

Отчёт по выполнению внешнего курса

Операционные системы

Идрисов Джрафер Арсенович

Содержание

Цель работы	1
Начало работы	1
1 раздел	1
2 РАЗДЕЛ	12
3 РАЗДЕЛ	23
Выводы	44

Цель работы

Овладеть базовыми принципами операционной системы Linux и развить умения, нужные для успешной работы в ней.

Начало работы

1 раздел

Я дал понять, что мне ясна суть и название курса.

Задачи

Важной частью курса является закрепление изученного материала через решение задач. И именно по результатам решения задач будет поставлена оценка за курс. На шагах с задачами рядом с полем ответа приводится число баллов, которое вы получите за её решение, а также набранный вами балл.

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 задания.

Вопрос: как называется этот курс? Чтобы ответить, выберите правильный ответ нажмите на зелёную кнопку ниже.

Выберите один вариант из списка

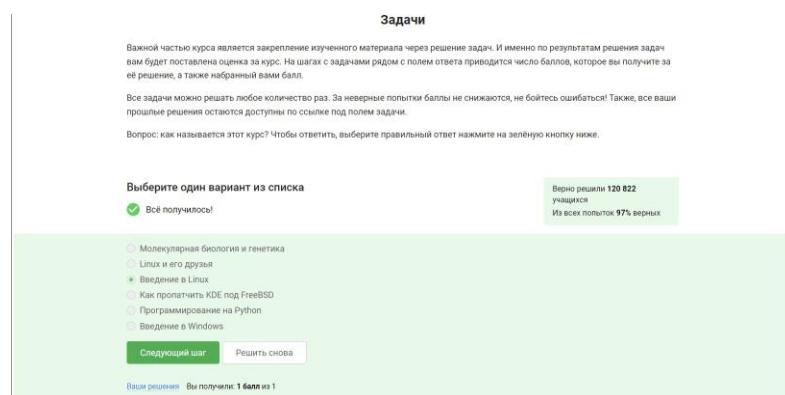
Все получилось!

Верно решили 120 822 участника
Из всех попыток 97% верных

Молекулярная биология и генетика
 Linux и его друзья
 Введение в Linux
 Как проплатить KDE под FreeBSD
 Программирование на Python
 Введение в Windows

Следующий шаг Решить снова

Ваше решение Вы получили 1 балл из 1



Осознал 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 наберете 115 баллов, в течение суток Вам присят сертификат, а если будете решать задания и получать баллы дальше – он будет автоматически обновляться.

дедлайнов 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](#).

Удачи!

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

Выберите все подходящие ответы из списка

Правильно, молодец!

Верно решили 187 011 учащихся
Из всех попыток 50% верных

За каждого неверную попытку снимается 1 балл, но баллы не могут стать меньше 0

Для получения баллов по курсу задачи нужно сдавать до дедлайна

Я не буду распространять и выкладывать в открытом доступе свои решения задач курса, чтобы другим оставалось интересно изучать самостоятельно

Дедлайнов по курсу нет, но я постараюсь проходить уроки регулярно, чтобы изучить Linux

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

Следующий шаг **Решить снова**

Выбрал подходящее

1.2 Как установить Linux 8 из 10 шагов пройдено 1 из 3 баллов получен

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

Выберите все подходящие ответы из списка

Верно.

Верно решили 95 278 учащихся
Из всех попыток 100% верных

Linux
 OS X
 Windows
 Другую

Следующий шаг **Решить снова**

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

Указал, что знаком с виртуальной машиной

1.2 Как установить Linux 9 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 учащихся
Из всех попыток 97% верных

Специальная программа для запуска одной ОС на другой ОС
 Вид операционной системы (ОС)
 Монитор
 Автомобиль будущего

Следующий шаг **Решить снова**

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

Указал, что я могу запустить Линукс.

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

Смогли ли вы запустить на своем компьютере Linux?

Выберите один вариант из списка

Верно.

Верно решили 88 497 учащихся
Из всех попыток 96% верных

Нет
 Да

Следующий шаг **Решить снова**

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

Для выполнения этого задания я создал документ в OpenOffice/LibreOffice Writer, где написал строку “Hello, Linux!” с использованием шрифта FreeMono. Если этот шрифт не был бы доступен, я бы использовал Arial или Times New Roman. После этого я сохранил документ в формате XML (Microsoft Word 2003 XML) или FODT (OpenDocument Text: Flat XML), как требовалось в задании. Такой подход позволяет сохранить структуру документа в формате XML или его аналоге, что упрощает совместимость с другими программами и облегчает обмен файлами.

Это задание дало мне возможность освоить основные функции текстового редактора OpenOffice/LibreOffice Writer и понять, как сохранять документы в различных форматах, включая XML и FODT. Теперь я готов к дальнейшему изучению Linux и использованию его в своих проектах и задачах.

1.3 Осваиваем Linux 7 из 10 шагов пройдено 2 из 6 баллов получено

Создайте документ в OpenOffice/LibreOffice Writer (аналог Microsoft Word) и напишите в нем шрифтом **FreeMono** (если такого шрифта у вас нет, то используйте **Arial** или **Times New Roman**) одну единственную строку: Hello, Linux!

После этого сохраните этот документ в формате **XML** (Microsoft Word 2003 XML) или в формате **FODT** (OpenDocument Text: Flat XML) и загрузите в форму ниже.

Подсказка: те из вас, кто пользуется Linux в виртуальной машине (см. [первое занятие](#)), могли заметить, что из вашей основной системы (Windows или OS X) не видно папки и файлов, созданных внутри Linux, а в Linux не видно файлов основной системы. На самом деле виртуальную машину VirtualBox можно настроить так, чтобы у обеих систем появился общий путь, но это не так просто. Поэтому для выполнения этого задания мы предлагаем вам обменяться небольшими файлами между вашими системами с помощью интернета, например, отправив их на почту из Linux и получив в основной системе или, например, это задание вы можете выполнить залив на этот прием из Linux.

Если же вас такое положение дел с общим файлами никак не устраивает и вы готовы действовать сразу “с места в карьер”, то смотрите [специальное видео](#) в этой недели по настройке VirtualBox. Однако мы рекомендуем перед просмотром пройти хотя бы начальные занятия первой недели курса (до “Термины, основы” включительно).

Подсказка 2: если после загрузки файла отображается “ERROR”, значит файл был сохранен не в XML или FODT формате. Переосохраните в нужном формате и попробуйте снова.

Напишите текст
✓ Все получилось!

Верно решили 46 381 учащийся
Из всех попыток 38% верных

Hello_Linux.xml (8 kB)

Следующий шаг Решить снова

В Linux, особенно в дистрибутиве Ubuntu, установочные пакеты обычно имеют расширение “.deb”. Этот формат пакетов применяется в Debian и его производных дистрибутивах, включая Ubuntu. Такие пакеты включают в себя программное обеспечение, а также сведения о зависимостях и инструкции по установке, что облегчает процесс установки и управление программами в Ubuntu.

1.3 Осваиваем Linux 8 из 10 шагов пройдено 3 из 6 баллов получено

Какое расширение имеют установочные пакеты в Linux (Ubuntu)?

Выберите один вариант из списка
✓ Хорошая работа.

Верно решили 70 807 учащихся
Из всех попыток 74% верных

.txt
.exe
.dmg
✓ .deb
.ubuntu

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

1347 725 Шаг 6 Следующий шаг >

На вкладке “About” (О программе) в плеере VLC можно найти информацию о разработчиках. Первая фамилия в списке авторов — Aimar.

Поставьте себе в систему плеер VLC (любым способом: через Software Center или скачиванием установочного пакета с сайта VLC).

Запустите, откройте Help → About (или Shift+F1) и напишите ниже первую **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).

Напишите текст

 Прекрасный ответ.

Верно решили 50 037 учащихся

Из всех попыток 62% верных

Aimar

Следующий шаг Решить снова

Ваше решение Вы получили: 2 балла из 2

1. Update Manager не используется для обновления ссылок в Software Center. Он предназначен исключительно для управления обновлениями операционной системы и установленного на ней программного обеспечения.
2. Update Manager не подходит для установки новых программ. Он используется только для управления обновлениями, а не для установки нового ПО.
3. Да, основная задача Update Manager — обновление уже установленных программ и компонентов операционной системы. Это позволяет пользователям получать самые свежие версии и исправления для программ и пакетов, установленных на их компьютерах.

Для чего можно использовать приложение Update Manager?

Выберите все подходящие ответы из списка

 Всё получилось!

Верно решили 60 759 учащихся

Из всех попыток 37% верных

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

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

Использую знание русского языка

Выберите все синонимы для "командной строки".

Выберите все подходящие ответы из списка

 Всё получилось!

Верно решили 69 413 учащихся

Из всех попыток 80% верных

- Терминал
- Ассоль
- Термин
- Консоль

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

Команда `pwd` предназначена для отображения текущей рабочей директории в командной строке. Она выводит полный путь к директории, в которой вы находитесь в данный момент. Сокращение `pwd` расшифровывается как "print working

directory" (печать рабочей директории). Команда показывает, в какой директории вы находитесь относительно корневой файловой системы.

1.4 Terminal: основы 10 из 13 шагов пройдено 2 из 5 баллов получено

Какая команда напечатает в какой директории мы сейчас находимся?

Выберите один вариант из списка

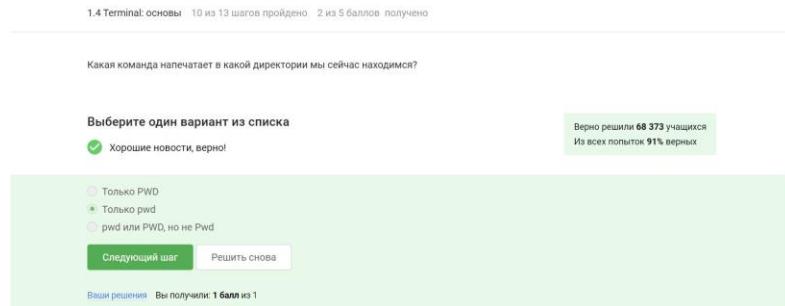
Хорошие новости, верно!

Только PWD
 Только pwd
 pwd или PWD, но не Pwd

Верно решили 68 373 учащихся
Из всех попыток 91% верных

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1



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

- Команда `ls --almost-all --human-readable -l /some/directory` использует опцию `--almost-all`, которая эквивалентна `-A`. Обе опции показывают все файлы и папки, включая скрытые, за исключением ссылок на текущую и родительскую директории.
- Команда `ls --human-readable -A -l /some/directory` - порядок опций не играет роли, поэтому `-A` и `-l` можно использовать в любом порядке.
- Команда `ls -lAh /some/directory` показывает, что опции могут быть объединены без пробелов, поэтому запись `-lAh` равнозначна `-l -A -h`.
- Команда `ls -h -A -l /some/directory` аналогична предыдущему примеру, но опции перечислены в другом порядке. Порядок перечисления опций не влияет на результат, так как `-h -A -l` эквивалентно `-l -A -h`.
- Команда `ls -Ahl /some/directory` показывает, что опции можно объединять без пробелов в любом порядке, поэтому запись `-Ahl` эквивалентна `-A -h -l`.

Все эти команды используют тот же набор опций, что и `ls -A --human-readable -l /some/directory`, и поэтому они полностью эквивалентны.

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 ls`.

Подсказка 2: в вопросах с чекбоксами/checkboxes может возникнуть ситуация, когда **все** предложенные варианты ответов 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 отмечать ни один** из них (ведь мы просим указывать только верные варианты) и нажимать кнопку "Отправить"/"Submit". Возможна и обратная ситуация, т. е. все предложенные варианты верны. В этом случае отмечайте их **всех** и нажимайте "Отправить"/"Submit".

