**Министр науки и высшего образования Российской̆ Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

Факультет информационных технологий и программирования

Лабораторная работа № 2

Управление разделами дисковой системы, создание и монтирование файловых систем

Выполнил студент группы № M3311

Ершова Мария

*Подготовка инфраструктуры. Создайте связанный клон виртуальной машины с обязательным изменением MAC адреса. Режим эмуляции сети задайте как «Сеть Nat». Переименуйте хосты как srver и client. К виртуальной машине server подключите* ***четыре*** *дополнительных IDE диска по 2 Гб в режиме динамического расширения.*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки. Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Чтобы изменить имя хостов:*

**hostnamectl set-hostname server**

**hostnamectl set-hostname client**

*К виртуальной машине server подключили* ***четыре*** *дополнительных SATA (должна быть sata вместо ide) диска по 2 Гб в режиме динамического расширения:*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**lsblk** – чтобы чекнуть диски.

*Убедитесь, что машины «видят» друг друга по сети:*

Ip server – 10.0.2.15/24

Ip client – 10.0.2.5/24

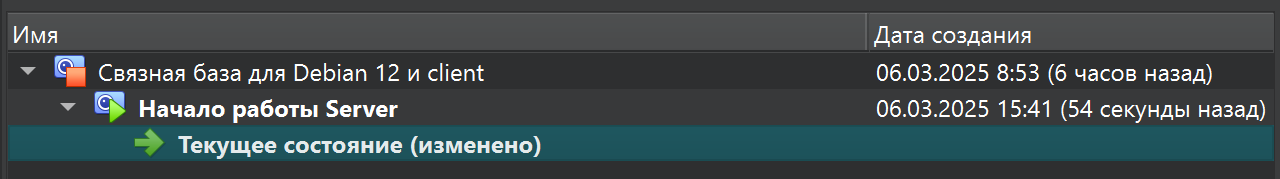
Изображение выглядит как текст, снимок экрана, Шрифт, черный

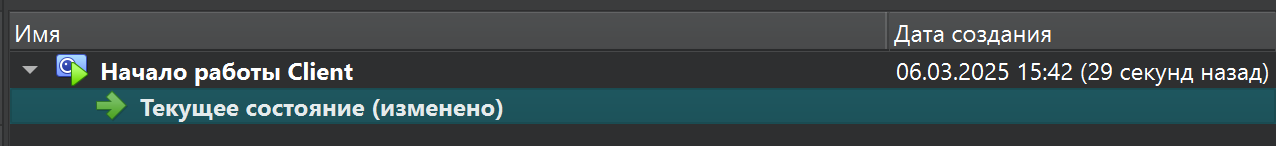
Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

*Перед началом работы создайте снимок виртуальной машины:*





*Создайте текстовый файл, в котором запишете последовательность команд для выполнения каждого из нижеследующих заданий. Для команд, имеющих интерактивный интерфейс – опишите последовательность выбора управляющих команд и их параметров. Если решение заключается в изменении конфигурационного файла – укажите название файла и вносимые или изменяемые строки.*

Мяу, мяу, мяу!

1. *На первом добавленном диске создайте новый раздел, начинающийся с первого свободного сектора и имеющий размер 500 МБайт:*

**fdisk /dev/sdb** - запуск утилиты fdisk на диске sdb.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

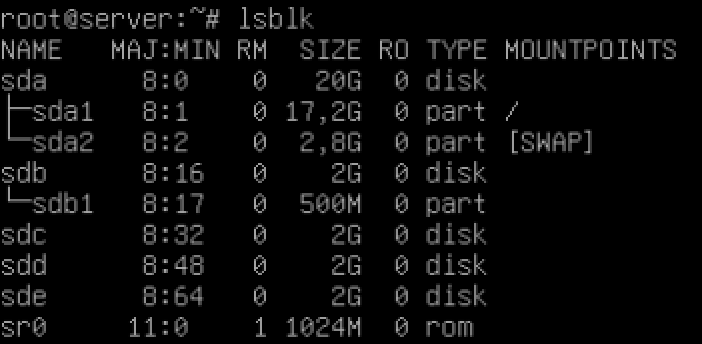
Дальше так: **n** -> **p** -> **1** -> **“enter”** -> **+500M**:

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Чтобы выйти из утилиты: **w**.

