# по вопросам из txt файла

### 1. Аббревиатура LTS в Ubuntu

англ. **Long Term Support**; «поддержка в течение длительного периода»

### 2. Что такое root

или суперпо́льзователь — это специальный аккаунт и группа пользователей в UNIX-подобных системах с идентификатором UID 0 (User IDentifier), владелец которого имеет право на выполнение всех без исключения операций

### 3. Что такое SSH

Secure Shell — «безопасная оболочка»

— сетевой протокол, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов)

### 4. Как посмотреть всех пользователей системы

В операционной системе Linux есть специальный файл в котором хранится информация обо всех пользователях. Это /etc/passwd.

**cat /etc/passwd**

### 5. Комбинация из трёх цифр которая выдаёт все права.

**chmod 777**

Когда используется 3-значное число, первая цифра представляет права владельца файла, вторая — группу файла, а последняя — всех остальных пользователей.

Разрешения на запись, чтение и выполнение имеют следующее числовое значение:

* r (читай) = 4
* w (запись) = 2
* x (выполнить) = 1
* нет разрешений = 0

Цифра разрешений определенного пользовательского класса является суммой значений разрешений для этого класса.

Каждая цифра номера разрешений может быть суммой 4, 2, 1 и 0:

* 0 (0 + 0 + 0) — Нет разрешения.
* 1 (0 + 0 + 1) — только разрешение на выполнение.
* 2 (0 + 2 + 0) — только разрешение на запись.
* 3 (0 + 2 + 1) — разрешения на запись и выполнение.
* 4 (4 + 0 + 0) — только разрешение на чтение.
* 5 (4 + 0 + 1) — разрешение на чтение и выполнение.
* 6 (4 + 2 + 0) — права на чтение и запись.
* 7 (4 + 2 + 1) — разрешение на чтение, запись и выполнение.

Например, если номер разрешения установлен равным 750, это означает, что владелец файла имеет права на чтение, запись и выполнение, группа файла имеет разрешения на чтение и выполнение, а другие пользователи не имеют разрешений:

* Владелец: rwx = 4 + 2 + 1 = 7
* Группа: rx = 4 + 0 + 1 = 5
* Другие: rx = 0 + 0 + 0 = 0

### 6. Как выдать права на каталог и его содержимое.

Узнать Linux права пользователя на папку очень просто. Для этого используется команда:

ls -l имя каталога

chmod 640 имя\_файла

Чаще всего в практике применяются следующие варианты чисел для команды:

Смена владельца файла или директории

При изменении прав на директории можно использовать параметр -R. Он рекурсивно изменит владельца всех вложенных директорий и файлов. Представим, что нам нужно предоставить права на папку new-user для одноимённого пользователя и его группы. Команда для этого будет выглядеть так:

chown -R new-user:new-group /home/users/new-user

Второй инструмент — chgrp. В отличие от первого, меняет только группу-владельца:

chgrp [группа] [файл или папка]

Как и chown, может работать рекурсивно с помощью ключа -R.

Для работы непосредственно с правами используется команда chmod:

chmod [настройки прав] [файл или папка]

При работе с директориями опция -R позволит изменить права на все вложенные файлы и папки.

Настройки прав в chmod можно определять двумя способами:

1. Указав категорию (u — пользователь-владелец, g — группа-владелец, o — другие пользователи, a — все пользователи), модификатор (+, - , =) и, соответственно, нужные права (r, w, x).

Бывает, что нам нужно изменить права только на все папки или только на все файлы в определённой директории. Часто с этим можно столкнуться при настройке прав на файлы сайтов. Вот вам пара однострочников на этот случай:

* Смена прав на 755 для всех папок внутри /var/www/sites/example.com:  
  find /var/www/sites/example.com/ -type d -exec chmod 755 {} \;
* Смена прав на 644 для всех файлов внутри /var/www/sites/example.com:  
  find /var/www/sites/example.com/ -type f -exec chmod 644 {} \;

### 7. Как сохранить в редакторе VI

**просто сохранить, это - :w**

Команда для сохранения файла в Vim и выхода из редактора есть :wq .

### 8. С каким ключём нужно запустить команду df для отображения инфы в человекочитаемом виде.

Опция -h позволяет указать команде df что мы хотим получить информацию в человекочитаемом виде.