Выберите все подходящие ответы из списка

Правильно, молодец!

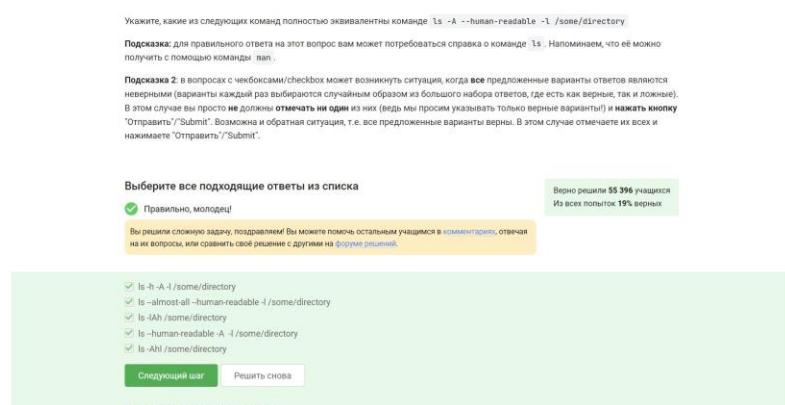
Вы решите сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свой ответ с другими на [форуме решений](#).

Верно решили 55 396 учащихся
Из всех попыток 19% верных

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

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1



- Команда `ls ~/Downloads` отображает содержимое директории `Downloads`, находящейся в домашней директории пользователя. Символ `~` представляет собой сокращение для домашней директории текущего пользователя.

2. Команда `ls ..`/Downloads покажет содержимое директории Downloads, находящейся в родительской директории относительно текущей. Это означает, что Downloads расположена на уровень выше текущей директории, например, в /home/bi.
3. Команда `ls /home/bi/Downloads` выведет содержимое директории Downloads, используя абсолютный путь к этой директории.
4. Команда `ls ../../Downloads` отобразит содержимое директории Downloads, используя относительный путь. Сначала она поднимается на уровень выше относительно текущей директории, затем переходит в Downloads.

Все эти команды позволяют отобразить содержимое директории Downloads, находясь в /home/bi/Documents, при этом исключая содержимое других директорий.

Предположим, что вы находитесь в директории /home/bi/Documents, причем /home/bi — ваша домашняя директория. Какая(ие) команда выведет содержимое /home/bi/Downloads, при этом не показывая содержимое других директорий?

Подсказка: если у вас не получается ответить на этот вопрос с использованием только теоретических знаний, то можете попробовать воспроизвести эту ситуацию на практике в своем терминале. Для того, чтобы оказаться в директории /home/bi/Documents нужно использовать команду `cd /home/bi/Documents` (она будет рассмотрена подробнее уже в следующем видеоролике).

Подсказка 2: в вопросах с чекбоксами/checkboxes может возникнуть ситуация, когда **все** предложенные варианты ответов 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 **отмечать ни 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 отмечаете их **всех** и нажимаете "Отправить"/"Submit".

`rm` — это команда для удаления файлов или директорий.

`-r` — это опция команды `rm`, обозначающая рекурсивное удаление. Это значит, что команда `rm -r` удалит все файлы и поддиректории в указанной директории, включая саму эту директорию. Благодаря этой опции `rm` может удалять целые директории вместе с их содержимым.

При вводе команды `firefox` в терминале запускается браузер Mozilla Firefox в новом процессе. Если после этого ввести команду `exit` в том же терминале, это завершит текущий терминальный сеанс, но не повлияет на процессы, запущенные из него, такие как Firefox. Таким образом, закрытие терминала не приведет к закрытию браузера Firefox.

Что произойдет, если ввести в терминал команду `firefox` (для запуска одноименного браузера), а затем ввести туда же команду `exit`?

Примечание: перед вводом этих команд в терминал у вас в системе не должен быть запущен Firefox!

Выберите один вариант из списка
 Так точно!

Верно решили 54 317 участников

Из всех попыток 53% верных

- Firefox закроется, Terminal продолжит работу
- Никто не закроется
- Terminal закроется, Firefox продолжит работу
- Обе программы закроются

Следующий шаг

Решить снова

Ваше решение Вы получили 1 балл из 1

Если запустить команду с символом `&`, программа начнет работать в фоновом режиме, как если бы она была запущена, не блокируя терминал.

Ctrl+Z: Это сочетание клавиш приостанавливает выполнение текущего процесса в терминале. Если программа была запущена без символа `&`, использование `Ctrl+Z` приостановит выполнение программы. Если программа была запущена с символом `&`, то при нажатии `Ctrl+Z` процесс также будет приостановлен.

Команда `bg` используется для возобновления приостановленного процесса в фоновом режиме после его приостановки с помощью `Ctrl+Z`.

Чему эквивалентен запуск программы с `&`?

Выберите один вариант из списка
 Верно.

Верно решили 51 830 участников

Из всех попыток 64% верных

- Запуск, `Ctrl+C, bg`
- Запуск, `Ctrl+C, fg`
- Запуск, `Ctrl+Z, bg`
- Запуск, `Ctrl+Z`

Следующий шаг

Решить снова

Ваше решение Вы получили 1 балл из 1

Запустил и увидел

Скачайте [файл](#) с программой; сделайте её исполняемым, запустите и скопируйте то, что он выведет на экран, в форму ниже.

Напишите текст

Здорово, всё верно.

Верно решили 43 240 участников

Из всех попыток 58% верных

0000-00-00 00:00:00

Control sum: 910

Следующий шаг

Решить снова

Ваше решение Вы получили 2 балла из 2

При запуске программы в терминале, её стандартный вывод (`stdout`) и поток ошибок (`stderr`) обычно направляются в терминал. Это позволяет пользователю видеть все ошибки или сообщения, которые программа может выдавать, и реагировать на них соответственно.

1.6 Ввод / вывод 5 из 7 шагов пройдено 1 из 4 баллов получен

Куда по умолчанию выводится поток ошибок из программы, запущенной в терминале?

Выберите один вариант из списка

Верно. Так держать!

Верно решили 49 142 учащихся
Из всех попыток 71% верных

- В файл err.txt
- Нигде
- В файл stderr
- На экран

Следующий шаг

Решить снова

Ваше решение Вы получили: 1 балл из 1

1. Команда `program 2>> file.txt` перенаправляет поток ошибок программы `program` в файл `file.txt`. Оператор `>>` добавляет ошибки в файл, сохраняя при этом существующие данные или создавая новый файл, если такого файла нет.
2. Команда `program 2> file.txt` также перенаправляет поток ошибок программы `program` в файл `file.txt`. Однако оператор `>` перезаписывает файл, если он уже существует, вместо добавления новых данных. Таким образом, при первом запуске команда создаст файл `file.txt` и запишет в него поток ошибок из программы `program`, но при последующих запусках файл будет перезаписываться, а не дополняться.

1.6 Ввод / вывод 6 из 7 шагов пройдено 2 из 4 баллов получено

Какие (какие) из команд создадут файл `file.txt` и запишут в него поток ошибок программы `program`? Считайте, что в момент запуска программы файл `file.txt` не существует.

Выберите все подходящие ответы из списка

Так точно!

Верно решили 45 805 учащихся
Из всех попыток 30% верных

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

- `program << file.txt`
- `program 2>> file.txt`
- `program < file.txt`
- `program 2> file.txt`
- `program >> file.txt`
- `program file.txt<2`

Следующий шаг

Решить снова

Ваше решение Вы получили: 1 балл из 1

Когда программы объединяются в конвейер, стандартный вывод (`stdout`) первой программы передается на стандартный вход (`stdin`) второй программы. Тем не менее, поток ошибок (`stderr`) каждой программы не передается через конвейер и продолжает выводиться в терминале. Это означает, что если ошибки не перенаправляются специально, они будут отображаться на экране, а не передаваться через конвейер.

1.6 Ввод / вывод 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 скажите показанную на занятии программу `interceptor.py`, создайте файл `names.txt` со списком имен (записав часть из них с ошибкой, т. е. с маленькой буквы) и постройте конвейер, аналогичный тому, что был показан на видео. Запустите конвейер и посмотрите на результат! Обратите внимание, что указанная программа работает корректно только со именами, набранными латинскими буквами!

Выберите один вариант из списка

Абсолютно точно.

Верно решили 46 719 учащихся
Из всех попыток 68% верных

- Исчезают (никуда не выводятся)
- Выводятся на экран
- Записываются в файл `err.txt`

Следующий шаг

Решить снова

Ваше решение Вы получили: 2 балла из 2

Команда `cd /home/alex/` меняет текущую директорию на `/home/alex/`.

Команда `wget -P /home/alex/Pictures -O 1.jpg http://example.com/example.jpg` скачивает изображение с указанного URL (`http://example.com/example.jpg`) и сохраняет его в директорию `/home/alex/Pictures` под именем `1.jpg`. Флаг `-O 1.jpg` указывает, что файл будет сохранен под этим именем в указанной директории.

1.7 Скачивание файлов из интернета 5 из 7 шагов пройдено 1 из 4 баллов получен

В каком файле на диске окажется картинка, если для её 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

Всё правильно.

Верно решили 46 359 учащихся
Из всех попыток 36% верных

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

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

-q или --quiet: Эти опции подавляют вывод сообщений при выполнении команды `wget`, что позволяет ей работать в тихом режиме без отображения лишней информации на экране.

1.7 Скачивание файлов из интернета 6 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... и т.д.)?

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

Выберите один вариант из списка

Хорошие новости, верно!

Верно решили 45 653 учащихся
Из всех попыток 65% верных

-nI или -no-verbose
 -v или -verbose
 -q или -quiet

Следующий шаг Решить снова

Ваше решение Вы получили: 2 балла из 2

`wget -r:` Эта опция указывает `wget` рекурсивно следовать по ссылкам на веб-странице, чтобы загрузить связанные файлы.

-l 1: Эта опция ограничивает глубину рекурсии до 1 уровня, что означает, что `wget` загрузит только те файлы, которые напрямую связаны с начальной страницей.

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

Таким образом, `wget` скачает все файлы с расширением `.jpg` и HTML-файлы на глубине 1. Однако, поскольку используется опция `-A jpg`, все скачанные HTML-файлы будут удалены после завершения загрузки, поскольку они не соответствуют заданному фильтру.

1.7 Скачивание файлов из интернета 7 из 7 шагов пройдено 4 из 4 баллов получено

Пусть на некоторой web-странице есть ссылки на картинки в форматах png и jpg, а также ссылки на другие страницы сайта (обычные html файлы). Какие файлы будут скачаны на компьютер, если запустить wget -r -l 1 -A jpg и передать в качестве аргумента ссылку на эту web-страницу? Выберите наиболее полный ответ!

Подсказка: для правильного ответа на этот вопрос, вам может не хватить справочной информации от man wget, т.к. в поведении опции -A есть некоторые исключения. Рекомендуем просмотреть соответствующий раздел в полном описании утилиты wget на сайте разработчиков.

Выберите один вариант из списка

Отличное решение!

Верно решили 43 677 учащихся
Из всех попыток 49% верных

- Будут скачаны jpg и html файлы, но все html будут удалены
- Не будет скачано ничего
- Будут скачаны только картинки (jpg и png), но все jpg будут удалены
- Будут скачаны только jpg файлы
- Будут скачаны png и html файлы, но все html будут удалены
- Будут скачаны только картинки (jpg и png), но все png будут удалены

Следующий шаг

Решить снова

Ваше решение Вы получили: 1 балл из 1

gzip: Это архиватор, который сжимает файлы в формат gzip (.gz). Он работает с отдельными файлами, не сохраняя структуру директорий. Обычно после распаковки архива в формате gzip сам архив удаляется по умолчанию.

zip: Это архиватор, который сжимает файлы в формат zip (.zip). В отличие от gzip, zip сохраняет не только один файл, но и структуру директорий, что позволяет архивировать и извлекать целые папки. После распаковки архив zip по умолчанию не удаляется, оставаясь в исходном месте.