Проверяем, что все fine:



1. *Создайте файл в домашнем каталоге пользователя root и сохраните в него UUID созданного раздела:*

Делаем (**blkid /dev/sdb1 > /root/uuid.txt**) и сразу проверяем:

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

1. *Создайте на созданном разделе файловую систему ext4 с размером блока 4096 байт:*

Изображение выглядит как текст, снимок экрана, черный, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**mkfs.ext4 -b 4096 /dev/sdb1**

Сделали и радуемся, но не забываем проверять :3

Ну а сейчас я пойду пить кофе.

Mini-справочник:

**lsblk** - для просмотра информации о блочных устройствах, собственно диски и их иерархию и т.д.

**blkid** - получение информации о файловых системах и UUID.

**mkfs.ext4** - создаёт файловую систему ext4.

1. *Выведите на экран текущее состояние параметров, записанных в суперблоке созданной файловой системы:*

**tune2fs -l /dev/sdb1**

Эта штучка выводит всю информацию, записанную в суперблоке фс:

*Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.*

1. *Настройте эту файловую систему таким образом, чтобы ее автоматическая проверка запускалась через 2 месяца или каждое второе монтирование файловой системы.*

**tune2fs -i 2m -c 2 /dev/sdb1**

Хыхыхыхы, **tune2fs** помогает нам настраивать настраиваемые параметры фс.

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

1. *Создайте в каталоге /mnt подкаталог newdisk и подмонтируйте в него созданную файловую систему*

Тыкнули ls /, увидели, что каталог /mnt существует (ну а куда бы делся).

**mkdir /mnt/newdisk**

**mount /dev/sdb1 /mnt/newdisk**



Надо бы проверить:

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

О, еще нашла такую штуку – **df -h**

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Как мы видим все смонтировалось.

1. *Создайте в домашнем каталоге пользователя root ссылку на смонтированную файловую систему:*

**ln -s /mnt/newdisk /root/meeeeeeow** – ну всё, сделали мягкую ссылку и радуемся:

**Изображение выглядит как текст, Шрифт, снимок экрана, черный

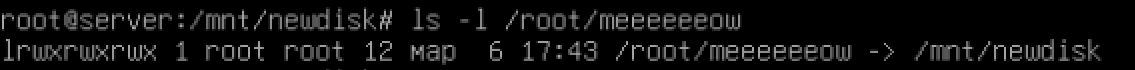
Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.**

И там, и сям этот каталог lost+found.

А вообще есть и такая штучка: ls -l /root/meeeeeeow



1. *Создайте каталог с любым именем в смонтированной файловой системе:*

**mkdir /root/meeeeeeow/cat**

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Вот такая темка.

1. *Включите автомонтирование при запуске операционной системы созданной файловой системы в /mnt/newdisk таким образом, чтобы было невозможно запускать исполняемые файлы, находящиеся в этой системе, а также с отключением возможности записи времени последнего доступа к файлу для ускорения работы с этой файловой системой. Перезагрузите операционную систему и проверьте доступность файловой системы. Проверьте невозможность запустить исполняемый файл, если он хранится в этой файловой системе:*

Делаем **blkid /dev/sdb1** и берем UUID:



Копируем UUID и записываем в /etc/fstab, добавляя еще некоторые темки:

**nano /etc/fstab**

**UUID** – уникальный идентификатор раздела

**/mnt/newdisk** – точка монтирования

**Ext4** – версия фс

**Defaults** – разрешает чтение-запись, использование устройств и суид-битов, автоматическое монтирование и асинхронный ввод-вывод.

**noexec** – запрещает выполнение исполняемых файлов в этом разделе

**noatime** – отключает обновление времени последнего доступа

**0** – отключает резервное копирование через dump

**2** – проверять после корневой (приоритет проверки фс при загрузке)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**mount -a** – монтирование всех фс в fstab

**reboot**

**mount | grep /mnt/newdisk**

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Все fine, радуемся!