### 9. Что такое /var/log/dmesg

**Лог загрузки системы**

/var/log/dmesg — драйвера устройств. Одноименной командой можно просмотреть вывод содержимого файла. Размер журнала ограничен, когда файл достигнет своего предела, старые сообщения будут перезаписаны более новыми. Задав ключ --level= можно отфильтровать вывод по критерию значимости.

### 10. За что отвечает PPID в выводе команды top

PID — идентификатор процесса;

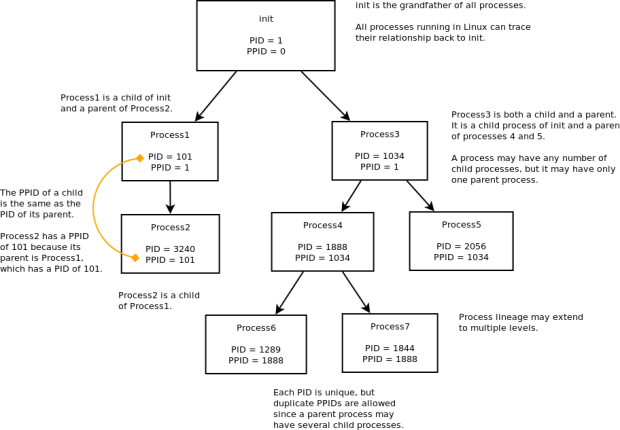
Кратко, схема процессов в **Linux** (да и любой другой **UNIX**-like системе) выглядит как дерево. Первым всегда запускается процесс init с номером 1, который порождает все остальные процессы.

Запущенный в обычном режиме процесс получает собственный PID (*Process ID*), а в качестве PPID (*Parent Process ID* – *ID родительского процесса*) – получает номер процесса, “от которого” он был запущен. При завершении работы родительского процесса – процесс-потомок так же прекратит свою работу, т.к. его “родительский процесс” отправит команду на завершение работы всех своих “потомков”.

What is the PPID?

In addition to a unique process ID, each process is assigned a parent process ID (**PPID**) that tells which process started it. The PPID is the PID of the process’s parent.

For example, if process1 with a PID of 101 starts a process named process2, then process2 will be given a unique PID, such as 3240, but it will be given the PPID of 101. It’s a parent-child relationship. A single parent process may spawn several child processes, each with a unique PID but all sharing the same PPID.



Why is the PPID Important?

Occasionally, processes go bad. You might try to quit a program only to find that it has other intentions. The process might continue to run or use up resources even though its interface closed. Sometimes, this leads to what is called a zombie process, a process that is still running, but dead.

One effective way to kill a zombie process is to kill its parent process. This involves using the ps command to discover the PPID of the zombie process and then sending a kill signal to the parent. Of course, any other children of the parent process will be killed as well.

### 11. Чему равна маска подсети 172.168.4.10/8

Маски подсети являются основой метода бесклассовой маршрутизации. При этом подходе маску подсети записывают вместе с IP-адресом в формате IP-адрес/количество единичных бит в маске. Число после знака дроби (т. н. длина префикса сети) означает количество единичных разрядов в маске подсети.

<https://ipcalc.co/>

IP-адрес: 11000000 10101000 00000001 00000010 (192.168.1.2)

Маска подсети: 11111111 11111111 11111110 00000000 (255.255.254.0)

Адрес сети: 11000000 10101000 00000000 00000000 (192.168.0.0)

Легенда:

* часть маски, определяющая адрес сети и состоящая из единиц;
* адрес сети, который определяется маской подсети;
* диапазон адресов устройств в этой сети.

Маска подсети — это тоже 32-бита. Но в отличии от IP-адреса, нули и единицы в ней не могут чередоваться. Всегда сначала идет сколько-то единиц, потом сколько-то нулей. Не может быть маски

120.22.123.12=01111000.00010110.01111011.00001100.

Но может быть маска

255.255.248.0=11111111.11111111.11111000.00000000.

Сначала N единиц, потом 32-N нулей. Несложно догадаться, что такая форма записи является избыточной. Вполне достаточно числа N, называемого длиной маски. Так и делают: пишут 192.168.11.10/21 вместо 192.168.11.10 255.255.248.0. Обе формы несут один и тот же смысл, но первая заметно удобнее.