1.8 Работа с архивами 5 из 7 шагов пройдено 1 из 3 баллов получен

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

Здорово, всё верно.

Верно решили 44 111 учащихся
Из всех попыток 78% верных

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

Следующий шаг

Решить снова

Ваше решение Вы получили: 1 балл из 1

zip: Команда zip позволяет архивировать файлы и директории в формате .zip.

tar: Команда tar создает архивы, которые могут содержать файлы и директории, сохраняя их структуру и иерархию.

1.9 Работа с архивами 6 из 7 шагов пройдено 2 из 3 баллов получено

Какие из перечисленных программ-архиваторов могут создать архив из директории с файлами?

Выберите все подходящие ответы из списка

Так точно!

Верно решили 43 962 учащихся
Из всех попыток 39% верных

- gzip
- zip
- tar

Следующий шаг

Решить снова

Ваше решение Вы получили: 1 балл из 1

-c: Создание нового архива.

-j: Применение алгоритма сжатия bzip2 для сжатия файлов.

-v: Отображение подробной информации о процессе создания архива.

-f: Указание имени архива.

1.8 Работа с архивами - 7 из 7 шагов пройдено 3 из 3 баллов получено

Какой набор опций нужно указать программе `tar`, чтобы запаковать файлы в `my_archive.tar.bz2`?

Выберите один вариант из списка
✓ Абсолютно точно.

Верно решили 42 625 учащихся
Из всех попыток 60% верных

- cf
- zf
- rtf
- <zf
- >zf

[Следующий шаг](#) [Решить снова](#)

Ваше решение Вы получили: 1 балл из 1

Маска `*.jpg` находит все файлы, чьи имена заканчиваются на `.jpg`, и не включает файлы, например, с именем `Alexey.jpg`, потому что они не соответствуют этому шаблону.

1.9 Поиск файлов и слов в файлах - 4 из 6 шагов пройдено 1 из 4 баллов получено

Какая маска команды `find` НЕ найдет файл `Alexey.jpg`?

Выберите все подходящие ответы из списка
✓ Абсолютно точно.

Верно решили 40 928 учащихся
Из всех попыток 22% верных

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

- Alex*
- *?
- **
- * jpg
- Alexey.jpg
- Alexey*

[Следующий шаг](#) [Решить снова](#)

Ваше решение Вы получили: ... из 1

Ищем, учитывая нижний регистр.

1.9 Поиск файлов и слов в файлах - 5 из 6 шагов пройдено 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
✓ Так точно!

Верно решили 40 042 учащихся
Из всех попыток 27% верных

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

- The beautiful-world is not enough
- world
- World
- The beautifulworld is not enough
- The world is not enough
- The World Is Not Enough
- The "world" is not enough
- The word is not enough

[Следующий шаг](#) [Решить снова](#)

Ваше решение Вы получили: 1 балл из 1

Программа выполнилась успешно.

1.Ч Поиск файлов и слов в файлах 5 из 6 шагов пройдено 4 из 4 баллов получено

Скачайте архив с произведениями Шекспира. Вам нужно сконвертировать файл, в котором будут все строчки из этих произведений, содержащие "love", и загрузить этот файл в форму.

Подсказка: для того, чтобы результаты поиска записались сразу в файл, можно воспользоваться перенаправлением вывода (см. занятие Ввод/Выход).

Напишите текст

Хорошая работа.

Верно решили 34 497 учащихся

Из всех попыток 57% верных

outlet.txt (7 KB)

Ваше решение Вы получили: 2 балла из 2

1. Выполнение ресурсоемких вычислений: Удаленные серверы с более мощным оборудованием и большим объемом вычислительных ресурсов могут использоваться для сложных задач, требующих много памяти и времени.
2. Хранение общедоступных данных: Удаленные серверы пригодны для хранения информации, к которой нужен доступ всем пользователям интернета, например, веб-контента, публичных файлов и прочего.
3. Сохранение больших объемов данных: Удаленные серверы обладают большим пространством для хранения, что делает их подходящими для больших массивов данных, таких как архивы, резервные копии или обработка больших данных.
4. Хранение конфиденциальной информации: Удаленные серверы можно настроить для хранения данных, доступ к которым должен быть ограничен определенной группой лиц. Сюда входят данные клиентов, финансовая информация или другие чувствительные сведения.

2 РАЗДЕЛ

2.1 Знакомство с сервером 5 из 6 шагов пройдено 1 из 2 баллов получен

Для каких задач можно использовать удаленный сервер?

Выберите все подходящие ответы из списка

Хорошая работа.

Верно решили 41 258 учащихся

Из всех попыток 54% верных

- Хранение конфиденциальных данных (т.е. доступ к ним должны иметь только ограниченный круг лиц)
- Хранение общедоступных данных (например, доступных для всех пользователей интернета)
- Хранение больших объемов данных
- Выполнение сложных (затратных по памяти и времени) вычислений

Ваше решение Вы получили: 1 балл из 1

Ключ `id_rsa.pub` — это открытый ключ, который используется для проверки подписи, а не для её создания. Он не содержит приватных данных, поэтому его можно безопасно распространять и применять для аутентификации на удаленных серверах.

Предположим программа ash-keygen создала вам два ключа: id_rsa и id_rsa.pub. Какой из этих ключей можно без опаски пересыпать по интернету?

Выберите один вариант из списка
 Хорошие новости, верно!

Верно решили 40 966 учащихся
 Из всех попыток 75% верных

- Оба
- Ни один нельзя
- id_rsa.pub
- id_rsa

[Следующий шаг](#)

[Решить снова](#)

Ваше решение Вы получили: 1 балл из 1

- **scp:** Эта команда предназначена для копирования файлов между компьютерами через SSH.
- **-r:** Этот флаг говорит scp выполнить рекурсивное копирование, что означает, что папка stepic будет скопирована со всем её содержимым, включая подпапки.
- **stepic:** Имя папки, которую требуется скопировать.
- **username@server:** Указывает имя пользователя и адрес сервера, на который нужно скопировать файлы.
- **~/:** Это символизирует домашнюю директорию пользователя на сервере, куда будут скопированы файлы.

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

Выберите один вариант из списка
 Так точно!

Верно решили 38 014 учащихся
 Из всех попыток 57% верных

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

[Следующий шаг](#)

[Решить снова](#)

Ваше решение Вы получили: 1 балл из 1

1. **Проверка и восстановление интернет-соединения:** Первым делом убедитесь, что компьютер подключен к интернету. Если соединение отсутствует, его следует настроить или восстановить.
2. **Обновление списка пакетов APT:** Команда sudo apt-get update обновляет список пакетов из источников APT. Это полезно, если информация о доступных пакетах устарела или изменилась.

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

Выберите все подходящие ответы из списка

Правильно, молодец!

Верно решили 36 659 учащихся

Из всех попыток 21% верных

Вы решите сложную задачу, поздравляем! Вы можете помочь оставшимся в [комментариях](#), отвечая на них вопросы, или сравнить своё решение с другими на [форуме решений](#).

- `sudo apt-get install --only-upgrade program`
- Проверка интернет соединения и его установка, если соединения нет.
- `sudo apt-get update`
- Проверка места на диске и его очистка, если диск переполнен.

Следующий шаг

Решить снова

Ваше решение Вы получили: 1 балл из 1

Image

1. **Передача файлов с компьютера на сервер:** Filezilla позволяет загружать файлы с локального компьютера на удаленный сервер.
2. **Передача файлов с сервера на компьютер:** Filezilla также может скачивать файлы с удаленного сервера на локальный компьютер.

Filezilla позволяет просматривать содержимое директорий как на локальном компьютере, так и на удаленном сервере. Её основная функция - передача файлов между локальным компьютером пользователя и удаленным сервером.

Для чего можно использовать программу Filezilla?

Выберите все подходящие ответы из списка

Отлично!

Верно решили 36 387 учащихся

Из всех попыток 49% верных

- Для просмотра содержимого директорий на своем компьютере
- Для копирования файлов со своего компьютера на сервер
- Для копирования файлов с сервера на свой компьютер
- Для запуска программы на сервере
- Для установки программы на сервер

Следующий шаг

Решить снова

Ваше решение Вы получили: 1 балл из 1

1. **Настроить сервер для отображения графики на удаленном компьютере:** Это может быть сделано различными способами, например, через удаленный рабочий стол или графический сервер (например, X11), что позволит программе выводить графический интерфейс на удаленный компьютер.
2. **Узнать, есть ли терминальная версия этой программы:** В некоторых случаях для программы может существовать командная (CLI) версия, которую можно запустить на сервере без графического интерфейса. Если такая версия доступна, её можно использовать для выполнения задач на сервере, не полагаясь на графический интерфейс.

Что можно сделать, если требуется запустить на сервере программу, для работы которой нужен не терминал, а экран?

Выберите все подходящие ответы из списка

Верно. Так держать!

Верно решили 35 473 учащихся

Из всех попыток 42% верных

- Настроить сервер, чтобы он поддерживал вывод информации на экран компьютера
- Ничего сделать нельзя
- Проверить, есть ли другая версия этой программы (специально для терминала)
- Запустить программу на своем компьютере

Следующий шаг

Решить снова

Ваше решение Вы получили 1 балл из 1

1. **man program:** Команда `man` (сокращение от “manual”) отображает справочные страницы программы в терминале. Когда вы вводите `man program`, открывается подробное руководство с описанием программы, доступными опциями и примерами использования.
2. **help program:** Эта команда предоставляет встроенную справку о командах в командной оболочке (например, в `bash`). Однако, она может быть недоступна для всех программ, работая только для команд, встроенных в оболочку.
3. **program -help:** Многие программы поддерживают опцию `--help`, которая выводит краткое руководство по использованию программы и её параметрам. В некоторых случаях встречаются другие варианты опций, такие как `-help`, `-h`, или `-H`.

Как обычно можно вызвать справочную информацию о программе `program ?`

Выберите все подходящие ответы из списка

Отличное решение!

Верно решили 34 902 учащихся

Из всех попыток 22% верных

Вы решите сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить свой ответ с другими на [форуме решений](#).

- `man program`
- `program --help` (в некоторых программах бывает еще `-help` или `-h`)
- `help program`
- `program ?`

Следующий шаг

Решить снова

Ваше решение Вы получили 1 балл из 1

Чтобы узнать, какие форматы данных поддерживает программа FastQC, нужно обратиться к её справочной информации. Это можно сделать, запустив команду `./fastqc --help` в каталоге, где находится исполняемый файл `fastqc`.

Однако, после просмотра справочной информации, оказывается, что в ней нет прямого указания на поддерживаемые форматы данных. Программа FastQC предназначена для анализа качества секвенированных данных и обычно работает с файлами в форматах FASTQ и BAM. Но без точного текста справки невозможно определить, какие форматы данных программа FastQC поддерживает на входе.

Помогите стране по программе FastQC (имеется в виде варианта для запуска в Терминале) и определите, какие форматы данных он может принимать **из ввода**.

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

Подсказка: если программа FastQC еще нет на вашем компьютере, то ее можно установить командой `sudo apt-get install fastqc` (или в некоторых версиях `sudo apt-get install fastqc`) или найти ее в Software Center по запросу `fastqc`. К сожалению, на некоторых дистрибутивах Linux у вас может не получится установить FastQC обычным способом (по ключевым словам `fastqc`, `fastqc`, `fastqc` ничего не будет найдено). В этом случае установка будет сложнее, описанная ей подробнее.

1. Откройте терминал, попробуйте выполнить команду java. Если получите сообщение, что такая команда не найдена, то переходите к шагу 2, иначе пропустите шаг 3.