**ДЕЛАЕМ СНИМОК!!!1111**

1. *Увеличьте размер раздела и файловой системы до 1 Гб. Проверьте, что размер изменился.*

Устанавливаем утилиту Parted – помогает работать с размерами фс.

**parted /dev/sdb**

**resizepart 1 1250MB** (там могут быть траблы (ошибки и служебные данные) с resize2fs, поэтому больше)

**quit**

**resize2fs /dev/sdb1 1G**

Проверяем:

Изображение выглядит как текст, снимок экрана, черный

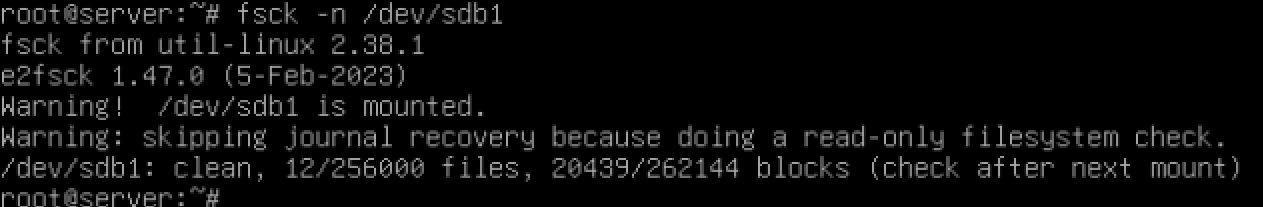
Контент, сгенерированный ИИ, может содержать ошибки.

1. *Проверьте на наличие ошибок созданную файловую системы "в безопасном режиме", то есть в режиме запрета внесения каких-либо изменений в файловую систему, даже если обнаружены ошибки.*

Нужен fsck

Используем параметр -n, так как так проверим потенциальные ошибки в фс, но не исправим их – безопасный режим.

**fsck -n /dev/sdb1**



Все хорошо!

**ДЕЛАЕМ СНИМОК СНОВА!!!1111**

1. *Создайте новый раздел, размером в 12 Мбайт. Настройте файловую систему, созданную в пункте 3 таким образом, чтобы ее журнал был расположен на разделе, созданном в этом пункте.*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Создали и радуемся.

**umount /dev/sdb1**

**tune2fs -O ^has\_journal /dev/sdb1** - отключаем журнал

**mke2fs -O journal\_dev -b 4096 /dev/sdb2** - теперь новый раздел - устройство для журнала (слава снимкам!!!)

**tune2fs -J device=/dev/sdb2 /dev/sdb1** - привязываем новый журнал к файловой системе

**tune2fs -O has\_journal /dev/sdb1** - включаем журнал обратно

**mount /dev/sdb1 /mnt/newdisk**



1. *Создайте на 2 и 3-м добавленном диске разделы, занимающие весь диск. Инициализируйте для LVM все созданные разделы.*

Устанавливаем lvm2: **apt install lvm2.**

Для дисков sdc и sdd создаем разделы sdc1 и sdd1 с помощью fdisk. Выбираем n и дальше enter, enter, enter, enter, чтобы разделы заняли максимум возможного места.

Вводим t чтобы изменить тип раздела, выбираем L, находим код для Linux LVM – 8e и вводим.

w – чтобы выйти.

Изображение выглядит как текст, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Инициализируем разделы для LVM

**pvcreate /dev/sdc1**

**pvcreate /dev/sdd1**

**pvcreate - создаст в начале раздела дескриптор группы томов**

Изображение выглядит как текст, снимок экрана, меню, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. *На дисках 2 и 3 создайте чередующийся LVM том и файловую систему ext4 на весь том.*

**vgcreate vg14 /dev/sdc1 /dev/sdd1** -создания группы томов

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

**lvcreate -n lv14 -i 2 -I 64 -l 100%FREE vg14** - создаём чередующийся LVM том -n lv14 (имя), -i 2 (количество устройств) -I 64 (размер страйпа в килобайтах) -l 100%FREE (размер логического тома).