Чтобы определить границы подсети, компьютер делает побитовое умножение (логическое И) между IP-адресом и маской, получая на выходе адрес с обнуленными битами в позициях нулей маски. Рассмотрим пример 192.168.11.10/21:

11000000.10101000.00001011.00001010

11111111.11111111.11111000.00000000

----------------------------------------------

11000000.10101000.00001000.00000000 = 192.168.8.0

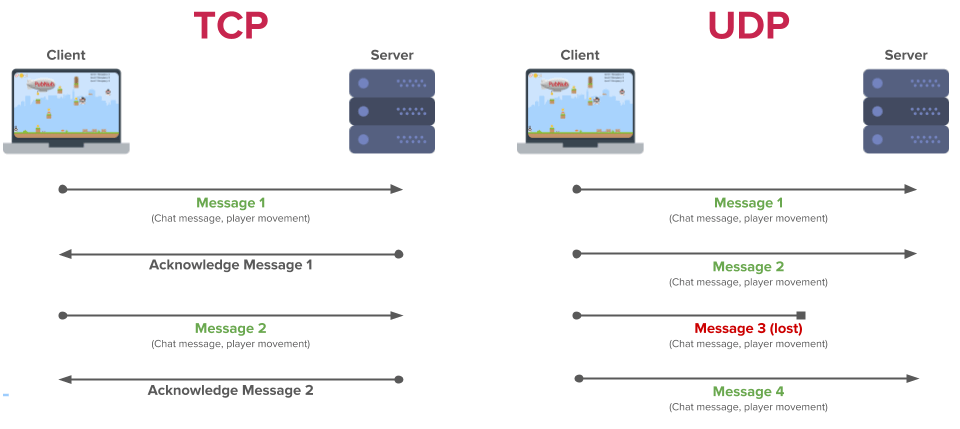
### 12. Отличие tcp от udp

это сетевые протоколы транспортного уровня

**TCP** – транспортный протокол передачи данных в сетях TCP/IP, предварительно устанавливающий соединение с сетью.

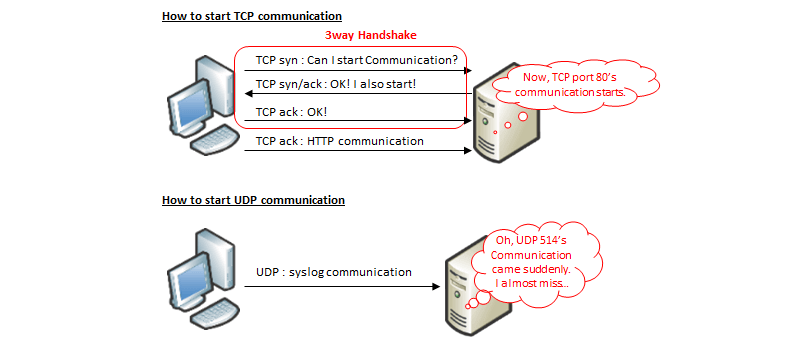
**UDP** – транспортный протокол, передающий сообщения-датаграммы без необходимости установки соединения в IP-сети.





Протокол TCP (Transmission Control Protocol) – это сетевой протокол, который «заточен» под соединение. Иными словами, прежде, чем начать обмен данными, данному протоколу требуется установить соединение между двумя хостами. Данный протокол имеет высокую надежность, поскольку позволяет не терять данные при передаче, запрашивает подтверждения о получении от принимающей стороны и в случае необходимости отправляет данные повторно. При этом отправляемые пакеты данных сохраняют порядок отправки, то есть можно сказать, что передача данных упорядочена. Минусом данного протокола является относительно низкая скорость передачи данных, за счет того что выполнение надежной и упорядоченной передачи занимает больше времени, чем в альтернативном протоколе UDP.

Протокол UDP (User Datagram Protocol), в свою очередь, более прост. Для передачи данных ему не обязательно устанавливать соединение между отправителем и получателем. Информация передается без предварительной проверки готовности принимающей стороны. Это делает протокол менее надежным – при передаче некоторые фрагменты данных могут теряться. Кроме того, упорядоченность данных не соблюдается – возможен непоследовательный прием данных получателем. Зато скорость передачи данных по данному транспортному протоколу будет более высокой.