2. Вам нужно установить языка Java, на Ubuntu это можно сделать с помощью `sudo apt-get install default-jre`.

3. Скачайте и распакуйте архив с FastQC (можно это сделать прямиком в терминале с использованием `wget` и `tar`).

4. Файл запуска FastQC называется `fastqc` и лежит той директории, куда произведена распаковка архива, например, `fastqc/bin/fastqc` (если директория `bin` лежит в любой другой директории, то вам нужно сделать исполнимым (прави помошь файлу) `chmod +x fastqc`). Перед первым запуском его нужно сделать исполняемым (прави помошь файлу). Если запустить его без параметров, то будет открыта графическая версия программы, а если указать опции или аргументы, например, `-help`, то будет запущена версия для терминала.

Выберите все подходящие ответы из списка

Всё получилось!

Было решено 32 124 учащихся
Из всех попыток 25% верных

fasta
 bam, sam
 fastq
 seq

Следующий шаг Решить снова

clustalw: название исполняемого файла Clustal, представляющего версию программы, которая работает в терминале.

test.fasta: имя входного файла, содержащего последовательности для множественного выравнивания.

-align: опция, указывающая Clustal на проведение множественного выравнивания последовательностей.

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`.

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

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

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

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

Напишите текст

Абсолютно точно.

Верно решили 28 700 учащихся
Из всех попыток 41% верных

clustalw test.fasta -align

Следующий шаг Решить снова

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

1. Команда `fg %1` выводит задачу номер 1 в передний план, возвращая управление программой `program1`. Однако мы немедленно прерываем ее выполнение комбинацией клавиш `Ctrl+C`.
2. Затем, `fg %2` переводит задачу номер 2 в передний план, делая активной программу `program2`. Мы приостанавливаем эту программу, нажав `Ctrl+Z`.

После этих действий информация о программах program1 и program2 больше не отображается в списке `jobs`, и остаётся информация только о программе program3.

2.4 Контроль запускаемых программ 8 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

Правильно, молодец!

Верно решил 34 021 учащийся
Из всех попыток 61% верных

Только о program1 и program2
 Только о program3
 Только о program2 и program3
 Только о program1 и program3

Следующий шаг Решить снова

Идентификаторы, используемые в утилитах `jobs`, `top`, и `ps`, имеют разные назначения и работают по разным принципам:

1. **jobs:** Команда `jobs` выводит список заданий, запущенных в текущем сеансе терминала. Здесь идентификаторы представляют собой порядковые номера заданий в рамках этого сеанса. Они могут не быть уникальными и могут меняться по мере завершения или приостановки заданий.
2. **top:** Эта утилита показывает список процессов, работающих в системе. Здесь идентификаторы — это уникальные процессные идентификаторы (PID), которые операционная система присваивает каждому процессу. Эти идентификаторы уникальны в пределах всей системы и не изменяются, пока процесс не завершится.
3. **ps:** Утилита `ps` также показывает список процессов в системе. Идентификаторы процессов здесь — это те же уникальные PIDs, которые присваиваются каждому процессу операционной системой, оставаясь неизменными в течение всего жизненного цикла процесса.

Таким образом, идентификаторы в `jobs` не совпадают с идентификаторами в `top` и `ps`, потому что `jobs` показывает информацию о заданиях в текущем сеансе терминала, в то время как `top` и `ps` отображают данные о процессах в системе в целом.

2.4 Контроль запускаемых программ 9 из 11 шагов пройдено 2 из 4 баллов: получено

`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`?

Выберите один вариант из списка

Всё правильно.

Верно решили 33 797 учащихся
Из всех попыток 52% верных

Одинаковые только у `jobs` и `ps`
 У всех разные
 Одинаковые только у `ps` и `top`
 У всех одинаковые

Следующий шаг Решить снова

Ваше решение: Вы получили: 1 балл из 1

Команда `kill` в UNIX-подобных системах, включая Linux, предназначена для отправки сигналов процессам. Сигнал номер 9, также известный как SIGKILL, применяется для принудительного завершения процесса. Когда процесс получает этот сигнал, он не может его обработать или выполнить какие-либо действия перед завершением, что приводит к немедленному завершению процесса.

2.4 Контроль запускаемых программ 10 из 11 шагов пройдено 3 из 4 баллов получено

С помощью какой команды можно мгновенно завершить остановленный процесс?

Выберите один вариант из списка

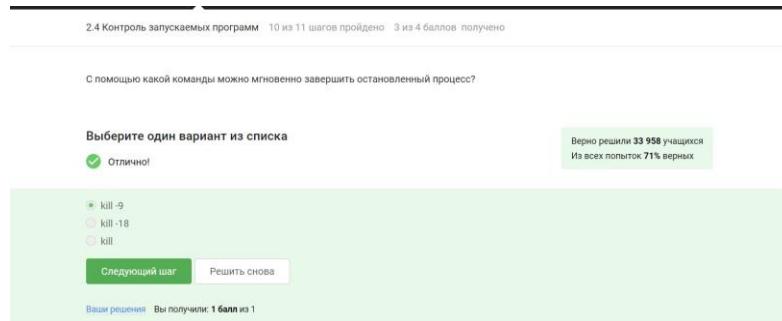
Отлично!

Верно решили 33 958 учащихся
Из всех попыток 71% верных

kill -9
 kill -18
 kill

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1



Когда процесс приостанавливается с помощью `Ctrl+Z`, он переходит в состояние приостановки (Suspended). Если после этого использовать команду `kill` без дополнительных опций, то процесс завершится, как только будет возобновлен. То есть при вводе команды `fg` или `bg` для возобновления приостановленного процесса, он сразу завершится.

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

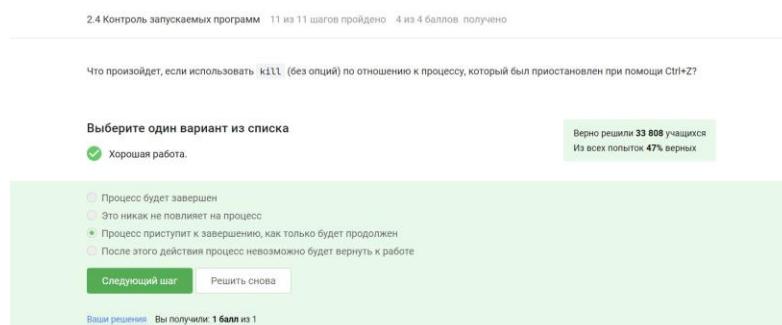
Хорошая работа.

Верно решили 33 808 учащихся
Из всех попыток 47% верных

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

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1



Когда многопоточное приложение приостанавливается с помощью `Ctrl+Z`, все его потоки останавливаются, и оно перестает потреблять процессорные ресурсы. В результате, использование CPU для этого приложения снижается до 0%, поскольку ни один из его потоков не выполняется.

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 рекомендую вам ненадолго отложить этот шаг и досмотреть занятие до конца. В следующих видеоФрагментах и заданиях будет показан пример многопоточного приложения (программы `boinc2`). Тестовые данные для запуска этой программы можно найти в последнем задании этого урока.

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

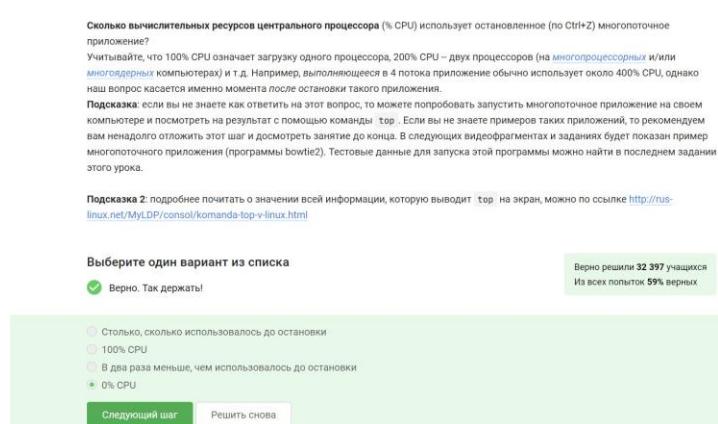
Выберите один вариант из списка

Верно. Так держать!

Верно решили 32 397 учащихся
Из всех попыток 59% верных

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

Следующий шаг Решить снова



Когда приложение приостанавливается с помощью `Ctrl+Z`, оно не освобождает память, которую использовало в момент приостановки. Следовательно, объем занятой памяти останется на том же уровне, что и в момент остановки приложения.

2.5 Многопоточные приложения 11 из 14 шагов пройдено 2 из 6 баллов получено

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

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

Выберите один вариант из списка
✓ Прекрасный ответ.

64 KB
По 64 KB на каждый поток
Несколько
* Столько, сколько оно потребляло в момент остановки

Верно решили 32 287 учащихся
Из всех попыток 56% верных

Следующий шаг Решить снова

Завершение отдельного потока в многопоточном приложении может быть сложной задачей, так как потоки взаимодействуют с общими ресурсами и данными. Принудительное завершение одного потока может привести к непредсказуемым последствиям для приложения в целом. В большинстве случаев, потоки завершаются вместе с завершением всего приложения. Однако если нужно остановить многопоточное приложение, обычно используется команда `kill` с идентификатором процесса, чтобы завершить его целиком.

2.5 Многопоточные приложения 12 из 14 шагов пройдено 3 из 6 баллов получено

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

Выберите все подходящие ответы из списка
✓ Всё получилось!

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](#).

Сочетанием клавиш `Ctrl+C`
Командой `threadkill`
✓ Никак
Командой `kill -lthread`

Верно решили 31 649 учащихся
Из всех попыток 32% верных

Следующий шаг Решить снова

Из двух инструментов `Bowtie2`, только `bowtie2` поддерживает многопоточность. Подпрограмма `bowtie2-build`, предназначенная для предварительной индексации референсного генома, выполняется последовательно и не может работать в нескольких потоках одновременно.

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 вызывать при помощи `--help`) и ответьте на вопрос – какой(ие) из этих шагов можно выполнить в несколько потоков?

Выберите один вариант из списка

Здорово, всё верно.

Верно решили 31 625 учащихся
Из всех попыток 58% верных

Оба
 Только bowtie2
 Никакой
 Только bowtie2-build

Следующий шаг **Решить снова**

Ваши решения Вы получили: 1 балл из 1

Данный отчет представляет собой результат работы программы bowtie2 и содержит информацию о выравнивании прочтений на референсный геном. Вот что означают эти числа:

- “306174 reads” означает, что обработано всего 306174 прочтения.
- “306174 (100.00%) were unpaired” указывает на то, что все прочтения были одиночными, а не парными.
- “11 (0.00%) aligned 0 times” показывает, что 11 прочтений не удалось выровнять ни разу.
- “305580 (99.81%) aligned exactly 1 time” указывает, что 305580 прочтений были выровнены ровно один раз.
- “583 (0.19%) aligned >1 times” означает, что 583 прочтения были выровнены более одного раза.
- “100.00% overall alignment rate” означает, что общее значение выравнивания составляет 100%.

Эта информация отражает качество и количество выравнивания прочтений на геном.

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

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

Примечание: если у вас не очень сильный компьютер, то работа bowtie2 на предложенных данных может занять достаточно продолжительного времени. Если вы не хотите ждать, то можете использовать альтернативные (сильно уменьшенные) версии референсного генома (`reference`) и рядов (`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
 Абсолютно точно.