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

mkfs.ext4 /dev/mapper/vg14-lv14

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

1. *Смонтируйте том в каталог /mnt/vol01 и настройте автомонтирование.*



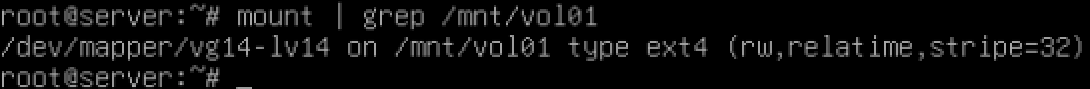
Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**mount -a**

**reboot**

**mount | grep /mnt/vol01**



1. *Получите информацию LVM о дисках, volume group и volume.*

**pvdisplay, vgdisplay, lvdisplay**

Изображение выглядит как текст, снимок экрана, меню

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

1. *Расширьте раздел на дополнительный диск используя туже volume group, что и в п. 14. Расширьте том на 100% нового диска.*

Для начала создаем два раздела с помощью fdisk - sde1 и sde2:

Изображение выглядит как текст, снимок экрана, меню, Шрифт

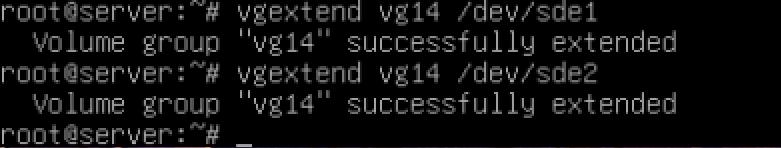
Контент, сгенерированный ИИ, может содержать ошибки.

Теперь делаем pvcreate для /dev/sde1 и /dev/sde2:

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Расширяем vg группу добавляя в нее все те же sde1 и sde2:



Ну и теперь то зачем мы тут собрались:



Ура! Все сработало:

Изображение выглядит как текст, снимок экрана, меню, Шрифт

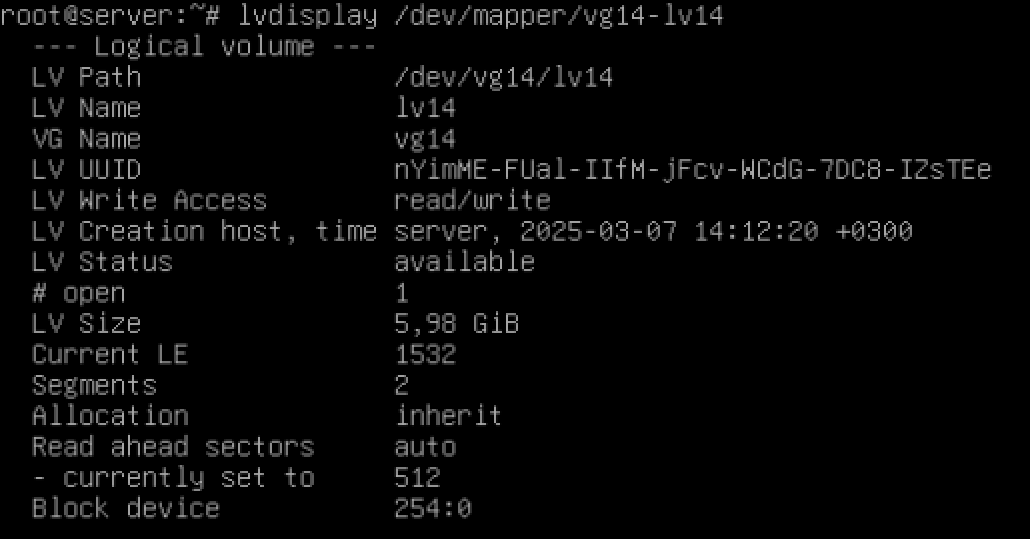
Контент, сгенерированный ИИ, может содержать ошибки.

1. *Расширьте файловую систему на 100% нового диска (обратите внимание, что вам не пришлось отмонтировать раздел)*

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Все прошло успешно, но, чтобы проверить делаем lvdisplay /dev/mapper/vg14-lv14



Видим, что все точно прошло славно, ура, ура!

1. *Получите информацию LVM о дисках, volume group и volume.*

Изображение выглядит как текст, снимок экрана, меню

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

1. *На машине server установите службу nfs-kernel-server, разрешите запуск и запустите ее. При необходимости разрешите доступ через сеть к этой службе.  
   Примечание: сделать это можно командами*

*Apt install nfs-kernel-server*

*systemctl enable nfs-server*

*systemctl start nfs-server*

Ну просто туда-сюда и все команды прописали.