* **Надежность**: в этом случае предпочтительнее будет протокол TCP, за счет подтверждения получения данных, повторной отправки в случае необходимости, а также использованию такого инструмента как тайм-аут. Протокол UDP такого инструментария не имеет, а потому при получении отправленные данные могут приходить не полностью;
* **Упорядоченность**: опять будет предпочтительнее TCP, поскольку этот протокол гарантирует передачу пакетов данных именно в том порядке, в котором они были отправлены. В случае с UDP такой порядок не соблюдается;
* **Скорость**: здесь уже лидировать будет UDP, так как более тяжеловесному TCP-протоколу будет требоваться больше времени для установки соединения, подтверждения получения, повторной отправки данных и т.д. ;
* **Метод передачи данных**: в случае с TCP данные передаются потоково, границы фрагментов данных не имеют обозначения. В случае с UDP данные передаются в виде датаграмм – проверка пакетов на целостность осуществляется принимающей стороной только в случае получения сообщения. Также пакеты данных имеют определенные обозначения границ;

Сравнивая оба протокола, очевидно, что протокол TCP – это, можно сказать, «снайпер». Прицелился, выстрелил, зафиксировал попадание, ищет следующую цель. UDP – это, скорее, «пулеметчик» - выставил ствол в направлении врага и начал долбить очередями, не слишком заботясь о точности. Как в войсках важны обе эти воинские специальности, так и в интернете важны оба этих протокола. TCP применяется там, где требуется точная и подтверждаемая передача данных – например, отправка фотографий, или переписка между пользователями. UDP, в свою очередь, нужен для общения в голосовом формате, или при передаче потокового видео, например, с веб-камер или IP-камер.

### 13. Определить какой ip является частным ()

У каждого хоста в сети Интернет должен быть уникальный адрес. Если ваши сети изолированы от Интернета (например, связывают два филиала), для хостов без проблем можно использовать любые IP-адреса. Однако, уполномоченной организацией по распределению нумерации в сети Интернет (IANA) специально для частных сетей зарезервированы следующие три блока IP-адресов:

* 10.0.0.0 — 10.255.255.255
* 172.16.0.0 — 172.31.255.255
* 192.168.0.0 — 192.168.255.255

### 14. Сколько портов доступно в системе

номер порта — число в диапазоне от 1 до 216-1=65 535.

Так, порты с номерами до 1023 Линукс и Unix-подобными ОС считаются за “критически важные” для сетевой деятельности системы, так что для доступа к ним и службам, с ними связанными часто требуются *root* права.

Порты от 1024 до 49151 имеют гриф “готовые к регистрации”. Это означает, что данные порты зарезервированы или могут быть зарезервированы за определёнными службами. К счастью или сожалению, они за этими сервисами не закреплены прочными правилами, однако могут дать ключ для распознавания запущенной программы на стороне хоста. Остальные (начиная с 49152) порты не зарегистрированы и используются по усмотрению пользователей ОС и имеют название “динамические” порты

### 15. Выбрать уровень, входящие в модель OSI (из 12 вариантов, правильных было 7)

### 

### 16. Системный порт

**0 – 1023 System Ports**

| **Диапазон номеров портов** | **Назначение** |
| --- | --- |
| **от 0 до 63** | Стандартные функции для всей сети |
| **от 64 до 127** | Функции хоста |
| **со 128 по 239** | Зарезервировано для будущего использования |
| **от 240 до 255** | Любая экспериментальная функция |

### 17. Что такое localhost

localhost (так называемый, «местный» от англ. local, или «локальный хост», по смыслу — этот компьютер) — в компьютерных сетях, стандартное, официально зарезервированное доменное имя для частных IP-адресов **(в диапазоне 127.0.0.1 — 127.255.255.254 \8 маска**

### 18. Где находится dhcpd.conf

Для настройки сервера DHCP используется конфигурационный файл dhcpd.conf, который обычно располагается в каталоге /etc или /etc/dhdcp.

-cf <filename> — Определяет размещение файла конфигурации.