Верно решили 23 694 учащихся
Из всех попыток 65% верных

```
echo "306174 reads; of these:  
306174 (100.00%) were unpaired; of these:  
11 (0.00%) aligned 0 times  
305580 (99.81%) aligned exactly 1 time  
583 (0.19%) aligned >1 times  
100.00% overall alignment rate" > bowtie.log
```

Следующий шаг **Решить снова**

Ваши решения Вы получили: 2 балла из 2

Когда вторая вкладка терминала выполняет команду `fg`, она пытается вернуть в активное состояние приостановленный процесс, который был запущен в первой вкладке. Однако, поскольку вкладки терминала изолированы друг от друга, вторая вкладка не может управлять процессами из первой. В результате терминал выдаст

ошибку, указывая, что нет процессов для возобновления, поскольку команда `fg` не может находить процессы из других вкладок.

2.6 Менеджер терминалов tmux 13 из 19 шагов пройдено 1 из 7 баллов получен

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
✓ Отличное решение!

Верно решили 30 754 учащихся
Из всех попыток 74% верных

Процесс переместится во вторую вкладку и продолжит работу
 Процесс вернется к работе в исходной вкладке
 Процесс переместится во вторую вкладку, но останется в режиме "приостановки"
 Терминал сообщит, что нет процесса для запуска в `fg`

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

Если в последней вкладке tmux ввести команду `exit`, это приведёт к закрытию этой вкладки и, следовательно, к завершению сеанса tmux, поскольку больше не останется активных окон или вкладок.

2.6 Менеджер терминалов tmux 14 из 19 шагов пройдено 2 из 7 баллов получено

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

Выберите один вариант из списка
✓ Абсолютно точно.

Верно решили 30 421 учащийся
Из всех попыток 76% верных

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

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

При закрытии терминала соединение с сервером разрывается, но процессы внутри tmux на сервере продолжают работать, так как tmux функционирует в фоновом режиме и сохраняет состояние сессии даже после потери соединения.

2.6 Менеджер терминалов tmux 15 из 19 шагов пройдено 3 из 7 баллов получено

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?

Выберите один вариант из списка
✓ Верно. Так держать!

Верно решили 30 220 учащихся
Из всех попыток 63% верных

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

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

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

Что произойдет, если запустить процесс в фоновом режиме в одной из вкладок tmux, а затем принудительно закрыть эту вкладку (`Ctrl+B, X`)?

Выберите один вариант из списка

Верно. Так держать!

Верно решили 30 092 учащихся

Из всех попыток 61% верных

tmux выдаст предупреждение и не даст закрыть вкладку

Вкладка закроется, а вместе с ней пропадет и запущенный в ней процесс

Вкладка закроется и процесс перейдет во вкладку, ближайшую из открытых (если есть, то слева, иначе справа)

Следующий шаг

Решить снова

Ваше решение Вы получили: 1 балл из 1

Комбинация клавиш `Ctrl+B, ,` используется в tmux для переименования текущей вкладки.

Задание на самостоятельное изучение tmux.

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

Выберите один вариант из списка

Правильно.

Верно решили 29 445 учащихся

Из всех попыток 54% верных

Ctrl+B и 0

Ctrl+B и . (точка)

Ctrl+B и r

Ctrl+B и t

Ctrl+B и . (запятая)

Следующий шаг

Решить снова

Ваше решение Вы получили: 1 балл из 1

1. **Вкладки можно разделять как горизонтально, так и вертикально:** tmux поддерживает разделение вкладок двумя способами, что позволяет создавать горизонтальные и вертикальные области для одновременной работы с несколькими процессами.
2. **Можно закрывать отдельные “части” вкладки:** tmux позволяет закрывать конкретные области во вкладке, освобождая место или завершая процессы в них.
3. **Команды для разделения применяются только к активной вкладке:** Разделение в tmux действует только на текущую вкладку, не затрагивая другие, что обеспечивает независимое управление каждой вкладкой.
4. **Если разделить вкладку сначала по горизонтали, а затем по вертикали, получится три “части”:** tmux поддерживает комбинированное разделение, что позволяет создавать сложные макеты с тремя или более областями в одной вкладке.

Утверждение, что можно перемещаться между областями в разделенной вкладке с помощью стрелок, неверно, потому что tmux использует другие комбинации клавиш для этого.

Задание на самостоятельное изучение tmux.

Кроме создания нескольких вкладок, tmux умеет еще и разделять (split) одну вкладку на несколько, например, горизонтальной чертой на верхнюю и нижнюю или вертикальной чертой на левую и правую. Разделение может быть полезно, например, чтобы запустить процесс в верхней половине вкладки, а продолжать работу в нижней и одновременно следить за тем, что происходит с процессом. Для "горизонтального" разделения используется (Ctrl+B и), а для "вертикального" – (Shift+B и).

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

Выберите все подходящие ответы из списка

Правильно.

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

Верно решили 24 656 учеников
Из всех попыток 23% верных

По половинкам "разделенной" вкладки можно перемещаться при помощи обычного нажатия на стрелочки (без использования Ctrl+B)

Можно закрыть одну из "частей" вкладки вытолкивая (Shift+B и)

Команды "разделений" действуют только в текущей вкладке tmux, а не во всех вкладках одновременно

Вкладку можно разделить и горизонтально, и вертикально, и даже по нескольку раз – просто используем нужные команды "разделений" необходимое количество раз

Команды "разделений" действуют сразу во все вкладках tmux одновременно

По половинкам "разделенной" вкладки можно перемещаться при помощи (Ctrl+B и стрелочек)

[Следующий шаг](#) [Решить снова](#)

3 РАЗДЕЛ

В редакторе Vim команда :q используется для выхода. Если в файле были внесены изменения, и вы хотите выйти без их сохранения, можно воспользоваться командой :q!. После ввода команды необходимо нажать клавишу Enter.

3.1 Текстовый редактор vim 8 из 12 шагов пройдено 1 из 7 баллов получен

Вы прошли больше 80% курса, оставьте отзыв

Оставить отзыв Нет, спасибо

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.

Верно решили 32 523 ученика
Из всех попыток 69% верных

":" , затем "Enter"

":" , затем "q"

"Ctrl", затем "x"

":" , затем 'q', затем "Enter"

"q"

[Следующий шаг](#) [Решить снова](#)

Ваша рецензия Вы получили 1 балл из 1

- Разница между `w` и `W` в том, что `w` перемещает курсор по "словам", которые отделены пробелами или другими знаками пунктуации, тогда как `W` перемещает курсор по "большим словам", которые разделены только пробелами. Следовательно, `W` игнорирует все символы, кроме пробелов.
- После 10 нажатий на `W` курсор действительно окажется там же, где и после 10 нажатий на `w`, если строки не содержат "больших слов".

При перемещении в vim "то слова" есть небольшая разница в том, используем мы маленькую (к. е. `w`) 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 `w` и `WORD`.

А для того, чтобы убедиться, что вы разобрались, отмечте ниже все верные утверждения про следующую строку:

`Strange... TEXT 1x1, page 1, 2x2 YES`

Примечание: во всех утверждениях имеется ввиду, что мы находимся в редакторе vim, включен нормальный режим работы и курсор находится в самом начале строки.

Подсказка: чтобы вызвать vim-справку, например, перемещению `w`, нужно открыть vim и ввести команду :help w . Вы попадете в то место справки, где описано это перемещение, а так как все перемещения описаны рядом, то двигаясь по тексту вверх и вниз можно прочитать о про `e` и про `b` и, самое главное, про `word` и `WORD`. Кроме того, можно вызвать сразу справку по термину `word` при помощи :help word . Чтобы закрыть справку, нужно ввести команду :q .

Выберите все подходящие ответы из списка

Верно.

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 25 385 ученика
Из всех попыток 20% верных

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

Чтобы попасть в конец строки, нужно одинаковое число нажатий, что на W, что на w

Нажимая только на W, нельзя переместить курсор на "

В этой строке 5 "слов" (word)

В этой строке 5 "больших слов" (WORD)

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

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

Предположим, что в текстовом файле записана одна единственная строка:

```
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 выполняют такое редактирование? В этих наборах нажатие на клавишу `Esc` обозначается как `<Esc>` (т.е. знаки “`<`” и “`>`” не несут отдельного смысла).

Примечание: во всех утверждениях имеется в виду, что мы находимся в редакторе vim, включен нормальный режим работы и курсор находится в самом начале строки.

Выберите все подходящие ответы из списка

Отличное решение!

Верно решили 23 655 учащихся
Из всех попыток 16% 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](#).

x2wwywPp
 ddithree four four four five<Esc>
 d2dwywPp
 d2wwywPp
 xxxxxxxxwywPp
 d2wwifour four <Esc>

[Следующий шаг](#)

[Решить снова](#)

- % указывает, что замена будет применена ко всему файлу.
- `s/Windows/Linux/` означает замену всех вхождений строки “Windows” на “Linux”.
- g указывает на глобальную замену в каждой строке, то есть все случаи “Windows” в строке будут заменены на “Linux”, а не только первое вхождение.

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 на `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

Всё получилось!

Верно решил 24 631 учащийся
Из всех попыток 57% верных

%s/Windows/Linux

[Следующий шаг](#)

[Решить снова](#)

Когда вы находитесь в нормальном режиме Vim и нажимаете клавишу “v”, вы переходите в режим визуального выделения (Visual). В этом режиме можно выделять текст с помощью клавиш навигации или других команд, например, W, e, \$. Выделенный фрагмент будет подсвечен, и в нижней части редактора появится надпись “VISUAL”, указывающая, что вы находитесь в режиме выделения.

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

Хорошие новости, верно!

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 свою решение с другими на [форуме решений](#).

- Чтобы выйти из режима выделения, нужно ввести :d
- Режим выделения открывается из любого другого режима по нажатию "v"
- Выйти из режима выделения можно, нажав клавишу Esc два раза
- Режим выделения открывается при помощи команды visual
- В режиме выделения можно использовать команды f (удалить) и u (скопировать)
- Режим выделения открывается из нормального режима по нажатию "v"

[Следующий шаг](#)

[Решить снова](#)

Когда вы используете стрелки вверх/вниз для перемещения по истории команд в терминале, вы просматриваете историю команд текущей оболочки. Если вы запускаете оболочку sh из bash, а затем из sh запускаете bash, и вводите команды в каждой из этих оболочек, то истории команд в них будут отдельными и не будут пересекаться.

Поэтому, когда вы используете стрелки вверх/вниз в самой последней запущенной оболочке (bash), вы будете перемещаться только по истории команд этой конкретной оболочки, то есть по истории, созданной в наборе С.

3.2 Скрипты на bash: основы 7 из 10 шагов пройдено 1 из 6 баллов получен

Вы прошли больше 80% курса, оставьте отзыв

Оставить отзыв Нет, спасибо

Надеемся, что однажды 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

Абсолютно точно.

Верно решили 30 266 учащихся
Из всех попыток 65% верных

Только из набора B

Из наборов B и C

Никакие команды появляться не будут

Только из набора A

Только из набора C

[Следующий шаг](#)

[Решить снова](#)

Ваше решение Вы получили 1 балл из 1

После запуска скрипта файл file1.txt будет создан в директории, определенной перед командой touch . Поскольку перед командой cd указан абсолютный путь /home/bi/ , файл file1.txt будет создан в директории /home/bi/ .

Таким образом, полный путь к файлу file1.txt будет /home/bi/file1.txt .

Вы прошли больше 80% курса, оставьте отзыв

Оставить отзыв Нет, спасибо

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

Предположим, что вы находитесь в директории /home/b1/Documents/ и запускаете в ней скрипт следующего содержания:

```
#!/bin/bash
cd /home/b1/
touch file1.txt
cd /home/b1/Desktop/
```

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

Выберите один вариант из списка

Правильно! молодец!

Верно решили 29 905 учащихся
Из всех попыток 76% верных

/home/b1/Desktop/file1.txt
 /home/b1/Documents/file1.txt
 Никаких файлов file1.txt не будет существовать после завершения работы скрипта
 /home/b1/file1.txt

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

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

Подсказка: если все варианты ответов являются неверными, то не отмечайте ни один из них и нажмите кнопку "Отправить"/"Submit".

Выберите все подходящие ответы из списка

Хорошая работа.

Верно решили 27 188 учащихся
Из всех попыток 25% верных

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.

VARIABLE
 variable
 __variable
 variable_123
 var/able
 var.table
 variab\$le

Следующий шаг Решить снова

Ваше решение Вы получили: 1 балл из 1

1165 151 Шаг 7 Следующий шаг >

В данном скрипте:

1. **#!/bin?/bash:** Эта строка определяет путь к оболочке (в данном случае, к Bash). Обычно она используется в начале исполняемого скрипта, чтобы указать системе, какой интерпретатор должен быть применен для выполнения скрипта. Ошибка в пути (например, опечатка в #!/bin?/bash) может привести к неправильному выполнению скрипта или к ошибке "Неверный интерпретатор".
2. **var1=\$1 и var2=\$2:** Эти строки назначают значения переменным var1 и var2, соответствующим первому и второму аргументам командной строки соответственно. Переменная \$1 содержит первый аргумент, а \$2 - второй.
3. **echo "Arguments are: \\$1=\$var1 \\$2=\$var2":** Эта команда выводит сообщение в стандартный вывод (консоль). Используется экранирование символов \\$1 и \\$2, чтобы они не трактовались как переменные, а отображались как текст. В результате этой команды будет напечатано: "Arguments are: \$1=значение_первого_аргумента \$2=значение_второго_аргумента", где переменные заменяются на значения, переданные скрипту в качестве первого и второго аргументов.

The screenshot shows a terminal window with the following text:

```

Напишите скрипт на bash, который принимает на вход два аргумента и выводит на экран строку следующего вида
Arguments are: $1имя_документа $2название

Например, если ваш скрипты называется ./script.sh, то при запуске его ./script.sh one two, на экране должно появиться
Arguments are: $1one $2two

а при запуске ./script.sh three four будет:
Arguments are: $1three $2four

Подсказка: в случае проблем с решением задачи, обратите внимание на наши рекомендации по написанию скриптов.

Напишите программу. Тестируется через stdin -> stdout
 Отлично!
 Плохо
 У还可 доступен Ваше решение, где вы можете сравнить свое решение с другими или спросить совета

1 #!/bin/bash
2 var1=$1
3 var2=$2
4 echo "Arguments are: '$1-$var1 $2-$var2'"
5
6
7
8
9
10

```

Below the terminal window, there is a progress bar indicating 29 093 users solved the task, with 41% being correct.

- **5 -ge 5:** Это условие проверяет, больше или равно ли число 5 числу 5. Поскольку 5 равно 5, условие истинно.
- **-e \$0:** Это условие проверяет, существует ли файл с именем, указанным в переменной **\$0**. В **\$0** хранится имя скрипта, который выполняется. Если файл с таким именем существует, условие верно.
- **-s \$0:** Это условие проверяет, существует ли файл с именем, указанным в **\$0**, и не является ли он пустым. Если файл существует и не пуст, условие будет истинно.
- **! (4 -le 3):** Эта конструкция означает “не” перед условием. Так как 4 не меньше или равно 3, условие ложно, но отрицание ложного условия даст истину.
- **\$# -ge 0:** Это условие проверяет, передано ли в скрипт ноль или больше аргументов. Поскольку всегда существует хотя бы ноль аргументов, условие будет верно.
- **\$:** Этот символ используется для обращения к переменной в bash. В данном контексте он не может быть условием.
- **\$var1 ==:** Это некорректное условие, так как оператор сравнения равенства (==) должен иметь значение справа от него.
- **\$var2 ||:** Это некорректное условие, так как оператор логического “или” (||) должен также иметь значение справа от него.
- **\$var1 !=:** Это также некорректное условие, потому что оператор неравенства (!=) должен быть с значением справа.
- **\$var2:** Это обращение к переменной, но без контекста это не является условием.

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте branching1.sh.

Предположим, вы пишите скрипт на Bash и хотят использовать в нем конструкцию `if` в следующем фрагменте:

```
if [ ... ]
then
else
fi
```

Вы можете угадать значение “`else`” скрипта: `[]` – и не ошибаетесь при проблемах после `[]` перед `]` любое из перечисленных ниже условий. Однако мы прошли все выборы только из них, при которых `else` начинавшийся на экране. Тогда, как зависимости от того, сколько параметров был запущен ваш скрипт и какие в нем есть переменные.

Например, условие `B` – из `B if/elif/else`, и `C`, когда всегда равно куплю мне зависимости от аргументов и переменных внутри скрипта и на экране будет начинаться `else`. Так же из аргумента `var1` – из `B if/elif/else`, так как в первомай `var1`, как может быть значение `None` (тогда будет начинаться `else`), или `None` и не было (перед этим начинаться не будет).

Примечание: если вы планируете проверять варианты ответов у себя в терминале, обратите внимание на то, что сокращение символа `$` честно может изменяться при копировании – не забудьте отредактировать их в соответствии с изображением на экране. Это связано с особенностями написания `$` в некоторых видеокодеках на экране.

Выберите все подходящие ответы из списка

Задача все верно.

Было решено 22 158 учеников
Из всех попыток: 16% верных

Была дана стартовая задача, позволяющая Вы можете начать оставшимся ученикам в [коммит проекта](#), отмечая на их вопросы или давайте свое решение в дополнение [форума проекта](#).

→ 50
 → 4
 5 > 5
 5 =gt; 5
 14 >= 3
 14 = 30

[Следующий шаг](#) [Решить снова](#)

Была дана стартовая задача, позволяющая Вы можете начать оставшимся ученикам в [коммит проекта](#), отмечая на их вопросы или давайте свое решение в дополнение [форума проекта](#).

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

Когда переменная `var` установлена в значение 5, скрипт также выведет “four”. Это происходит потому, что в блоке `if-elif-else` скрипт проверяет условия последовательно, и после нахождения первого подходящего условия остальные проверки игнорируются. В случае с `var=5`, первое условие `[[$var -gt 5]]` возвращает `false`, а остальные условия также не выполняются, потому что 5 не меньше 3 и не равно 4. Поэтому выполняется ветвь `else`, которая и выводит “four”.

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"
else [[ $var -lt 3 ]]
then
echo "two"
else [[ $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=5`, а затем запустили еще раз, но уже с `var=3`.

Выберите один вариант из списка

Правильное, молодец!

Было решено 25 138 учеников
Из всех попыток: 64% верных

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

[Следующий шаг](#) [Решить снова](#)

Этот скрипт на Bash демонстрирует использование оператора `case` для выбора действий на основе значения переменной.

1. **`#!/bin/bash`:** Эта строка указывает на то, что скрипт должен интерпретироваться оболочкой Bash.
2. **`v=student`:** Переменной `v` присваивается значение “student”, которое будет использоваться в последующих строках вывода.
3. **`case $1 in`:** Здесь начинается блок `case`, который будет проверять значение первого аргумента (`$1`), переданного скрипту при его 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 в диапазоне от 2 до 4 включительно, переменной res присваивается строка, отражающая соответствующее количество студентов, например, "2 students", "3 students", или "4 students".
7. ***) `res="A lot of ${v}s";;`**: Если значение \$1 не соответствует ни одному из перечисленных условий, переменной res присваивается строка "A lot of students".
8. **`esac`**: Завершает блок case.
9. **`echo "$res"`**: Выводит значение переменной res на экран.

Таким образом, скрипт проверяет значение первого аргумента при запуске и в зависимости от него выводит строку, которая указывает на количество студентов.

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 1: выводить нужно только строку справа, т.е. ">" выводить не нужно.

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

Верно решили 23 310 учащихся

✓ Абсолютно точно.

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 $1 in
4 *) res="${v}s"; #res - это результат для вывода
5 1) res="$1 ${v}";;
6 [2-4]) res="$1 ${v}s";;
7 *) res="A lot of ${v}s";;
8 esac
9 echo "$res" #вывести результат
10
11
12
13
14
```

Напишите программу. Тестируется через stdin → stdout

Верно решили 23 310 учащихся
Из всех попыток 38% верных

✓ Прекрасный ответ.

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 $1 in
4 *) res="${v}s"; #res - это результат для вывода
5 1) 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 проходит по списку из следующих строк: “`a`”, “`,`”, “`b`”, “`,`”, “`c_d`”. В каждой итерации сначала выводится слово “`start`”, затем выполняется проверка условия `if`.

1. Если переменная `str` равна “`a`”, условие `$str > "c"` не выполняется. Выводится “`start`”, затем “`finish`”.
2. Если переменная `str` равна “`,`”, то условие также не выполнено, и снова выводятся “`start`” и “`finish`”.
3. Если переменная `str` равна “`b`”, выводы аналогичны: “`start`” и “`finish`”.
4. Если переменная `str` равна “`,`”, результат тот же: “`start`” и “`finish`”.
5. Если переменная `str` равна “`c_d`”, условие `$str > "c"` выполняется (поскольку “`c_d`” больше “`c`”). Оператор `continue` прерывает текущую итерацию, пропуская вывод “`finish`” и переходя к следующей итерации.
6. Если переменная `str` равна “`,`”, условие `$str > "c"` не выполняется, поэтому выводятся “`start`” и “`finish`”.

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

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

Отличное решение!

Верно решили 24 582 учащихся

Из всех попыток 45% верных

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

Этот скрипт демонстрирует использование циклов `while`, условных операторов `if-elif-else` и операторов сравнения в bash.

1. В первых трех строках определяются переменные `child`, `adult` и `stdout`, которым присваиваются определенные значения.
2. Основной цикл `while` выполняется до тех пор, пока `stdout` не станет равным 1.
3. Внутри этого цикла пользователь получает приглашение ввести свое имя (“`enter your name:` ”), и введенное значение сохраняется в переменную `name`.
4. После этого проверяются условия:
 - Если введенное имя пустое или равно 0, выводится “`bye`”, переменная `stdout` становится 1, и программа завершает работу.
 - Если введенное имя не пустое, далее запрашивается возраст (“`enter your age:` ”), который сохраняется в переменную `age`.
5. Далее следует вложенный цикл `while`, который также выполняется, пока `stdout` не станет 1.
6. Внутри вложенного цикла проводится проверка возраста:

- Если возраст равен 0 или пуст, выводится "bye", stdout устанавливается в 1, и программа завершает работу.
 - Если возраст меньше или равен значению переменной `child`, выводится сообщение о том, что пользователь относится к группе "child".
 - Если возраст больше переменной `adult`, выводится сообщение о том, что пользователь относится к группе "adult".
 - В других случаях, если возраст находится в диапазоне от 17 до 25 лет, выводится сообщение о том, что пользователь в группе "youth".
7. После каждой итерации вложенного цикла оператор `break` завершает выполнение вложенного цикла и возвращает программу в основной цикл.
8. Если пользователь ввел имя, но не указал возраст, программа вновь попросит ввести возраст.
9. Если возраст введен, программа завершится, 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 (опять 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:

№1

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

№2:

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

Напишите программу. Тестируется через stdin → stdout

Абсолютно точно.

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](#).

Верно решили 21 670 учащихся

Из всех попыток 23% верных

```
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 [[ (-n $name) || ($name = 0) ]]; then
10       echo "bye"
11       stdout=1
12     elif [[ -n $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 -gt ${adult} ]]; then
22           echo "$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   done
29
```

Эти примеры показывают арифметические операции, в которых значение переменной *a* увеличивается на значение переменной *b*. Результат записывается обратно в *a*.

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [math1.sh](#), [math2.sh](#).

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

Выберите все подходящие варианты!

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

Верно.

Верно решили 22 116 учащихся

Из всех попыток 20% 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](#).

