# **Задание 1**

1. Создайте каталог ~/test и в нём файл test\_123 с любым содержимым.
2. Создайте символическую ссылку на каталог ~/test по пути /tmp/soft\_link.
3. Используя ссылку /tmp/soft\_link, скопируйте файл test\_123 в каталог /tmp с тем же именем. Создайте жёсткую ссылку на файл /tmp/test\_123 с именем /tmp/hard\_link.

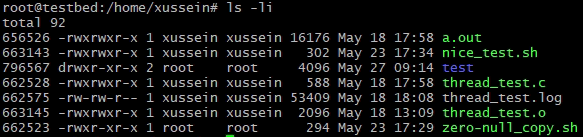
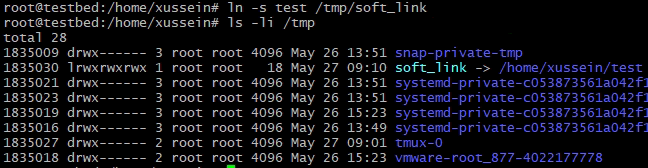
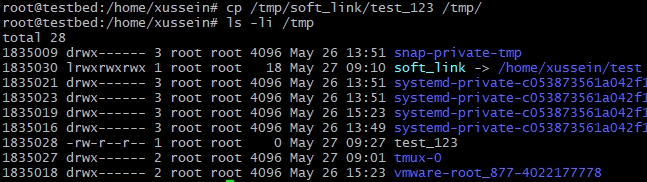
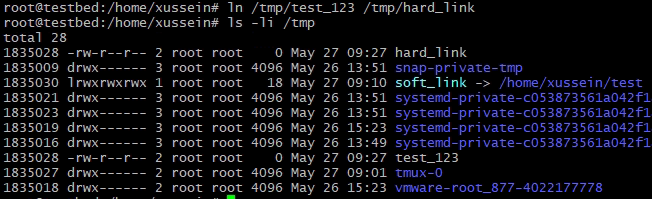
*Вопрос 1. Файл ~/test и /tmp/test\_123 — это один и тот же файл (одинаковые inode)?*

*Вопрос 2. Файл /tmp/soft\_link и /tmp/hard\_link — это один и тот же файл (одинаковые inode)?*

*Вопрос 3. Файл /tmp/test\_123 и /tmp/hard\_link — это один и тот же файл (одинаковые inode)?*

*Вопрос 4. Докажите, что одна из ссылок символическая, а другая жёсткая. Обязательно приложите к ответу скриншоты команд, которые иллюстрируют различия ссылок разного типа, или, если не уверены, ход решения задания.*

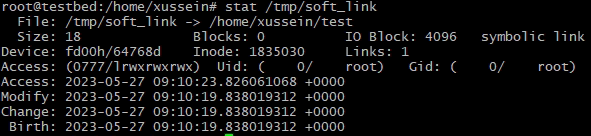
# **Решение 1**

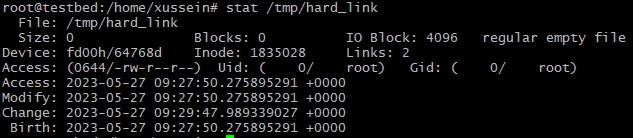
1.   
     
     
   
2. 
3.   
   

Вопрос 1. Inode у них разные, хотя бы потому что первый - каталог (inode=796567), второй - файл (inode=796581).

Вопрос 2. Нет, soft\_link inode=1835030, hard\_link inode=1835028.

Вопрос 3. Да, inode совпадают (inode = 1835028).

Вопрос 4. Команда для создания символической ссылки - ln -s file link. Через выглядит как конструкция типа link -> file. При этом ссылка получает свой inode. Команда для создания жесткой ссылки - ln file link и получает inode оригинального файла. Через ls отличить 2 хардлинка невозможно. Но через stat можно определить их количество.



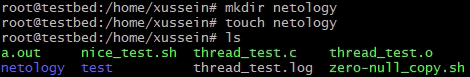
# **Задание 2**

1. Создайте файл ~/netology.
2. Создайте каталог ~/netology/.
3. Поместите файл netology в каталог netology.

*Какое или какие из трёх действий невозможно выполнить? Почему?*

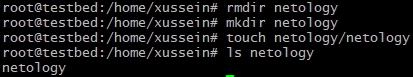
# **Решение 2**



Поскольку сначала создан файл, система не позволяет создать файл-директорий с таким же именем. В обратную сторону: 

При попытке создать файл система просто “трогает” каталог, не меняя его.

Поэтому для помещения файла netology в каталог netology его нужно создать в каталоге.