### 19. Правила iptables (как пробросить UDP порты 15000-15002)

Проброс порта — это специальное правило в роутере, которое разрешает все обращения извне к определенному порту и передает эти обращения на конкретное устройство во внутренней сети.

| | iptables -L INPUT | | --- | |
| --- | --- |

Если написано вида Chain INPUT (policy ACCEPT), то все порты по умолчанию доступны снаружи (за исключением правил, что ниже)

Перечень основных действий, для выполнения которых используется Iptables:

-A – добавить правило в цепочку;

-C – проверить применяемые правила;

-D – удалить текущее правило;

-I – вставить правило с указанным номером;

-L – вывести правила текущей цепочки;

-S – вывести все активные правила;

-F – очистить все правила;

-N – создать цепочку;

-X – удалить цепочку;

-P – установить действие «по умолчанию».

В качестве дополнительных параметров используются опции:

-p – вручную установить протокол (TCP, UDP, UDPLITE, ICMP, ICMPv6, ESP, AH, SCTP, MH);

-s – указать статичный IP-адрес оборудования, откуда отправляется пакет данных;

-d – установить IP получателя;

-i – настроить входной сетевой интерфейс;

-o – то же самое в отношении исходящего интерфейса;

-j – выбрать действие при подтверждении правила.

### какое правило ipdates открывает доступ к UDP портам

​​

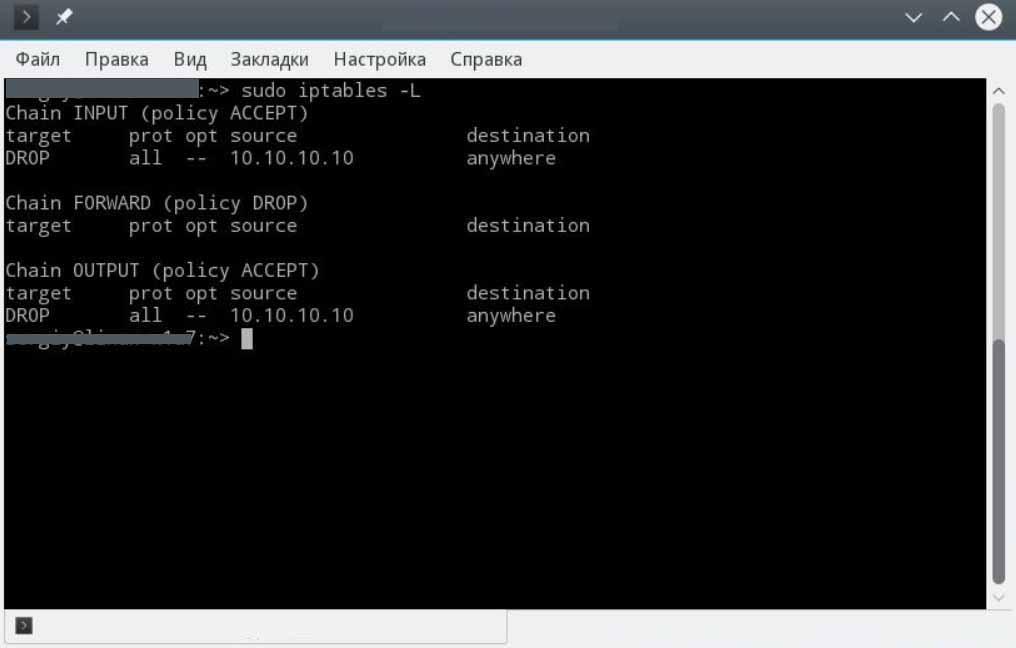
Подобными инструкциями можно давать разрешения для подключений, использующих несколько портов. Например, так выглядит команда на открытие целого диапазона портов (32810-32814):

$ sudo ufw allow 32810:32814/udpКак

### 20. удалить правила iptables

При удалении правил в Iptables вводится команда с опцией **-D**. Но перед этим может понадобиться просмотреть перечень правил:

$ sudo iptables -L

Пример удаления правила:

$ sudo iptables -A OUTPUT -s 10.10.10.10 -j DROP

Если требуется полное удаление правил, применяется команда:

$ sudo iptables –F

Очистка правил

При любых ошибках в работе Iptables, чтобы исключить нарушение функционирования ядра перед другими действиями, требуется очистить правила, «обнулить» вводные данные. Выполняется эта процедура командой:

$ sudo iptables –F

Если речь идет об определенной цепочке, то она будет выглядеть иначе:

$ sudo iptables -F Input

Перечисленные действия выполняются для таблицы **filter**, которая подключена изначально.