- let "a = a + b"
- let a+=b
- a+=\\$b
- let "a=\\$a+\\$b"
- let "a+=b"

Команда `echo` отобразит строку, представляющую результат выполнения команды `pwd`, которая показывает текущий рабочий каталог. В этом сценарии скрипт сначала переходит в директорию `/home/bi/`, а затем выводит ее полный путь с помощью `pwd`.

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [programs.sh](#).
Пусть вы находитесь в директории `/home/b1/Documents/` и запускаете в ней скрипт следующего содержания:

```
#!/bin/bash  
cd /home/b1/  
echo "pwd"
```

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

Выберите один вариант из списка

Всё правильно.

Верно решили 23 677 учащихся

Из всех попыток 51% верных

- `/home/b1/Documents`
- `"pwd"`
- `/home/b1`
- Код возврата команды `pwd` (0 в случае успешного выполнения и не 0 в случае ошибок)
- `pwd`

[Следующий шаг](#) [Решить снова](#)

Если требуется сначала запустить программу, а затем проверить её код возврата, можно воспользоваться следующим подходом:

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

Если программа выводит данные в `stdout`, но нужно узнать её код возврата, можно перенаправить вывод в файл, а затем проверить содержимое файла в условии `if`. Однако, это может быть неэффективным, особенно если программа генерирует большой объем данных или если вы не хотите сохранять вывод в файл. Пример:

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

В этой ситуации программа `program` будет запущена, ее вывод будет перенаправлен в файл `output.txt`, а затем код возврата программы будет проверен в условии `if`.

Мы рассказали, что можно проверить код возврата внешней программы прямо в конструкции `if` при помощи `if program arguments` (действия внутри `if` выполняются, если программа закончилась с кодом 0). Однако это не всегда правда! Если запуск внешней программы выводит что-то в `stdout`, то в проверку `if` поступят именно этот вывод, а не код возврата! Вы можете убедиться в этом, написав простой bash-скрипт с использованием, например, `if "pwd"`.

Однако как быть, если хочется всё-таки запустить программу `program`, которая пишет что-то в `stdout` и потом выполнить какие-то действия если ее код возврата равен 0? Выберите **все верные** утверждения или правильно работающие конструкции `if`.

Примечание: во всех вариантах ответов, где есть кавычка, используется именно косая кавычка (`'`), а не обычная (`"`) или двойная (`''`).

Выберите все подходящие ответы из списка

Абсолютно точно.

Верно решили 21 426 учащихся

Из всех попыток 20% 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](#).

- `if [['program' -eq 0]]`
- Сначала `cat "program"`, затем `if [[$var -eq 0]]`
- Ничего сделать нельзя
- Сначала запустить `program`, затем `if [[$? -eq 0]]`
- `if 'program' > some_file.txt'`

[Следующий шаг](#) [Решить снова](#)

Команда `echo "counters are $c1 and $c2"`, находящаяся в скрипте после десяти вызовов функции `counter` с аргументами от 1 до 10 включительно, выведет строку:

`counters are 55 and 110`

Вывод команды echo "counters are \$c1 and \$c2" в скрипте после десяти вызовов функции counter с аргументами от 1 до 10 включительно будет следующим:

- После первого вызова с параметром 1, c1 увеличивается на 1, а c2 удваивается, получая 2.
- После второго вызова с параметром 2, c1 увеличивается на 2, а c2 удваивается, становясь 4.
- После всех десяти вызовов функциям counter, где последний аргумент 10, c1 суммирует значения от 1 до 10, что составляет 55, а c2 суммирует удвоенные значения, получая 110.

Таким образом, когда вы используете команду echo "counters are \$c1 and \$c2", она выведет:

"counters are 55 and 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 "c1+=\$1"
    let "c2+=\$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 правильно.

Верно решили 20 009 учащихся

Из всех попыток 28% 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](#).

counters are 55 and 110

Этот Bash-скрипт вычисляет наибольший общий делитель (НОД) двух чисел, введенных пользователем. Основная логика скрипта заключена в бесконечном цикле while, который продолжает работать, пока условие [true] истинно.

- Сначала скрипт запрашивает два числа n1 и n2, которые пользователь вводит с клавиатуры.
- Если n1 пустое (т.е. пользователь нажал Enter, не вводя никакое значение), то скрипт срабатывает на условии `if [-z \$n1]`

и выводит сообщение "bye", после чего завершает цикл с помощью команды break.

- Если переменная n1 не пустая, то скрипт определяет функцию gcd, которая использует алгоритм Евклида для вычисления НОД двух чисел. Алгоритм заключается в серии делений с остатком, пока не получен остаток ноль. - После завершения вычислений, скрипт выводит "GCD is \$n1", где \$n1 – результат работы функции gcd`.

Таким образом, скрипт запрашивает ввод двух чисел для вычисления их НОД и повторяет этот процесс до тех пор, пока пользователь не введет пустую строку, нажав Enter.

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 выводить соответствующее сообщение на экран (см. выше). Иначе нужно сравнить 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
15 25
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 sshin → sshout
Отлично!
Верно решили 18 148 участника
Из всех попыток 35% верных

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

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

Этот скрипт работает следующим образом:

- В бесконечном цикле `while true` выполняется команда `read`,читывающая ввод пользователя.
- Затем проверяется, совпадает ли первый операнд со строкой "exit". Если это так, выводится сообщение "bye", и цикл завершается.
- Далее проверяется, являются ли оба операнда целыми числами. Если хотя бы один из них не целое число, выводится сообщение "error", и цикл прекращается.
- Если оба операнда целые, выполняется соответствующая математическая операция, основываясь на операторе (+, -, *, /, %, **).
- После выполнения операции ее результат выводится на экран.
- Если оператор не соответствует ни одной из допустимых операций, скрипт выводит "error" и завершает цикл.

