как `'s/шаблон/замена/флаги'`. Здесь `g` означает, что замена должна производиться глобально для каждого вхождения шаблона в строке (то есть не только для первого вхождения).
3. `[A-Z]\{2,\}`: Это регулярное выражение, которое ищет последовательность символов в верхнем регистре (от `A` до `Z`) длиной два символа или более (`\{2,\}`). Знак пробела после закрывающей фигурной скобки указывает, что после последовательности символов должен идти пробел.
4. `abbreviation`: Это строка, на которую заменяется найденная последовательность символов. В данном случае, если найдется последовательность символов в верхнем регистре, она будет заменена на “`abbreviation`”.
5. `g`: Этот флаг указывает на глобальное применение замены ко всей строке. Без этого флага замена будет осуществлена только для первого вхождения шаблона в строке.

Таким образом, данная команда `sed` найдет все последовательности символов в верхнем регистре, состоящие из двух символов или более, за которыми следует пробел, и заменит их на строку “`abbreviation`” в файле `input.txt`. Результат будет записан в файл `edited.txt`.



- состоит только из больших букв латинского алфавита,
- состоит из хотя бы двух букв,
- окружено одним пробелом с каждой стороны.

При этом будем считать, что в тексте **не может быть две "аббревиатуры" подряд**. Например, текст "YOU YOU and YOU!" является **некорректным** (в нем есть две "аббревиатуры", но они идут подряд) и на таких примерах мы проверяя вашу инструкцию **не будем**.

Пример: если у вас был текст "Hi, I heard these songs by ABBA, TLA and DR !", то он должен быть преобразован в "Hi, I heard these songs by ABBA, abbreviation and abbreviation !".

Примечание: после вашей замены "аббревиатуры" на слово "abbreviation" количество пробелов в тексте **не должно меняться!**

Внимание! Во время проверки мы **не запускаем команду**, которую вы ввели на реальном файле с "аббревиатурами" (это небезопасно, можно же ввести `ls -rf /*`!) Вместо этого мы сперва анализируем структуру вашей инструкции (например, что в ней использовано именно `sed` и сделано это ровно один раз, что на вход подается `input.txt`, а результат будет записан в `edited.txt` и т.д.), а затем запускаем её **смысловую часть** (т.е. поиск по регулярному выражению и замена на "abbreviation") на тестовых примерах. К сожалению, наш запуск не идеально повторяет `sed`, но он очень близок к нему. Главная "несовместимость" заключается в том, что наша проверка не понимает идущие подряд символы, отвечающие за количество повторений (т.е. *, +, ? и {}). Однако эту "несовместимость" легко исправить указав, кому помочь ("и") какой из символов к чему относится! Например, регулярное выражение `a+?` (оль или один раз по одной или более букве "a") нужно записать как `(a+)?` (при этом запись `(a+)+?`, конечно же, не поможет).

Напишите текст

Всё правильно.

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

Верно решили **16 632** ученика
Из всех попыток **34%** верных

```
sed %/[A-Z]\{2,\}/g input.txt > edited.txt
```

Эта опция позволяет сохранить все нарисованные графики после закрытия `gnuplot`, таким образом, они не будут автоматически закрыты. Это полезно, если вам нужно сохранить графики для последующего просмотра или анализа.



3.6 Строим графики в `gnuplot` 3 из 10 шагов пройдено 1 из 7 баллов получен

Вы можете скачать и попробовать применить `gnuplot` к файлу, который мы показали в видеофрагменте: [authors.txt](#).

Какую опцию нужно указать при запуске `gnuplot`, чтобы при его закрытии не были автоматически закрыты и все нарисованные в нём графики?

Выберите один вариант из списка

Так точно!

Верно решили **18 785** ученика
Из всех попыток **51%** верных

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

[Следующий шаг](#)

[Решить снова](#)

Ваше решение: Вы получили **1 балл из 1**

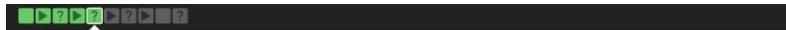
[Следующий шаг >](#)

435

296

Шаг 3

Название ряда данных будет взято из первого значения во втором столбце, так как в данном случае опция `autotitle columnhead` используется для автоматического определения заголовков столбцов. На графике будет нарисовано 9 точек, так как точка из первой строки файла данных будет пропущена при построении графика.



3.6 Строим графики в gnuplot 5 из 10 шагов пройдено 2 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

Отлично!

Верно решили 17 975 учащихся

Из всех попыток 32% верных

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](#).

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

[Следующий шаг](#)

[Решить снова](#)

Решение проверено. Вы получили 1 балла из 1.