[Перечисление и удаление правил брандмауэра Iptables | DigitalOcean](https://www.digitalocean.com/community/tutorials/how-to-list-and-delete-iptables-firewall-rules-ru)

### 21. Какой командой нельзя попасть в домашнюю директорию (я открывал консоль и вписывал команды)

| **cd** | переносит вас в ваш домашний каталог |
| --- | --- |
| **cd ~** | также переносит вас в ваш домашний каталог |
| **cd /** | переносит вас в корневой каталог всей системы |
| **cd /root** | переносит вас в домашний каталог пользователя root, или супер-пользователя, учётной записи, созданной во время установки системы |
| **cd /home** | переносит вас в каталоге home, в котором находятся каталоги пользователей |
| **cd ..** | переносит вас на один уровень выше |
| **cd ~otheruser** | переносит вас в домашний каталог пользователя *otheruser*, если только пользователь *otheruser* дал вам соответствующие разрешения |
| **cd /dir1/subdirfoo** | Вне зависимости от того, где вы находитесь, этот абсолютный путь перенесёт вас прямо в каталог **subdirfoo**, подкаталога, вложенного в **dir1** |
| **cd ../../dir3/X11** | Этот относительный путь сначала поднимет вас на два уровня выше, затем спустит в dir3, и наконец, в каталог X11. |

### 22. Что является комментарием (в вариантах ответа была написана какая-то неведомая штука)

**помимо # еще можно комментировать через**

**[COMMAND] <<[-] 'DELIMITER'**

**HERE-DOCUMENT**

**DELIMITER**

### 23. Какая команда не создает папки с именем sus1, sus2, sus3 (так же открывал консоль и тестил)

Ответы: mkdir sus{1-3}, mkdir sus{1..3}, mkdir sus{1, 2, 3}

Мой ответ: mkdir sus{1-3}

### 24. Что выведет $ true $ echo $?

### Ответы: 1, 0, true, $

### Мой вариант: 0

// если в консоли написать man true

### 25. Как удалённо зайти на компьютер student под ником aboba по SSH

**ssh aboba@student**

### <https://selectel.ru/blog/ssh-ubuntu-setup/>

### 26. В вопросе был написан bash-скрипт в одну строку и спрашивают, что он выведет

### 27. Вопрос про регулярки (мне попалось такое: как записать любое количество цифр)

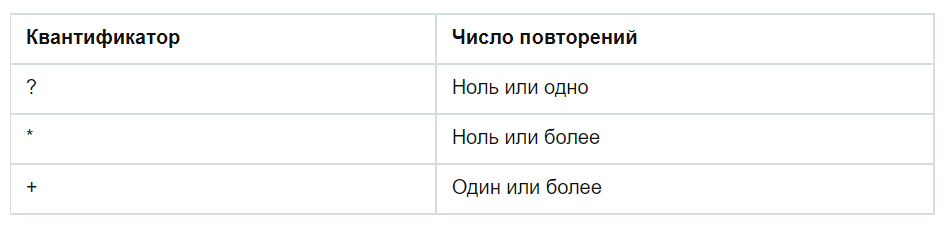
**^[0-9]+**

^ — начало строки.

Квадратные скобки [] помогают перечислить варианты для одного символа.

\* если 0 цифр тоже

+ если нужна хотя бы одна цифра

****

[Регулярные выражения (regexp) — основы / Хабр](https://habr.com/ru/post/545150/)

### 28. Какая команда заменяет маску

*replace?*

Следующая команда изменяет сетевой интерфейс с именем «eth0» для использования IP-адреса 102.168.0.1 и назначает маску подсети 255.255.255.0:

| 1 | sudo ifconfig eth0 192.168.0.1 netmask 255.255.255.0 |
| --- | --- |

Если вам также необходимо изменить шлюз по умолчанию, используемый сетевым интерфейсом, вы можете использовать команду «route». Следующая команда, например, устанавливает шлюз по умолчанию для интерфейса «eth0» для 192.168.0.253:

| 1 | sudo route add default gw 192.168.0.253 eth0 |
| --- | --- |

Чтобы увидеть новую настройку, вам нужно будет отобразить таблицу маршрутизации. Введите следующую команду и нажмите «Ввод»:

| 1 | **route -n** |
| --- | --- |

### 29. Какая команда копирует только файлы

find **source** -type f -exec cp -f **source** \*dest

При копировании каталогов нужно добавлять параметр «**-R**», так как без этого «**cp**» будет выдавать ошибку. Поэтому, для копирования каталогов нужно ввести команду «**cp -R**» и указать исходный и новый каталог. Например, для того чтобы скопировать каталог «**folder**» в «**newfolder**» вам нужно выполнить вот такую команду:

cp -R folder/ newfolder/

-R , -r , --recursive

Рекурсивное копирование каталогов.

-b

Создание резервной копии для всех файлов, которые могли быть перезаписаны или удалены при копировании.

-d

Копирование символических ссылок, а не файлов, на которые они ссылаются.

-L

Копировать файлы, а не символические ссылки.

### 30. Снова был написан скрипт в одну строку и вопрос о его выводе

### 31. Что такое Dockerfile

Dockerfile — это текстовый файл с инструкциями, необходимыми для создания образа контейнера. Эти инструкции включают идентификацию существующего образа, используемого в качестве основы, команды, выполняемые в процессе создания образа, и команду, которая будет выполняться при развертывании новых экземпляров этого образа контейнера.

### 32. Как удалить докер образ

**docker rmi image\_name\_or\_id**

Удаление одного или нескольких конкретных образов

Используйте команду docker images с флагом -a, чтобы найти идентификатор удаляемых образов. Эта команда покажет вам все образы, включая промежуточные слои образов. Когда вы определитесь с составом удаляемых образов, вы можете передать их идентификаторы или теги в docker rmi:

Список:

docker images -a

Удаление:

docker rmi Image Image

Очистка всех неиспользуемых или не связанных с контейнерами образов, контейнеров, томов и сетей

В Docker имеется команда, очищающая все не связанные с контейнерами ресурсы, в том числе образы, контейнеры, тома и сети:

docker system prune

Чтобы удалить все остановленные контейнеры и неиспользуемые образы (а не только образы, не связанные с контейнерами), добавьте в эту команду флаг -a:

docker system prune -a

### 33. Как загрузить образ

Загрузка образа в среду Docker

Для выполнения загрузки используется обратная операция **docker load**:

### 34. Какой командой можно войти в контейнер

**docker exec -it bash**

### 35. Что такое FROM в dockerfile

**creates a layer from the container\_id Docker image**

Инструкция FROM задает образ контейнера, который будет применяться при создании нового образа. Например, при использовании инструкции FROM mcr.microsoft.com/windows/servercore полученный образ является производным и зависимым от базового образа ОС Windows Server Core. Если указанный образ отсутствует в системе, где выполняется процесс сборки Docker, подсистема Docker попытается скачать его из общедоступного или частного реестра образов.

Формат инструкции FROM выглядит следующим образом:

FROM <image>

Ниже приведен пример команды FROM.

Чтобы загрузить Windows Server Core версии ltsc2019 из Реестра контейнеров (Майкрософт):

FROM mcr.microsoft.com/windows/servercore:ltsc2019

### 36. Вопрос про запуск pipeline

### 37. Выбрать этапы CD

**Развертывание**

**Поддержка и мониторинг**

**Планирование**

### 38. Вопрос про команды gitlab-ci (конкретно у меня был вопрос, какая команда хранит артефакты сроком 10 дней)

**expire\_in: 10 day**

### 39. На каком порту работает Grafana

По умолчанию, Grafana работает на **порту** 3000.

### 40. Вопрос про метрики (Combined Log Format %h %l %u %t\%r\ %&gt;s %b\%{Refer}i\\%{User-agent}i\

| Client Hostname | DNS hostname of the host making the request | %h |
| --- | --- | --- |

| Remote Logname | Remote logname (from identd). This field is almost always null ("-") | %l |
| --- | --- | --- |

| Authenticated User Name | Name or identifier of the authenticated user | %u |
| --- | --- | --- |

| Request Date and Time | Date and time at which the request was received by the server (in Common Log time format) | %t |
| --- | --- | --- |