1. *Сделайте так, чтобы к каталогу /mnt/vol01можно было получить доступ через NFS, при этом установите параметры, которые:*
   1. *Разрешают доступ к каталогу только с IP адресов сети ваших виртуальных машин.*
   2. *Разрешают монтировать каталог для записи.*

Заходим в /etc/exports и в конце прописываем /mnt/vol01 10.0.2.0/24(rw,sync,no\_subtree\_check). Сохраняем и выходим.

10.0.2.0/24 — разрешает доступ только для IP-адресов из сети 10.0.2.0/24

rw — разрешает чтение и запись

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Сохранили изменения, перезапустили и установили права доступа:



777 - любой пользователь может читать, записывать и выполнять файлы в этом каталоге

1. *На компьютере client осуществите монтирование сетевого ресурса в каталог /var/remotenfs.*

apt install nfs-common – оп и скачали

Теперь создаем точку монтирования, убеждаемся в том, что каталог экспортирован сервером NFS:

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Видим, что все fine.

Время для монтирования **mount 10.0.2.15:/mnt/vol01 /var/remotenfs**:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Ну вот и проверили заодно все ли смонтировалось: и файл lost+found видим, и df -h показал ожидаемый результат, УРА!

1. *Убедитесь, что монтирование удалось. Скопируйте в каталог remotenfs любой файл.*

И так, берем и идем в /var/remotenfs и создаем там файл kitik:  
  
Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Идем теперь на сервер и чекаем, что там у нас интересного:

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Все прошло успешно!

1. *Получите информацию о inode любого файла, выведите информацию о логическом размещении файла на диске*

Ну не отходя от кассы посмотрим файл kitik:

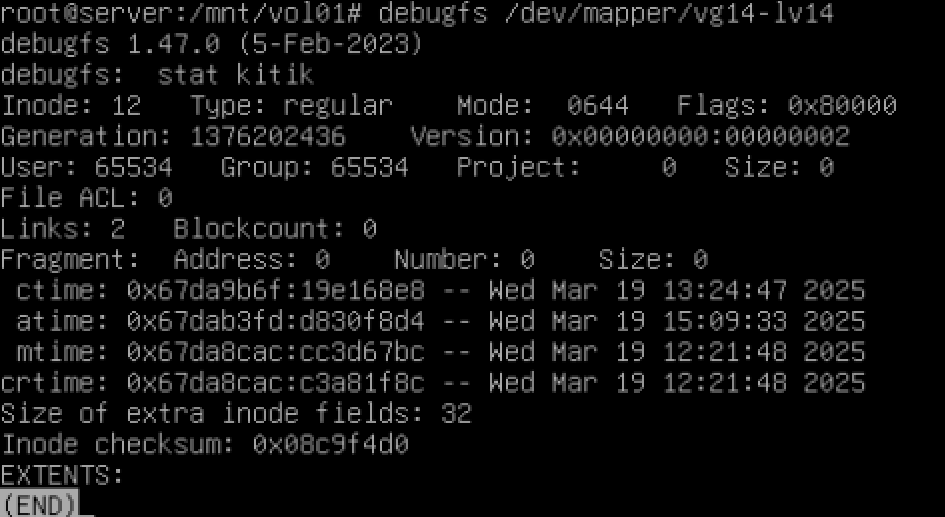
Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Номер inode файла можно узнать с помощью ls -i kitik.

filefrag -e kitik для отображения подробной информации о логическом расположении файла.

После общения с Маятиным стало понятно, что filefrag не то, что нам нужно. Необходимо использовать утилиту debugfs:



Ну и вроде все fine.

1. *Создайте мягкую и жесткую ссылки на файл, посмотрите информацию о их логическом расположении, inode, найдите отличия и сходства, чем они объясняются*

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Создаем мягкую и жёсткую ссылки на файл kitik:

ln kitik myHardLink

ln -s kitik mySoftLink

И снова смотрим что там по debugfs:

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Всё отлично!!!