- `"point 1, value ".x1 x1` означает, что для первого деления на оси x будет установлена метка "point 1, value", а ей будет соответствовать значение `x1`.
- `"point 2, value ".x2 x2` означает, что для второго деления на оси x будет установлена метка "point 2, value", а ей будет соответствовать значение `x2`.
- `"point 3, value ".x3 x3` означает, что для третьего деления на оси x будет установлена метка "point 3, value", а ей будет соответствовать значение `x3`.



Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: `plot.gnu`, `plot_advanced.gnu`, `plot_advanced2.gnu`. Все эти скрипты 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 OX (по возрастанию). Вы хотите, чтобы на этих 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 конкатенации, 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

Хорошая работа.

Верно решили 13 925 учащихся

Из всех попыток 44% верных

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

[Следующий шаг](#)

[Решить снова](#)

1. `a=a+1`: Эта строка увеличивает переменную `a` на 1.
2. `zrot=(zrot+350)%360`: В этой строке переменная `zrot` увеличивается на 350 и затем берется остаток от деления на 360. Это позволяет создавать эффект вращения вокруг оси Z.
3. `set view xrot,zrot`: Эта команда устанавливает углы поворота вокруг осей X и Z для просмотра графика.
4. `splot -x**2-y**2`: Эта команда строит трехмерный график функции `-x**2-y**2`.

- pause 0.1: Эта команда делает паузу в скрипте на 0.1 секунду.
- if (a<50) reread: Эта строка проверяет условие $a < 50$. Если оно выполняется, то выполняется команда `reread`, которая перечитывает текущий скрипт. Таким образом, скрипт будет выполнятся снова и снова, пока a меньше 50.

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

Если вы не скачали на предыдущем шаге файлы `animated.gnu` и `move.rot`, то скачайте их теперь, т.к. они понадобятся для выполнения задания.

Указанные файлы использовались в последнем видеофрагменте для создания вращающегося графика. Измените инструкции в файле `move.rot` (т.е. добавьте и удалить инструкции `newline`) таким образом, чтобы:

- График отразился **зеркально** относительно горизонтальной поверхности. То есть там, где была точка $(10, 10, 200)$, станет точка $(10, 10, -200)$, где была точка $(-10, -10, 200)$ станет $(-10, -10, -200)$ и т.д. При этом точка $(0, 0, 0)$ останется на месте.
- Изображение стало **вращаться в обратную сторону**. То есть если раньше вращалось "влево", то теперь станет "вправо".
- Вращение стало **в два раза быстрее**. То есть станет в два раза больше перерисовок графика на каждую секунду вращения.

Измененный файл загрузите в форму ниже.

Примечание: наша система проверки **не может** запустить на вашем файле `move.rot` программу `gnuplot` и сравнить полученный график с заданным. Вместо этого мы **анализируем команды**, которые вы указали в файле. Поэтому если вы видите, что ваш скрипт в `gnuplot` работает точно по условию, а мы отвечаем "Incorrect/Неверно", то попробуйте упростить свою модификацию `move.rot` и отправить его еще раз.

Напишите текст

Верно. Так держать!

Верно решили 12 854 учащихся
Из всех попыток 47% верных

```
a=a+1
zrot=(zrot+350)%360
set view xrot,zrot
splot x**2-y**2
pause 0.1
if (a>=360) reread
```

Следующий шаг Решить снова

- chmod 764 file.txt: Эта команда устанавливает права доступа на чтение, запись и выполнение для владельца файла (7), на чтение и запись для группы (6) и на чтение для остальных (4).
- chmod ug+w file.txt; chmod u+x file.txt: В этом варианте сначала добавляются права на запись для владельца и группы (ug+w), а затем устанавливается право на выполнение только для владельца (u+x).

3.7 Разное 4 из 15 шагов пройдено 1 из 7 баллов получен

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

Примечание: запись вида `команд1; команда2; команда3` означает, что в терминале последовательно выполнились все три команды (сначала `команд1`, затем `команда2` и, наконец, `команда3`).

Выберите все подходящие ответы из списка

Здорово, всё верно.

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

Верно решили 16 484 учащихся
Из всех попыток 21% верных

chmod 777 file.txt
 chmod 764 file.txt
 chmod ug+w file.txt; chmod u+x file.txt
 chmod u-wx file.txt; chmod g-w file.txt
 chmod o-wx file.txt; chmod g-x file.txt; chmod a+wx file.txt

Следующий шаг Решить снова

Ваша решениеВаши решения: Вы получили 1 балл из 1

1681 105 Шаг 4 Следующий шаг >

- sudo chown user dir: Эта команда изменит владельца каталога `dir` на пользователя `user`.

2. `sudo chmod o+w dir`: Эта команда добавит право на запись для остальных (other) пользователей для каталога `dir`.
3. `sudo chown user:group dir`: Эта команда изменит как владельца, так и группу каталога `dir` на `user` и `group` соответственно.
4. `sudo chmod a+w dir`: Эта команда добавит право на запись для всех (владельца, группы и остальных) пользователей для каталога `dir`.