Напишите **калькулятор** на bash. При запуске ваш скрипт должен ожидать ввода пользователем 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 операнд (число). В этом случае нужно произвести указанную операцию над заданными числами и вывести результат на экран.
3. Любая другая команда из одного аргумента или из трех аргументов, но с операцией не из списка. В этом случае нужно 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. `./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
bye
```

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").

[Подсказка: в случае проблем с решением задачи, обратите внимание на наши рекомендации по написанию скриптов.](#)

Верно. Так держать!

Из всех попыток 36% верных

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

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

Команда `find /home/bi -iname "star*"` ищет все файлы в каталоге `/home/bi`, названия которых начинаются со строки "star", без учета регистра. Таким образом, она найдет файлы, такие как "Star_Wars.avi", "star_trek_OST.mp3", "STARS.txt", и "stardust.jpeg".

В отличие от этого, команда `find /home/bi -name "star*"` также ищет файлы, начинающиеся с "star", но уже с учетом регистра. Это значит, что файлы, которые начинаются с "Star" или "STARS", пропускаются. Следовательно, эта команда найдет только те файлы, в названии которых слово "star" написано строчными буквами.

Таким образом, файлы, которые будут найдены первой командой, но не будут найдены второй, это "STARS.txt" и "Star_Wars.avi".

Пусть в директории /home/bi лежат файлы Star_Wars.avi, star_trek_OST.mp3, STARS.txt, stardust.mpeg, Eddard_Stark_biography.txt.

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

Выберите все подходящие ответы из списка

✓ Прекрасный ответ.

✓ STARS.txt
☐ stardust.mpeg
☐ star_trek_OST.mp3
✓ Star_Wars.avi
☐ Eddard_Stark_biography.txt

Верно решили 20 547 учащихся
Из всех попыток 36% верных

Следующий шаг Решить снова

1. Если в команде поиска заменить `-name` на `-path`, результат поиска иногда может остаться прежним. **- Это утверждение верное.** Это потому, что оба параметра (`-name` и `-path`) используются для поиска по имени файла, но `-name` проверяет только название самого файла, а `-path` включает в себя полный путь к файлу. Если в результатах поиска нет вложенных директорий, то замена `-name` на `-path` не изменит результат.
2. В некоторых случаях, `find` с параметром `-name` найдет меньше файлов, чем с аналогичным запросом, но с параметром `-path`. **- Это утверждение верное.** Это возможно, если некоторые файлы находятся в вложенных директориях, а их полные пути содержат искомую строку, даже если сами имена файлов не соответствуют ей. В этом случае `-path` найдет больше файлов, потому что он учитывает весь путь, а не только имя файла.

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

Выберите все подходящие ответы из списка

✓ Верно. Так держать!

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 18 450 учащихся
Из всех попыток 22% верных

✓ В некоторых случаях `find` с `-name` найдет больше файлов, чем `find` с таким же запросом, но с `-path`
✓ Если заменить в команде поиска `-name` на `-path`, то результат поиска иногда может остаться таким же
☐ 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`
☐ Опция `-path` используется только для поиска директорий, а `-name` только для поиска файлов

Следующий шаг Решить снова

Команда `find /home/bi -mindepth 2 -maxdepth 3 -name "file*"` ищет файлы в каталоге `/home/bi/` и его подкаталогах с глубиной от 2 до 3 уровней, имена которых начинаются с “file”.

1. **file1:** Будет найден, так как находится на глубине 2 и его имя начинается с “file”.
2. **file2:** Будет найден, потому что расположен на глубине 3 и его имя также начинается с “file”.
3. **file3:** Не будет найден, поскольку находится на глубине 4, которая выходит за пределы указанного диапазона глубины поиска.

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

Здорово, всё верно.

Верно решили 20 711 учащихся
Из всех попыток 41% верных

- Все кроме `file3`
- Все кроме `file2`
- Все кроме `file1`
- Только `file2`
- Только `file1`

Эти четыре команды создают файл `results.txt` одинакового размера, поскольку количество символов, записываемых в файл, не меняется из-за характера команд:

- `grep "word" file.txt > results.txt`: записывает все строки из файла `file.txt`, которые содержат слово “`word`”, в `results.txt`.
- `grep -A 1 "word" file.txt > results.txt`: добавляет в `results.txt` строки, содержащие “`word`”, а также одну строку после каждой строки, содержащей “`word`”.
- `grep -B 1 "word" file.txt > results.txt`: добавляет в `results.txt` строки, содержащие “`word`”, и одну строку перед каждой строкой, содержащей “`word`”.
- `grep -C 1 "word" file.txt > results.txt`: сохраняет в `results.txt` строки, содержащие “`word`”, а также одну строку перед и одну строку после каждой строки, содержащей “`word`”.

Таким образом, все четыре команды создадут файл `results.txt` одинакового размера.

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

Так точно!

Верно решили 20 237 учащихся
Из всех попыток 41% верных

- `grep -A 1 "word" file.txt > results.txt` и `grep -B 1 "word" file.txt > results.txt`
- `grep -A 1 "word" file.txt > results.txt`
- Все, кроме `grep -C 1 "word" file.txt > results.txt`
- `results.txt` будет одинакового размера во всех случаях
- `grep -C 1 "word" file.txt > results.txt`

[Следующий шаг](#)

[Решить снова](#)

Команда `grep -E "[xk1XKL]?[uU]buntu$"` текст.txt будет искать строки в файле `text.txt`, которые заканчиваются на “`ubuntu`” или “`Ubuntu`”, с опциональным символом перед этим окончанием. Этот дополнительный символ, если он есть, может быть одним из следующих: “`x`”, “`k`”, “`l`”, “`X`”, “`K`”, или “`L`”.

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

Выберите все подходящие ответы из списка

Так точно!

Верно решили 18 768 учащихся
Из всех попыток 23% 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](#).

- Lubuntu is better than Ubuntu
- Uuuuubuntu!
- Kubuntu
- Well, xubuntu is OK
- Lubuntu is better than Windows
- Mac OS X 10.9, Windows XP; Ubuntu 12.04

Следующий шаг

Решить снова

Если в команде `sed "/[a-z]*p" text.txt` не использовать опцию `-n`, `sed` по умолчанию будет выводить каждую строку после обработки, независимо от того, была ли она изменена.

Команда `sed "/[a-z]*p" text.txt` будет искать строки, содержащие ноль или более символов в нижнем регистре, и печатать их. Однако, поскольку опция `-n` не указана, `sed` также напечатает оригинальные строки из файла, которые не были изменены, что приведет к удвоенному выводу.

В итоге каждая строка из файла будет выведена дважды: сначала оригинальная строка, затем строка после обработки `sed`.

Что произойдет, если в команде `sed -n "/[a-z]*p" text.txt` не указывать опцию `-n`?

Выберите один вариант из списка

Отлично!

Верно решили 19 784 учащихся
Из всех попыток 39% верных

- Каждая строчка будет выведена два раза
- На экран будет выведено всё содержимое файла `text.txt`
- Будут выведены все строки файла `text.txt`, в которых есть только большие буквы латинского алфавита
- Появится сообщение об ошибке

Следующий шаг

Решить снова

Ваше решение. Вы получили: 1 балл из 1

Эта команда `sed` заменяет последовательности из двух или более прописных букв, за которыми следует пробел, на строку "abbreviation".

Вот объяснение, как эта команда работает:

1. **sed**: Это инструмент в системах Unix/Linux, который используется для редактирования текста в потоке.
2. '**s/.../.../gsed. Первый набор точек - это шаблон, который мы хотим найти, второй набор - текст, который его заменяет. Флаг `g` указывает, что замена выполняется для всех вхождений шаблона в строке.**
3. **[A-Z]\{2,\}**: Это регулярное выражение, обозначающее последовательность из двух или более прописных букв (от A до Z) с пробелом в конце. Число в фигурных скобках указывает на минимальное количество повторений (2 или более).

4. **abbreviation:** Это строка, которая заменяет найденные последовательности. В данном случае она заменяет найденные последовательности прописных букв на "abbreviation".
5. **g:** Этот флаг говорит, что замена применяется ко всей строке, а не только к первому вхождению.

Таким образом, эта команда `sed` ищет все последовательности из двух или более прописных букв, за которыми идет пробел, и заменяет их на "abbreviation" в файле `input.txt`, записывая результат в файл `edited.txt`.

Запишите в форму ниже инструкцию `sed`, которая заменит все "аббревиатуры" в файле `input.txt` на слово "abbreviation" и запишет результат в файл `edited.txt` (на экран при этом ничего выводить не нужно). Обратите внимание, что в инструкции должны быть указаны и сам `sed`, и оба файла!

Под "аббревиатурой" будем понимать слово, которое удовлетворяет следующим условиям:

- 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 проверять вашу инструкцию **не будем**.

Пример: если у вас был текст "Hi, I heard these songs by ABBA, TLA and DM!", 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 `rm -rf /*`!). Вместо этого мы сперва анализируем структуру вашей инструкции (например, что в ней использованы имена `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 при помощи "(" и ")" какой из символов к чему относится! Например, регулярное выражение `a{?}` (ночь или один раз по одной или более букве "a") нужно записать как `(a{+})?` (при этом запись `(a{+})?`, конечно же, не поможет).

Напишите текст

Хорошая работа.

Верно решили 16 632 учащихся
Из всех попыток 34% верных

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](#).

`sed 's/[A-Z]{2,}/abbreviation/g' input.txt > edited.txt`

Эта опция сохраняет все созданные графики после закрытия `gnuplot`, так что они не будут автоматически закрыты. Это удобно, если вы хотите сохранить графики для дальнейшего просмотра или анализа.

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

Верно.

Верно решили 18 785 учащихся
Из всех попыток 51% верных

`-s, -show-plots-after-exit`

Такой опции не существует.

Графики и так не закрываются автоматически при закрытии `gnuplot`!

`-p, -persist`

[Следующий шаг](#)

[Решить снова](#)

Название ряда данных будет установлено по первому значению во втором столбце, поскольку используется опция `autotitle columnhead` для автоматического определения заголовков столбцов. На графике появится 9 точек, потому что точка из первой строки файла данных не будет учтена при построении графика.

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

Хорошая работа.

Верно решили 17 975 учащихся

Из всех попыток 32% верных

- Всеми решими сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).
- Название "номате", нарисовано 10 точек
 - Название – первое значение из второго столбца, нарисовано 10 точек
 - Название – первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)
 - Название "data.csv" using 1:2, нарисовано 10 точек
 - Название – первое значение из первого столбца, нарисовано 9 точек (точка из первой строки пропущена)

[Следующий шаг](#)

[Решить снова](#)

- `"point 1, value ".x1 x1` указывает, что для первого деления на оси x будет установлена метка "point 1, value", которая будет соответствовать значению `x1`.
- `"point 2, value ".x2 x2` означает, что для второго деления на оси x будет установлена метка "point 2, value", соответствующая значению `x2`.
- `"point 3, value ".x3 x3` предполагает, что для третьего деления на оси x будет метка "point 3, value", которая будет соответствовать значению `x3`.

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [plot.gnu](#), [plot_advanced.gnu](#), [plot_advanced2.gnu](#). Все три скрипта 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 этой оси было только три деления (т.е. три черточки) в этих самых координатах, а подписи 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 **конкатенация**, 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

Абсолютно точно.

Верно решили 13 935 учащихся

Из всех попыток 44% верных

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

1. `a=a+1`: Эта строка увеличивает значение переменной `a` на 1.
2. `zrot=(zrot+350)%360`: В этой строке переменная `zrot` увеличивается на 350, затем к ней применяется оператор остатка при делении на 360, что позволяет реализовать вращение по оси Z в рамках 360 градусов.
3. `set view xrot,zrot`: Эта команда устанавливает углы поворота графика вокруг осей X и Z.
4. `splot -x**2-y**2`: Эта команда строит трехмерный график функции `-x**2-y**2`.

5. **pause 0.1:** Эта команда делает паузу в скрипте длительностью 0.1 секунды.
6. **if (a<50) reread:** Эта строка проверяет условие $a < 50$. Если условие выполняется, команда `reread` перечитывает текущий скрипт, чтобы он продолжал выполняться. Таким образом, цикл повторяется до тех пор, пока a не достигнет значения 50.

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. добавлять и удалять инструкции нельзя!) 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

Здорово, всё верно.

Верно решили 12 854 учащихся
Из всех попыток 47% верных

```
a=0+1
zrot=(zrot+350)%360
set view xrot,zrot
splot -x**2-y**2
pause 0.1
if (a<50) reread
```

1. **chmod 764 file.txt:** Эта команда задает права доступа к файлу `file.txt`, предоставляя владельцу полные права (чтение, запись и выполнение), группе — права на чтение и запись, а всем остальным — только право на чтение.
2. **chmod ug+w file.txt; chmod u+x file.txt:** Здесь сначала добавляются права на запись для владельца и группы (`ug+w`), а затем устанавливается право на выполнение только для владельца (`u+x`).

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

Примечание: запись вида `команда1; команда2; команда3` означает, что в терминале последовательно выполнились все три команды (сначала `команда1`, затем `команда2` и, наконец, `команда3`).

Выберите все подходящие ответы из списка

Правильно.

Верно решили 16 484 учащихся
Из всех попыток 21% 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](#).

chmod 467 file.txt
 chmod 777 file.txt
 chmod u-wx file.txt; chmod g-w file.txt
 chmod ug+w file.txt; chmod u+x file.txt
 chmod rwxrwr-- file.txt
 chmod o-wx file.txt; chmod g/x file.txt; chmod a+wx file.txt

1. **sudo chown user dir:** Эта команда назначит пользователя `user` владельцем каталога `dir`.
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 владельца и группу каталога `dir` на `user` и `group` соответственно.
4. `sudo chmod a+w dir`: Эта команда предоставит право на запись всем пользователям (владельцу, группе и другим) для каталога `dir`.

Предположим вы использовали команду `sudo` для создания директории `dir`. По умолчанию для `dir` были выставлены права доступа `rwxr-xr-x` (владелец `root`, группа `root`). Таким образом никто кроме пользователя `root` не может ничего записывать в эту директорию, например, не может создавать файлы в ней.

После выполнения какой команды `user` из группы `group` всё-таки сможет создать файл внутри `dir`? Укажите все верные варианты ответов!

Примечание: считаем, что все команды выполняются от имени `user`, если явно не указано, что команда выполнена с `sudo`.

Примечание 2: мы выбрали пример с директорией, а не с файлом не случайно.

Дело в том, что если создать при помощи `sudo` файл с правами `rw-r--r--` в директории, которая принадлежит пользователю, то возникнет любопытная ситуация. С одной стороны пользователь может удалить этот файл (т.к. ему разрешено удалять **все** файлы внутри его директории) и может прочитать его содержимое (т.к. право "r" у файла установлено для всех), с другой стороны он не может этот файл редактировать (т.к. право "w" у файла есть только для `root`). При этом некоторые "умные" редакторы, например, `vim` позволят даже редактировать этот файл, но сделают они это своеобразно: через удаление оригинала и создание копии уже с нужными правами (удалить мы можем, а раз можем читать, то и копию создать не сложно). Итого получается, что несмотря на права `rw-r--r--`, пользователь может сделать с этим файлом почти всё что угодно!

В случае же, когда речь идет о директории созданной `root`, ситуация будет проще: пользователь сможет смотреть её содержимое (у него есть право "r"), 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 (грубо говоря их **назования**). Таким образом право "r" у директории дает возможность просматривать "записи", т.е. просматривать её 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 Antipovsky** за помощь в оформлении **Примечания 2**!

Выберите все подходящие ответы из списка



Отличное решение!

Верно решили 14 683 участников
Из всех попыток 15% 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](#).

- `sudo chown user dir`
- `sudo chown user:group dir`
- `sudo chown :group dir`
- `sudo chmod a+w dir`

- **Количество символов:** Это общее количество всех символов в файле, включая пробелы, табуляции и символы новой строки.
- **Количество слов:** Это общее количество слов в файле. Слово определяется как последовательность символов, отделенных пробелами, табуляцией или символами новой строки.
- **Количество строк:** Это количество строк в файле. Каждая строка заканчивается символом новой строки.
- **Размер файла в байтах:** Это полный размер файла в байтах, включая все символы и структуры данных.

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

Выберите все подходящие ответы из списка

Абсолютно 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](#).

Верно решили 17 158 учащихся
Из всех попыток 21% верных

- Количество символов
- Количество строк
- Размер файла в байтах
- Количество слов
- Длину самой длинной строки

[Следующий шаг](#) [Решить снова](#)

- **-s (или --summarize)** указывает команде выводить только общий размер директории, а не перечислять размеры всех вложенных файлов и поддиректорий.
- **-h (или --human-readable)** обозначает вывод информации в удобном для чтения формате, с использованием человекочитаемых единиц, таких как Кб, Мб, Гб.

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 2048 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 800 Кбайт и две поддиректории в каждой из которых лежит по файлу в 400 Кбайт, то загаданная команда должна вывести на экран одно число: 2.4к (также на экране может быть выведен еще и символ **, обозначающий, что это размер именно текущей директории).

Напишите текст

Хорошая работа.

Верно решили 16 381 учащийся
Из всех попыток 53% верных

du -h -s

Команда `mkdir dir{1..3}` создаст три поддиректории в текущем каталоге. Их имена будут `dir1`, `dir2` и `dir3`, сгенерированные с помощью фигурных скобок и числового диапазона.

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

Хорошие новости, верно!

Верно решили 16 720 учащийся
Из всех попыток 40% верных

`mkdir dir{1..3}`

[Следующий шаг](#) [Решить снова](#)

Ваше решение: Вы получили 2 балла из 2

Выводы

Я понял тему и готов продолжить изучение Linux.