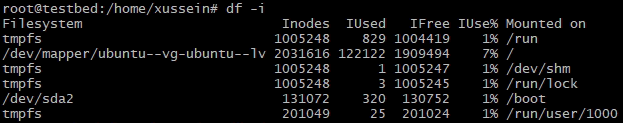


**Задание 3**

1. Как посмотреть количество inodes?
2. В каких файловых системах не может возникнуть проблемы нехватки inodes?

*Запишите ответ в свободной форме.*

# **Решение 3**

1. Утилитой df с ключом -i/. По порядку: максимальное количество inodes, количество использованных, количество свободных и процент использованных. 
2. Файловые системы btrfs и zfs “из коробки” поддерживают динамическое увеличение количества inodes. Также при создании файловых систем xfs и ext4 можно зарезервировать некоторое пространство для увеличения дефолтного количества inodes при необходимости.

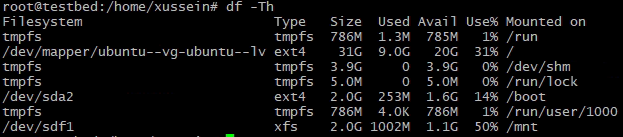
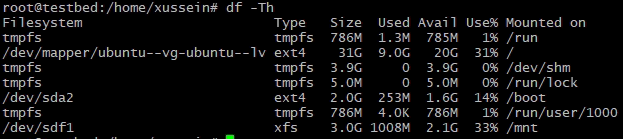
# **Задание 4**

Задание не предполагает использования LVM.

1. Подключите к системе новый диск 3 Гб.
2. Создайте на диске один раздел размером 2 Гб.
3. Разметьте раздел как xfs.
4. Смонтируйте раздел по пути /mnt. Создайте любой файл на смонтированной файловой системе. Сделайте скриншот вывода команды df -hT.
5. Увеличьте раздел до 3 Гб.
6. Расширьте файловую систему на новое свободное пространство.
7. Убедитесь, что после всех манипуляций созданный вами файл остался внутри раздела и файловой системы.
8. Сделайте скриншот вывода команды df -hT.

*В качестве ответа приложите два сделанных скриншота.*

# **Решение 4**

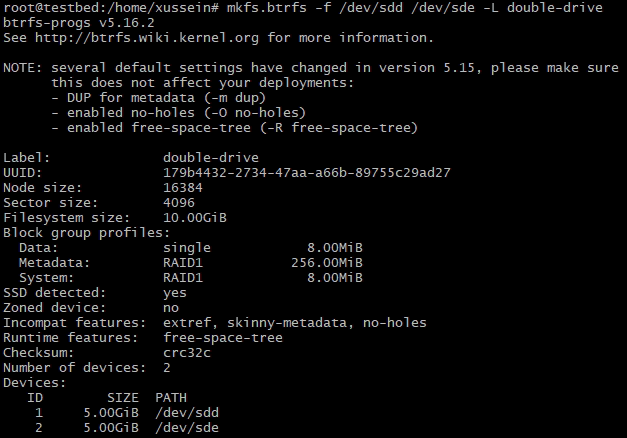
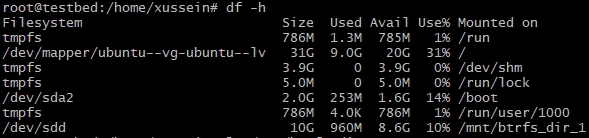
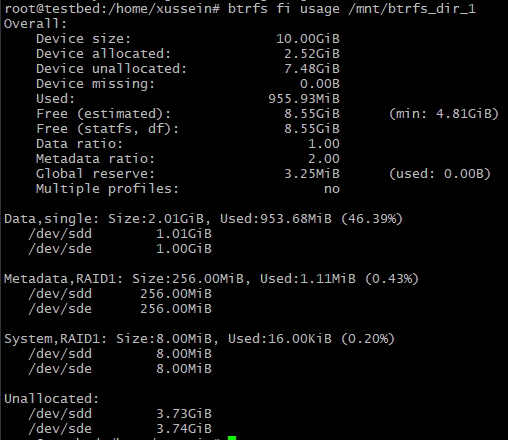
1. 
2. файл размером 1 гб все еще на месте после операций growpart и xfs\_growfs  
     
   

# **Задание 5**

Создайте файловую систему Btrfs на двух дисках по 5 Гб каждый.

*Сколько места будет доступно для работы с файлами? Сколько места займут метаданные?*

# **Решение 5**

1. 
2. df сообщает что для данных доступно 10 гигабайт. Но так как мы создали raid1 то реально доступно 5 гигабайт.  
   

Соответственно, использовано пространства 1 гигабайт (файл-пустышка с нулями) и метаданные - 1.11 мегабайт.