В случае же, когда речь идет о директории созданной `root`, ситуация будет проще: пользователь сможет смотреть её содержимое (у него есть право "У"), но удалять и создавать файлы в ней не сможет (права "W" у него нет).
Важно отметить, что директории в Linux это в каком-то смысле файлы. Содержимое такого "файла" – это записи о файлах и поддиректориях этой директории (такую говорят и называют). Таким образом, право "У" у директории дает возможность просматривать "записи", т.е. просматривать ее состав. Право "W" у директории дает возможность удалять/добавлять новые "записи", т.е. удалять/создавать файлы/поддиректории в ней.
На самом деле и это еще не все. Существует так называемый **sticky bit** (атрибут файла или директории), выставление которого меняет описание выше поведение. Файлы (или директории) с таким атрибутом смогут удалять только их владелец вне зависимости от прав, установленных у директории, в которой эти файлы (или директории) лежат!

Отдельное спасибо слушателю курса **Alexey Antipovskiy** за помощь в оформлении [Примечания 2](#)

Выберите все подходящие ответы из списка

Всё правильно.

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](#).

sudo chmod g+w dir
 sudo chown -group dir
 sudo chown user dir
 sudo chmod o+w dir
 sudo chown user:group dir
 sudo chmod a+w dir

Следующий шаг **Решить снова**

Ваше решение Вы получили 1 балл из 1

1681 105 Шаг 5 **Следующий шаг >**

- **Количество символов:** Это общее количество символов в файле, включая пробелы, символы табуляции и символы новой строки.
- **Количество слов:** Это количество последовательностей символов, разделенных пробелами, табуляцией или символами новой строки. Обычно слова разделяются пробелами, но команда `wc` считает и другие символы разделителями слов.
- **Количество строк:** Это количество строк в файле, где каждая строка заканчивается символом новой строки.
- **Размер файла в байтах:** Это общий размер файла в байтах, включая все его содержимое.



3.7 Разное 7 из 15 шагов пройдено 3 из 7 баллов получено

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

Выберите все подходящие ответы из списка

Отлично!

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 17 158 учащихся
Из всех попыток 21% верных

- Количество предложений
- Количество символов
- Количество слов
- Количество строк
- Размер файла в байтах

[Следующий шаг](#) [Решить снова](#)

Ваше решение Вы получили: • из 1

1681 105 Шаг 7

[Следующий шаг >](#)

- **-s (или --summarize)** означает вывод суммарного размера директории, а не всех поддиректорий и файлов внутри неё.
- **-h (или --human-readable)** означает вывод в удобном для чтения формате, в человекочитаемых единицах, таких как Кб, Мб, Гб.



3.7 Разное 8 из 15 шагов пройдено 5 из 7 баллов получено

Впишите в форму ниже команду, которая выведет сколько места на диске занимает текущая директория (при этом **размер** нужно вывести в **удобном для чтения формате** (например, вместо 2848 байт надо выводить 2.0К) и **больше** на экран выводить **ничего не нужно**). В команде указывайте **только необходимые** для выполнения задания **опции и аргументы**, лишних опций указывать не нужно!

Пример: если в текущей директории есть два файла по 888 Кбайт и две поддиректории в каждой из которых лежит по файлу в 488 Кбайт, то загаданная команда должна вывести на экран одно число: 2.4M (также на экране может быть выведен еще и символ "+", обозначающий, что это размер именно текущей директории).

Напишите текст

Правильно, молодец!

Верно решил 16 381 учащийся
Из всех попыток 53% верных

du -h -s

[Следующий шаг](#) [Решить снова](#)

Ваше решение Вы получили: 2 балла из 2

1681 105 Шаг 8

[Следующий шаг >](#)

Команда `mkdir dir{1..3}` создаст в текущей директории три поддиректории с именами dir1, dir2 и dir3, используя возможность генерации имен поддиректорий с помощью фигурных скобок и диапазона чисел.

3.7 Разное 10 из 15 шагов пройдено 7 из 7 баллов получено

Впишите в форму ниже максимально короткую команду (т.е. в которой минимально возможное число символов), которая позволит создать в текущей директории 3 поддиректории с именами dir1, dir2, dir3.

Если вы придумали команду, которая выполняет эту задачу, а система проверки сообщает вам "incorrect"/"Неверно", то скорее всего вы придумали не самую короткую команду из возможных!

Напишите текст

Правильно, молодец!

Верно решили 16 720 учащихся
Из всех попыток 40% верных

```
mkdir dir(1..)
```

Следующий шаг Решить снова

Ваше решение Вы получили **2 балла** из 2

1681 105 Шаг 10 Следующий шаг >

Выводы

Я усвоил материал и готов дальше изучать Linux.